Hypovigilance Detection and Assistance to Vehicle Drivers

Djamel Eddine Benrachou

الجـمهـوريــة الجـزائريــة الديمقـراطيــة الشعـبيــة
وزارة التعـليم العالي والبحث العلمي

BADJI MOKHTAR- ANNABA UNIVERSITY
UNIVERSITE BADJI MOKHTAR - ANNABA

جامعة باجي مختار- عنابة

Faculté : Sciences de l'ingéniorat **Année : 2017/2018**
Département : Electronique

THÈSE

Présentée en vue de l'obtention du diplôme de Doctorat 3ème Cycle

Intitulé

Hypovigilance Detection and Assistance to Vehicle Drivers

Option : *Automatique et Signaux*

Par : BENRACHOU Djamel Eddine

Directeur de Thèse : BENSAOULA Salah MCA Univ. Annaba

Soutenue publiquement le 18 / 04 / 2018 devant le jury composé de :

PRESIDENT :	**BOUGHAZI Mohamed**	**Prof**	**Univ. Annaba**
EXAMINATEURS:	**FEZARI Mohamed**	**Prof**	**Univ. Annaba**
	BOUDEN Toufik	**Prof**	**Univ. Jijel**
	BOUCHRIKA Imed	**MCA**	**Univ. Souk-Ahras**
INVITÉ :	**BOULEBTATECHE Brahim**	**MCB**	**Univ. Annaba**

الملخص

تعتبر غفوة السائق من أحد الأسباب الرئيسية لحوادث الطرق اذ تعد مراقبة سلوكه للكشف عن النعاس مشكلة معقدة، حيث تتضمن هذه المراقبة على عناصر السائق الفسيولوجية والسلوكية. إن المتابعة عن طريق الكمبيوتر تسمح بمراقبة حالة الإنسان بدون التدخل في مهمة السياقة ويمكن الحصول على تقدير دقيق لحالة السائق من خلال تحليل تعابير الوجه، حالة العينين، وتقدير مستوى إغلاق الجفون و كذلك تحديق النظر.

يتكون نظام مراقبة حالة السائق -اعتمادا على حالات العينين- من ثلاث خطوات أساسية: 1) كشف الوجه 2) كشف موقع العينين بدقة 3) التعرف على حالة العينين مفتوحة أو مغلقة. هاته المراحل التي تكون عملية و في وضعيات قيادة فعلية تقدم استجابة دقيقة. إن نظام المراقبة بالكمبيوتر المخصص لرصد حالة السائق، يستخدم منطقة الوجه كحد للمعالجة. ان نظاما كهذا يتطلب تقنيات التقاط الصورة و المعالجة المناسبة و المتينة التي تضمن لنا تشغيل مستقر.

في هاته الأطروحة، اقترح في المقام الأول استعمال Gabor's wavelets وتحليل المكونات الأساسية PCA للكشف عن الوجه. تسمح هاته الطرق بالتمثيل الجيد لملامح الوجه. فيما يتعلق بمرحلة التصنيف والتفسير تشكل الخصائص المستخرجة مدخل المصنف الذي يكون من نوع فاصل على هامش واسع SVM. إن هذه المرحلة الأولي، تتضمن إستراتيجية جديدة لدراسة الأوجه عن طريق معالجة الصور والعمليات المرفولوجية. هذا ما يسمح بمعرفة الموقع الصحيح للوجه في ظروف واقعية. تتطلب مرحلة الاختبار استعمال قاعدتي بيانات عاميتين هما قاعدة بيانات ORL و CMU-MultiPIE. و لقد قمنا أيضا بجمع قاعدة بيانات خاصة بنا تمثل صور لعدة أشخاص ملتقطة في ظروف بيئية ذات إضاءة طبيعية و غير مراقبة، وذلك بغية تقييم أداء التعميم لطريقتنا المقترحة وفعاليتها للتغيرات المحيطة. تحتوي قواعد البيانات الثلاث على الظروف البيئية التي تعكس ظروف السياقة اليومية، حيث نقص الدقة أو عدم كشف الوجه يشكلان دوما عائقا لا يستهان به لعمل النظام الكلي، مما يعني أنه من المستحيل تحليل ملامح الوجه إذا كان هذا الأخير غير مكشوف بدقة أو غير مكشوف تماما. إن الطريقة التي استعملت في هاته الأطروحة تعتمد على المعلومات المستخرجة من خصائص الصور. وهذا يقتضي استخدام أنماط الثنائية المحلية LBP ، والتي من المعروف أنها تكون شديدة التمييز، ومقاومة لتغيرات كل من القوام و الإضاءة المحيطة .
تركز المرحلة الثانية على كشف وتحديد موقع العينين. تم تطوير ثلاث طرق للعمل مع الصور الثابتة وتسلسل الفيديو، و التي تم الحصول عليها من جهاز كمبيوتر محمول مع كاميرا غير معايرة، وتحت ظروف إضاءة مختلفة.

الطريقة الأولى :
تستخدم الطريقة الأولى واصف الخصائص اعتمادا على الرسم البياني للنماذج الثنائية المحلية المحسنة فضائيا، حيث النتيجة تعطى على شكل مدخل إلى خوارزمية التعلم العميق على أساس الشبكات العصبية المتكررة (RNN)، ولا سيما نموذجي Long Short-Term Memory (LSTM) و SVM. تم الكشف عن منطقة العين بنجاح، مع دقة 98.1 ٪ في تسلسل الفيديو في الوقت الفعلي، في مدة زمنية تقدر 0.562 ثانية.

ومع ذلك، فإن هذه الطريقة قد لا تسمح بالكشف عن موقع العينين بشكل صحيح في الظروف القصوى للدوران المحوري للرأس(سواءا أفقيا أو عموديا). وبالإضافة إلى ذلك، بعض القوام من صورة العين ليسموصوفا بشكل جيد، بسبب تغيير المنظور، وعدم قدرة eLBPH على تمييزبعض النماذج في هذه الحالة.

الطريقة الثانية :

يتم حل هذه المشاكل من خلال الحفاظ على ثبات التغيرات في العالم الحقيقي، على طول الخوارزمية الثانية المحددة للعينين. ان eLBPH يجمع بين طريقة فيولا-جونز للكشف وتتبع موقع العينين، LBP المنتظمة والمسافة الإحصائية من نوع شي مربع (Sχ2).

هذا المزيج يعزز أداء كاشف فيولا-جونز الكلاسيكي، وموفرا تقديرا أفضل لموقع العين. ويمكن أيضا التغلب على بعض المشاكل التي واجهتها الخوارزمية السابقة.يتم التحقق من أداء الخوارزمية الحالية عن طريق ثلاث قواعد بيانات عامة، وهي قاعدة بيانات الوجه (الوجه GI4E)،قاعدة البيانات Yale-B الموسعة وتسلسل الفيديو من (GI4E وضعية الرأس). الخوارزمية تعمل دون الكشف المسبق للوجه وفي ظروف إضاءة الحقيقية. هذه الخوارزمية تحدد موقع العينين بدقة 97.35٪ ، بزمن حساب قريب من الوقت الحقيقي.

الطريقة الثالثة :

في الطريقة الثالثة، يقترح قاموس للميزات المحلية الثابتة لتمثيل منطقة العين، ويدعى الرسم البياني LBP الهرمي المحسن فضائيا، المقترح.حيث يكونهذا الواصف في قلب خوارزمية جديدة تسمى EyeLSD، والتي اقترحت للتحديد الدقيق لموقع العين والكشف عن حالتها (سواء مفتوحة أو مغلقة). تحتوي خوارزمية EyeLSD على ثلاث خطوات رئيسية، في الخطوة الأولى نقوم بالمعالجة المسبقة للصورة عن طريق الحد من التشويش وتحسين قوام الصورة. الخطوة التالية تهدف إلى دمج اثنين من المصنفات، SVM و Perceptron متعدد الطبقات (MLP)، لتصنيف ثنائي لصور العين أوغيرها. من أجل تحسين عملية الكشف عن العين تم الاعتماد على سلسلة من الخطوات، في مرحلتي ما قبل و بعد المعالجة. حيث قمنا بتقييم هذه الخوارزمية بناءا على ثلاث قواعد بيانات عامة، CAS PEAL-R1 ، BioID وقاعدة بيانات حقيقية للعين ZJU Eyeblink. و قد قمنا كذلك بجمع قاعدة بيانات خاصة بنا مختلفة من حيث حالات العين و كذا تعبيرات الوجه. النتائج التي تم الحصول عليها تبين أن طريقة EyeLSD فعالة في تحديد موقع العينين بدقة قدرها 98.12 ٪ في سيناريوهات مختلفة من العالم الحقيقي.

تركز المرحلة الثالثة والتي تعتبر الخطوة الأخيرة في خوارزمية EyeLSD على التعرف على حالة العين، و يكون ذلك بوضع إستراتيجية التعلم الفعال لتفسير صور العين الملتقطة بواسطة الواصف متعدد TPLBP المقترح. متعدد TPLBP يجمع بين قدرة الدقة المتعددة LBP بغية تقديم وصف ثري عن المعلومات المستخرجة من صور العيون (فيما يتعلق بالقوام الدقيق و الكلي لنموذج العين على حد سواء.). ويهدف الواصف متعدد TPLBP إلى تحسين متانة النموذج ضد التغيرات السابق ذكرها. وقد أسفرت خطوة اكتشاف حالة العين عن نتائج واعدة مع دقة تقدر بـ 95.18 ٪.

المفاتيح

تحليل الفيديو في وقت حقيقي، غفلة، غفوة، مساعدة السائق، نظام غير تدخلي، سائق السيارة، كشف.

Abstract

Driver drowsiness is one of the main causes of road accidents. Monitoring the behavior of the driver for the detection of drowsiness is a complex problem, which involves physiological and behavioral elements. Computer vision provides the ability to monitor the person without interfering with the driving task. An accurate estimate of the driver state, can be obtained by analyzing the facial expressions, including the eye states : eyelid closeness, blinking, or gaze fixation. A driver monitoring system by analyzing the eye conditions has three basic steps : (1) face detection ; (2) eye detection and localization ; (3) recognition of the eye states (open or closed). These steps being operational under real driving conditions, must provide a highly accurate detection response. A computer vision system, dedicated to driver monitoring, uses the driver's face as a treatment area. Such system is governed by appropriate and robust acquisition and processing techniques that ensure stable operation.

In this thesis, the proposed scheme for face detection uses Gabor's wavelets, Principal Component Analysis (PCA), to characterize the facial region with optimal data and a Support Vector Machine (SVM) classifier for the classification phase. This first step involves a new analysis strategy using image processing methods and morphological operations, This allows to recognize the exact position of the face in real conditions. Two public databases are involved in the test phase, namely the ORL face database and the CMU-MultiPIE database. We also built our own database, representing different subjects under real and uncontrolled lighting conditions, in order to evaluate the generalization performance of our approach and its robustness to ambient changes. These three databases include the most common environmental conditions in daily driving. However, degraded detection or loss of face detection is an obstacle to the overall functioning of the system, i.e., it is impossible to analyze facial features. This occurs when the driver does not maintain a frontal position to the camera. Textures information-based method has been used in this thesis, this choice was made on the Local Binary Pattern (LBP) technique, known to be highly discriminative and robust to different environmental and textures changes. In the second step, three methods are proposed, to detect the eyes in images and video sequences, obtained from a laptop with a Web camera, under different lighting conditions.

First approach

This first approach, uses a spatially enhanced LBP histogram-based feature descriptor (eLBPH), the result of which is given as input to a deep learning algorithm based on recurrent neural networks (RNN), particularly the Long Short-Term Memory (LSTM) model and the SVM classifiers. The ocular region is detected successfully, with an accuracy of 98.1% in real-time video sequences, with a computation time of 0.562 seconds. However, this method may fail to correctly detect the eyes under conditions of extreme axial (horizontal or vertical) head rotation. In addition, some image textures are not well described, because of the perspective change, and the inability of the eLBPH to discriminate certain patterns in these cases.

Second approach

The problems encountered in the first approach are solved by preserving the invariance of changes in the real world. In this approach we combine the Viola-Jones method for eye detection and tracking,

uniform LBPs and a chi-square statistical similarity distance ($\mathbb{S}_{\chi^2}$). This combination enhances the performance of the classic Viola-Jones detector, providing a better estimate of eye locations. It can also overcome some of the problems encountered by the first approach. The present algorithm is validated with three public databases, namely the face database (Face GI4E), the extended Yale-B database and video sequences of (GI4E Head Pose). The algorithm works, without prior detection of the face and in real lighting conditions. This algorithm locates the eyes with an accuracy of 97.35%.

Third approach

In the third approach, a dictionary of invariant local features, called the spatially enhanced LBP Pyramidal histogram (ePLBPH *), is proposed to represent the ocular region. The ePLBPH* descriptor is the core of a new algorithm called EyeLSD, which we have proposed for ocular localization and state detection (open or closed). The EyeLSD algorithm consists of three main stages, the first stage pre-processes the image by reducing noise and improving textures. The second stage integrates two classifiers, SVM and Perceptron Multilayer (MLP), for a binary classification of eye and non-eye images. A series of preprocessing and post-processing steps are implemented to improve the eye detection stage. We evaluated this algorithm on three public databases, BioID, CAS PEAL-R1 and a real world eye database ZJU Eyeblink. We also acquired and annotated our own database for different eye conditions and facial expressions. The results obtained show that the EyeLSD method is effective for locating the eyes with an accuracy of 98.12% in real scenarios. The third step, which is the final stage of the EyeLSD algorithm, focuses on recognizing the state of the eyes (open or closed), establishing an effective learning strategy for interpreting the detected eye images with the descriptor Multi-TPLBP proposed. Multi-TPLBP combines LBP's multiple resolution capability for a rich description of eye patch information (regarding both the micro- and macro-textures of the eye model). The multi-TPLBP descriptor also aims to improve the robustness of the model with different conditions of acquisition and environment. The eye state detection step has yielded promising results with an accuracy of 95.18% and can also treat a very wide range of eye appearance than other methods compared with.

Keywords : Real-time video analysis, hypovigilance, drowsiness, driver assistance, non-intrusive system, vehicle driver, detection.

Résumé

La somnolence du conducteur est l'une des principales causes des accidents de la route. Surveiller le comportement du conducteur pour la détection de la somnolence est un problème complexe, qui implique des éléments physiologiques et comportementaux. La vision par ordinateur permet de surveiller la personne sans interférer avec la tâche de conduite. Une estimation précise de l'état du conducteur peut être obtenue en analysant les expressions faciales dont l'état des yeux : la mesure du niveau de fermeture des paupières ainsi que le clignement ou la fixation du regard. Un système de surveillance de l'état du conducteur basé sur l'états des yeux se compose de trois étapes fondamentales : (1) détecter le visage ; (2) la détection et la localisation des yeux ; (3) la reconnaissance de leurs états (ouvert ou fermé). Ces étapes étant opérationelles dans des conditions réelles de conduite, doivent fournir une réponse de détection très précise. Un système de vision par ordinateur, dédié à la surveillance de l'etat du conducteur, utilise la zone faciale comme limites de traitement. Un tel système est régi par des techniques d'acquisition et de traitement appropriées et robustes garantissant un fonctionnement stable.

Dans cette thèse est proposé en premier lieu l'utilisation des ondelettes de Gabor et l'Analyse de Composantes Principales (ACP) pour la détection du visage. Ces deux méthodes permettent une représentation optimale des caractéristiques du visage. Ainsi, les caractéristiques extraites constituent l'entrée d'un classifieur de type séparateurs à vaste marge (Support Vector Machine, SVM) pour les phases de classification et d'interprétation. Cette première étape, comporte une nouvelle stratégie d'analyse par des méthodes de traitements d'images et des opérations morphologiques. Ce qui permet de reconnaitre la position exacte du visage dans des conditions réelles. La phase de test implique l'utilisation de deux bases de données publiques, à savoir la base de données de visage ORL et CMU-MultiPIE. Nous avons également collecté notre propre base de données, représentant différents sujets dans des conditions d'éclairage réelles et non-contrôlées, afin d'évaluer la performance de généralisation de notre approche et sa robustesse aux changements ambiants. Ces trois bases de données incluent les conditions environnementales les plus courantes qui reflètent celles de la conduite quotidienne. Cependant, une détection dégradée ou une perte de détection du visage, constitue un obstacle au fonctionnement global du système : c'est-à-dire qu'il est impossible d'analyser les traits du visage. Ceci se produit lorsque le conducteur ne maintient pas une position frontale à la caméra. La méthode basée sur l'information de textures a été utilisée dans cette thèse. Ceci implique l'utilisation des motifs binaires locaux (Local Binary Patterns, LBP), qui sont connus pour être hautement discriminants, robustes aux changements d'éclairage environnemental et celles des textures. La deuxième étape se focalise sur la détection et la localisation des yeux. Trois méthodes sont dévelopées pour fonctionner avec des images statiques et des séquences vidéo, obtenues à partir d'un ordinateur portable avec une caméra Web, et dans conditions d'éclairage diverses.

Première approche

La première approche utilise un descripteur de caractéristique basé sur les histogrammes LBP spatialement améliorés (eLBPH), dont le résultat est donné en entrée à un algorithme d'apprentissage profond basé sur des Réseaux de Neurone Récurrents (RNN), en particulier le modèle *Long Short-Term Memory* (LSTM) et les classifieurs SVM. La région oculaire est détectée avec succès, avec

une précision de 98.1% dans les séquences vidéo en temps réel, avec un temps de calcul de 0.562 secondes. Cependant, cette méthode peut ne pas détecter correctement les yeux dans des conditions de rotation axiale (horizontale ou verticale) extrême de la tête. De plus, certaines textures de l'image oculaire ne sont pas bien décrites, en raison du changement de perspective, et de l'incapacité des eLBPH à discriminer certains modèles dans ces cas de figure.

Deuxième approche

Les problèmes rencontrés dans la première approche, sont résolus en préservant l'invariance des changements dans le monde réel. Dans cette approche nous combinons la méthode de Viola-Jones, pour la détection et le suivi des yeux, les LBP uniformes et une distance de similarité statistique, de type khi carré ($\mathbb{S}_{\chi^2}$). Cette combinaison augmente les performances du détecteur classique Viola-Jones, fournissant une meilleure estimation des emplacements des yeux. Elle peut également surmonter certains des problèmes rencontrés par la première approche. Le présent algorithme est validé avec trois bases de données publiques, à savoir la base de données de visage (Face GI4E), la base de données Yale-B étendue et des séquence vidéos de (GI4E Head Pose). L'algorithme fonctionne, sans détection préalable du visage et dans des conditions d'éclairage réels. Cet algorithme localise les yeux avec une précision de $97,35\%$.

Troisième approche

Dans la troisième approche, un dictionnaire des caractéristiques locales invariantes, appelé histogramme LBP Pyramidal spatialement amélioré (ePLBPH*), est proposé pour représenter la région oculaire. Le descripteur ePLBPH* est au coeur d'un nouvel algorithme appelé EyeLSD, que nous avons proposé pour la localisation oculaire et la détection d'état (ouvert ou fermé). L'algorithme EyeLSD comprend trois étapes principales, la première prétraite l'image en réduisant le bruit et en améliorant les textures. La seconde étape intègre deux classificateurs, SVM et Perceptron multicouche (MLP), pour une classification binaire des images oculaires et non-oculaires. Une série d'étapes de prétraitement et de post-traitement est mise en oeuvre pour améliorer le processus de détection des yeux. Nous avons évalué cet algorithme sur trois bases de données publiques, BioID, CAS PEAL-R1 et une base de données oculaires réelles ZJU Eyeblink. Nous avons également acquis et annoté notre propre base de données pour différentes conditions oculaires et expressions faciales. Les résultats obtenus montrent que la méthode EyeLSD est efficace pour localiser les yeux avec une précision de $98,12\%$ dans des scénarios réels. La troisième étape, qui est la dernière étape de l'algorithme EyeLSD, se concentre sur la reconnaissance de l'état des yeux (ouverts/fermés), en établissant une stratégie d'apprentissage efficace pour interpréter les images des yeux détectées avec le descripteur Multi-TPLBP proposé. Multi-TPLBP combine la capacité de résolution multiple LBP pour une riche description des informations de patchs oculaires (concernant à la fois les micro et macro-textures du modèle de l'oeil.) Le descripteur multi-TPLBP vise également à améliorer la robustesse du modèle aux différentes conditions d'acquisition et d'environment. L'étape de détection de l'état des yeux a permis l'obtention de résultats prometteurs avec une précision de $95,18\%$ et peut également traiter une très large gamme d'apparence de l'oeil par rapport aux méthodes récentes.

Mots clés : Analyse vidéo en temps réel, hypovigilance, somnolence, assistance conducteur, système non-intrusif, conducteur véhicule, détection.

Acknowledgement

The work presented in this thesis was carried out at the Laboratory of Automatic and Signals of Annaba (LASA) at Badji Mokhtar University of Annaba, and at INESC Porto - Institute for Systems and Computer Engineering of Porto at the Faculty of Engineering of the University of Porto (FEUP). For that, I take this opportunity to thank all the people who contributed to the realization of this work and the success of this scientific journey.

Let me begin, by expressing my deepest gratitude to my Supervisor Dr. Salah Bensaoula and my Co-supervisor Dr. Boulebtateche Brahim. Thanks to their encouragement, their pedagogy and valuable advice, they were able to guide me to carry out my research. I express my respect for them.

would express my sincere thanks to Prof. António Paulo Moreira (manager of ROBIS- Robotics and Intelligent Systems Unit/INESC-TEC Porto), who welcomed me into his laboratory during my scientific visits and gave me access to research facilities. He supported my achievement in my journey and paved the path for me to achieve my goals. I would also like to express my gratitude to my friend and Co-author, Dr. Filipe Neves Dos Santos, who found the time and the energy to help me despite his busy schedule. His kindness, support, encouragement and invaluable advices have greatly contributed to the development of this work.

I am honored by the presence of Prof.Mohamed Boughazi as a chair of this jury. Please find here the expression of my deepest respect and sincere gratitude. I'm also thankful to Prof.Mohamed Fezari who accepted kindly to review this work. Please find here, my deepest sympathy for your encouragement and support. My thanks also go to Prof. Toufik Bouden and Dr. Imed Bouchrika who kindly agreed to participate in this thesis committee and for taking the time to review this work.

Thank you to all my friends and colleagues from the LASA laboratory, in particular to Billel Amouri, Nacer Kouadria, Mohamed Amine Bouguerra, Tarek Melahi, Seif Allah ElMesloul Nasri, Chouaeb Chakour, Abdeljalil Larab, Mouad Kezih and Fethi Amara. I always remeber the good time we had.

Sincere thanks to the entire ROBIS staff, especially for Héber Sobreira, André Araújo, Luís Oliveira, Bruno Ferreira, Jorge Barbosa, André Figueiredo, and Tiago Pereira. From a simple visit, a friendly relationship was created, and it has been a rewarding experience.

Last but not least, I would like to thank my family for all their love and encouragement. For my grandmother, my father and my mother who raised me with a love of science and supported me in all my pursuits, for my brother and sister. Their support has been unconditional during all these years. Thank you.

List of Tables

List of Figures

List of abbreviations

- **AAM**: Active Appearance Model
- **ACC**: Classification Accuracy
- **ADAS**: Advanced driver-assistance system
- **ANN**:Artificial Neural Network
- **ASFA**: French Association of Highway Companies
- **ASM**:Active Shape Model
- **AUC**: Area Under Curve
- **BPTT**: Back-propagation through time algorithm
- **BRINT**: Binary Rotation Invariant and Noise Tolerant
- **CARRS-Q**: Center for Accident Research and Road Safety- Queensland
- **CCD Camera**:Charge Coupled Device Camera
- **CEC**: Constant Error Carousel
- **CFB**: Correlation Filter Bank
- **CLBP-S**: Completed LBP-Signe
- **CLBP**: Completed LBP
- **CLM**: Constrained Local Model
- **CRC**: Correlation Representation Classifier
- **CV**: Cross-Validation
- **DAC**:Driver Alert Control
- **DCT**: Discrete Cosine Transform
- **DFT**: Discrete Foutier Transform
- **DMS**: driver monitoring system
- **ECG**: Electrocardiograph
- **EEG**: electroencephalography

- **ELM**: Extreme learning machine
- **EMG**: electromyography
- **EOG**: electrooculography
- **FN**: False Negative
- **FNN**: Feedforward Neural Network
- **FP**: False Positive
- **FPLBP**: Four-Patch LBP
- **GPF**: General Projection Function
- **GPS**: Global Positioning System
- **HOG**: Histogram of Oriented Gradients
- **HPOG**: Histogram of Principal Oriented Gradients
- **HSV**: Hue, Saturation, and Value
- **ICA**: Independent Component Analysis
- **IPF**: Integral Projection Function
- **IR**: Infrared
- **K-ELM**: kernelized ELM
- **LBP**$^{sri_s u2}$: Scale and Rotation Invariant SubUniform LBP
- **LBP-HF**: LBP and Fourier Features
- **LBP**: Local Binary Pattern
- **LBPF**: Local Binary Pattern Filtering
- **LBPH**: Local Binary Pattern Histogram
- **LBPV**: Local Binary Pattern Variance
- **LDA**:Linear Discriminant Analysis
- **LDW**:Lane Departure Warning
- **LFW**: Label Face in the Wild database
- **LPF**: Low-pass filtering
- **LSP**: Locally Selective Projection

- **LSTM**: Long Short-Term Memory
- **LTP**: Local Ternary Pattern
- **MLP**:multilayer perceptron
- **MRELBP**: Median Robust Extended LBP
- **Multi-LBP**: Multi-resolution LBP
- **Multi-TPBLP**: Multi-resolution Three-Patch LBP
- **MultiHPOG**: Multi-scale histogram of principal oriented gradients
- **NHTSA**: U.S.National Highways Traffic Safety Administration
- **NIR**: Near-Infrared
- **PCA**:Principal Component Analysis
- **PDM**:Personalized driving model
- **PERCLOS**: Percentage of Eye Closure
- **PLBP**: Pyramid LBP
- **PS**: Pictorla Structure
- **RBF**: Radial Basis Function
- **RGB**:Red, Green, Blue color space
- **RNN**: Recurrent Neural Network
- **ROC**: Receiver Operating Characteristic
- **ROI**:Region Of Interest
- **SBELM**: Sparse Bayesian Extreme Learning Machine
- **SDM**: Supervised Descent Method
- **SIFT**: Scale Invariant Feature Transform
- **SRC**: Sparse Representation Classifier
- **SVM**:Support Vector Machine
- **TCL**:Time-to-lane crossing
- **TN**: True Negative
- **TP**: True Positive

- **TPLBP**: Three-Patch Local Binary Pattern
- **TPLBPF**: low-pass filtering Three-Patch LBP
- **VF**: Variance Filter
- **VPF**: Variance projection Function
- **WRSI LBP**: weighted rotation and scale invariant LBP
- **eLBPH**: specially enhanced Local Binary Pattern Histogram
- **ePLBPH***: enhanced Pyramid Local Binary Pattern Hisotgram

Contents

CHAPTER 1

General Introduction

Contents

1.1 General context

Over the past three decades, scientific research, automotive industry, and government agencies have invested enormous efforts to improve driving conditions and driver safety. However, a significant number of serious accidents are still occurring. Driver hypovigilance−drowsiness and cognitive distraction−results in critical and harsh accidents. According to "the European accident research and safety report" of 2013, established by Volvo Truck Corporation [1] about 1.2 million of deaths are caused by road accidents. The same source reported about 90% of accidents are due to hypovigilance. Another report published in 2011 by *Center for Accident Research and Road Safety-Queensland* [2] (CARRS-Q) [1] showed that 30% of road deaths are caused by drowsy driver [2]. This rate can reach up to 50% in specific cases (e.g., fatal accidents involving a single vehicle). The French Association of Highway Companies (ASFA)[2] [3] stated that drowsiness was the leading cause of accidents on highways (1 accident out of 3), followed by drunk driving (1 accident out of 4) and speeding (1 accident out of 8). The *National Highway Traffic Safety Administration* [4] (NHTSA)[3] claimed that 100.000 accidents are related to drowsiness of which 1550 are fatal and 40.000 cause serious injuries. NHTSA stated that drowsy driving caused between 2.2% and 2.6% of all fatalities each year. In UK, over 20% of hazards are caused by preceding reasons [5]. As a result, there is a real need for systems to constantly monitor and alert the driver of hazardous situations. The demand of the automotive industry to monitor drivers has already given rise to a number of solutions focused on driver monitoring systems. The technology of some of them is described in the following. One of the optimal solutions to improve the driver safety is to anticipate his dangerous behaviors to avoid

hazardous risks.

1.2 Motivation

Driver hypovigilance causes a considerable amount of highway hazards and safety-critical events. This occurs in thousands of severe physical, economic and psychological illnesses affecting drivers, customers and vehicles around and pedestrians as collateral casualties. Statistical data of collision reports show that driving in an hypovigilance states (drowsiness and cognitive distraction) cause a serious fatality rate on highways. So, to overcome this fact, mediation must be at the stage of these two parameters,i.e., drowsiness and distraction. This can be an effective way to prevent serious hazards. Uncontrollable need to sleep and deteriorations of driving performances, characterize the driver drowsiness state, e.g., low reaction moment, alertness loss, and shortfalls in information processing [6, 7]. Several state agencies establish a series of preventive laws and procedures, for preventing hypervigilance state and diminishing amount of accidents,e.g., compulsory rest in long distance drive. Today, certain vehicles exploit diverse implementations of safety technology. This machinery includes pretension seat-belt, airbag, antilock brake operation, traction control strategy, and electronic stability programs. However, such security mechanisms can protect customs in case of collisions only to a specific extent and still do not reduce the high hazard cost. So, to prevent vehicle crashes, scientists are targeting driver monitoring schemes and measuring the degree of drowsiness and cognitive distraction. Preventing systems include, driver monitoring system (DMS), advanced driver-assistance systems (ADAS), driver inattention monitoring schemes, and driver warning operations.

1.3 Thesis objective

The focus of this thesis is to design a groundwork based on the facial expression study of the individual to prevent hypovigilance. However, it is necessary to first identify the different forms of hypovigilance, namely inattention, fatigue, and drowsiness. These terms are considered well-established concepts. But from a scientific perspective, almost no clear definition of the phenomenon is provided. The lack of explicit definitions of hypovigilance is a major problem that often leads to misunderstanding and confusion between different studies. This also generates a false estimate of the actual impact that different forms of hypovigilance can have on driver states. As a result, it will be difficult to overcome this phenomenon and avoid thousands of road accidents. Therefore, we briefly define each state related to driver hypovigilance:

[1]. CARRS-Q is an Australian center dedicated to research and education in the field of road safety at national and international level. They evaluate the human, economic and material damage caused by road accidents.

[2]ASFA for Association des Sociétés Françaises d'Autoroutes, is a professional association bringing together all players in the concession, motorway and road construction sector in France.

[3]NHTSA is a US federal agency responsible for road safety established in 1970. It is responsible for defining and enforcing standards for the construction of road infrastructures and vehicles.

1.3.0.1 Driver hypovigilance: drowsiness and cognitive distraction

Drowsiness is an intermediate phase between two physiological reactions: Wakefulness and sleep [8]. During this transition, the organism shows observation and analysis faculties reduced and inhibited. Drowsiness can manifest at different scales that tend to vary between individuals and often in function of time. This variation can in part be brought to light by the interaction between two processes: the circadian rhythm and the homeostatic process [9]. The circadian rhythm is controlled by the internal biological clock of the human being, whereas, the homeostatic process results in an increase in sleep resistance with continuous hours of wakefulness [9]. The combination of the above processes and factors, generates adverse effects on the human organism, which results in sleep feeling of the individual at any time of the day, and sometimes throughout all day long [8]. Consequently, sleepiness phenomenon leads to catastrophic and hazardous situations, which may involved in a large number of road traffic accidents.

Somnolence or sleepiness is a safety-critical phenomenon, which is characterized by three parameters: the time of day (circadian rhythm), the duration of time awake and the prior sleep (homeostatic regulation) [10, 11]. Sleepiness may be defined by micro-sleep periods of 2 to 6 seconds that the person may have. The operational definition of sleepiness [12] is a physiological drive to fall asleep. In industrial and operational environment, sleepiness is a feared phenomenon and can be caused by workload and hours that worker spent at a specific task, principally if this task is monotonous kind. Sallinen et al. [13] found that monotonous work, can be detrimental to moderate sleep loss (4 hours of nocturnal sleep) for drowsiness and performance impairment [14]. This observation is perfectly appropriated in a driving environment.

Fatigue is described as the general overall feeling of tiredness and lack of energy. It is often confused with drowsiness state. Fatigue can be associated with many variables, such as work environment, health, sleep quality, and medications. It is also this feeling of maintaining focus to accomplish a specific task, because of its monotonous nature, e.g., highway driving. Thus, drowsiness and fatigue are closely related and difficult to dissociate. They are affected differently with other states of the person, such as chronic stress, mental load, and chronic pain.

Distraction is a lack of awareness when the driver is distracted from his/ her primary task that is driving [15]. Driver's intention or distraction, is also one of the major causes of traffic accidents that adds to driver fatigue and drowsiness. The driver's distraction may be separated into two main categories (visual and cognitive distraction) [16]. Visual distraction, occurs when drivers look away from the road (e.g., when the driver configures a GPS device or sound system). Cognitive distraction occurs when the driver turns away from the control of the vehicle by being lost in thinking or doing other tasks unrelated to driving (e.g., talking on a hand-free cell phone, sending a text message or planning a route). Cognitive distraction alters the driver's ability to recognize targets across the visual scene and focus on the center of the driving scene [16].So, according to mentioned definitions, somnolence is mostly a critical state of hypovigilance, especially in a driving situation. Extreme fatigue can evolve into a state of drowsiness or distraction, and sometimes both. These phenomena may have the same disastrous effects and involve a decrease in the ability of the individual to drive, for example, the reaction time of the driver would be longer, which will increase the risk of accident [17].

Drowsy people often exhibit inherent visual characteristics distinguishable across the face, such as eye states, eye blinking and many other visual features [17]. So, analyzing the eye state is crucial for drivers drowsiness detection [17, 18, 19, 20]. Driver's fatigue strongly correlates with a PERC-

LOS measure [21]. However, knowing eye localization is determinant to recognize their states.

1.4 State of the Art

Driving an automobile is a very complex task that requires to perform several subtasks simultaneously. The driver must find his trajectory, monitor and regulate the vehicle speed, avoid obstacles (this task becomes more complex in urban areas), respect the road rules and control the vehicle. Under such conditions, it is clearly evident that the vigilance of the operator must be optimal.

Approaches for monitoring the driver behavior are categorized into two groups [22]: the driver-based and the driving-based methods. Their classification relies on the type of signal used to derive the pilot's alertness level. Driver-based method, including the physiological measurements, such as brain signals, ocular signals or heart rate signals. And the physical measurements, such as facial expression analysis. Driving-based method, includes the analysis of the vehicle behavior on the road, including steering activities, pressure exerted by the driver on the vehicle pedal and vehicle reactions to specific events (lane departure events).

The study of the driver's physiological signals

The physiological method for drowsiness monitoring is a quantification of physiological signal changes (brain wave signals, heart rate signals, gaze direction, and skeletal muscle activity). These physiological signals are derived by appropriate sensors, such as electroencephalography (EEG), electrocardiography (ECG), electrooculography (EOG) and electromyography (EMG).

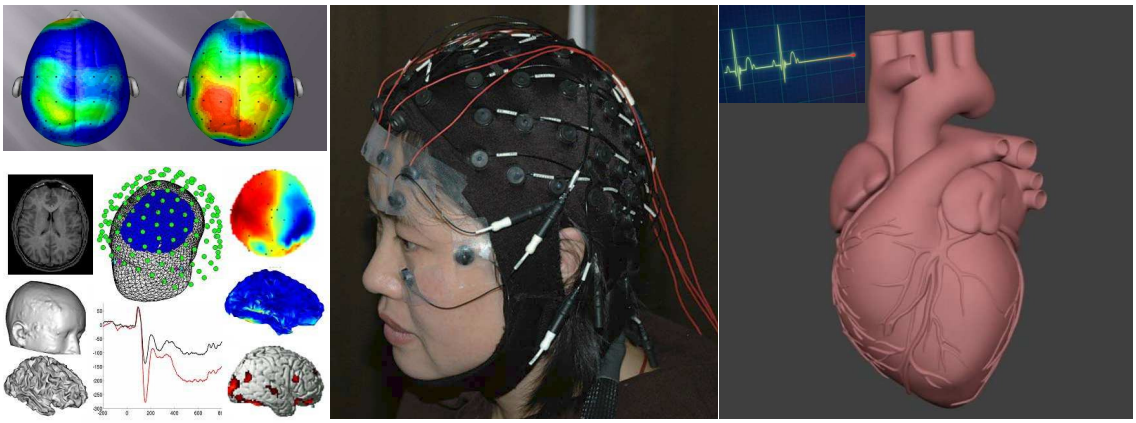


Figure 1.1: Physiological-based measures for drowsiness detection. In the midst, image shows a subject wearing the electrodes for recording both EEG and EOG signals [23].

The physiological approach produces fairly similar signals from one individual to another, e.g., EEG signals [24]. EEG measures the electrical activity of neurons through several electrodes placed on the scalp (Fig. 1.1). The frequencies of brain waves EEG [8, 25] are classified into four waveforms according to their frequency bands (delta, theta, alpha and beta). These components of the

EEG signal are closely related to the levels of drowsiness and help to interpret this particular state of the driver [25].

The EMG evaluates electrical activity recordings produced by skeletal muscles. The EOG detects drowsiness by measuring the corneoretinal potential in a specific area around the eye. In this area, a pair of electrodes are placed on the upper and lower part of the eye, or they can be placed on the left and right of the eye, as shown in Fig. 1.1 [23].

Shi et al. [26, 27, 28, 29] proposed various detectors that use EEG signals to recognize somnolent individual. Yu et al. [30] suggested a method for detecting drowsiness. The established approach extracts the rhythmic features of EEG signal by continuous wavelet coefficient method, then a learning model classifies these EEG features to decide whether the person is sleepy or not. Khushaba et al. [31] established a fuzzy-based approach combined with wavelet transform model to estimate the driver's drowsiness from a set of EEG, EOG and ECG signals. Zutao Zhang et al. [25] proposed a multilayer active safety system for detecting the pilot drowsiness and collision warning.

Ocular parameters are a reliable indicator widely used for detecting the driver drowsiness [21, 32]. Ma et al. [33] evaluate the impact of monotonous activities on the somnolence of the individual. The approach developed for this purpose uses the EOG signals for the interpretation of the physiological state of the person. Xuemin Zhu et al. [23] introduced an approach based on deep learning methods to detect drowsiness. Their approach combines the EOG signals and convolutional neural network (CNN). Clementine François and her team. [8] developed an automatic drowsiness estimator that combines physiological and physical indicators. Their system uses three ocular parameters: the mean duration of blinks [34, 35], PERCLOS measure [21] and percentage of micro-sleeps [8, 34]. Experimentations are validated in laboratory conditions that approximate the operational scenario. The approach proposed by Clementine François et al. has shown good results and can detect the drowsiness at different levels. The interpretation of somnolence by physiology is rather reliable because it gives a neuro-physiological definition of the driver state. However, they have several disadvantages,e.g., the EEG signals are very sensitive to artifacts and noise.

Physiological-based implementation for driver monitoring may be impractical in real-world conditions, because monitoring devices are invasive and inconvenient for the driver comfort. Recently some works were proposed to solve the intrusiveness problem. Jung et al. [36] proposed a non-intrusive drowsiness monitoring system based on biomedical signal measurement. Their approach uses the conductive electrodes attached on the steering-wheel to measure physiological signals from the driver. The physiological signals are obtained in a non-intrusive manner from the person during driving. The driver states (the degree of fatigue and heart rate) are monitored in addition to drowsiness, which is detected from the driver's ECG signals.

The study of the vehicle behavior

Monitoring the vehicle behavior may reveal mistakes indirectly caused by the driver's abnormal actions. Miscellaneous parameters are studied, such as force applied to the vehicle's pedals, speed variation, steering wheel movement and lanes keeping. A limited number of vehicle makes offer such optional systems for some models.

Today, a new automobile technology dedicated to driver safety is appearing and is being increasingly popularized. Going back at least 10 years ago, Toyota and some luxury vehicle brands like Mercedes, Lexus and Volvo launched their first Driver Monitor System (DMS). Volvo and Mercedes-

Benz (Fig. 1.2) have introduced systems that determine the condition of the driver, based on the response of the vehicle.

In 2008, Volvo designed the first device in Europe, to detect somnolence and alert the driver, which is called Driver Alert Control (DAC) [37]. It includes camera, several sensors and a computer hardware unit to manage different process. The camera constantly monitors the positioning of the vehicle in relation to the road markings (Fig. 1.3b). Embedded sensors record the movements of the car. The management unit stores collected information and calculates the risk of the vehicle's control loss. If the risk is considered to be high enough, the driver is informed by an audible signal. A text message and a coffee cup symbol, appear on the vehicle dashboard (information display) (Fig. 1.3a) and advising the driver to take a break.

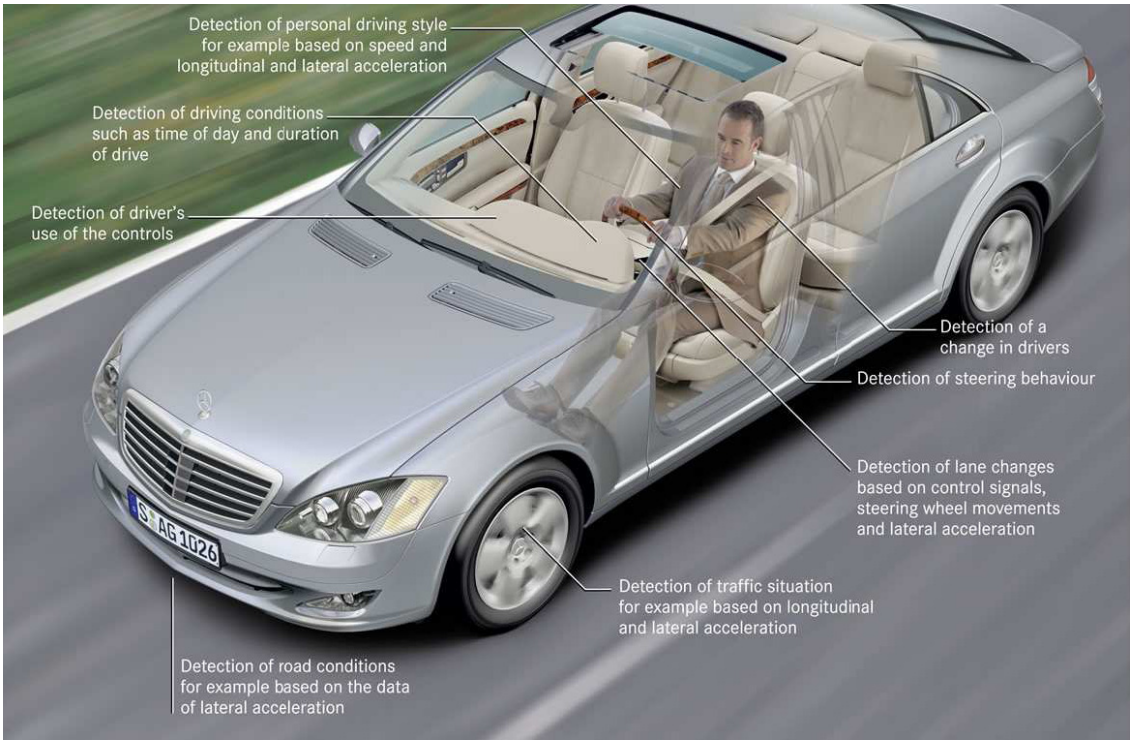
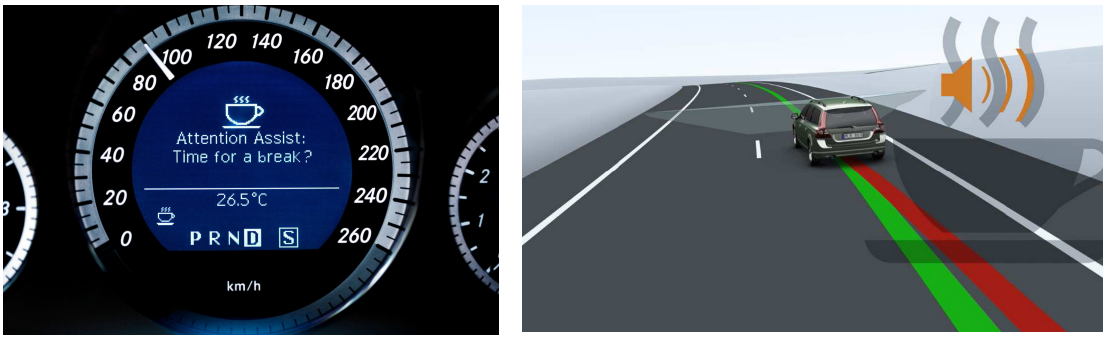


Figure 1.2: Mercedes-Benz's ADAS. (image from `https://www.mercedes-benz.com`.)

In the 2016, Lexus lunched a DMS with a lane keep assist, pre-collision system and all-speed dynamic radar cruise control. Lexus's DMS provides an alert to driver, if potential hazard is detected, e.g., the closing of driver's eyes or face appears to be turned away. It can also operate the pre-collision braking system. The system monitors driver and vehicle behaviors. In case of drowsiness, the system suggests to driver to take time to pull over for a break.

1.4.0.1 Lane Departure Warning and Lane Change Assistant

Lane Departure Warning (LDW) systems prevent road accident by expecting the vehicle uncontrolled deviations. The LDW tracks the car line keeping to maintain the driver's safety. The system provides



(a) (b)

Figure 1.3: Visual signal generated by DAC system in case of hypovigilance: (a) Visual alert (coffee cup and text message) displayed by the Volvo system on the vehicule dashboard. (b) monitoring of the car positioning on road markings.

an alert when a lane departure event is expected. Several LDW systems use computer vision approaches for tracking the marks on the roadway, by using a camera mounted on the rear-view mirror or on the car dashboard [38].

The camera is fixed for a frontal view of the road, with a wide range and an enlarged viewing angle. In case of the vehicle path converges to the track marking, LDW system may emit different alarms (auditory, steering wheel or seat vibration and visual form) [38].

The computer vision algorithms used in the LDW system must operate in real-time, under different atmospheric conditions. They must also ensure a good detection, while looking through a restricted field-of-view with a wide variety of markings including broken lines, continuous lines, double lines, white lines and yellow lines. These conditions increase the challenge for the road markings recognition task.

Therefore, one of the difficulty encountered by LDW systems, is the non conventional environmental conditions [38], such as wet asphalt, nighttime, lighting conditions, sunlight reflection, shadows, light-colored roadways, unmarked roads and damaged roads. Figure . 1.4 illustrates atmospheric conditions that the camera placed at the vehicle's outboard can face. Other implementations estimate that a vehicle will cross a lane boundary within a time threshold by analyzing non-imaging sensor information, including the vehicle speed, trajectory, kinematic data, and a model of the lane boundary.They can determine a time-to-lane crossing (TCL) [38, 39, 40, 41], which is the time duration available for the driver before lane-boundary crossing.

The LDW systems have been developed for a year. There is no progressive innovation stopped from expanding. Most recent LDW systems use TLC technology to provide an appropriate warning to the driver when the vehicle crosses the lane-boundaries. The LDW based on TLC generates a warning if the TLC condition is satisfied , i.e., a TLC value lower than a predetermined threshold. Saito et al. [22] emphasis that a non-exact value of the TLC threshold may increase the false alarms rate, which reduces the efficiency of the system and annoys the comfort of the driver. Therefore, the LDW based on the TLC estimate fails in a real driving scenario [22]. Wenshuo Wang et al. [43] proposed an LDW system based on the TLC estimate and a personalized driving model (PDM). Their framework uses predictive algorithms to estimate the trajectory of the vehicle to expect line departure events. Yuichi Saito et al. [22] proposed an adaptive driver assistance system, with two levels that

Figure 1.4: In the top row, monocular vehicle detection under challenging environmental and lighting conditions: detected vehicles are also labeled by monocular distance measure. In the bottom row, a transform from perspective view into a bird's eye view employed for monocular distance measures. [42]

operate simultaneously. The first stage monitors the departure event of the vehicle, and the second stage monitors the physical state of the driver.

The study of physical signals

The study of drowsiness from physical signals relies mainly on the treatment of the driver's video to measure the level of vigilance reflected by the change of the facial expressions. Drowsy people often exhibit some visual behaviors easily observable by changes in facial features, such as the eyes [46], the mouth and the head pose [47], as shown in Fig. 1.5.

There are other parts of the body that can be monitored by computer vision methods, to reveal the drowsiness state, such as the foot gesture and hand movements [48, 49], as shown in Fig. 1.6. Recent studies have shown that the ocular region is closely related to the level of vigilance. Percentage closure of the eye as a function of time "Percentage of Eye Closure" (PERCLOS) [21], is a widespread measure for the detection of somnolence of the driver. This measure determines the closure of the eyes corresponding to the somnolence [50]. The frequency of eye closure is also considered an effective indicator of somnolence as it allows the driver's micro-sleep periods [51] to be detected.

Thus, drowsiness is assimilated to visual changes in facial features, which are the consequence of the internal state change of the driver. The various facial features often monitored are analyzed in the following.

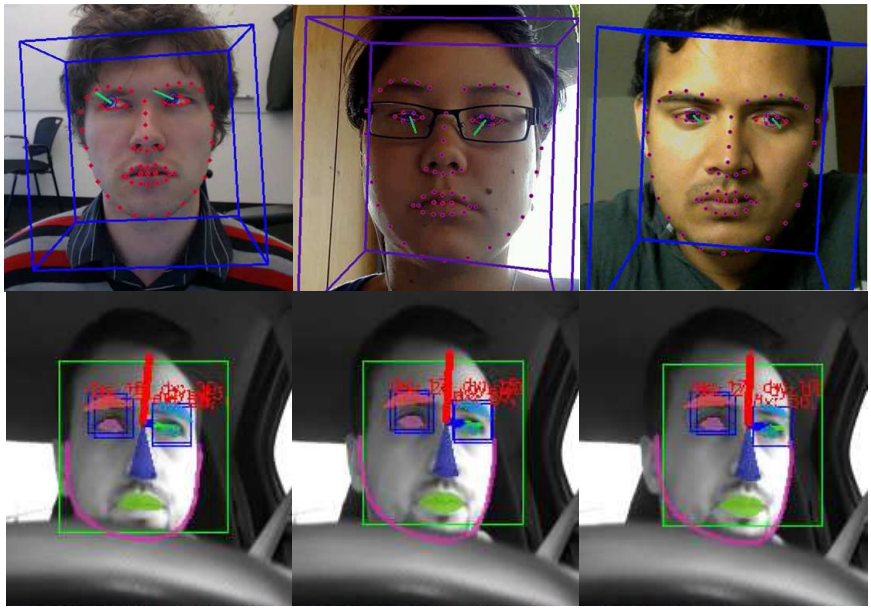


Figure 1.5: The top row shows some examples of gaze estimation on video sequences with open source facial behavior analysis toolkit (OpenFace) [44]. In the second row, driver monitoring system a real driving scenes, the video sequence is from HealthyRoad video set. [45] `http://healthyroad.pt/`.

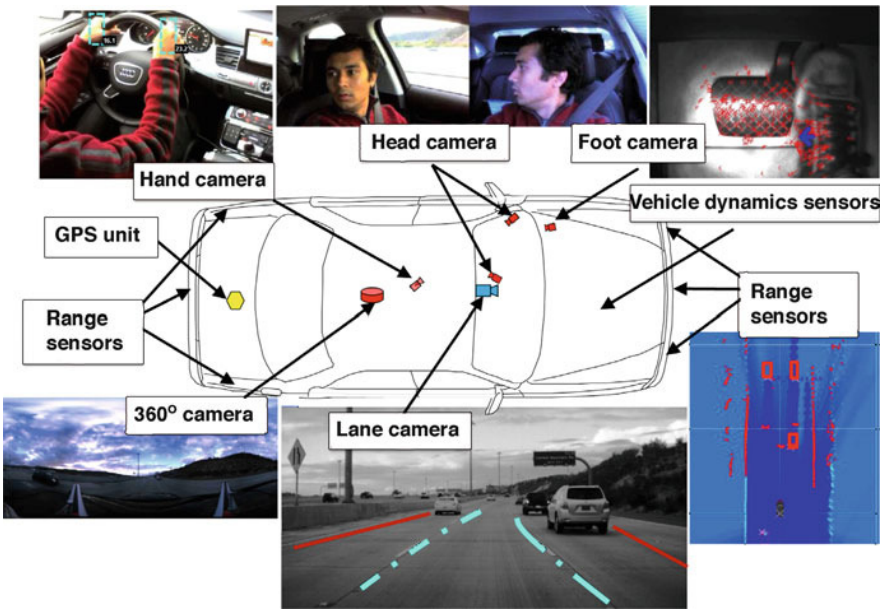


Figure 1.6: Scene understanding (vehicle cockpit and external environment of the vehicle), and monitoring of the driver behavior using several sensors for recording incoming and outgoing information. Driver awareness can be modeled by combining multiple information sources [42].

1.4.0.2 The analysis of opening and closing states of the eyes

Some methods detect driver drowsiness by counting the number of consecutive eye opening and closing sequences of the driver, from the monitoring videos recorded in the vehicle cockpit and captured with a low cost cameras (see Fig. 1.7).

Figure 1.7: Some of the employed monocular low-cost cameras.

Chao Sun et al. [52] proposed a real-time eye states recognition approach for detecting driver drowsiness. The algorithm comprises three steps: The first step consists of face detection using the Adaptive Boosting (AdaBoost) algorithm [53]. The second step aims to locate each eye in a given image. To do this, the ocular region (pupil localization) is defined by using the geometry of the face. Eye detection (step three) includes the Principal Component Analysis (PCA) approach, which extracts the eye characteristics and then recognizes the opening and closing of the eyes. At the end, PERCLOS measures the consecutive closing of the eyes through time to detect if the driver is drowsy or not. This approach was validated with thirty-two people, in a realistic driving scenario.

Jaeik Jo et al. [54] established a PS-DSM system that tracks the driver's behavior in real time. Their method includes five principal steps (facial features detection, head orientation angle, eye localization, and eye opening and closing detection). The eye states are detected by combining PCA and Linear Discriminant Analysis (LDA) approach, alongside a statistical characterization of the eye textures. Two methods were used sparseness and kurtosis of the projection histograms. Extracted features of the eye states, are used to fed SVM and distinguish an eye open from an eye closed in the input image. Obtained results of driver's eye states are interpreted by using PERCLOS measure to determine driver somnolence level.

Xinghua Sun et al. [55] implemented a driver monitoring system to alert the driver in case of drowsiness. This system uses three eye parameters (gaze direction, iris tracking, and eye states). The algorithm includes several stages, namely face detection, eye localization, iris tracking, and driver's eye assessment. Face detection is performed by the Viola-Jones algorithm. the location of the eyes is extracted from the facial area, which is improved by several preprocessing stages. Thereafter, the algorithm get the exact eye locations, extracts the iris locations and tracks them. The dynamic Bayesian network is used as the analysis tool to perform the reasoning of drowsiness.

Through the various works presented above, driver drowsiness detection systems based on image processing and machine learning methods proven their effectiveness for detecting dangerous behavior of the driver. However, some weakness of such systems must be highlighted in the following.

Driver's sleepiness is often related to the state of facial features, mainly eyes, mouth and head pose. However, to recognize the states of these features, they must be detected first. In this context, the detection of the face is a key and common step to facilitate the localization of its characteristics.

But, if the detection of the face fails, it would be impossible to establish the other steps, namely the location of the face characters and the recognition of their states. For example, the recognition of closing and opening states of the eyes is an excellent indicator of driver sleepiness and requires two preliminary steps, the detection of the eyes and their location.

Figure 1.8: Face and eye opening and closing monitoring at night under difficult environmental and imaging conditions: eye detection failure due to motion blur, one eye not located due to face rotation [42].

Face detection and its feature extraction are sensitive to the driver's environmental and imaging conditions, such as ambient lighting, strong light reflection, partial occlusion, head pose, image blur and noise (see Fig. 1.8). They significantly affect the system accuracy. They are often a challenge for any computer vision algorithm. In the following chapters all these steps are discussed in depth, with a solution provided at each stage to reach the final goal, which is correctly recognize the eye states (open and closed) under extreme conditions of the real world.

1.4.0.3 Eye Blinking Analysis

The eyes are the most expressive sign of a person's condition. Recent work has shown an efficiency of eye blink frequencies to detect sleepiness [56, 57]. In [58], blinking frequencies is employed to measure sleepiness levels. In drowsy state, the frequency of the driver's blinking may significantly change, as well as the duration of eyelid closing, which begins to extend involuntarily. More specifically, when the driver is fully vigilant, the frequency of blinking weakens and the closing time of the eyelids shows a slowness. It has been observed that the blinking frequency, increases exponentially with a duration of eyelids closure that becomes shorter and shorter. [59, 60].

1.4.0.4 Mouth states and yawning analysis

Yawning is an involuntary intake of the person's breathing, through a wide opening of the mouth. This state is usually a response of the human body to a physiological trigger state, such as fatigue or boredom.

Shabnam Abtahi et al. [61] proposed a driver's yawning detection approach for drowsiness detection. Their proposal aims to detect somnolence from the opening frequency of the driver's mouth. Their approach helps for background subtraction and can detect well facial features with invariance to skin color and lighting variations. However, it only operates under ideal conditions and remains sensitive to thresholding parametrization, and difficult angles of the head rotation. Nawal Alioua et al. [62] established a driver sleepiness monitoring system based on image processing and computer

vision methods. The overall scheme aims to detect drowsiness by counting the number of successive yawing state of the conductor, above a certain threshold, an alarm will be given to prevent hazards. Abtahi et al. [61] proposed an approach to locate and track the driver's mouth states, in real time by using a CCD camera mounted on the dashboard of the vehicle. The camera detects the ROI of the mouth from the face. Then, a mixture of a skin analysis method and LDA detects the mouth region, by separating skin pixels (face and lips) from non-skin ones (background). The detected mouth will constitute the input of a neural model to recognize the mouth states in three different ways, normal, yawning and talking state. Fan et al. [63] have also identified driver's yawning as an important physical sign to prevent drowsiness. They proposed an approach for tracking the mouth and recognize its state. Shabnam Abtahi et al. [64] developed a real-time driver monitoring system based on yawing frequencies. Their framework uses integrated intelligent camera platform (APEX^{TM}) for automotive vision systems, manufactured by CogniVue Corp. The system recognizes three mouth states (normal, talking or singing and yawning), to alert the driver when drowsiness is detected, by counting the number of yawning instances on the video stream. This approach got an accuracy of 60% for yawning detection, which is insufficient to ensure the driver's safety in case of somnolence.

Yawning is an excellent indicator of drowsiness, the mouth form may appear more larger, which facilitates its detection with a relatively high accuracy. However, if a driver speaks, screams or sings while driving, it will be hard to distinguish between these activities, where the three scenarios can lead to an open mouth and thus increase false alarms. Thus, yawning based drowsiness detection approaches, still suffer from a high false positive rate [65].

1.5 Thesis structure

This thesis is divided in six chapters, of which the present introduction is the first one. Chapter 2 presents a review of the state of the art in face detection and description of our approach for face detection. Chapter 3 describes approaches to eye detection and localization. This chapter focuses on our eye detection frameworks, and their performance tested on video sequence database. Chapter 4 introduces the EyeLSD algorithm, for eye localization and state detection. This chapter focuses on the EyeLSD's eye localization phase and evaluates its performance on different datasets. Chapter 5 presents EyeLSD algorithm to deal with the detection of the eye states (open and closed). Finally, chapter 6 contains the conclusions and main contributions of this work, and future research that may spring from it. A bibliography close this document.

1.6 List of publications

International journals

- **D.E.Benrachou**, F.N Dos Santos, B.Boulebtateche, and S.Bensaoula., ≪EyeLSD a Robust Approach for Eye Localization and State Detection≫, ***Journal of Signal Processing Systems***, vol. 90, no 1, p. 99-125, doi: https://doi.org/10.1007/s11265-016-1219-1, 2018, Springer Verlag.

- **D.E.Benrachou**, B.Boulebtateche, and S.Bensaoula., ≪Gabor/pca/svm-based face detection for drivers monitoring≫, ***Journal of Automation and Control Engineering (JOACE)***, vol.1, no.2, p. 115-118, doi: 10.12720/joace.1.2.115-118, 2013, IACSIT Press.

International Conferences

- **D.E.Benrachou**, F.N Dos Santos, B.Boulebtateche, and S.Bensaoula., ≪ EyeLHM: Real-Time Vision-Based approach for Eye Localization and Head Motion Estimation ≫, ***IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC'16)*** , Bragança, Portugal, 2016.

- **D.E.Benrachou**, B.Boulebtateche, and S.Bensaoula., ≪ Off-Line Driver's Eye Detection: Multi-Block Local Binary Pattern Histogram Vs. Gabor Wavelets ≫, ***1st International Conference on Automatic Control, Telecommunication and Signals (ICATS'15)***, Annaba, Algeria, 2015.

- **D.E.Benrachou**, F.N Dos Santos, B.Boulebtateche, and S.Bensaoula., ≪ Automatic eye localization; multi-block LBP vs. Pyramidal LBP three-levels image decomposition for eye visual appearance description ≫, ***Pattern Recognition and Image Analysis: 7th Iberian Conference (IbPRIA'15)***, Santiago de Compostela, Spain, 2015.

- **D.E.Benrachou**, F.N Dos Santos, B.Boulebtateche, and S.Bensaoula., ≪ Online vision-based eye detection: LBP/SVM vs LBP/LSTM-RNN ≫, ***11th Portuguese Conference on Automatic Control (CONTROLO'14)***, Porto, Portugal, 2014.

- **D.E.Benrachou**, B.Boulebtateche, and S.Bensaoula., ≪Gabor/pca/svm-based face detection for drivers monitoring≫, ***5th International Conference on Computer and Automation Engineering (ICCAE'13)***, Bruxelles, Belgium, 2013.

CHAPTER 2

Face detection

Contents

2.1 Introduction

Facial expressions convey the observable emotional state of the individual. It facilitates also non-verbal interaction between people. The analysis of facial expressions is an astonishing ability for the human being. For computer this task remains an open challenge, in spite of the efforts invested and results achieved to date. In computer vision, the comprehension of facial expression requires a multi-layer process implementation, including face detection, face tracking in a video stream, and facial traits extraction (nose, mouth, eyes). Face detection has wide potential applications, such as surveillance and facial recognition for biometric authentication of persons.

This chapter is devoted to driver's face detection under realistic conditions. First the fundamentals of object detection are introduced, then supervised and unsupervised learning approaches are

presented and discussed. Finally, a complete comparative study is made between the developed algorithm and most recent approaches for face detection.

2.1.1 Object detection and classification

In computer vision, object detection is a tedious and often complicated task. Detecting an object in a cluttered image can be revealed as an easy task and effortless for the human being, but very difficult for the machine. Object detection consists of identifying an element "target" in a digital image or a video sequence, using prior knowledge given by a series of features, also called attributes or parameters. This set of attributes, express the fundamental textures of an image, such as edges, lines, corners or their mixture. Several approaches can be employed to extract the desired object from its background. The most common approaches are based on shape, color or texture information of object. These constitute an identity specific to each object. Developing systems for object detection includes two principal steps: Feature extraction and classification.

Feature extraction with appearance modeling, aims to construct an appearance model using a set of features of a specific object. The model should represent a large range of visual appearance by minimizing the intra-class variations and maximizing the extra-class ones. However, inadequate feature representation, can give poor performance in the subsequent classification stage, which will fail to accomplish detection task.

Therefore, it is important to derive the optimal descriptor (set of features), which has to be immune to imaging and environment conditions. Scale and rotation are the main reasons for object detection failure. This challenges the search of an optimal descriptor. Classification is a subsequent step to feature extraction. It is used in the case of two or multiple objects in a scene, where it is important to precisely find the desired object in the input image. In the literature, there exist various feature extraction methods for object detection as well as classification methods. In this chapter we will focus on some of them.

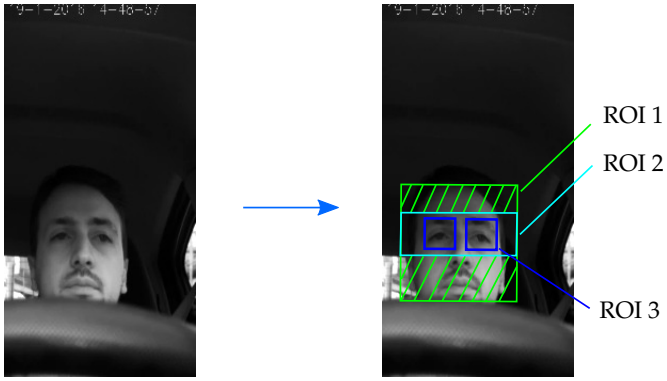
2.1.1.1 Combining Computer Vision with Machine Learning

The combination of computer vision and machine learning is an emerging discipline that has become popular for their adaptation in different vision-based applications. In computer vision and pattern recognition. Two main artificial learning techniques are employed: supervised learning and unsupervised one. In supervised machine learning, a set of data is defined by an expert in form of input-target pairs for the training. This is different from unsupervised learning, where no task-specific training signal is given. The algorithm attempts to estimate the data structure only by inspection. This chapter discusses a very popular unsupervised learning technique, namely principal component analysis (PCA) - Section 2.4.2, which is widely used in statistical pattern recognition [66], data reduction and feature extraction [66]. In supervised learning method, training subset S of the input pairs (x, y), where x is an element of the input feature vector X and y is an element of the target vector Y, and a disjoint test subset S', both drawn independently from the input-target distribution $D_{X\times Y}$. It is highly recommended to use validation subset, which is drawn from the training subset for validating the performance of the learned model. The learning task aims at minimizing a task-specific error measure E, which is iteratively calculated between what the learning model predicts and the actual target, the evaluation is performed on test subset. In parametric algorithms, such as neural networks and support

vector machine (SVM) - Section 2.4.3, the common approach to train model is to minimize training error. The ability of a learning algorithm to transfer the performance of the set of training data to the test subset, this phenomenon is called *generalization*, which is discussed later for neural networks and SVMs.

2.1.2 Detecting regions of interest (ROIs)

In this thesis, we focus on the problem of detecting driver drowsiness through eye closure analysis to improve road safety. However, for measuring the eye states, three important region of interests (ROIs) should be isolated first (as illustrated in Fig. 2.1).



(a) Settings the ROIs and the eyes are localized:
ROI 1- Face, ROI 2 Eye region, ROI 3 Eye localization

Figure 2.1: An illustration showing the importance of driver's face detection step (ROI 1), coarse estimation of the ocular region (ROI 2) and precise eye localization (ROI 3) [45].

Face and eye detection is an important step in many analysis systems [67, 68]. In this context, research has made much progress in model- and learning-based object detection methods. [69, 70]. Some popular face detection algorithms, including boosting-based detection with efficient use of integral image, Haar-like features and a cascade of weak classifiers, have defined high-performance systems. These approaches are discussed in subsection 2.3.4. The rough detection of the eyes (ROI 2) and their localization (ROI 3), are studied in the next chapter, where three algorithms were built for detecting and locating the ocular region of the driver under hard conditions of the real world. The next section describes the proposed driver's face detection algorithm.

2.2 Driver's Face detection

Driver assistance system must fulfill several requirements, such as monitoring the driver's behavior, while providing an optimal response, in terms of information processing speed and decision-making. The system must detect every suspicious behavior of the driver while anticipating hazard situations on the road. The ultimate objective is to prevent imminent crashes related to the driver fatigue and lack of vigilance.

Face detection is considered as a primary step to construct a driver's monitoring system, and is often required before detecting and analyzing facial features (mouth, nose, eyes) and for estimating head pose. The detection of the facial region, has been investigated for longer than four decades and includes several applications, such as driver drowsiness, person identification and expression analysis.

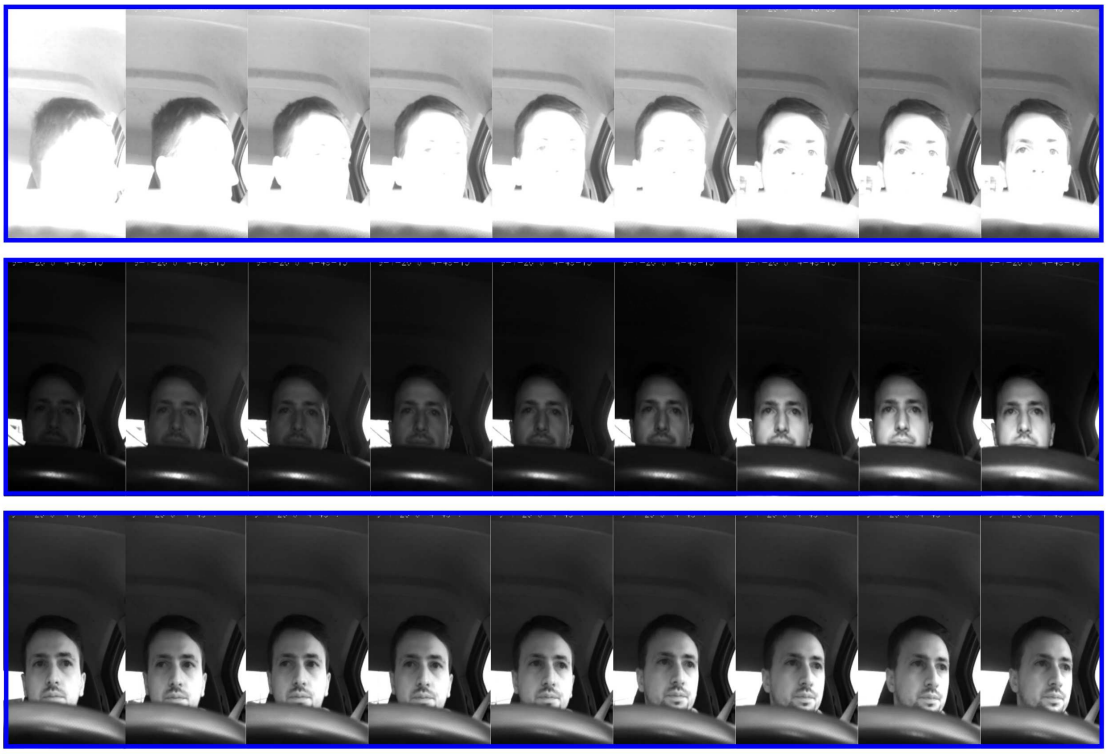


Figure 2.2: Examples of non-ideal conditions while driving, causing difficulties for driver's face and eyes monitoring (images from non-public video sequences provided by HealthyRoad company [45].

Some methods detect the face by combining its features, that is, the face is detected by establishing a spatial relationship between the eyes, nose and mouth. However, this type of technique generally detects each feature first and then adjusts the spatial constraint when all elements are detected. Face detection, can also be performed by a pattern matching approach. This method compares several face patterns with other regions of the image. The regions that correlate with the appearance of face have very low correspondence values. However, these methods are computational expressive, sensitive to view changes and ambient conditions. In addition, non-rigid facial textures include more variations, such as expressions (smiles, scream, laugh), and even lighting changes are accentuated in this case. These factors increase the detection difficulty.

Therefore, an explicit detection scheme is required to overcome these problems. Face detection is often dealt with statistical analysis and machine learning approaches, to build a robust method capable of distinguish faces from non-faces. For the design of a face detector, several problems have to be considered, which are presented below:

- Size: A face detector should be able to detect faces at different scales. This is usually achieved by using a multiple size of face images in the training set. Nevertheless, this process influences the reliability of the detection because small faces are very difficult to detect than large faces.
- Position: a face detector should detect faces at different positions within the image. This is done generally, by applying a sliding window method over the image, and computing features at each window that the system classifies it as face or non-face. The choice of the step size directly influences speed and accuracy of the detection system.
- Orientation: in real-world application, face appears in different orientations due to the camera angle and the head-pose. So, face should be detected in pitch, roll, yaw angels with respect to a frontal pose. The rotation problem is tackled generally by considering face images in different rotations, to construct the learning model.
- Expressions: facial expressions cause a significant change in the face appearance. A simple way to handle this change, is to consider face images with diverse expressions in the training dataset.
- Lighting and imaging conditions: the natural lighting conditions involve various changes, such as fluctuation in spectra, source distribution (different light sources can significantly modify the skin color) and intensity. Image conditions, including low resolution, blurring or missing texture details, can result in poor image quality. These cause an enormous challenge to the face and its characteristics (e.g., an eye) detection algorithms. The lighting conditions have an impact on the intensity of the pixels around the facial features and can fluctuate strongly in precise regions relative to others, such as the ocular region. This point plays an important role in ensuring high detection accuracy.
- Occlusions: partial occlusions represent a major challenge for most face detectors, since the face appearance may considerably change by the objects that the person wears (e.g., glasses, mask, hat), shadows and even when the person has a a beard or a mustache.

2.3 Related work

Face detection is considered as a key step in a driver monitoring system, as it primarily reduces the search field for facial features localization, e.g., the eyes, as shown in Fig. 2.1. The face detector must operate optimally under realistic conditions, while achieving high accuracy and maintaining its detection robustness, despite the environmental changes. The implemented face detector should be very fast, so that drowsiness can be detected before a disaster occurs. This section presents a brief review of the state-of-the-art approaches for face locating. In the literature face detection methods are classified into four categories [67, 68]: knowledge-based methods, invariant feature methods, template matching methods and appearance-based methods.

2.3.1 The knowledge-based methods

These methods require definition of rules to encode human knowledge of what constitutes a typically human face. The established rules should capture as many relationships as possible between the different face components.

2.3.2 The invariant feature-based methods

These approaches focus on the invariant facial features, including face components, texture, skin color and a their mixture. The aim is to find common structural features between faces in different ambient conditions. Face detection task can be fulfilled in different circumstances, such as poses and rotations.

Hashem Kalbkhani et al. [31] proposed a face detection approach that is inserted by initializing a rectangle in the input image (assumed to contain a face), this rectangle has a pre-established coordinates that roughly define the position of the face. Next, a skin-color algorithm is utilized for a binary transformation of the input image into black and white pixel values. The white pixels are the skin textures and the black pixels represent background. Connected component analysis is applied to overcome unwanted imaging conditions (noise, artifacts, color regions similar to skin color). Subsequently, a search method based on the sliding-window is integrated to define the ROIs (i.e., a face) according to the surface size delimited by white pixels. Yuseok Ban et al. [71] proposed a face detection method that use skin color likelihood and boosting algorithm to create a search map to better detect as face. A probabilistic approach is adapted to compute similarity between a color region and the skin color. So, based on the probability of skin color, authors enhance the color related to the skin and ignored the color that does not represent the skin (background). Local Binary Pattern (LBP) and Haar-like features are used to build cascaded classifiers. The boosted classifier is implemented, depending on the color of the skin, to locate the face in a color image. Their approach shows good invariance to changes in pose and cluttered backgrounds. However, occlusion causes a detection problem for skin color-based methods. This may fail to properly detect the face under these conditions.

2.3.3 The template-matching methods

The algorithm often used, analyzes an input image to find a face. It uses the sliding-window search method, which consists in defining a sub-image and at each iteration the algorithm calculates the correlation between the sub-image and a predefined face model. The algorithm can then approve or reject the similarity with a human face from a user-defined threshold. The detection accuracy may deteriorate. Because, the face patterns are different from one person to another. Detection is strongly influenced by the used face model, and imaging conditions. This type of method may be also costly in terms of computing capacity.

2.3.4 The appearance-based methods

These methods require a large capacity of face images with different variations such as, head pose, illumination, occlusion. When they are taken into account by the algorithm, they can reinforce the learning model and thus enhance the algorithm to handle diverse face appearance and fortify its ability to recognize faces with multiple variations, despite their absence in the learning database. This practice helps the learning model to be invariant to previous changes during the detection phase.

Face patterns are described through their visual appearance by combining extracted features and statistical classifiers. Generally, there is three steps to introduce; (1) preprocessing (image noise reduction, illumination correction, contrast enhancement), (2) feature extraction and normalization, (3) classification stage interprets the extracted features.

Appearance-based methods require that the input image be scanned at each location and at different scales. Thus, the number of test sub-windows can easily reach millions. However, the amount of scanned regions depends on the resolution of the input image and the used step offset parameters during the detection phase. These methods require a classifier with a high likelihood rate (detection rate), which must generate an extremely low false alarm rate. Chi Man Vong et al. [72] presented a rapid face detection algorithm based on appearance method. Face detection problem, is solved as a binary classification problem. The algorithm includes mainly two steps, principal components analysis (PCA) method to extract eigenface features, classification between face and non-face features is performed by sparse Bayesian extreme learning machine (SBELM) classifier, which is a neural-based method that combines extreme learning machine method and sparse Bayesian learning method. Searching for a face in the input image, is made by using a scalable sliding window-based method.

One of the most popular appearance-based face detector is that proposed by Viola-Jones [46]. This method scans the whole image to extract Haar-like features in overlapping rectangular areas. In the classification stage, Viola-Jones use a boosting selection of features that consists of several weak classifiers arranged in cascade, rather than using a single strong classifier. Each successive classifier is based on the rejection or acceptance result of the previous classifier.

Viola-Jones method gained popularity by its speed and robustness for face detection. It counts three principal advantages:

1. The integral image representation describes the image features (rectangle features) remarkably faster for being used in the node classifiers.

2. The cascade framework allows background patches to be filtered out quickly.

3. The AdaBoost classifiers is composed of several weak classifiers trained in form of cascade nodes. The AdaBoost algorithm is used to select the rectangle features and to combine them to form an ensemble classifier in a cascade node.

Despite its advantages, recent works have reported some weakness of Viola-Jones framework. Xiaohua et al. [73], highlight two disadvantages of the detector. First, during the training phase, the augmentation of the number of features can lead to a significant increase of the learning model complexity. Because in Adaboost classifier, each feature corresponds to a single weak classifier, a node of classifier corresponds to a learner combination of several weak classifiers. Second, if one of the class is not well covered in the training dataset, the classifier may not work well. Authors overcome these issues by selecting contextual features and combining two feature descriptors to represent the face at different scales. The image features are represented by a simplified Gabor features computed by means of integral images [74] with four orientations. A hierarchy of face regions of specific sizes is adopted, the classifier is trained with three hierarchical image levels, i.e., resolution of 18×18 pixels, 18×24 pixels, and 24×24 pixels. The 1^{st} level is constructed with a Haar-like features, The 2^{nd} and the 3^{th} levels use a simplified Gabor features. Gabor's simplified features perform better than Haar-like features. Face detector of [73] provides a high classification accuracies of 99.77% and 99.41% in FERET and BioID frontal face datasets, respectively. Their algorithm was only formed and tested with frontal face images.

However, although its excellent results. The mixture of simplified Gabor features and Haar-like features is not truly invariant to lighting changes, nor to pose and expression variations. Thus, the face detector may fail under these circumstances.

Viola-Jones fails with a sudden change in light and head rotation. Several works have been proposed to circumvent these disadvantages and improve the performance of the Viola-Jones detector may be cited [75, 76, 77].

2.4 Face detection with SVMs

In this section, we present our algorithm that detects face under real world conditions. The implementation is not optimized and was developed with Matlab environment, in intel Core2Duo^{TM}CPU based laptop system at 2.2 Ghz and 4Gb memory specs.

- **General scheme of the system**

Face detection algorithm is divided into three essential steps (as shown in Fig. 2.3). The first step pre-processes the raw input image, which significantly reduces image noise while improving texture quality. The second step extracts the relevant features in the predefined keypoints on the image. The last step uses an advanced classifiers to interpret the information collected at each keypoint.

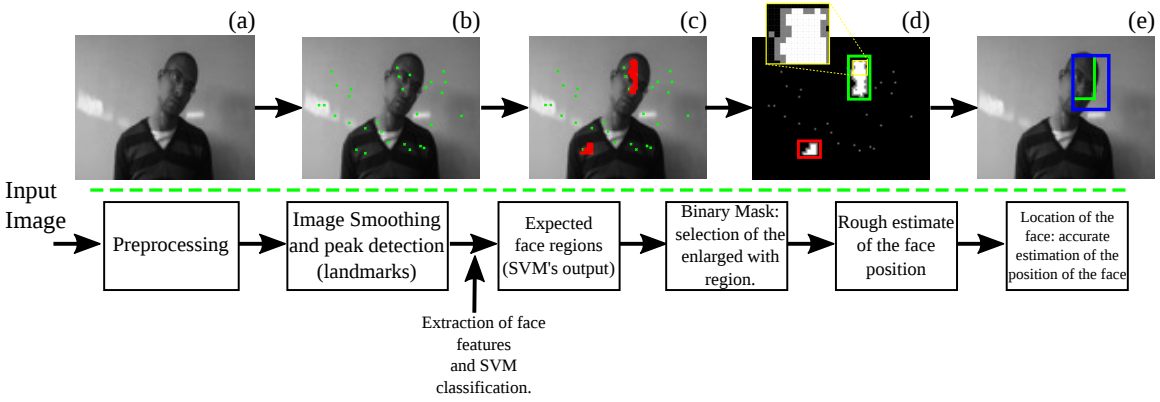


Figure 2.3: The proposed method for face detection. In the experiments, the proposed method is evaluated in face images. For clarity, we only show the search map with keypoints in step (b). In step (c) the Gabor wavelets encode the local regions of the face regions. Step (d), applies a region selection method, the white region of binary image represent the face while black region represents the non-face. In step (e), green and blue rectangles represent the results of the proposed method.

In stage one, three-channel RGB image is converted into gray-scale image Υ of 100×80 pixels, then image processing techniques are applied in original image. The features of face regions become more salient after pre-processing step, due to noise filtering out and texture enhancement. The pre-processing step consists of a normalization of the illumination that compensates for the variations of low-frequency lighting and suppresses the noise with a Gaussian filtering.

The outcomes of these preliminary steps create a search map with keypoints that highlight different face features and some background regions. This idea has been stressed by Roe1 Hoogenboom and Michael Lew [78] and successfully applied for face detection.

To minimize the quantity of regions to be analyzed by the SVM model, we must find the invariant characteristics of the face. This will greatly reduce the number of regions, while increasing the

chances that SVM scans the correct area in Υ . The facial structure is generally similar from one person to another. The eyes correspond to the dark areas (regions with small gray scale) and the nose shows strong reflection. The search map is set up by smoothing the input image and getting local maximum of pixels, whose intensity is higher than their neighbors intensity values. Local maximum of a face image, highlights the high-intensity region (e.g., due to nose). The processing step acts in the image thus bypassing the overlord generated by conventional sliding-window scheme.

In the first step, Fig. 2.3(a), uses the search map with the keypoints. The image is first blurred which increases the robustness of the feature extractor to image noise because the Gabor features are less sensitive to noise but not invariant [79]. The next step consists of extracting local maximum regions (peaks) from the preprocessed image, which are the regions of high-intensity. Fig. 2.3(b) These regions are the most likely places to contain the right face location, where the intensity of pixels is higher than the intensity of regions in their neighborhood. Connected-component labeling is used for detecting connected regions in peaks, while assigning them as landmarks or keypoints. Fig. 2.3 In the first step, Fig. 2.3(c), the Gabor wavelets are used to encode local features with forty Gabor filters in five scales and eight orientations (as shown in Fig. 2.4). The size of the sub-image used in our experiments is 27×18 pixels. The Gabor filter approach has shown low sensitivity to noise, small translation range, texture rotation, and change in scaling.

The second step, Fig. 2.3(d), is to apply a region selection method. To retain the most discriminated image parts that appear as a facial region, spatial structure of objects in a scene is used besides of binary morphological operations. Two phases constitute the morphology step, erosion and dilation [80]. The white region (pixels) of binary image are expanded, the black region (pixels) are diminished by erosion operation. Afterward, black region (pixels) of the area diminished by erosion operation, is expanded by dilation operation. This second process of sequentially erosion and dilation is called opening process. After the opening process applied two times, largest blob is generated for a face (enhanced area withing the green bounding-box, as shown in Fig. 2.3(d)), small blobs are generated for noise then rejected (false alarms, set within the red bounding-box, as shown in Fig 2.3(d)). The stage two, Fig. 2.3(d), measures the enhanced area structural proprieties (retained white pixels), applied upon binary image for choosing the widest discriminated surface. This surface represents the facial region. In last stage, Fig. 2.3(e), reports-back a bounding-box on the original image with a size of 30×20 pixels. The next subsections, explains in detail the implementation of the face detection algorithm.

2.4.1 Gabor wavelet transform

- **Theory of Gabor wavelet transform**

The proposed face detection method includes two important steps: the extraction of the main features of the face and their classification. The feature extraction step is to collect salient information about the face patterns. The classification stage consists of classifying these features into two categories: face and non-face. In the first step, Gabor features are implemented to extract the salient features of the input images. Gabor's 2-D filters are theoretically interesting for image interpretation thanks to their excellent computational properties.

Gabor's features are robust to texture rotation, scaling and translation. The descriptor also provides high tolerance to the photometric disturbances, such as changes in lighting and image noise.

All of these reasons make Gabor's representation one of the best image descriptors in terms of performance. Gabor's filters are widely applied in different studies, such as object recognition [81], face detection and recognition [82] and iris recognition [83].

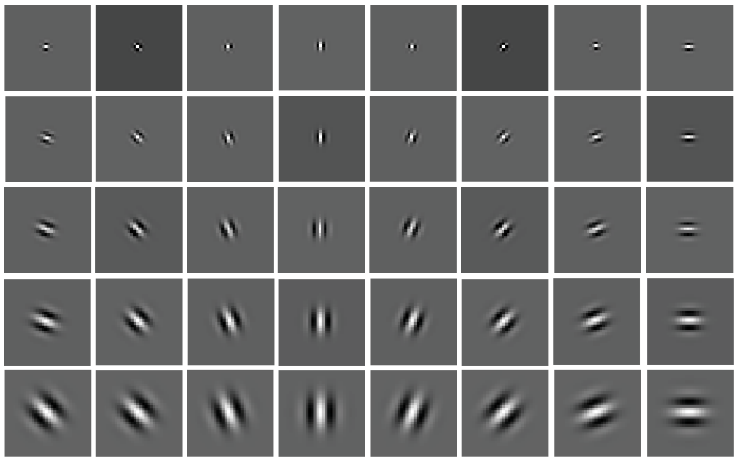


Figure 2.4: Gabor filters of five different scales and eight different orientations [18, 82].

The two dimensional Gabor is capable of obtaining micro features (e.g., facial expressions). The descriptor's kernel is similar to the response of the two-dimensional receptive field profiles of the human being simple cortical cell [84].

Gabor's filters represent face textures in different center frequencies (scales) and orientations, i.e., face textures are simultaneously represented in the spatial and frequency domain. Thus, Gabor representation allows an effective extraction of the visual appearance of the face that appears at different sizes and locations. In the spatial domain a two-dimensional Gabor wavelet based on Gaussian kernel function modulated by a complex sinusoidal plane wave, described as:

$$G(x, y, \omega_0, \theta) = \frac{1}{2\pi\sigma^2} e^{-\frac{1}{2\sigma^2}(R_1^2+R_2^2)} \times [e^{i(\omega_0 R1)} - e^{-\omega_0^2\sigma^2/2}] \tag{2.1}$$

where R_1 and R_2 are

$$R_1 = xcos\theta + ysin\theta$$
$$R_2 = -xsin\theta + ycos\theta$$

The standard deviations of the two-dimensional Gaussian function are σ_1 and σ_2 along the axis $x-$ and $y-$, respectively. x and y are the pixel coordinates in the spatial domain, ω_0 is the spatial frequency and θ is the orientation of the filter bank.

For a given input image I, the response of the Gabor filter G is calculated as the convolution of $G(x, y, \omega_0, \theta)$, with image I as follows:

$$C_{\Psi I} = I(x, y) * G(x, y, \omega_0, \theta) \tag{2.2}$$

where $I(x, y)$ is the intensity value of the input image I at (x,y), is the convolution. The response $C_{\Psi I}$ consists of real response or/ and imaginary response.

In our application of Gabor filters, the two filter parts are used, the real and imaginary response of $C_{\Psi I}$ for extracting the facial features. Let $C_{\Psi I}^{re}$ and $C_{\Psi I}^{im}$ are the real and the imaginary components of Eq. 2.2, respectively.

The mixture of both terms represents the information of the facial local textures, and can be derived with the amplitude response:

$$C_{\Psi I}(x, y, \omega_0, \theta) = \sqrt{\|C_{\Psi I}^{re}\|^2 + \|C_{\Psi I}^{im}\|^2} \tag{2.3}$$

The convolution process in Eq. 2.2 is efficiently implemented by fast Fourier transform (FFT), element-by-element multiplication and inverse Fourier transform (IFFT). The Gabor magnitude response represents the face image at different scales (σ_0) and orientations (θ). Each element of the face features, is normalized to zero mean and unit variance (ρ is the down-sampling factor). Finally, all responses are concatenated to form a unique feature vector to represent a single face image. The final feature vector will be expressed as follows:

$$C^{(\rho)} = [\mathbf{c}_{\Psi I}^{(\rho)}(x, y, \omega_0^1, \theta^1), \mathbf{c}_{\Psi I}^{(\rho)}(x, y, \omega_0^1, \theta^2)), \ldots, \mathbf{c}_{\Psi I}^{(\rho)}(x, y, \omega_0^1, \theta^m), \ldots, \mathbf{c}_{\Psi I}^{(\rho)}(x, y, \omega_0^n, \theta^m)]^T \tag{2.4}$$

In conducted experiments, the Gabor filter bank has eight orientations and five radial center frequencies (scales) [82] , $\sigma_n = \frac{\pi}{2\sqrt{2^2}}$ with $n \in \{0, 1, \ldots, 4\}$ and $\theta_m = \frac{\pi}{8}m$ with $m \in \{0, 1, 2, \ldots, 7\}$. It should be noted, that the only downside of Gabor traits is their large dimension. The classification of these large features will be computationally costly. Therefore, it would be more appropriate to introduce an additional step before classification, which consists of projecting the Gabor features from a large dimensional space to a smaller one.

2.4.2 Feature reduction

Gabor features set in section 2.4.1 resides in a large multi-dimensional space, where $C^{(\rho)} \in \Re^N$. Classification in large multidimensional space is not efficient, because the training set may include noises, insufficient details, a strongly correlated features (information is strongly redundant) and the number of training samples cannot match the dimensionality of the data to attain desired accuracy [85]. Moreover, computational time to learn high dimensional data is prohibitively high. Principal Component Analysis (PCA) aims at data transformation from high-dimensional feature space to low-dimensional one using projection basis which is optimal in terms of mean-squared error.

Orthonormal projection basis is derived by identifying the eigenvectors of the covariance matrix defined as:

$$\Sigma_C = \frac{1}{M}\sum_{n=1}^{M}(C_n^{(\rho)} - \mu)(C_n^{(\rho)} - \mu)^T \tag{2.5}$$

where $C_n^{\rho} \in \Re^N$ is an image inside the training set $n \in \{1, 2, \ldots, M\}$ and μ is the average image of the training set, $\mu = \frac{1}{M}\sum_{n=1}^{M}(C_n^{(\rho)})$. Equation 2.5 zero out the mean of the feature vector C^{ρ} and $\rho = 1$, which means no downsampling is made during feature extraction stage. PCA factorizes the covariance matrix into the following form:

$$\Sigma_C = \Phi\Lambda\Phi^T \tag{2.6}$$

where $\Phi \in \Re^{N \times m}$ is the desired orthonormal projection basis associated with non-zero eigenvalues, $\Lambda \in \Re_{m \times 1}$. The lower dimensional data is derived from linear projection as follow:

$$\Upsilon = \Phi^T (C - \mu) \tag{2.7}$$

The lower dimensional feature vector $\Upsilon \in \Re^m$, $m < N$, captures the most relevant features of the original data C important for the classification stage.

2.4.3 The support vector classifier

Support vector machine (SVM) is a widespread method frequently applied in industry and academia to solve a real-world problems, such as fault detection, object classification, object detection and pattern recognition. SVMs were first proposed by Cortes and Vapnik [86] and well known for their robustness in solving diverse classification problems.

The SVM approach intends to enlarge the margin boundaries between linearly separable classes. A large margin classifier (i.e., SVM) determines a particular solution, in which the solver classifies different patterns by widening the separation boundary between two or more classes. The classification boundary is defined by the largest separating margin that contains no sample that can be plotted around the decision boundary, as shown in Figure 2.5.

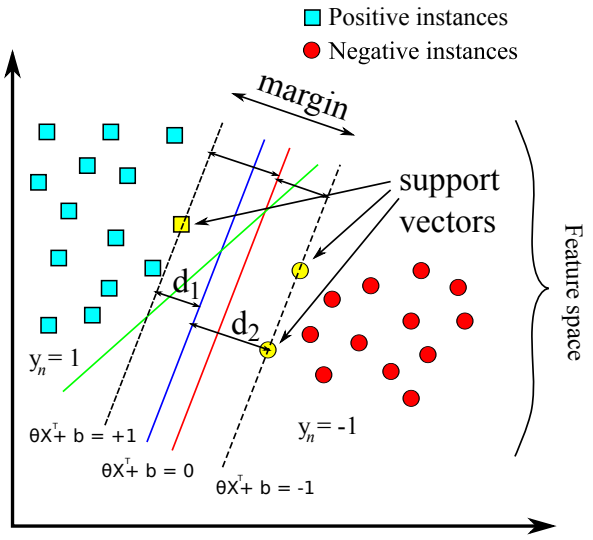


Figure 2.5: SVM representation: margin definition in case of a 2D feature space, $d_1 = d_2$ for the two symmetric lines defined by -1 and +1 [42, 86].

The data samples are represented as individual points in n-dimensional space, where n is the number of considered features. Each data items is a point in $\Re^n$. The SVM method can be seen as an optimization problem that seeks to find a linear classifier $g(x) = \theta^T X_i + b$ by minimizing a cost function. The vector θ is the *normal* of the hyperplane, and real number b is the hyperplane's offset from its origin (along the normal θ).

So, by assuming a set of training examples X_i, where $i = \underbrace{1 + \cdots + N_s}_{N_s}$ and N_s is the number of training samples. (which is not augmented with an extra element) for each sample a label is assigned $y_i \in \{1, -1\}$, indicating membership of each example for a specific class. The linear classifier $g(x) = \theta^T x_i + b$ is sought, such that:

$$\begin{cases} \theta^T x_i + b \geq 1 & if \quad y_i = +1 \\ \theta^T x_i + b \leq -1 & if \quad y_i = -1 \end{cases} for \quad all \quad i \tag{2.8}$$

These two constraints can be rewritten into a single inequality as follows:

$$y_n(\theta^T x_n + b) \geq 1 \tag{2.9}$$

The gradient vector of $g(x)$ is θ. Therefore, the square of the margin is inversely proportional to $\|\theta\|^2 = \theta^T\theta$. by minimizing $\|\theta\|^2$. Using *Lagrange multipliers*. The constraints 2.9 can be incorporated into the minimization Eq. 2.10.

$$L = \frac{1}{2}\|\theta\|^2 + \sum_{i=1}^{N_s} \alpha_i(y_i[\theta^T x_i + b] - 1), \qquad \alpha_i \geq 0 \tag{2.10}$$

So, the maximization of the margin is performed by minimizing L according to θ and b , and maximized according to the *Lagrange multipliers* α_n. The partial derivatives of Eg. 2.10 with respect to θ and b to zero results in the constraints:

$$\begin{gathered} \theta = \textstyle\sum_{i=1}^{N_s} \alpha_i y_i x_i \\ \textstyle\sum_{i=1}^{Ns} y_i \alpha_i = 0 \end{gathered} \tag{2.11}$$

The so-called *dual form* of this optimization problem can be reformulated by reintroducing Eq. 2.11 into Eq. 2.10

$$L = \sum_{i=1}^{N_s} \alpha_i - \frac{1}{2}\sum_{i=1}^{N_s}\sum_{j=1}^{N_s} y_i y_j \alpha_i \alpha_j x_i^T x_j, \qquad \alpha_i \geq 0 \tag{2.12}$$

L should be maximized with regard to the α_i. This is a quadratic optimization issue, for which regular software package are accessible. In this dissertation, the SVM implementation is performed with an integrated program for support vector classification library LIBSVM [87].

This interpretation of the SVM classifier is of restricted applicability, and deals with only a linear classification issue, i.e., a set of data of different classes that can be distinguished correctly with linear decision boundaries. However, in the real world, data are seldom linearly separable and this is due to the presence of noise and a very high dimensional data space. Therefore, SVMs may have a non-linear kernel that is set up in the hope of improving the linear separation of training data. The discriminant function introduced in the quadratic form,

$$\delta(x) = [x \quad 1 \quad x_0^2 \quad x_1^2 \ldots x_{N-1}^2 \quad x_0 x_1 \quad x_0 x_2 \ldots x_{N-1} x_N]^T \tag{2.13}$$

$$g_k(\delta) = \theta_k^T \delta(x) \tag{2.14}$$

Dataset is transformed from the measurement space to a new feature space. In Eq 2.12 training examples are coupled to other examples by an inner product. In other terms, the nonlinear kernel, K is built from an inner product in a feature space based on some mapping δ.

$$\delta(x_i)^T \delta(x_j) = (x_i^T x_j + 1)^2 = K(x_i, x_j) \tag{2.15}$$

The polynomial degree can be increased, so, instead of $(x_i^T x_j + 1)^2$ any integer degree can be implemented $(x_i^T x_j + 1)^d$ for $d > 1$. However, the most usual degree of polynomial is two, i.e., $d = 2$ (quadratic). A larger degrees tend to overfit the SVM model and make it instable.

The relationship between the features vector x to be classified and the classifier Θ can be rewritten as,

$$g(x) = \theta^T \delta(x) = \sum_{i=1}^{N_s} K(x, x_i) \tag{2.16}$$

So, the formulation above allows to replace the inner product by a more general formation expressed by the *kernel*. Besides to the polynomial kernel, another nonlinear kernel that allows an adaptive (nonlinear) decision boundaries is the Gaussian kernel which can be implemented with $\sigma^2 I$ as weighting matrix, radial basis function (RBF) kernel can be formulated:

$$K(x_i, x_j) = exp(-\gamma \|x_i - x_j\|^2) \tag{2.17}$$

and

$$\gamma = \frac{1}{\sigma^2}, \qquad \gamma > 0 \tag{2.18}$$

Then, kernel equation 2.17 is introduced in the question L, which is adapted to a nonlinear classification case. For small values of γ, the RBF kernel generates nonlinear classification boundaries, whereas, high values of this parameter give a smooth classification boundaries.

Inequality formulation (hard constraints) in Eq. 2.9 is replaced by soft constraints:

$$\begin{cases} \theta^T x_i + b \geq 1 - \xi_i & if \quad y_i = +1 \\ \theta^T x_i + b \leq -1 + \xi_i & if \quad y_i = -1 \end{cases}$$

Consequently, the optimization problem is changed into,

$$L = \frac{1}{2}\theta^2 + C \sum_{i=1}^{N_s} \alpha_i (y_i [\theta^T x_i + b] - 1 + \xi_i) + \sum_{i=1}^{N_s} r_i \xi_i, \qquad \alpha_i, r_i \geq 0 \tag{2.19}$$

Here, α_i and r_i are the Lagrange multipliers. ξ is a slack variable, $\xi_i \geq 0$. SVM is a quadratic programming problem, which aims to find an optimal hyperplane by minimizing the misclassification error, with respect to θ and b for a given set of labeled samples $[x_i, y_i]$. For $i = 1, \ldots, m$, where m is number of samples. Generally, this is a non-trivial mathematical issue, and there are various approaches that are characterized by accuracy and complexity. (e.g., by using quadratic programming)

Training an SVM model with an RBF kernel induces a maximization of ξ by combining it with θ^2. The parameter C establishes a sort of compromise that determines an equilibrium between having a large overall margin, at the cost of more training samples incorrectly classified or a small margin with less training samples incorrectly classified. The hyper-parameters C and σ of the RBF kernel can be

optimized by using grid search technique and the cross-validation group (CV). The best values over the CV group can then be used to build the learning model.

2.4.4 Training phase

2.4.4.1 Database description

(a) **(b)**

Figure 2.6: (a)."ORL Database of Faces"(Olivetti Research Laboratory, Cambridge) [88, 89], "AT & T laboratories Cambridge," Cambridge university computer laboratory, "the digital technology group", 1992-1998. (b)."CMU Pose, Illumination, and Expression (PIE) database" [90, 91], "THE ROBOTICS INSTITUTE", Carnegie Mellon university.

To analyze the face detection rate, two datasets were used:

- ORL database (Olivetti Research Laboratory, Cambridge) [88, 89], Fig. 2.6(a), consists of an assortment of face images, generally used in the context of face identification. This database is composed of various images of 40 distinct subjects. Images are captured under varied situations, fluctuations in illumination and facial grimaces (eye open / closed, smiling / not smiling), with presence or absence of particular elements (glasses/ no glasses).Images are taken with a dark homogeneous background and ten different head poses of each individual.

- CMU-MultiPIE (MPIE) database [90, 91], Fig. 2.6(b), is a CMU pose, illumination and expression dataset. It consists of various human faces, assembled between October and December 2000. It has 41 368 images of 68 people. Each individual's image is taken under 13 various light circumstances and with 4 distinct emotions, such person with neutral face, smiling,

eyelid blinking and talking.These specific expressions are assumed to be the most frequent 'expressions' in normal life.

Here it should be pointed out that the image sets of face and non-face images, used during our experiments are geometrically normalized into a size of 27×18 pixels.

2.5 Results and discussion

This section assesses the performance of the proposed face detector, principally, based on Gabor's features, PCA, and SVM classifier. Two sets of data were used for the evaluation protocol, namely the ORL Face database [88, 89] and the CMU Multi-Pie database (MPIE) [90, 91]. To measure the generalization capability of the algorithm, a new dataset of face images was acquired by using a normal laptop web-camera. In Gabor-based face detector, the face location is formulated as a 2-class classification problem, a state-of-art classifier was used, namely SVM classifier that is trained with linear and nonlinear kernels, to define the hyper-plane that maximizes the separation gap between the two classes (face and non-face). For a well benchmarking, the face detection algorithm is compared against most recent state-of-the-art appearance-based face detectors.

2.5.1 Classifiers and Parameters Settings

The face detection was treated by developing a model based on face appearance and a binary classification between 2 classes: the class of face and the class of non-face. Two classifiers were selected, linear kernel-based SVMs and non-linear kernel-based SVMs (RBF kernel) [86] were trained for maximizing the separating margin between the face and the non-face, while minimizing the number of error of the training set. The SVM's hyper-parameters C and σ of the RBF kernel are set during the conducted experiences. The cross-validation techniques CV is also implemented, the best values of the CV group were used to design the learning model.

- **Performance metrics**

Confusion matrix is used to measure the quality of the classification system, Fig. 2.1. Each column of the matrix corresponds to the number of occurrences of an estimated class, whereas, each row represent the number of occurrences of the real class. The interest of the confusion matrix is to allow a quick and simple visualization whether the system succeeds to realize a correct classification of the instances.
Table . 2.1 determines if in each studied image, the face, is correctly recognized. The columns present the values estimated by the algorithm presented in Section 2.4, while the lines correspond to the ground truth.
The values presented in the Table . 2.1 are explained by the following points:

- True Positive (TP): Face image and the algorithm detected it as Face.
- False Negative (FN): Face image but the algorithm detected it as Non-Face.
- False Positive (FP): Non-Face image and the algorithm detected it as Face.

		Estimated Class: 0	Estimated Class: 1
Observed Class	0	True Negative	False Positive
	1	False Negative	True Positive

Table 2.1: Confusion Matrix of face classification.

- True Negative (TN): Non-Face image and the algorithm detected it as Non-Face.

- **Statistics extracted from the confusion matrix**

The performance validation of the face detector and the final results are presented, including calculation of TP, TN, FP, and FN. The overall accuracy (Acc) for each experiment is reported in Table 2.2. The measurements used for evaluating the learning algorithm quality are the Receiver Operating characteristic (ROC) curves and the Area Under Curve (AUC) that is computed to show the probability of correct discrimination between different classes. Figure 2.7 show the AUC of each method's ROC graph tested on the test databases.The AUC metric is calculated using the *10-fold cross* validation technique. *Precision* (Prec) and *Recall* (Rec) are computed.

$$Prec = \frac{TP}{TP + FP}$$

$$Rec = \frac{TP}{TP + FN}$$

Thanks to those measures, $F_1 - Score$ metric is interpreted as a harmonic mean of the precision and recall for further comparison of the results.

$$F_1 score = 2 \times \frac{Prec \times Rec}{Prec + Rec}$$

If the learning model is predicting a positive class, a high recall and a low precision are mostly obtained, in contrast, if the trained model has high precision and low recall, it predicts a negative class most of the time. Consequently, a highest precision and recall values are often desirable to be obtained simultaneously by the training model.

2.5.2 Experimental Results and Discussions

This section presents the performance realized of our face detection algorithm. Two public databases were used to test our face detector: ORL database (Olivetti Research Laboratory, Cambridge) [88]

Method	**Database**	**TP [%]**	**FP [%]**	**TN[%]**	**FN [%]**	**Prec**	**Rec**	F_1Score	**Acc [%]**	AUC
Gabor/SVM(linear)	ORL	72.06	2.51	17.87	7.54	0.96	0.90	0.93	89.94	0.96
Gabor/SVM(linear)	MPIE	73.50	2.56	18.23	5.70	0.96	0.93	0.94	**91.96**	0.95
Gabor/SVM(RBF)	ORL	71.23	3.35	20.11	5.31	0.95	0.93	0.94	91.34	0.94
Gabor/SVM(RBF)	MPIE	72.02	2.77	21.05	4.15	0.96	0.94	0.95	**93.07**	0.97

Table 2.2: Statistical results for linear and RBF kernel in face detection.

and CMU-MultiPIE database [90]. The face detector employs forty Gabor filters in five scales and eight orientations, as shown in Fig. 2.4. Table 2.2 shows, the performance of the proposed method to detect face window on precedent datasets. Examples of successful detection of face windows from our database are shown in Fig. 2.8. The best results of 93.07% have been obtained with RBF-based SVM kernel on the CMU Multi-PIE face database.

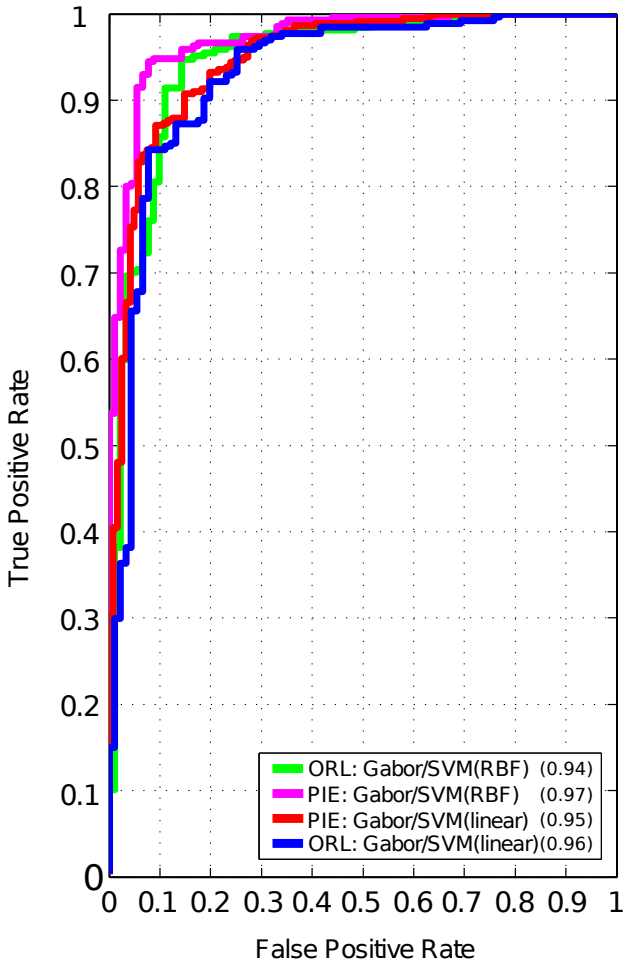


Figure 2.7: ROC curves of Gabor features and PCA using the SVM classifier ORL database and CMU MultiPIE (MPIE) database. AUC values are given at the end of corresponding legend texts.

Figure 2.8: Snapshots illustrate some successful face detection on pictures captured within the laboratory: frontal series, alternative expression (smiling, eye closed), illumination change, head rotation, occlusion (beard, myopia glasses), and distance.

On the other hand, we tested the method on a subset of the ORL database with variations. It achieved 91.96% successful detection rate, which means that the method is robust for these variations. There is clearly an improvement in detection rate using the CMU Multi-PIE face database. Compared to the ORL database, the CMU Multi-PIE face database, contains many more variations of lighting and head pose, as shown in Fig. 2.6(a) and Fig. 2.6(b).

At first glance, this improvement in the detection rate is due to the fact that Gabor's filter is a powerful feature descriptor, especially for non-rigid face textures [82]. In addition, by including other variations, a significant improvement in the detection accuracy can be achieved. However, it should be emphasized that there is a loss of information in the process of quantizing the Gabor feature vectors. The descriptor can only realize a rough representation of the overall shape information, i.e., a global representation of the face textures, by neglecting the local texture information (texture details). This makes the Gabor descriptor, neglects the discrimination of micro-texture or details of face textures, such as facial expressions.

Figure 2.7 shows the ROC curves of Gabor features on the two datasets. The best performance in terms of AUC values is achieved on CMU Multi-PIE face database. Figure 2.8 illustrates some examples of successful face detection in complicated conditions (lighting, pose, expression, and occlusions). Different colors of bounding box are obtained: the green rectangle corresponds to the initial estimate of the face area, the blue rectangle represents an accurate detection of the face, the rectangle is geometrically normalized at 30×20 pixels. We examine the ability of the detector to treat partial occlusions (pose, glasses, beard) and its robustness to light changes. Some typical results are shown in Fig. 2.7. the results obtained show the robustness of the method against precedent changes.

2.5.3 Performance comparison with five methods

It is really difficult to make a fair comparison between the proposed method and other methods because of the lack of common sets of data and the widely accepted evaluation protocol. In spite of these issues, table 2.3 lists four methods we are aware of, with corresponding experimental settings such as the dataset tested on and its main characteristics; the size of the image patches used for the training/ test phases, the number of images of the test partition and the performance realized by each

method. Table 2.3 shows four methods were assessed on the same datasets used in our experiments, namely as ORL [89], CMU Multi-PIE [91] and CMU Frontal Face dataset [92].

method	data	patch size (pixels)	# Test: face(+), non-face(-)	Acc (%)	challenge
Gabor/SVM (proposed [18])	ORL	27×18	(+)267, (-)199	91.96	
PCA/SVM ([72])	ORL	92×112	(+)160, (-)200	88.26 ±0.54	Expression, low-resolution, glasses, occlusions, slight-poses, lighting
PCA/ELM ([72])	ORL	92×112	(+)160, (-)200	94.50 ±2.27	
PCA/ K-ELM [72]	ORL	92×112	(+)160, (-)200	98.12 ±0.84	
PCA/ SBELM [72]	ORL	92×112	(+)160, (-)200	96.90 ±1.59	
Gabor/SVM (proposed [18])	CMU (MPIE)	27×18	(+)170, (-)191	93.07	Expression, lighting, occlusions, head- poses, distance.
PCA/SVM [72]	CMU (FF)	27×18	(+)404, (-)521	87.91 ±0.94	
PCA/ELM [72]	CMU (FF)	64×64	(+)404, (-)521	97.25 ±1.27	
PCA/ K-ELM [72]	CMU (FF)	64×64	(+)404, (-)521	99.99 ±0.01	Expression, lighting, occlusions, frontal pose.
PCA/ SBELM [72]	CMU (FF)	64×64	(+)404, (-)521	99.19 ±1.65	

Table 2.3: Comparison of Gabor/SVM method with exiting methods in terms of Accuracy (%) on ORL database, CMU-MultiPIE (MPIE) and CMU-frontal face (FF).

Withal, corresponding experimental settings are presented such as, image dataset, beside of its characteristics (number of test images with their corresponding size) and the final performance realized by each algorithm. What the previous methods have in common is that the feature descriptors adopted for the facial description are similar or identical to those mentioned in Section 2.3.4, they use the same databases, namely the ORL database and CMU Face databases. In [72], the PCA and SBELM mixture had a detection rate of 96.90% on ORL database, while with similar data, our method obtained a detection rate of 91.96%. One reason for this drop in the detection rate is due to the fact that all face training data are with size of 92×112 pixels, which is almost 5 times than training images used in our method. Indeed, large facial images contain much more texture details than small

images. This increases the classification accuracy. The PCA and SVM mixture got a detection rate of 88.26%. The proposed method outperforms it with a detection rate of 91.96% on ORL database. The PCA method for feature extraction is unable to handle complex variations, such as lighting changes and facial expressions [93]. Moreover, PCA is not invariant to the rotation of textures of the face [94]. These reasons make the Gabor method of a better descriptor than the PCA method for extracting facial features and also explains the improved detection rate made by our method. However, the discretization effects and imperfect filter symmetry of Gabor features, make them sensitive to small variations with the amount of rotation [95]. Therefore, Gabor features can tolerate texture rotations but not invariant to them. Some errors in Gabor's magnitude and phase responses are not dramatic for the feature extraction step. However, These can slightly decrease detection rate. The proposed method achieved a detection rate of 93.07% based on CMU Multi-PIE data. This result improves that achieved by the same method on the ORL database. This increase in performance is primarily related to the wealth of information offered by the CMU Multi-PIE database, which contains more diverse facial expressions, much more light variations and faces at different depth-of-field.

2.6 Conclusion

Face detection is a prerequisite driver-behavior surveillance step. To analyze the driver's face features, it is important that the established method should be accurate and robust. The main concern of this chapter was to develop a method able to detect a face in complicated conditions. The findings realized a significant performance, in terms of robustness to ambient variations and detection rate. The method developed to find the face includes three main stages. First, the Gabor descriptor extracts important facial features. Then, the PCA method reduces high dimensional features, while reducing redundant information. Finally, SVM model is designed to distinguish face from non-face images. The experiments were validated on three databases: ORL face database, CMU Multi-PIE face database, and pictures captured within the laboratory. The main advantages of using PCA method:

1. The method reduces the time and storage space required and hence accelerates the supervised learning stage, as SVM in our case.

2. It eliminates multi-collinearity and improves the performance of the SVM model.

3. And, it can deduces redundant information of the training set, which improves detection rate.

However, the PCA method does not only have advantages. The traditional formulation of PCA (Section 2.4.2) can be a serious disadvantage for an application that must operate in real-world conditions. PCA is sensitive to data changes. in the case of driver monitoring, the processed data changes statistically according to its environment. Face detection is an important step in the driver's monitoring and hazard detection scheme. However, if the facial area is not detected, it would be impossible to locate the facial features such as eyes, to establish a diagnostic of the driver's states. To solve this issue, we thought of designing an alternative approach to eliminate face detection step before detecting facial features. In the next Chapter, we design different approaches for locating the facial features, without detecting the face.

CHAPTER 3

Driver's Eye Detection without face detection in real-life scenarios

Contents

In the present chapter, we first detail the state-of-the-art eye-locating methods and then we explore the feasibility of implanting eye detectors under very uncontrolled conditions without face detection step.

3.1 Estimation of the eye location

3.1.1 Introduction

Eyes are one of the most expressive facial details, reflecting the person's emotional situations and her degree of awareness. The effective positioning of the eye region in a given face image is crucial for interpreting the behavior of the driver, as well as for many additional face-related research applications, such as face detection, face alignment, face recognition, eye closeness detection, gaze estimation,and

head-pose estimation. In last decades, the task of accurate eye localization gained considerable attention from scientists and engineers. This is principally because of the challenge associated with the eye's appearance change. This change may be due either the intrinsic dynamic features of the eyes or simply to the surrounding natural fluctuations. Several factors can considerably affect the appearance of the eyes and convey a significant change that often causes a real difficulty for an accurate detection of the eyes. These challenges are showed through the Label Face in the Wild database (LFW) [96], and illustrated in Fig. 3.1 with various conditions that affects image quality, such as expressions, occlusion, pose, imaging conditions and lighting. These variations reduce the accuracy of the localization of the eyes and make this task particularly difficult in an uncontrolled scenario such as that of driving environment.

Figure 3.1: Images of Label Faces in The Wild database (LFW [96]) to illustrate the major challenges encountered in the phase of eye localization in general, especially, under uncontrolled conditions: variation of pose, occlusion, change of light, facial expression.

- *Facial expression:* the change in facial expressions usually introduces a radical change of shape and appearance of the eyes, which are highly sensitive to this kind of variation. For example, laughing causes a complete closing of eyes and screaming can distort the shape of eyes as well.
- *Occlusion:* the eyes are frequently occluded by hair, sunglasses and myopia glasses. The partial occlusion is often encountered in real world applications. Under these circumstances, the eyes are sometimes hard to detect and can definitively not detected at all.
- *Pose:* the change of the head-poses (yaw, pitch, roll) introduces a change in the eye appearance, and the eyes are sometimes completely hidden in a profile pose.
- *Imaging conditions:* ambient environmental conditions, such as lighting (varying of the spectral scale, distribution of light source and variation in pixel intensity) imply a significant change to

the visual aspect of the eye. In addition, factors attributed to real-world applications, including low image resolution, blurred image, or poor texture details of the image, generate a poor image quality. Moreover, variations of perspective and depth-of-field transform the appearance of the image textures and involve an unexpected fluctuations.

Eye localization is closely related to several applications based on facial analysis, such as eye detection, eye tracking, eye gaze fixation and blink detection. Some differences between those applications should be, however, highlighted. This difference is clarified by giving a brief description of each application:

- *Eye detection*, consists to generate a rough estimation of eye location in an input face image.
- *Eye localization*, however, requires a much more accurate prediction of eye positions (with high detection accuracy, very low estimation error, and requires a post-processing phase). The localization of the eyes is generally treated as a subsequent fine adjustment step after eye detection.

Eye localization and eye detection are different problems. Eye detection aims to roughly find the eye in a face image. In contrast, eye localization accurately estimates the center position of the eyes [97, 98].

- *The eye tracking*, considers further parameters in the localization of the eyes, namely time and redundant information in consecutive frames. This is usually explored to facilitate the heuristic localization of the eyes in videos.
- *The estimation of the gaze*, focuses on assessing the person's attention through an analysis of pupil motion.
- *The detection of eye blinking*, is the detection of the dynamics of the eye, namely the opening and closing of the eye. These actions are analyzed in order to estimate the physical states of the individual.

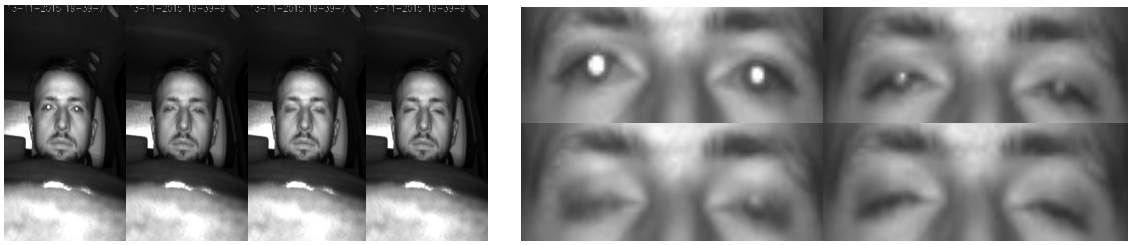
3.2 Related work

Various dedicated approaches for the localization of the eyes have been proposed in the literature, these are generally classified into three categories [98, 99, 100]: the measurement of eye characteristics, the learning of a statistical appearance model, and the exploitation of structural information. These methods depend on the nature of the information used for the development of the detection model.

3.2.1 Methods based on measuring the characteristics of the eyes

These methods explore the inherent characteristics of the eyes by treating them as a special facial characteristics. Many eye-specific features are used in practice, such as the shape of the eyes, the intensity contrast between the white area of the eyes, and the eye pupil. These characteristics could be a very reliable indicators to find the exact eye location. However, they tend to be relatively reliable

under uncontrolled environmental conditions. In addition, further characteristics can be used, such as the light reflected by the eyes in the case of infrared (IR) camera, as shown in Fig. 3.2a and Fig. 3.2b but these methods depend heavily on the used hardware part and may even require human expertise.



(a) **(b)**

Figure 3.2: (a) Images of the driver captured with IR camera. (b) Eyes examples under active near-infrared lights `http://healthyroad.pt/`.

3.2.1.1 Model-based on shape characteristics

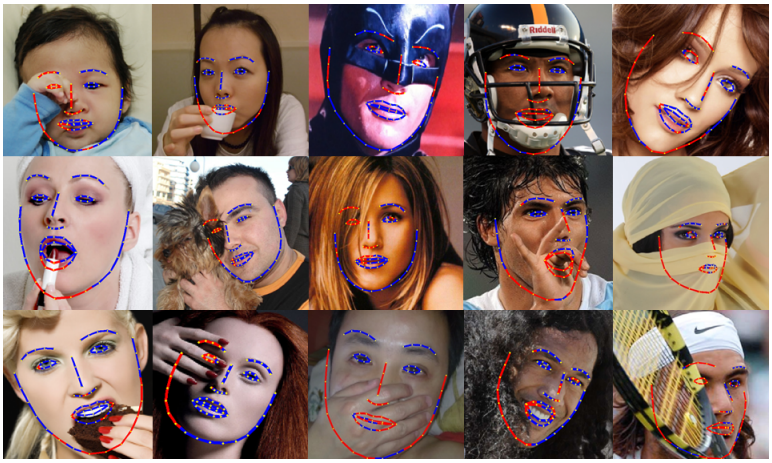


Figure 3.3: ASM based facial landmark localization algorithms [101].

The construction of a pattern-based eye shape requires four key components: the eyelids, the white region of the eye, the iris and the eye pupil. Each of these parts is geometrically unique, e.g., the eyelids have an elliptical shape, while the iris and pupil have a rather circular shape. For an appropriate geometric design of these forms, two representations are generally adopted: a continuous representation and a discrete one. The deformable model [102] is the most widely used algorithm for designing the continuous shape model. Nonetheless, the eyes are rather well expressed in a discrete way with the Active Shape model (ASM) [52, 101].

ASM is a well-known method for modeling the structural information of the eye, by automatically locating key-points that determine the shape of different facial details, covering the eyes [103, 104, 105]. The shape of the eyes can be represented with a series of discrete landmarks of the ASM

method applied to the ocular region (see Fig. 3.3). The model obtained represents the overall structural information of the eyes (overall shape) and not the local structural deformations (local shape) of the eyes. In [106], suggested a local shape-based model to interpret the local structure of the eyes, especially, the ocular components with a circular shape. Hough's circular transformation can also identify the circular shape of the iris and pupil. However, this method is computationally expensive and unable to handle discontinuities of the eye shape.

3.2.1.2 Model based on Intensity contrast characteristics

The distinguishable intensity response of the eye is another helpful hint for eye localization. This special cue enhances the existence of eyes in an image, since an eye open has a high intensity of contrast, distinguishable between different eye components. whereas, the eye pupils emit a much lower gray intensity than that of the iris and the white-eye.

Traditional methods measure the intensity contrast response of the ocular region are Integral Projection Function (IPF) [94] and Variance Projection Function (VPF), both of which are merged in General Projection Function (GPF) [79]. These methods yield a good localization results but most of them show a high efficiency only with normalized face images (slight change in eye scales and texture rotations). Moreover, they tend to be less efficient under uncontrolled conditions and facing poor image quality, which may result in great performance loss. [78] try to overcome a weakness of such methods, by accumulating locally smoothed version of pixel intensity, which tends to be more stable compared to the global one. However, the influence of environmental conditions, such as occlusion and illumination, on the overall appearance of the eye can lead to failure of the projection method. In addition, the projection response is sensitive to the rotation of the texture. Zheng et al. [107] solved precedent projection-related issues by proposing a locally selective projection (LSP) algorithm for eye localization.

3.2.1.3 Model-based on infrared illumination

The near-infrared (NIR) imaging techniques can deal with the ambient illumination fluctuations, and highlights the facial features, including the eyes. At the night time, pupil and iris can be captured at different illumination spectra. The eye pupil usually exhibits a large reflection rate than the iris, resulting in a bright spot at pupil position. This bright spot is a good indicator for eye localization [108, 109] (see Fig. 3.2a and Fig. 3.4).

In practice, NIR light source with a precise wavelength range, can meet the requirements of most in-door application scenarios. Due to its robustness against visible lighting changes. This method is widely adopted for monitoring the condition of the driver. Jaeik Jo et al. [110] proposed a driver monitoring system, which operates day and night. The overall system incorporates NIR light to capture the driver's face under optimal light conditions. Figure 3.4 shows the lighting configuration, used by the authors and tested with the driver wearing black sunglasses at wavelengths of 700 nm, 850 nm and 950 nm. This configuration is operational for the driver wearing myopia glasses and sunglasses. Nonetheless, their system is relatively expensive due to the equipment used. In addition, some restrictions have to be satisfied to ensure good performance, such as opened eye states and the on-axis light, together with NIR imaging hardware.

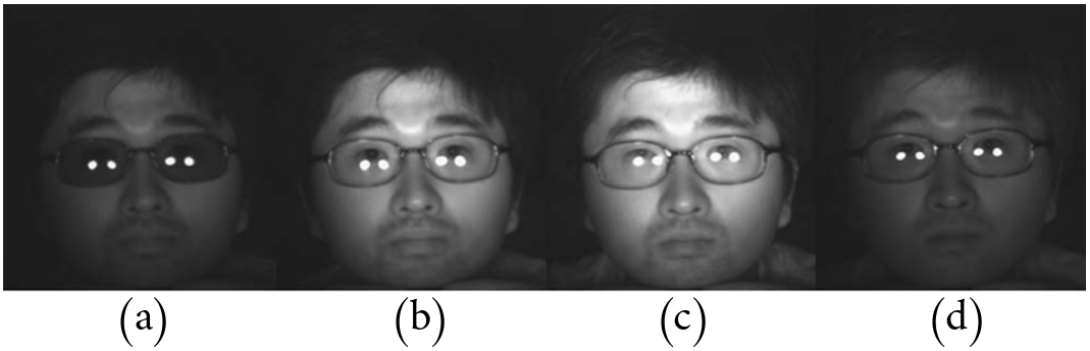


Figure 3.4: Images obtained in various wavelengths using near-infrared camera for driver wearing sunglasses: (a) 700 nm, (b) 750 nm, (c) 850 nm, and (d) 950 nm [110].

3.2.2 Methods based on the structural aspect of the eyes

The structure-based approaches place emphasis on the study of the spatial structure of the eye intrinsic components and the geometric regularity between eyes and other facial features in the face context. The eye structure-based approaches, infer successfully, the location of an eye by estimating the locations of its parts, and show robustness against occlusion and large eye variations. However, NIR lighting method is expensive in calculation and often involves a preliminary calibration phase. Moreover, eyes could not be detected when the sunlight intensity is higher than that of the IR reflection.

Figure 3.5: Three illustrations of the training set indicating the position of the marked features and the structure of the pictorial model learned [111].

As was mentioned earlier, the techniques involved in the three types of approaches for the localization of the eyes overlap with each other. Thus, the Active Shape model (ASM) [52], the active appearance model (AAM) [112], the constrained local model (CLM) [113] and the pictorial structure model (PS) [114] are representative techniques developed under this approach. Circular Hough transform technique realized good performance for eye locating [115, 116]. However, this technique is computational expensive. The pictorial structure design, is quite appropriate approach for eye localization. This approach represents the object of interest, by its components and their spatial relations, as shown in Fig. 3.5. The PS method considers the components of the object of interest, in the context of its internal global configurations, to locate of each component and its spatial structure. However, conventional PS framework provides acceptable performance only under ideal conditions, while in practice the configuration between facial parts may be deformed mostly due to different variations

(scale, rotation and expression). Xiaoyang Tan et al. [114] enhance the conventional PS method for modeling complex structural appearance changes of the eyes, under uncontrolled conditions.

3.2.3 Methods based on the learning statistical appearance model

The methods reviewed in this section concentrate on statistical design of information content, computed from the photometric appearance of ocular image patches. Appearance-based schemes, in general, potentially handle further information than other approaches showed above (those based on characteristics and structures). The photometric information of the eyes encompasses both information on the characteristics of the eyes (e.g., the eye shape) and other consistent textural materials that may be neglected or difficult to measure by the precedent methods.

The eye patterns are defined through their visual appearance by combining derived features and statistical classifiers. Three steps outline these methods:

1. **pre-refining or pre-processing** that consists of noise reduction, illumination correction and texture enhancement.

2. **feature extraction and normalization step**, the most relevant features of the eyes can be derived by applying simple or more complex photometric assemblies. Each set of features is derived from a mathematical transformation of a set of neighboring raw pixel values, extracting important features that must be retained despite the changes in the original image. Nevertheless, no unique feature descriptor can satisfy all invariance requirements because image variations are numerous and primarily transform image textures non-linearly. Selecting the ones to use in practice is mostly application-driven and the changes taken into account, textures information preserved despite these changes, and other affine transformations [98] (features encoded, discrimination retained, computational efficiency).

3. **classification** interprets statistically the derived features by establishing a learning-model.

The advantage of these methods is that richer and more reliable information can be obtained from eye models, even with poor quality facial images [117], i.e., with non-ideal acquisition conditions (e.g., noise, low spatial resolution, non-uniform lighting conditions). Nevertheless, appearance-based methods have some limitations.

To summarize, most of the aforementioned methods, are capable of giving acceptable solutions to eye localization under restricted settings. The essence of eye localization is mainly used to help eye detection with finer estimation of eye location, whatever changes that faces or eyes can undergo. Ito et al. [118] use a circular Hough transform for detecting the eyes. The verification stage is carried out by using Histogram of Oriented Gradient (HOG) transform. The eye localization approach introduced by Monzo et al. [119] uses a Haar-like feature based method to detect the eye first, and locates the eyes using Histogram of Oriented Gradient. Yan Ren et al. [97] proposed a learning method for precise eye localization. They combine a two-class sparse representation classifier (SRC) and scale invariant feature transform (SIFT) features to keep invariance to arbitrary scale and rotation. The search for an eye location is tackled by creating a heat-map with SRC output and pyramid-like locating method that discriminates the eye from a non-eye under variant resolution, while reducing the amount of searching regions. Their method shows feasible eye localization without the assistance

of face detector. Shiming Ge et al. [120] formulates the eye localization as an optimization problem. Their method is based on correlation filter bank (CFB) trained with EM-like adaptive clustering technique. The final model can find the exact eye location under pose and illumination changes. Mingcai Zhou et al. [121], investigate eye localization problem, by using coarse-to-fine searching strategy and improving Supervised Descent Method (SDM), joint to multiple nonlinear features that enhance the accuracy, while maintaining a certain invariance. Their approach is called coarse-to-fine multi-feature SDM (CF-MF-SDM). The CF-MF-SDM algorithm achieves better localization accuracy when compared to other methods, but fails to locate eyes with large rotation angles in-plane.

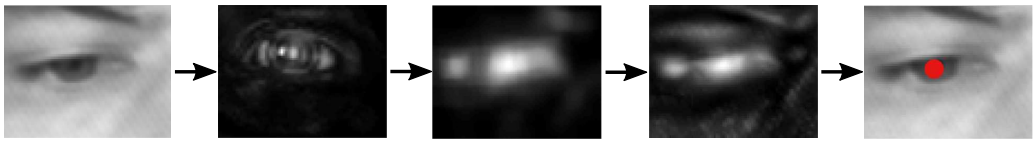


Figure 3.6: a) region containing a human eye; b) the corresponding accumulator space by Self-Similarity image; c) the corresponding accumulator space derived from differential analysis of the image intensity; d) joint space smoothed with Gaussian kernel; e) localized pupil [122].

Marco Leo et al. [122] proposed a method able to find the eye center without resorting to any advanced classifier, a general diagram of their algorithm is given in Fig. 3.6. Their method works well under restricted conditions, i.e., the eyes must be fully open and the faces in front view. Hyunjun Kim et al. [123] proposed an accurate eye localization method that tolerates head-pose and scale variations. Figure 3.7 presents some results of presented method, where the eyes are precisely detected under extreme rotation of the head.

3.2.3.1 Extraction and representation of the photometric appearance features

The key challenge for eye localization is to find the optimal descriptor (set of features). This descriptor has to fulfill some requirements [98]: (1) It should be immune to certain changes, such as fluctuations in lighting, scale variations, texture rotation, orientation, and other affine transformations. (2) The discriminant propriety of the descriptor should be preserved by the information encoded. (3) The descriptor should be computational efficient.

Generally, most existing appearance-like feature extraction methods can be decomposed into two categories: (1) Feature descriptors that represent the transformation of the images in frequency domain, e.g., Discrete Cosine Transform (DCT) [124], LogGabor [125] which is a feature descriptor that localize the frequency information of the image textures, Haar wavelets features, Gabor features. (2) Feature descriptors that represent texture information in spatial domain, such as LBP [126], Scale Invariant Feature Transform (SIFT) [127], HOG transform [128], Local Ternary Patterns (LTP) [129], and Histogram of Principal Oriented Gradients (HPOG) [130]. Some popular feature descriptors are

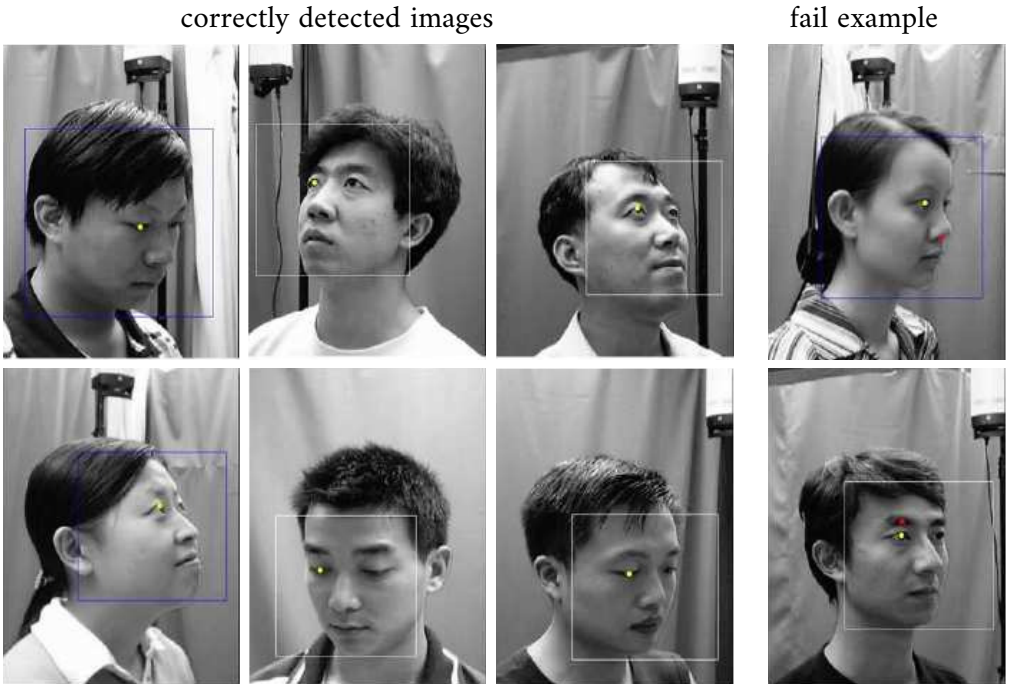


Figure 3.7: Eye localization results on CAS-PEAL database [123].

illustrated in Fig 3.8 and applied for ocular region description. All these feature descriptors have their pros and their cons as well. For example, the traditional LBP features, encode only the local features of the textured eye image. When it is more advantageous for a descriptor to represent the textures of eye patterns with their local- and global-features. Gabor features extract the global shape of the eye appearance, over a range of coarse scales. The extracted features are rich in terms of information. And hence, they are widely used in facial analysis domain, but they are computationally inefficient. Gradient-based feature descriptors, such as HOG transform is a local shape descriptor [128]. The descriptor merges the local orientation (shape) information, instead of the magnitude of small image patches. The eye image is partitioned into arrays of small spatial sub-images, and several neighboring sub-images cover a larger local region, which is the primary component of the descriptor. The local shape information is first computed on every pixel of a sub-image by measuring its gradient, and is merged in that sub-image as well as in other sub-images within the same local region with different weights, corresponding to the spatial distance, which helps to enhance the resulting distribution representation. The final histogram of each local-region is prepossessed by normalizing its contrast, then concatenated to establish the final descriptor. These processing steps enhance the descriptor robustness against illumination or shadowing fluctuations. However, HOG does not consider the unstableness of the computed gradients [130]. Since, the pixel-wise gradients are susceptible to appearance variations generated by image blur, noise, low resolution. [130] addressed this issue, by proposing HPOG descriptor. So, instead of using the pixel-based gradient computation directly, they consider the possibility to do this in a larger scope.

The variation of scales and the rotation of the image textures, are two main reasons for eye

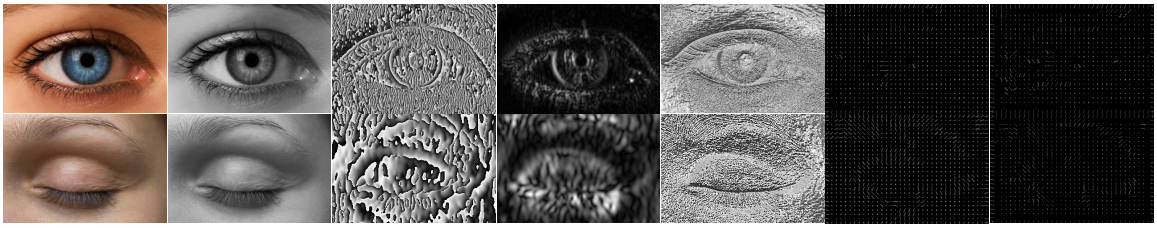


Figure 3.8: Illustration of feature sets for eye patterns. From left to right: color image, gray intensity, Gabor features (phase filter response), Gabor features (magnitude filter response), Local Binary Patterns [126], Histogram of Oriented Gradients (HOG) transform [128], and HPOG transform [130].

localization failure. This observation suggests that reasoning at multiple levels of abstraction and scale is necessary, mirroring other problems in computer vision where reasoning across multiple levels has proven beneficial. Using a straightforward image decomposition strategy, such as spatial pyramid method [131], is a step towards a descriptor invariant to resolution changes. G. Mahalingam and K. Ricanek Jr [132] developed a multi-resolution hierarchy of patch-based feature descriptors for periocular recognition. Their approach combines a hierarchical pyramid-like image and Three-Patch Local Binary Patterns (LBP) [126] feature descriptors (TPLBP) [133]. It can accurately describe periocular features. Also, Turtinen and Pietikäinen [134] have used spatial pyramid-like image for coding local texture features. Their work validates this approach for processing arbitrary spatial resolutions of the rigid-textures in challenging conditions. [135, 136] present approaches for rigid scene and texture classification. The classification enhancement is due to their spatial pyramid LBP and multi-resolution LBP approaches.

3.2.3.2 Statistical appearance models

Among the presented feature descriptors, some of them are more advantageous, particularly the statistic-based ones are more stable and handle well the uncertainty of the image data. However, they are computationally more expensive and need more representative training data to ensure good performance. It is clear that an appropriate representation of the characteristics of the eye is strongly recommended. However, this is not sufficient for a representative model of the eye. An effective learning classifier is another condition, which can handle residual variability, from a few training samples, captured under non-ideal conditions.

The following introduces some popular and important classifiers introduced for eye localization. The classification phase introduces tow types of classifiers, generative models and discriminative models [137].

- **Generative and Discriminative Models**

  - The success of **discriminative models** lies in the employed algorithms, which estimate the class probabilities $p(C_k|x)$ of a K classes of a given input feature x. (also referred as posterior class probabilities of the class labels C_k) For eye localization problem, they establish a discriminative function to segregate positive instances (eye class) from the negative ones (non-eye class), in a form of decision map for interpreting the different types of visual features of

training samples. This discriminative function separate the two classes, by hyperplane (e.g., SVM), or threshold function (e.g., sigmoid non-linear decision function in the MLP). Eye detection and localization are two complementary problems, which often solved by a binary classification way, when typical classifiers are used, including SVM [138], AdaBoost [139] and neural networks [140]. Another way to explicitly find the locations of the eyes, is to formulate this task as a regression rather than a binary classification problem. Nenad Markuš et al. [141] suggest an eye pupil localization method based on an ensemble of randomized regression trees. Their method can be executed from device with a very limited processing power, such as a smartphone. However, some failures were observed during the localization of the center of the eyes, especially when the user wears glasses, or with a rather pronounced head rotation.

- **Generative models**, include algorithms that calculate first the class conditional densities $p(x|C_k)$ and then apply Bayes' theorem, those formulations are combined with the prior class probabilities $p(C_k)$, to yield the posterior values $p(C_k|x)$, where $p(x) = \sum_k p(x|C_k)p(C_k)$. The generative models include Hidden Markov Models (HMMs) [142] and Gaussian Mixing Models (GMM). More details on generative models for eye location are available in [98].

This thesis focuses on the development of computer vision algorithms that rely on discriminative models and help solve various problems related to driver safety.

3.3 Online Eye Detection with Recurrent Neural Network

- **General scheme of the system**

This section gives a complete scheme for automatic eye detection in a video stream. Enhanced local binary patterns characterize the ocular region and extracted features, which are arranged in a sequential form. The later is then given as an input to the recurrent neural Long Short-Term Memory network for classification. The interest of using a recurrent network is that the temporal dependencies present in the image sequences can be taken into account during the classification phase. Since the full process is automatic, and the recurrent networks operate an online prediction of the eye location.

3.3.1 Feature descriptor based on traditional Local Binary Pattern (LBP) and Spatially enhanced Local Binary Pattern Histogram (eLBPH)

- **Theory of traditional Local Binary Pattern (LBP)**

The Local Binary Pattern (LBP) is a powerful gray-level invariant texture primitive. The non-parametric LBP operator was firstly mentioned by Harwood et al. [38], and then introduced by Ojala et al. [23] for texture description. Figure. 3.10 shows the original LBP operator applied without any preprocessing step and works with a 3×3 square neighborhood. The pixel values of P neighbors that are evenly distributed in angle on a circle of radius R centered at c, and are thresholded by comparing the pixel's gray value g_c with the gray values of its P neighbors $\{g_n\}_{n=0}^{P-1}$ with respect to the center pixel and considering only the sign information to form a local binary pattern.

$$LBP_{R,P}(c) = \sum_{i=0}^{P-1} s(g_i - g_c)2^i, \quad s(x) = \begin{cases} 1, & \text{if } x \geq 0. \\ 0, & \text{otherwise.} \end{cases} \tag{3.1}$$

The so-thresholded binary values are weighted by powers of two and summed to generate the LBP code, as shown in Fig. 3.9. So, by given an image I of a size $N \times Mpixels$, the original $\text{LBP}_{R,P}(c)$ is calculated at each pixel c, such that a textured image can be described by representing the whole image I by LBP histogram vector h (see Fig. 3.10), where $h(k) = \sum_{i=1}^{N} \sum_{j=1}^{M} \delta(LBP_{R,P}(i,j) - k)$, and $0 \leq k < d = 2^P$ is the number of LBP patterns. Basically, the parameters of the LBP operator are set to the values of $R = 1$ and $P = 8$, by modifying their values, LBP features can be extracted for any quantization of the angular space and for any spatial resolution.

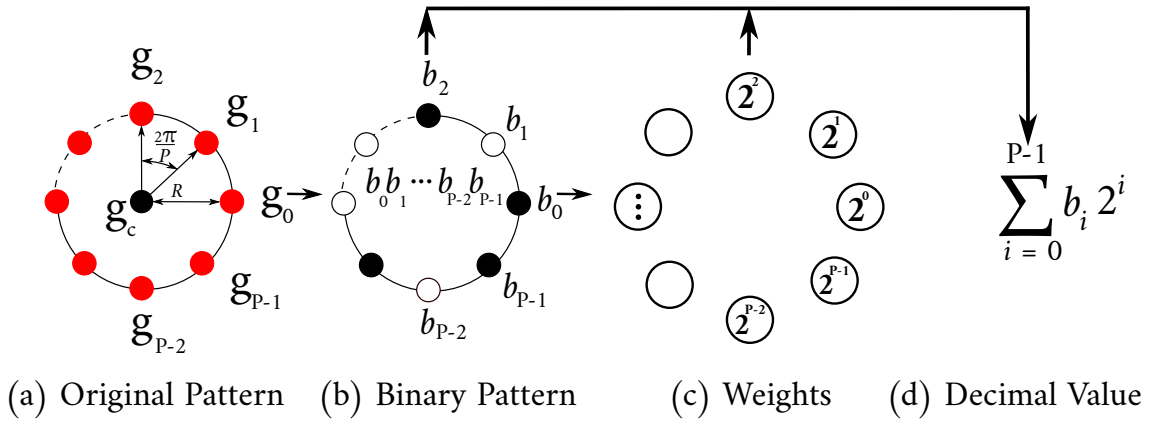


Figure 3.9: (a) A regular (R, P) neighborhood type applied to determine a LBP operator: central pixel g_c and its P circularly and evenly spaced neighbors $g_0, \ldots, g_{P-1}$ a circle of radius R [143].

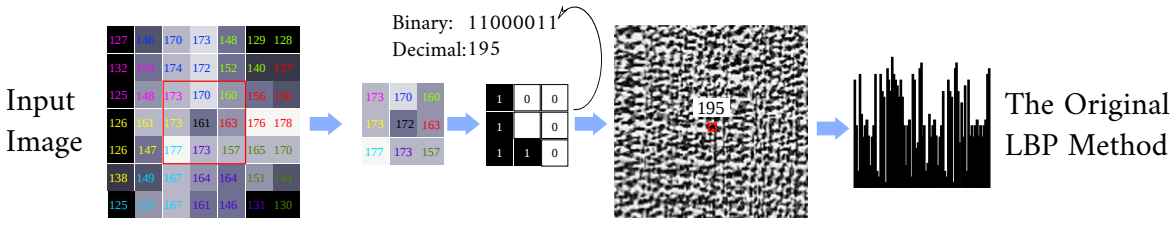


Figure 3.10: The conventional flowchart used to extract Local Binary Pattern like features [143].

Although the original LBP method is limited in several aspects, due to their questionable efficiency and flexibility, the overall LBP architecture is still very popular and widely applied in several areas. However, in spite of its popularity, the standard LBP operator counts important weakness [143]:

1. The original operator produces a rather long, large scale histogram, even for small neighborhoods. This results in a decrease of the discriminative power of the descriptor and important requirements in terms of computation and storage resources.

2. Only local texture features are captured and the large-scale texture information (global features) is definitively not discriminated.

3. The original LBP is highly sensitive to image texture rotation.

4. The original descriptor is not invariant to image noise.

5. The original LBP operator has the disadvantage of losing local texture information, mainly through the use of hard, fixed and coarse quantization scheme, and only signs of differences of neighboring pixels are used.

A large number of extensions and modifications have appeared [143], in order to improve the robustness and the discriminative power of the descriptor.

- **Theory of Spatially enhanced Local Binary Pattern Histogram (eLBPH)**

A simple extension of the LBP, denoted by $\text{LBP}_{P,R}$ is to use neighborhoods of different sizes [144]. The extension can take any radius (R) and neighbors (P) around a center pixel, by using a circular neighborhood and the bilinear interpolation whenever the sampling point does not fall in the center of a pixel.

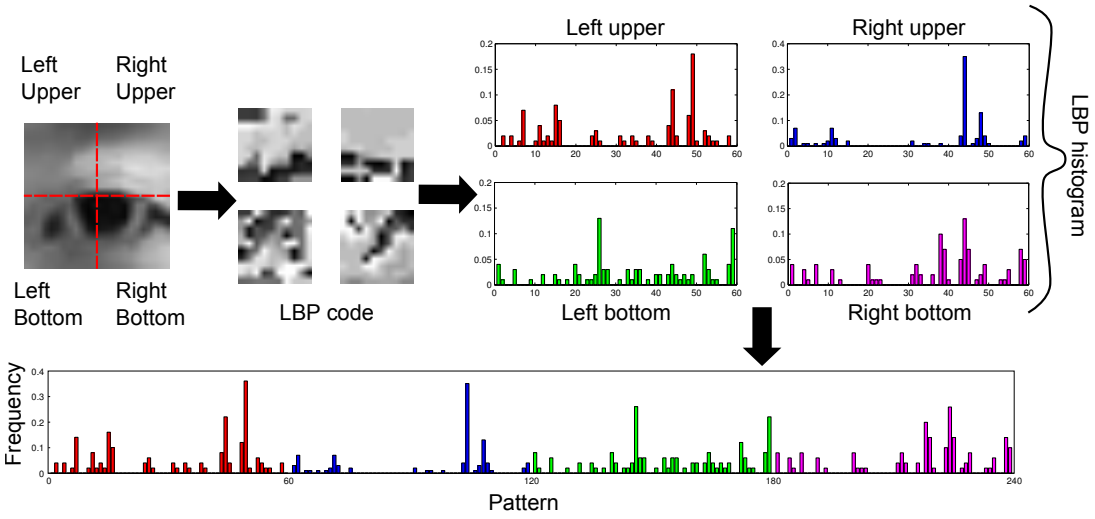


Figure 3.11: Enhanced LBP histogram (eLBPH). The eLBPH is used to describe the ocular region. It is formed on the basis of an eye image with size $24 \times 24 pixels$, which is divided into 4 non-overlapped sub-blocks of size $12 \times 12 pixels$ [140].

Another extension is the so-called uniform patterns $\text{LBP}^{u2}_{P,R}$. A LBP code is called uniform if it contains at most two bitwise transitions from 0 to 1 and conversely when the bit pattern is moved in a circular binary form [144].

For the computation of LBPH, the uniform patterns are used such that each uniform pattern has an individual bin and all non-uniform patterns are assigned to a separate bin. So, with 8 neighbors, the numbers of bins for LBPH are $256 bins$ and $59 bins$ for uniform patterns LBPH (LBPH^{u2}), respectively.

Clearly, the uniform patterns reduce the length of feature vectors, without a significant information loss.

The ocular region is considered as dynamic and non-rigid object, highly sensitive to imaging conditions and environmental variations. So, by pre-processing the image patch into several sub-region (sub-block), we can mitigate these large variations to a certain extent. The resulting texture descriptor is called enhanced LBP histogram (eLBPH), which is chosen to describe the eye with a LBPH^{u2}. The eLBPH proposed by Ahonen et al [145], is a reference for LBP based face recognition techniques. The eLBPH implementation for facial area description consists of the following procedure: first divide the facial image into d sub-regions $\{R_0, R_1, \ldots R_{d-1}\}$ and from each sub-region the LBPH is calculated individually, then the resulting d sub-regional LBPHs are concatenated to form the eLBPH, in the same order of the regional division applied to the image. The eLBPH descriptor has a length of $d \times l$, where l is the length of the sub-regional LBPH. Figure 3.11 shows an illustrative example of the sub-regional division and histogram concatenation strategy for eye representation. LBP^{u2} is statistically stable and less sensitive to noise [146]. All sub-regional LBPH are concatenated to form the eLBPH of $236bins$ ($59bins \times 4$) (see Fig. 3.11). These parameter settings were suggested by [147] for ocular region description.

They have shown that the eye is effectively represented by eLBPH in three different forms:

1. The labels of the local histogram contain information about the eye at a pixel-level.
2. The labels are summed over a sub-blocks level.
3. The sub-block histograms are concatenated to build a spatial enhanced description of the eye.

Our method, the ocular features are derived from a sequence of pictures rather than from single pictures. The employed classifier inherently considers temporal forms of the training dataset.

3.3.2 Classification

Subsection 3.3.3 presents the multilayer perceptron (MLP) applied to interpret extracted image features and locate the eyes in a given face. Subsection 3.3.4 examines Long-Short Term Memory (LSTM) recurrent recurrent networks and their application for sequence labeling and classification of eye and non-eye image sequences. Subsection 3.3.7 presents experimental setup of the constructed network architectures and their generalization capacity for eye detection.

3.3.3 Multilayer Perceptrons (MLP)

In this section, we exclusively deal with supervised classification methods. From feature extraction step, we can generate a set of labeled training data S of x input and t target, each input $x \in \Re^M$ is a real-valued feature vector with a specific length M. Each target z is a unique class drawn from a set of K classes and corresponds to each element in the feature vector.

- **Artificial Neural Networks**

Many varieties of artificial neural networks (ANNs) have emerged, each type of network has different properties and generally application-driven. Two distinctions between different types of neural

networks exist; the first one whose connections form a state feedback (cycles), and the one whose connections do not have state feedback (acycles). ANNs with cycles are referred to as recursive, or recurrent neural networks (RNNs), whereas, those with an acycle architecture (architecture, means the way that the neurons are connected together) are referred to as feedforward neural networks (FNNs). The most commonly utilized design for pattern recognition, is the multilayer perceptron (MLP) [148, 149, 150], which is the one we focus on in the follow.

- **Theory of the MLP**

MLP can be viewed as a logistic regression classifier, where a particular output is determined from I input vector $x \in \Re$, translated using a linear combination of the form $x^T w$, corresponding to its input weights w and then setting the output through some nonlinear activation functions. Throughout the thesis, the logistic sigmoid function (see Eq. 3.2) is retained during experiments.

$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{3.2}$$

For eye and non-eye classification problem, the activation of the output units is normalized by using Softmax function [151] that forces the network output to represent a probability distribution of each output class, i.e., this function ensures that all the output values of the network are set to values between 0 and 1 and their sum is equal to 1.

$$y_k = \frac{e^{a_k}}{\sum_{k'=1}^{K} e^{a_{k'}}} \tag{3.3}$$

Equation. 3.3 shows the output of each neuron depends on all other neurons adders Softmax group, where the sum of the output values are equal to one. The network is trained with gradient descent by differentiating the objective function with respect to the output. this step is refereed as a forward pass. Then, back-propagation algorithm finds the partial derivatives, with respect to the network weights, this step is refereed as backward-pass.

The MLP for eye classification is simply constructed by feeding the input vector, activate the network, and then choose the class label corresponding to the most active output unit (either 1 for eye or 0 for non-eye class). In this thesis, the objective function of cross-entropy is used to train different neuronal architectures. The cross-entropy objective function is represented in the following forms:

$$O = -\sum_{x,z \in S} \sum_{k=1}^{K} z_k \; ln \; y_k \tag{3.4}$$

where y_k is the corresponding output of K classes and a_k is the the input connected to each output unit k.

MLP classifier is trained by minimizing any differentiable objective function [150], by using gradient descent algorithm. The convenient procedure for network training is to reduce the learning error via an optimization of the objective function, while adjusting the weights, because the weights need to have the right relative values to work properly.

Cross-entropy error terms are obtained from the sum on the input-target component in the training set. Therefore, their derivatives are also a sum of these distinct terms. The derivatives of an objective function, mean the derivatives for one particular input-target component. Gradient descent algorithm finds the derivative of objective function, regarding each element of the weight vector. This process

fits the weight values and reduces the learning error in the direction of the negative slope. Gradient descent is deeply discussed in [152].

3.3.4 Recurrent Neural Networks and Long Short-Term memory

Section 3.3.3 presents briefly MLP networks as a simple methods able to map contextual information through the network structure, by propagating data from input layer to output layer, through hidden layers.

By adding recurrent connections between neurons, the network can process a sequential data form. This particular neural architecture, is referred as Recurrent Neural Networks and constitutes a cyclic connection between the different hidden neurons of the network. Based only on the recurring architecture parameter of the RNNs, the transition from an MLP model to an RNN model may seem somewhat trivial. However, the implications for the sequential learning process is another parameter that is far-reaching and should be taken into consideration.

Unlike the MLP models that only map information from input to output vectors, RNN is able to map entire historical information from the previous inputs to each output. Because RNN hidden units are not depending only on the current layer at instant t, but they are also depending on the output of the layer at one step back in time (i.e., $t-1$), so the information propagates in two directions through the network as shown in Fig. 3.12. RNN are a powerful neural models that can approximate any measurable sequence-to-sequence input signal, with a sufficient number of hidden units.

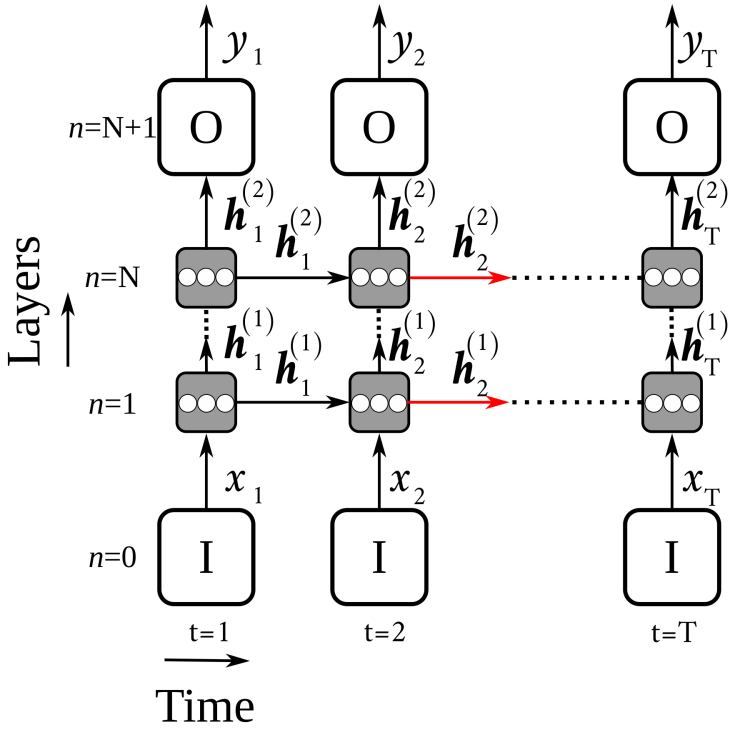


Figure 3.12: Standard Recurrent Neural Network architecture.

- **Recurrent Neural Networks**

RNNs are intrinsically a deep classifiers in time [153], by the interconnection that exists between the different hidden neurons. The particular architecture of the network with recurrent connections permits to the information collected at the output to be dependent on the previous input: the information that circulates constantly inside the network, can greatly contribute to the change of the network output. RNNs can also model dynamically an input information, and are therefore able to easily manage information with a sequential aspect, extracted from the short-term analysis of an input signal [153].

However, the great weakness of recurrent networks for sequential classification data, lies in the gradient-based training approach. The temporal aspect of information can be retained only a few moments due to vanishing gradient problem [154]: The variation of the back-propagated error on a temporal scale, depends exponentially on the amount of network weights. Thus, the error tends to explode (blow up) or disappear (vanish) because it is back-propagated in time, resulting in an oscillation of the weights, or weights that remain almost constant. In both cases, the learning stage is unsuccessful and the network neglects to learn long-term information dependencies, i.e., it has a fixed range of context.

Fortunately, a particular neural architecture with promising properties can deal with previous problems, namely Long Short Term Memory (LSTM) pioneered by [155]. The established approach used for training, is to use conventional time-based back-propagation or back-propagation through time algorithm (BPTT) [156].

RNN models are generally perceived as a feedforward neural network extended over time. Training an RNN with a BPTT algorithm is, however, not a simple task and may even involve some training drawbacks mentioned above. In the backward-pass, the error signal tends to explode, which can lead to the oscillation of the weights, or disappear, learning to bridge long-time lags. In this case, the learning phase may take an unacceptable duration or not work at all [155]. It should be noted that the exponential temporal evolution of the backpropagated error is related to the size of the weights. [157]. LSTM is designed to keep down this error back-flow problems.

3.3.5 Learning procedure for a Recurrent Neural Networks

- **Forward Pass**

In the forward-pass of the RNN training process, the activations reach the hidden layer from both, the external input at a time step t and the hidden layer activation at $t-1$. The forward-pass is applied through the network structure, that is calculated for the length T of the input sequence x, starting at on step back in time $t = 1$, and incremented at each time step until $t = T$.

By considering a sequence x presented to an RNN with I input units, H hidden units, and K output units. x_i^t is a value of input i at time t. a_j^t and b_j^t are the network input to unit j at time t and the activation of unit j at time t, respectively.

The network input to hidden unit j at time t is calculated as follows

$$a_h^t = \sum_{i=1}^{I} w_{ih} x_i^t + \sum_{h'=1}^{H} w_{h'h} b_{h'}^{t-1} \tag{3.5}$$

The element-wise activation functions are applied then

$$b_h^t = \theta_h(a_h^t) \tag{3.6}$$

To activate the overall hidden units, Eq. 3.5 and Eq. 3.6 are calculated recursively at each time step t, starting with a $t = 1$ and then the t value is incremented at each time step.

At the same time step as the hidden activations, the network inputs to the output units can be computed

$$a_k^t = \sum_{h=1}^{H} w_{hk} b_h^t \tag{3.7}$$

For our eye and non-eye sequence frame classification, the activation function and the output activation function used to build the RNN model, are softmax and logistic sigmoid for network outputs and hidden-to-hidden activation function, respectively, with the classification targets typically presented at the ends of the sequences. It follows that the same objective functions can be used too.

- **backward Pass**

The backward pass consists of repeatedly computes the partial derivatives of the objective function (Eq. 3.4) with respect to the network outputs, then we need to compute the derivatives with respect to the weights [152]. The computation of weight derivatives between recurrent connexions requires the application of BPTT algorithm. The general concept of BPTT is very simple and the algorithm is computationally efficient [152].

BPTT is similar to the established back-propagation algorithm that evolves in time (i.e., it includes the parameter t, for time step), Thus, BPTT consists of a recursively apply the chain rule. The objective function of the RNN model, depends on the activation of the hidden layer through its influence on the output layer and the hidden layer at the next time step. So, by applying the chain rule we obtain the following formulation

$$\delta_h^t = \theta'(a_h^t)\Big(\sum_{k=1}^{K} \delta_k^t w_{hk} + \sum_{h'=1}^{H}\sum_{h'=1}^{H} \delta_{h'}^{t+1} w_{hh'}\Big) \tag{3.8}$$

where

$$\delta_j^t \overset{\text{def}}{=} \frac{\partial O}{\partial a_j^t} \tag{3.9}$$

The δ terms can be computed by recursively applying Eq. 3.8, at each time step, starting at $t = T$, then t is decremented at each iteration or step.

The achieve the calculation of the final partial derivative, that consists of deriving the objective function with respect to each of the network weights, two parameters should be taken into account [152]:

1. $\delta_j^{T+1} = 0 \forall\, j$, which means that beyond the end of each sequence, the received error is evaluated to zero.

2. The weight values from and to each unit in the hidden layer are similar at each time step.

$$\frac{\partial O}{\partial w_{ij}} = \sum_{t=1}^{T} \frac{\partial O}{\partial a_j^t} \frac{\partial a_j^t}{\partial w_{ij}} = \sum_{t=1}^{T} \delta_j^t b_i^t \tag{3.10}$$

3.3.6 Long Short-Term Memory

The LSTM architecture is built exactly as an RNN with improvement in the internal architecture. This improvement consists of replacing hidden non-linear units (i.e., sigmoid logistic units) by the memory blocks. The hidden layer could be attached to another type of conventional nonlinear hidden layers and any types of differentiable output layer.

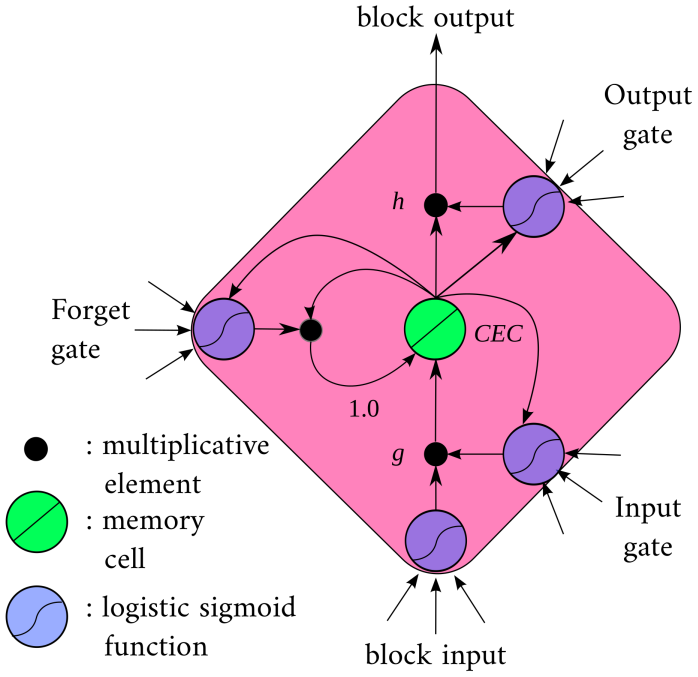


Figure 3.13: LSTM memory block with a single cell. The internal state of the cell is maintained with a recurrent connection of fixed weight 1.0. The three gates collect activations from inside and outside the block, and control the cell via multiplicative units (small circles): **The input gate** and **output gate** scale the input and output of the cell, **the forget gate** scales the internal state. The cell input and output activation functions (g and h) are the multiplicative gates and applied at the indicated places [152].

The established RNN architecture imposes a limited access in the contextual information range, the problem arises from the influence of the input conveyed on the hidden layer, therefore, on the network output. Either vanish or explode exponentially as long as the information cycles through the network recurrent connections (see Section 3.3.4). Fortunately, the LSTM scheme might be able to learn and bridge contextual informations over long time steps, even in case of noisy or/and incomprehensible input sequences, without loss of short time lags capabilities [155]. This is achieved by using an appropriate gradient based algorithm for an architecture enforcing constant error flow through internal state of spacial units, represented as Constant Error Carousel (*CEC*) and translated through a central self-connection linear unit, as shown in Fig. 3.13.

The special architecture of LSTM uses the ability of the internal recurrent connection sub-nets, i.e., memory blocks. Each block consists of one or further self-connected memory cells and three multiplicative components: **Input gate**, **output gate** and **forget gate**, as represented in Fig. 3.13. The multiplicative gates help LSTM memory cell to save and access information over long period of time, while avoiding to the gradient error of vanishing. The incoming information is introduced into the memory cell, through the opening state or the activation of the multiplicative input gate, which transmits information into the memory cell. In addition, this gate gives a sort of security for the memory cell, from irrelevant incoming information flow from the external units.

Once the information is introduced into the *CEC*, it can be kept when the value of the effective weight is set to 1, which is determined by the forget gate. The information stays inside and cycles around as long as needed. As soon as, the rest of the network, gets to read the stored information, the weight value is set to 0 and the information will disappear "reset operation", i.e., the forget gate has the potential to reset themselves. The output gate regulates the entry to the memory cell and for reaching the saved information in *CEC* from the connected external units of the network. Activating the read gate ensures that stored information can be read and comes out, to set up the rest of the network.

- **Equations of Long Short-Term Memory**

Long Short-Term Memory consists of recurrently interconnected memory blocks, each block is composed from one or several *CEC*, their number defines the dimension of the memory block. The implemented network topology uses a unidirctional LSTM blocks [152, 155], with a single memory cell (see Fig. 3.13).

This section gives the calculation of the activation (forward pass) and gradient calculation (background pass) of an LSTM hidden layer within a recurrent neural network, using the backpropagation through time algorithm.

The input of the network, to some unit j at time step t is set a_j^t, the same unit after activation is set b_j^t. The weight w_{ij} is the connection between unit i to unit j.

The mathematical representation of the LSTM network is given for a single block of memory. For a higher number of LSTM blocks, the calculations are repeated in any order for each block. The subscripts **input**, **forget** and **output** are the input gate, forget gate and output gate, respectively. The subscripts c refers to one of the C memory cells. s_c^t is the *state* of cell c at time t. The activation function of the gates is set $f(.)$, and g and h are the multiplicative gates (i.e., the input and output activation functions of the LSTM cell, respectively). The differentiable function g squashes the memory cell input a_c^t, while the function h scales the memory cell output b_c^t. Let I be the number of inputs, K be the number of outputs and H be the number of cells in the hidden layer.

The LSTM memory block is not fully connected with the other blocks in the layer. Only the *cell outputs* b_c^t (block outputs in Fig. 3.13) are connected to the external blocks in the layer. The other LSTM activations, namely the states, the cell inputs, or the gate activations, are set inside the block. There, h refers to cell outputs from other blocks in the hidden layer. As for standard hidden units (depending on the application, standard units and LSTM blocks can be combined in the same hidden layer).

As described for the standard RNNs in Section. 3.3.4, the forward pass is calculated for input sequence x of a length T. The process begins at $t = 1$ and recursively applying the update equations,

while incrementing t, until $t = T$, and recursively calculating the unit derivatives while decrementing t (see Section 3.3.4 and refer to [152] for details).

- **Forward pass**

Input Gates

$$a^t_{\mathbf{input}} = \sum_{i=1}^{I} w_{i\mathbf{input}} x^t_i + \sum_{h=1}^{H} w_{h\mathbf{input}} b^{t-1}_h + \sum_{c=1}^{C} w_{c\mathbf{input}} s^{t-1}_c \tag{3.11}$$

$$b_{\mathbf{input}^t} = f(a^t_{\mathbf{input}}) \tag{3.12}$$

Forget Gates

$$a^t_{\mathbf{forget}} = \sum_{i=1}^{I} w_{i\mathbf{forget}} x^t_i + \sum_{h=1}^{H} w_{h\mathbf{forget}} b^{t-1}_h + \sum_{c=1}^{C} w_{c\mathbf{forget}} s^{t-1}_c \tag{3.13}$$

$$b_{\mathbf{forget}^t} = f(a^t_{\mathbf{forget}}) \tag{3.14}$$

Memory cells

$$a^t_c = \sum_{i=1}^{I} w_{ic} x^t_i + \sum_{h=1}^{H} w_{hc} b^{t-1}_h \tag{3.15}$$

Internal states of the memory cells

$$s^t_c = b^t_{\mathbf{forget}} s^{t-1}_c + b^t_{\mathbf{input}} g(a^t_c) \tag{3.16}$$

Output Gates

$$a^t_{\mathbf{output}} = \sum_{i=1}^{I} w_{i\mathbf{output}} x^t_i + \sum_{h=1}^{H} w_{h\mathbf{output}} b^{t-1}_h + \sum_{c=1}^{C} w_{c\mathbf{output}} s^t_c \tag{3.17}$$

$$b_{\mathbf{output}^t} = f(a^t_{\mathbf{output}}) \tag{3.18}$$

Memory cell outputs

$$b^t_c = b^t_{\mathbf{forget}} h(s^t_c) \tag{3.19}$$

The cell output b^t_c allows connection between the current memory block, to another. Here, h is the output of the external block in the hidden layer.

- **Backward pass**

The backward pass is a recursive operation, which consists of repeating the application of the chain rule to determine the partial derivation at each time step, starting at $t = T$ that is decremented at each iteration t.

$$\varepsilon^t_c \stackrel{\text{def}}{=} \frac{\partial O}{\partial b^t_c} \qquad\qquad \varepsilon^t_s \stackrel{\text{def}}{=} \frac{\partial O}{\partial s^t_c} \tag{3.20}$$

Memory cell outputs

$$\varepsilon^t_c = \sum_{k=1}^{K} w_{ck} \delta^t_k + \sum_{h=1}^{H} w_{ch} \delta^{t+1}_h \tag{3.21}$$

Output Gate

$$\delta^t_{\mathbf{output}} = f'(a^t_{\mathbf{output}}) + \sum_{c=1}^{C} h(s^t_c)\varepsilon^t_c \tag{3.22}$$

Internal states of the memory cells

$$\varepsilon^t_s = b^t_{\mathbf{output}} h'(s^t_c)\varepsilon^t_c + b^{t+1}_{\mathbf{forget}}\varepsilon^{t+1}_s + w_{c\mathbf{input}}\delta^{t+1}_{\mathbf{input}} + w_{c\mathbf{forget}}\delta^{t+1}_{\mathbf{forget}} + w_{c\mathbf{output}}\delta^t_{\mathbf{output}} \tag{3.23}$$

Memory cells

$$\delta^t_c = b^t_{\mathbf{input}} g'(a^t_c)\varepsilon^t_s \tag{3.24}$$

Forget Gates

$$\delta^t_{\mathbf{forget}} = f'(a^t_{\mathbf{forget}}) \sum_{c=1}^{C} s^{t-1}_c \varepsilon^t_s \tag{3.25}$$

Input Gates

$$\delta^t_{\mathbf{input}} = f'(a^t_{\mathbf{input}}) \sum_{c=1}^{C} g(a^t_c)\varepsilon^t_s \tag{3.26}$$

3.3.7 Experimental Setup

- **Network Architecture**

In our experimentations several neural architectures were tested, before setting a well fitted one for eye/non-eye sequence classification. Unidirectional LSTM, with one hidden LSTM layer, containing 12 one-cell memory blocks (memory cell block of size 1), trained forwards and backwards with no target delay. Each memory block is composed from one cell unidirectional LSTM fully interconnected and fully connected to the rest of the network. In the output layer, we used the Softmax function, to ensure output values are in range $[0, 1]$, and their sum is equal to 1 at each time step. The number of the output units is equally related to the number of classes. For the recurrent networks, the hidden layers were also fully connected to themselves. The LSTM blocks had the following activation functions: logistic sigmoid functions for input and output activation functions of the cell (g and h), as shown in Fig. 3.13, and in the range $[0, 1]$ for the gates. PyBrain [158], a modular Machine Learning Library for Python was used for the LSTM network implementation.

- **Network Training**

For LSTM-RNN presented architecture, we calculated the gradient error using BPTT at each time-step, and trained the weights using online steepest descent with momentum. For the optimization phase, we initialize the weights with a very small values in a flat random distribution with range $[-0.1, 0.1]$, regularization terms are used to improve the convergence velocity. Which is a vital step to get a good performance with RNNs and makes them less exposed to over-fitting. The weight-decay is valued to 0.01, momentum of 0.9 and learning rate equal to 10^{-3}.

3.3.7.1 Personal video database to detect the eyes

A set of video sequences of the ROBIS-Robotics and Intelligent Systems, INESC-TEC laboratory researchers at the campus of Faculdade de Engenharia da Universidade do Porto (FEUP), Universidade do Porto, Portugal. These are the subjects that integrate the database used to validate our experiences.

The video recording was used to collect a large quantity of images of each individual, under conditions very similar to those of the real conditions. Video records were taken with day light plus artificial light lamps (indoor conditions), creating lighting conditions very similar to those present in the real-world scenario; we allowed large variations in facial expressions and moderate poses. After processing the video, we got a total of 4000 images for different members of the laboratory. The database includes 2000 eye images and 2000 non-eye images that are manually cropped. 2800 eye and not-eye images are rearranged into a training subset, and the remainder 1200 images with positive and negative samples are confounded and equally divided between test and validation subsets. These images are automatically captured and geometrically normalized into a size of 27×18 pixels.

The database was collected using a Python code with a web camera and the Intel Open Source Computer Vision Library (OpenCV) [159]. The images were captured with a video camera Web (Logitech QuickCam Ultra Vision), recording 25 frames per second with a total image size of 320×240 pixels (although the target faces are within one meter of distance, reflecting the actual driving scenario)

3.3.8 Results and Discussion

In this section, we compare between the classification performance of two appearance-based methods for detecting the eyes in a video stream. The two approaches were based on spatially enhanced local binary pattern histogram (eLBPH) descriptor, which has attained an established position in the field of non-rigid texture description (e.g., a face) [160]. The implementation of the eLBPH follows the procedure presented in Section 3.3.1 (see Fig. 3.11) : first divide a full eye image of size 27×18 pixels, into six regions (sub-images), and each sub-image has a size of 9×9 pixels. Second individually extract uniform LBP histogram for each region. Third concatenate all these regional histograms into a single (global) histogram for final recognition. The LBP histogram (LBPH^{u2}) of a single sub-image generates a feature vector of 59 distinct elements, and the global spatial histogram of the eye image has 354 distinct outputs.

In LBP based non-rigid texture representations, the eLBPH presented by Ahonen et al. [161, 162], gains popularity because the following approaches adopt the similar ideas [145, 163]. LBPHs are effective for rigid texture images description [160], because of such holistic textures are rather small variational, and hence their corresponding LBP histograms are statically stable and reliable. Rigid textures can often be seen as reproductions, symmetries and mixtures of various basic local patterns with random fluctuations related to their sensitivity to natural conditions, such as lighting, shadow and orientations.

Rigid textures, either natural or artificial, are easily measurable (described) thanks to the specificity of distributions of different texture local models, since such textures tend to be uniform, and hence statistically stable and less variational. Whereas, LBPHs are not entirely suitable for non-rigid texture images (e.g., a face) description, principally due to their large variational nature. Eye images are perceived as dynamic non-rigid object with large changes, such as illuminations, occlusion, ex-

pressions [145]. Moreover, LBPH descriptors are sensitive to noise in near-uniform image regions. These regions are the most likely places to contain the true position of facial traits (ocular region).

For non-ideal or large-variational texture images, such as an eye. The LBPH-based texture classification performance, generally declines seriously if no pre-refining is carried out [164, 165, 166], which reinforces the undesirability of LBPH for extensive-variational eye images.

Eye patterns in real-world are often full variational. The changes of their appearances are larger than rigid texture images. Hence, the LBPH is not quite favorable for eye image depictions, and the adoption of LBP histogram feature needs requiring special preprocessing such as region division and histogram concatenation for satisfactory performance, since the changes within sub-images are less than the whole eye images. As described in Section 3.3.1, eLBPH represents the eye image textures in three affective and complementary ways:

1. The local labels of the LBP histogram (uniform LBPH of $59bins$), contain information of eye textures at a pixel-level.

2. The labels are summed over a sub-images level.

3. The sub-image LBP histograms (each sub-image of 9×9pixels represented with an LBPH of $59bins$), are concatenated for a spatial enhanced description of the eye pattern (eLBPH of $59bins \times 6 = 354bins$).

Hence, eLBPH features represents, both micro- and macro-structures of the eye pattern, which are required for effective texture extraction and discrimination. In addition, region division can greatly attenuate variations in the image of the eye to some extent, thus indicating the success of eLBPH for ocular region description, which extends to a certain degree.

- **The sensitivity of eLBPH to the sub-block parameter**

In this experience, eLBPH is formed on the basis of an eye image with size 27×18 pixels, which is divided into 6 non-overlapped sub-blocks of size 9×9 pixels.

The so-called region division method enhances the classification performance. Both the stability and effectiveness of this method, depends on the preprocessing of region division. However, despite of its advantage, eLBPH is sensitive to the number of so-divided regions. By gradually increasing or reducing the number of divided regions. classification performance and descriptive precision can constantly change vis-à-vis [160].

Through obtained results, effectiveness of the eLBPH is clearly observed. This enhancement of the results, is not limited only to the adopted image decomposition strategy, but it also concerns the LBP variant used to build the descriptor. The ocular region is inherently highly sensitive to imaging conditions and environmental variations. So, by preprocessing the image patch into several sub-images, we can mitigate these large variations to a certain extent.

The eLBPH^{u2} is chosen to describe the eye with a LBP^{u2} histograms. The uniform patterns reduce the length of feature vectors, without a significant information loss. So, our ocular feature descriptor can compactly represented in a histogram of $354bins$. However, it is important to note that eLBPH only relatively rather than absolutely preserves the spatial relationship between eye sub-images. The full eye picture is partitioned into six parts. The corresponding six regional histograms

are concatenated into the eLBPH, which sustains each region spatial relations for the eye. However, the LBP codes in some image parts are settled into the same bin, as noted by [160] for face recognition, while they respectively come from other different areas. From the relative spatial preservation of eLBPH for eye pattern, we can easily determine that LBPH cannot retain any spatial information of eye due to the fluctuation of its histogram extracted from the full picture. Moreover, the so-called spatial preservation of eLBPH is strictly associated to the number of sub-images [160].

So, with a full eye image divided into d parts, the spatial relations of the d regions for an eye image are rightly preserved in the eLBPH. Therefore, the greater number of so-divided parts is, the better the spatial relation of eye is relatively conserved, and the eye detection performance also is consequentially increased. By the high details level of the eye pattern, it is necessary to be able to preserve its features, dividing the whole image into relatively small sub-images of 9×9 pixels is relatively correct, which enhances the classification accuracy of eLBPH, by increasing the number of sub-images. However, beyond this resolution, we will face a problem of loss of contextual information of the eye details. This will result in significant information loss and accuracy decreasing.

Classifier	Database	**TP [%]**	**FP [%]**	**TN[%]**	**FN [%]**	**Prec**	**Rec**	$F_1 Score$	**Acc [%]**	Times [s]
SVM(linear)	PV	48.50	1.67	49.67	0.17	**0.996**	**0.997**	**0.980**	**98.1**	**0.562**
SVM(RBF)	PV	48.33	2.83	48.67	0.34	0.930	0.993	0.944	96.8	0.642
LSTM	PV	47.16	2.67	48.67	1.50	0.970	0.948	0.960	95.8	0.697

Table 3.1: Eye and non-eye classification on Personal Video (PV) Database for eye detection.

Four common evaluation measures were computed by considering all the frames of the test subset, to assess the performance of online eye detection approaches. The classification accuracy (Acc) is the proportion of frames correctly classified. Recall (Rec) is the proportion of frames labeled as eyes in the ground truth that are estimated as eyes by the algorithm. Precision (Prec) is the proportion of frames estimated as eyes by the algorithm that are effectively eyes in the ground truth. Finally, the $F_1 - measure$ is a global performance measure corresponding to the harmonic mean of precision and recall.

Table 3.1 shows the performance of deep LSTM network with forget gate [167] and peephole connections [168] for online eye detection.

The network is trained with an objective function of crossed-entropy, for the minimization of the error-entropy (EEM) and helps a flexible and rapid classification error convergence, compared to the traditional mean square error (MSE). Training with EEM can significantly reduce the number of training epochs to achieve convergence [169].

Several network configurations were tested, varying the number of hidden LSTM units and classification epochs. The exhibit configuration is valid for an optimal compromise between classification performance, error convergence and detection time in a given video sequence. LSTM network was trained with 1000 training epochs, which were required for the best classification result. Through our experience, no observations of a wide change in the classification performance generated by extending the number of training epochs.

LSTM network does not require fine-tunning of its weights to access long range contextual information [152], comparing to other well established network architectures, e.g., MLPs and standard RNNs. The deep LSTM network extracts discriminative information from low-grade details to in-

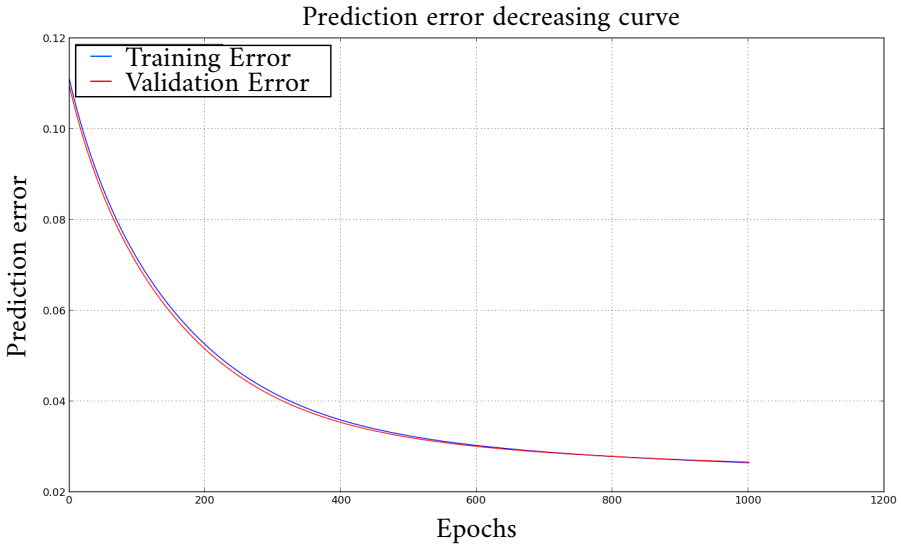


Figure 3.14: Prediction error (training error and validation error) decreasing curves of LSTM-RNN for online eye detection.

crease both recall and precision of 97% and 94.8%, respectively. This is remarkable and explains the high $F_1 - test$ of 96%. Potential explanation, deep-LSTM network learns and bridges the contextual informations over long time-steps of the ocular eLBPH features. LSTM network performance remains reliable without missing the short time lags capacities, even if the input sequential information is incomprehensible or includes image noise generated by eLBPH details because the descriptor is less susceptible to noise but not invariant.

As well as being faster, the RBF SVMs are slightly more accurate than LSTM network. The final difference in performance between SVM(RBF) and LSTM network on eye/ non-eye classification task is quite small 1.0% of difference in the classification accuracy, for a difference in a detection time of 0.055 seconds. These differences are not larger, which means that LSTM network is favorable for eye detection task because comparing to SVMs, LSTM network is very efficient for preserving long time dependencies, which are highly required for eye detection task in a video stream. The best classification rate of 98.1% was achieved by combining eLBPH and SVMs based on a linear kernel, for a detection time of 0.562 seconds, which is relatively faster than the previous two approaches, but because of the simplicity of the classifier this is an insufficient gain.

LSTM is better adapted to long-range patterns learning and pattern ordering in a given sequence, than an SVM classifier. We expect a greater distinction between the two classifiers on the classification of the eyes, in particular on cluttered background of impractical conditions.

- **Detection results**

Our system provides good performance regarding the eye detection in different variations. We have studied two classification methods of different nature. LBP features have been selected. The use of gray-scale with spatial enhancement of LBP features as inputs to the SVM and LSTM classifiers

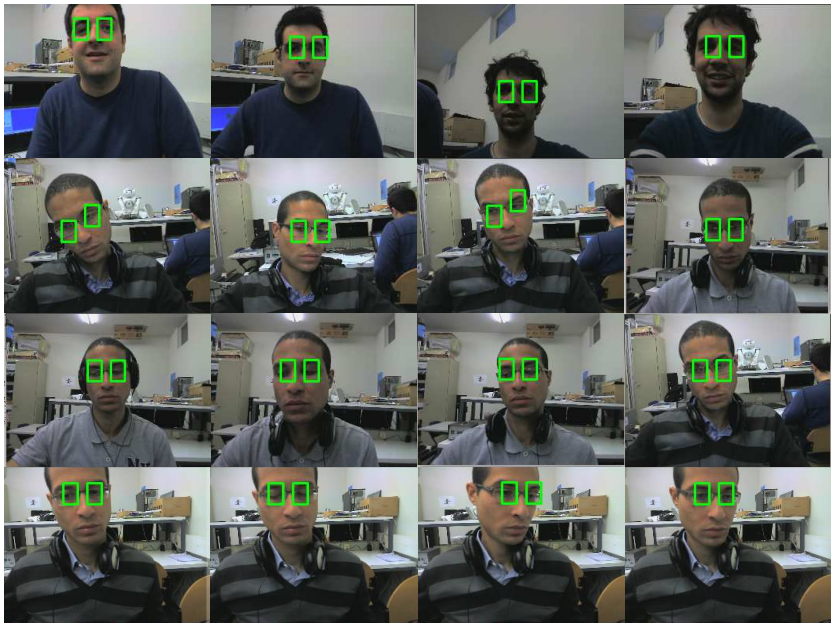


Figure 3.15: Snapshots illustrate some successful eye detection: individuals wear glasses and accessories, slight angles of head rotation, facial expressions (smiling, eye closed change of depth-of-field, cluttered background, different illumination conditions) . Pictures were captured with a web-camera indoor (within the laboratory) [147].

makes the system have a good performance regarding illumination changes, presence of structural components (e.g., the eye glasses) and variations of the eye states. For the detection performance, our system works well with uncontrolled daytime illumination and cluttered background, being robust to both strong and faint daylight. The characterization of complex elements such as the eyes in an uncontrolled environment is helped by the computation of eLBPH features. These features are able to deal with all the variations in the eye images.

The results showed in Table 3.1 demonstrate that our method gives good detection accuracy on our database. Quantitative results can be seen Fig. 3.15 and Fig. 3.16. From the quantitative results, it can be observed that the LBPH descriptor based on a regional division decomposition of the original image, greatly improves accuracy. As we are primarily interested in accurate eye location, we conclude that the eLBPH approach offers many important advantages for this purpose.

However, we are convinced that the results can be improved by exploring other methods of image decompositions for the extraction of LBP features, because by applying standard supervised learning methods (e.g., the SVM classifier) and of sequential supervised methods (e.g., the LSTM network), we did not observe a large gap between the accuracy of detection generated by the two methods for eye detection.

We noticed a few sources of discrepancies with annotated data (examples can be seen in Fig. 3.16):

- **Presence of eyeglasses.** In some images, the appearance of the eyes is significantly deteriorated by a strong reflection of the embracing light on the glasses. Our method may partially fail under these circumstances, sometimes only one eye can be detected and sometimes several eyes are

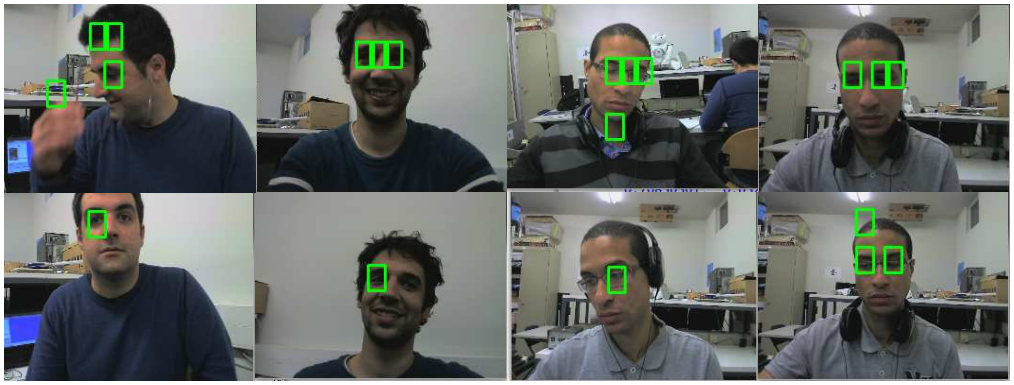


Figure 3.16: Snapshots from captured pictures illustrating some typical failure of our eye detector: individual with extreme head rotation angles, facial expressions and with accessories (myopia glasses, expressions, indoor lighting conditions) [147].

detected in the same face.

- **Facial expression and closed eyes.** Eye detector works well with open and closed eyes. Nevertheless, a well-pronounced facial expression, such as a broad smile, may occasionally result in the detection of more than two eyes on the same face.

- **Extreme head rotation.**The combination of a wide angle of head rotation and facial expression can also promotes the emergence of false positives.

- **Eye region estimation failure.** Since our approach does not detect face at first, which increases the notion of challenge. The researching space is thus very large. Some bad results are observed through a false detection, for example the detection of some regions expected similar by their appearance to the eye, on the face (the hair, region between the eyes) of the user or in the background. An example can be seen in the Fig 3.16.

In the next chapter, all these problems associated with the appearance of false positives in the eye detection algorithm are analyzed. These drawbacks are overcome in the algorithm of eye localization presented in Chapter 4, where we explore the possibility of precise localization of the eyes in intense conditions of the real world.

Figure 3.17: Three segment of video sequence demonstrating the success of our detector in real world conditions. The video is recorded inside the research lab with different lighting conditions, head rotation, cluttered background, facial expressions and accessories (e.g., glasses) [147].. (Zooming is recommended for digital visualization).

3.4 Conclusion

In the first part of this chapter, we provided a comprehensive review of ongoing research in the area of eye detection and localization. We focused on the difficulties and global challenges in real life scenarios and how to deal with them. We pay a particular attention to practical problems for the development of a robust eye localization approach with different preprocessing and post-processing methods.

In the second part, we developed an eye region estimator based on specially enhanced LBP (ePLBPH) descriptor to extract information of the appearance of each eye to be estimated. Then, we defined a probabilistic learning, which makes it possible to learn the LBP features in order to establish a correspondence between the trained model and a new input image. We conducted a series of tests on our video sequence database, to evaluate the effectiveness of our approach and its ability to overcome ambient difficulties. Our estimator is able to detect the eyes accurately, without having recourse to face detection beforehand, and with a very short calculation time. However, there are other techniques that can give better results. Some weaknesses have been identified, such as the sensitivity of the system to light in the user's glasses and even facial expressions, sometimes a combination of these three factors. These issues are not tolerated in the case of a real driving scenario, when the driver is permanently exposed to a dangerous situation by a simple loss of concentration or a falling asleep (period of micro-sleep).

A robust model for eye localization is essential step to recognize the eye states. Indeed, it allows to build an accurate representation of the appearance of the eye by taking into account most of relevant variations of the real world to make the system more confident.

CHAPTER 4

Driver's Eye Localization without face detection in real-life scenarios

Contents

We propose in this chapter an algorithm for eye localization, which imporves the performance of the precedent approach presented in Chapter 3. The algorithm overcomes the variations of the real world and maintain a good detection accuracy.

4.1 EyeLSD algorithm for eye localization

4.1.1 General scheme of the EyeLSD system

We proposed the EyeLSD algorithm, to locate the eye and detect its states (open and closed), without detecting face. The experiments are made within the framework of a realistic reproduction of the conditions that drivers undergo while being in the vehicle cockpit. EyeLSD combines the strength of several feature descriptors that properly represent the eye appearance. Their combination provides richer and consistent information of the eye.

EyeLSD algorithm is decomposed in three main stages, as shown in Fig. 4.1. The first stage consists to pre-process the original image by reducing noise and enhancing textures (steps a and b), the second stage extracts features in key-points of the image (steps c, d, e, and f), and then the final stage uses statistical classifiers to interpret gathered information (step g).

In stage one, Fig. 4.1(a), the 3-channel RGB image is converted into gray-scale image Υ, then image processing techniques are applied in Υ, to filter out noise and further enhance the localization.

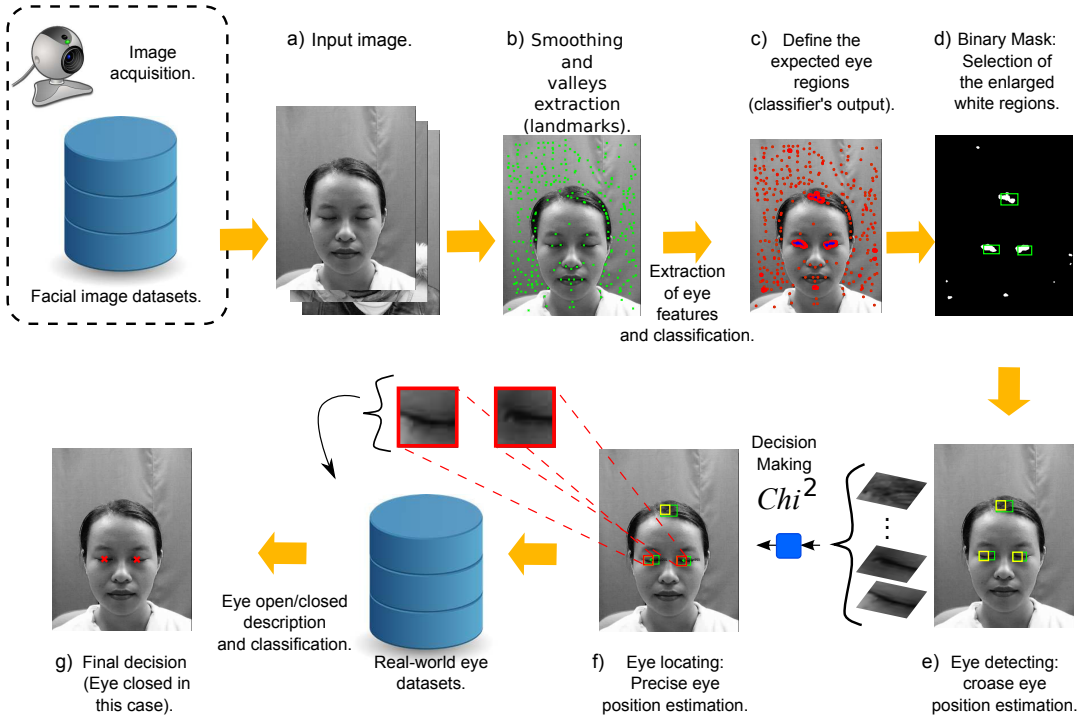


Figure 4.1: EyeLSD flowchart; eye localization and open/ closed state estimation [140].

After a searching map is established, with key-points that highlights specific facial traits (e.g., nose tip, mouth corners, eye centers, eyebrow, and lips). This pre-refining step is adopted as locating strategy to satisfy requirement of localization, instead of applying traditional sliding-window design, at risk of missing relevant image regions, e.g., the ocular region. In stage one, Fig. 4.1(b), the searching map is set up with key-points, the image Υ is firstly smoothed because, LBP based descriptors are sensitive to noise in near-uniform image regions. Then, local minimum regions (valleys) are derived from the pre-processed image, corresponding to the lower gray-scale regions. These regions are the most likely places to encompass the true position of facial traits. Connected-component labeling is performed to detect connected regions in valleys, while assigning them landmarks (key-points) for the next steps. In stage one, Fig. 4.1(c), the spatially enhanced Pyramid-like method, i.e., ePLBP*, Section 4.1.2, is used to encode local features with different scales within key-points. The ePLBP* descriptor reduces the influence of illumination and noise change, while moderates the variation in scales and rotation degrees.

The second stage consists of using region selection methods, to choose the most discriminated image parts that stand for ocular regions. The spatial structure of objects in a scene is used, besides of binary morphological operations [80]. The white region (pixels) of binary image, are expanded and the black region (pixels) are diminished by erosion operation. Afterward, black region (pixels) of the eye area diminished by the erosion operation is expanded by dilation operation. This second process of sequentially erosion and dilation is called opening process. After the opening process applied two times, the largest blob is generated for an eye (enhanced area) and small blobs are generated for noise then rejected. In stage two, Fig. 4.1(d), the measure of the structural proprieties of the enhanced

area (retained) is applied upon binary image to choose the widest surface, i.e., the silhouette of the eye. The area's center of mass (moment) is considered as an eye center. In stage two, Fig. 4.1(e), a bounding-box is reported-back on the original input image and detected regions are geometrically normalized to $24 \times 24 pixels$. The eye candidates are classified as true or false based on similarity value *Chi-square* ($\mathbb{S}_{\chi^2}$), which boosts the performance of the ocular detector. This post classification phase increases the possibility to localize eyes, whatever their states as shown in Fig. 4.1(f).

The final stage (step g) consists of detecting the eye open and closed status. This presented in Chapter 4, which explains in details each step of the eye states detection.

4.1.2 EyeLSD: Ocular region feature computation and classification

- **Descriptor for Ocular Region Detection**

In this chapter, we focus to design a method for eye localization, which is a fundamental stage for the later one (eye states detection step). The ocular representation should satisfy three principal criteria: (1) Precision of the features descriptor for representing eye image textures. (2) Robustness against various imaging and environmental conditions, i.e., eye representation must be invariant to variations that degrade the eye appearance. (3) Effectiveness in improving the accuracy of detection model.

Recently, there has been a significant emergence of some local image descriptors based on LBP method. These descriptors combine the discriminant strength of conventional LBPs and provide structural improvements that enhance the LBP invariance to certain changes, such as texture rotation for example.

Thereby, the used methods are divided into three subcategories [143]: (1) Pre-process the input image before extracting the LBP features, mainly using filtering techniques (2) Combination of several LBP feature descriptors to obtain a more powerful one; (3) Extraction of LBP and non-LBP features in parallel, then combining the two types of features into a single one.

1. **Preprocessing**
   the preprocessing phase is important in LBP for improving classification results and providing robustness to the features descriptor against texture changes, such as rotation and resolution variations.

   Gabor filtering method is the most widely descriptor combined with preprocessed LBP features. The consolidation of these two forms provides complementary information. LBP histogram captures small and fine details of image textures, while Gabor filters encode local appearance texture information over a broader range of scales. [170] combines Gabor filtering and LBP descriptor to reinforce the discriminant capacity of both representations for texture description. Li et al. [171] proposed a Scale and Rotation Invariant SubUniform LBP (LBP^{sri_su2}) descriptor for texture analysis. LBP^{sri_su2} is partially based on conventional LBP descriptor. LBP^{sri_su2} is robust against scales and rotation variations. Davarzani et al. [172] proposed weighted rotation and LBP invariant scale (WRSI LBP), to overcome the sensitivity of LBPs to the texture rotation. The spatial pyramid method, is a step toward a descriptor that is invariant to resolution changes. Qian et al. [135] proposed an LBP descriptor that transforms an image in special pyramid domain then computes the traditional LBP descriptors. Their method

is named as Pyramid LBP (PLBP). Mahalingam and Ricanek Jr [130] have set up a local image descriptor with a hierarchical multi-resolution struture. This descriptor is based on patches for periocular recognition. This approach is a combination of hierarchical pyramidal image and feature descriptors (TPLBP) [133] of Three-Patch Local Binary Patterns (LBP) [126]. The derived descriptor is able to accurately describe the periocular characteristics. Turtinen and Pietikäinen [136] used a pyramidal spatial image to encode the local texture characteristics. Their work validates this approach to handle arbitrary spatial resolutions of rigid textures under difficult acquisition conditions. [135, 136] current approaches for the rigid classification of scene and texture. The improvement in classification is due to their LBP pyramidal spatial and multi-resolution LBP approaches.

2. **Combining multiple LBP-like features**

   Zhenhua Guo et al. [173] suggested a completed LBP (CLBP) scheme developed for texture analysis. CLBP combines multiple LBP type features, namely CLBP-Sign (CLBP_S), CLBP-Magnitude (CLBP_M) and CLBP-Center (CLBP_C). The combination of these three type LBP based descriptors provides significant improvement that can be made for rotation invariant texture analysis. They proved that the sign factor is further prominent than the magnitude factor in keeping the local change information, which can reveal why the CLBP_S(i.e. regular LBP) features are better useful than the CLBP_M features. Moreover, by combining the CLBP_S,CLBP_M and CLBP_C methods, all of which are in binary sequence form, either in a joint or in a hybrid fashion, often better texture classification accuracy than the state-of-the-arts LBP designs were achieved. Liu et al. introduced BRINT [174] and MRELBP [175]. BRINT is constructed by combining separately three descriptors BRINT_S, BRINT_M and BRINT_C. BRINT employs all the invariable models to textures rotation, to avert the leading proportion of the uniform models [143]. As with traditional LBP descriptor, in BRINT, pixels are examined in a circular neighborhood, but holding the amount of bins in a single-scale LBP histogram, so that constant and small, and that arbitrarily large circular neighborhoods is be examined and described on different scales. BRINT derives features from several low-dimensional features. And it shown robustness to noise.

3. **Combining LBP features with non-LBP features**

   The original LBP operator was developed as a complementary measure of the local image contrast, and the joint histogram of the two complementary features, LBP and the variance (VAR), was proposed to deal with the textural rotation problem and invariance of the original operator [144]. Guo et al. [176] considered it necessary to introduce a supplementary step to the development of the LBP operator, which is the quantification. Their feature descriptor is called local binary pattern variance (LBPV). Ahonen et al. [177] combines the features of Discrete Fourier Transform (DFT) and LBPs (LBPHF). LBPHF is invariant to the global textures rotation.

Therefore, by the various points mentioned above, it is found that the discriminant power of the LBP descriptor can be significantly improved in different ways of preprocessing. The proposed LBP descriptor has been enhanced to give optimal performance, while preserving a certain invariance to the real world changes, including texture rotations, viewing angle, capturing texture information at different depths of field, and perspectives.

The next section presents a new descriptor for eye representation called enhanced Pyramidal Local Binary Pattern Histogram (ePLBPH) [140].

4.1.3 Theory of the Enhanced Pyramidal Local Binary Patterns (ePLBP)

Local binary pattern in spatial pyramid domain (PLBP) is a powerful multi-resolution analysis method. Over the pyramid transformation, each pixel in the low spatial layer of the pyramid, is generated by down sampling the low-pass filtered high resolution image at the pyramidal level just bellow as shown in Fig. 4.2. So, in images of low-resolution, a pixel corresponds to a region in its high-resolutions. This region is described by [136, 178] as an *"effective area"* of the filtered pixel. Please see [135] for more details about LBPs representation in spatial pyramid domain.

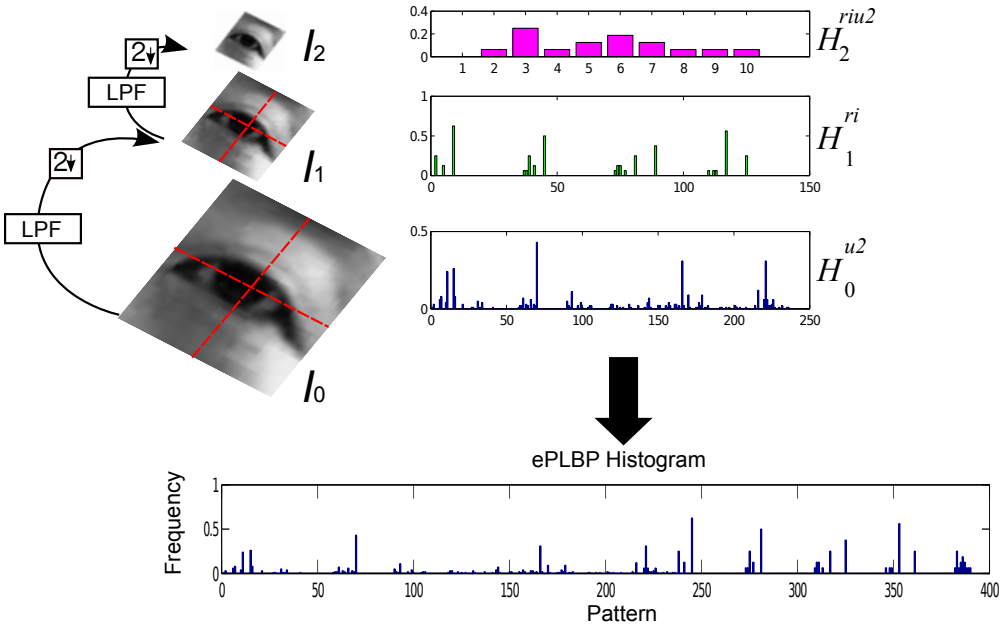


Figure 4.2: The pyramid decomposition and corresponding LBP signatures. The diagram of pyramid sampling in neighboring 3 resolutions. The down sampling ratios in each x and y directions are both 2, the resolution variation of neighboring two pyramids is with a factor 4 [140].

The pyramid generation approach consists of low-pass filtering (LPF) and down sampling images at the pyramid level just below. The pyramid image is recursively constructed as follows: $G_1(x, y) = I(x, y)$ for pyramid level $l = 1$, where $I(x, y)$ is the original image, $l = \{1, \ldots, L\}$, and L is the number of layers in the pyramid. In a general definition of pyramid construction process and for pyramid level $l > 1$:

$$G_l = \sum_m \sum_n f_G(m, n) G_{l-1}(R_x x + m, R_y y + n) \tag{4.1}$$

where R_x and R_y are the down sampling ratios in x and y directions, respectively. $(R_x, R_y > 1)$ in case of down sampling is used during the pyramid image generation. Otherwise, $R_x = R_y = 1$ if no

spatial sampling is used. x and y are the image coordinates, whose values are expressed in a Cartesian coordinate system, f_G is a 2-D isotropic Gaussian (circularly symmetric).

In the spatial pyramid domain, feature extraction of texture Γ are produced through a combination of texture information of all pyramid levels. Let Γ^{k} represent the texture information of the kth pyramid, ($k = 1, \ldots, N$), g$_c^k$ corresponds to the central pixel of the kth pyramid.

$$\Gamma^k = t(g_c^k, g_k^0, \ldots, g_k^{p-1}) \approx \gamma(s(g_0^k - g_c^k), \ldots, s(g_k^{p-1} - g_c^k)) \tag{4.2}$$

The resulting binary code is denoted as $\mathrm{LBP}_{P,R,k}$, which is the LBP code of a pixel at the kth spatial pyramid and expressed as follows:

$$LBP_{P,R,k} = \sum_{p=0}^{P-1} (s(g_k^p - g_c^k)2^p) \tag{4.3}$$

The final PLBP is a combination of several LBP histograms for N-spatial pyramid images:

$$PLBP_{P,R} = \cup_k LBP_{P,R,k} = \{LBP_{P,R,1}; \ldots; LBP_{P,R,N}\} \tag{4.4}$$

In our enhanced LBP spatial pyramid architecture ePLBP (proposed for EyeLSD), the eye pattern is down-sampled twice. The spatial pyramid is generated with 3 levels of image sequences $I = \{I_0, I_1 \ldots, I_{L-1}\}$, Fig. 4.2(i.e., $L = 3$). The pyramid images are denoted as I_0, I_1, and I_2 for the pyramid's basis, first and second level of the pyramid, respectively. The size of the n^{th} level image has the half size of the $(n-1)^{th}$ level image, Fig. 4.2. Based on Ojala et al's rules, the enhanced $\mathrm{PLBP}^{*}_{P,R}$ (ePLBPH) can be constructed, where $*$ stands for $\{u2\ , ri\ , riu2\}$ patterns.

The original image, represents the 0^{th} level image I_0 of the pyramid, LBP^{u2} is performed upon the 0^{th} pyramid's image with size $24 \times 24 pixels$. The uniform patterns are used because they tolerate rotation better, since they contain fewer spatial transitions exposed to unwanted changes upon image rotation, besides being highly descriptive. LBP^{u2} is statistically stable and less sensitive to noise [146]. I_0 is the highest resolution image in our pyramid and contains more details about the eye appearance. To enhance the discriminative capability of the applied descriptor, the 0^{th} image of the pyramid is equally divided into 4 non-overlapped sub-regions. The global spatial histogram is computed by concatenating all sub-regional LBPH^{u2}. The obtained LBPH^{u2} signature is denoted as H_0^{u2} with a length of $(59 bins \times 4)$.

At the 1^{st} level of the image pyramid I_1 with size $12 \times 12 pixels$, LBP^{ri} code [179] is generated, to handle the invariance to texture rotation. This pyramid image level is pre-processed in same way as I_0. The LBP histogram obtained is denoted as H_1^{ri} with a length of $36 bins \times 4$. The I_1 is a smoothed image of size $12 \times 12 pixels$ and contains less details about the eye appearance. So, the extracted information may not be very discriminant. To solve this issue, we proposed to use a sub-region-division and histogram concatenation strategy.

The highest level image I_2 may be sparse and unstable, due to its small size of $6 \times 6 pixels$, and thus the sub-region division cannot be used therein. At this level of the pyramid, the feature descriptor used should be highly descriptive, while generating a small length of LBPH. The LBP^{riu2} [180] provides a good discrimination in comparison to the "$non - uniform$" patterns, and this leads to differences in their statistical properties [144]. However, in this pyramid image level the spatial

preservation of LBPH does not preserve any spatial information of eye, due to its histogram statistic over the whole eye image patch. The LBP^{riu2} is by definition gray-scale invariant measure, ensures the invariance to rotation, and considers only uniform patterns, this is a fundamental proprieties of texture. The LBP^{riu2} generates a histogram denoted as H_2^{riu2} with a length of $10bins$. Finally, all histograms H_0^{u2}, H_1^{ri} and H_2^{riu2} are concatenated, to form the enhanced pyramidal eye signature **F**; $F = \{H_0^{u2}, H_1^{ri}, H_2^{riu2}\}$. After that, for each image patch, we could estimate its corresponding features. In this work, the performance of the ePLBPH has been compared with LBP^{riu2} pyramid histogram (ePLBP^{riu2}) of $((10bins \times 4) + (10bins \times 4) + 10bins)$ dimensions, which processed in the same way as ePLBPH*.

4.1.4 Distance thresholding for pair matching

In the following, we explain the learning protocol of the thresholding procedure to apply the pair matching (same-not-same binary classification) setting for a refined estimate of the eye location.

- **Similarity measurement for precise eye localization**

This subsection presents the similarity test to sustain the performance of our EyeLSD eye detector. The estimation of the similarity distance between two descriptor-based histograms is a particularly relevant issue in vision-based applications [133]. The most straightforward way for pair matching using image descriptors is to analyze the distance between a pair of feature vectors that encode an object appearance, i.e., given two eye images I_1 and I_2, which are encoded using some image descriptor g as $g(I_1)$, $g(I_2)$, the pairs are considered matching if $d(g(I_1, I_2)) < \tau$, where d is a distance function and τ is threshold. In order to establish the value of the threshold τ which corresponds to the problem of locating the eyes.

Initial threshold values are set in a fixed range (established during our experiments), then the best threshold value is the one that achieves the highest eye recognition score on three test databases: GI4E database [181, 182], GI4E head-pose (GI4E-HP) [182, 183] and Extended Yale-B face database (EYaleB) [184, 185].

- **TP**. output of $\mathbb{S}_{\chi^2}$ for confirming the real position of detected eye. Some examples are illustrated in Fig. 4.4 and Fig. 4.3. The real positives correspond to the bounding boxes that appear in blue color.
- **FP**. output of $\mathbb{S}_{\chi^2}$ for confirming the supposed real position of the eye in current frame but in fact the detected items are; either a background image patch or other facial features(noise, mouth, eyebrow).
- **TN**. images that do not correspond to the eyes and successfully rejected by $\mathbb{S}_{\chi^2}$.
- **FN**. images correspond to the eyes and falsely rejected by $\mathbb{S}_{\chi^2}$, in this case we can say that $\mathbb{S}_{\chi^2}$ fails to differentiate between eye and non-eye texture referring to the designed template.

EyeLSD uses *Chi-Square* distance ($\mathbb{S}_{\chi^2}$) for pair matching because, $\mathbb{S}_{\chi^2}$ performs better than other statistical distances for LBP-based feature histograms comparison [145]. A comparison of the histogram pairs that are supposed to represent the eye (open or closed), a first features histogram of

Data	$\mathbb{S}_{\chi^2}$ (τ)	TP [%]	FP [%]	TN[%]	FN [%]	Prec	Rec	$F_1 Score$	Acc [%]
GI4E	0.30	75.80	21.73	2.40	0.0	0.78	1.0	0.87	78.26
GI4E	0.25	75.05	20.17	4.76	0.0	0.79	1.0	0.88	79.82
GI4E	0.20	75.13	17.81	7.04	0.0	0.81	1.0	0.89	82.18
GI4E	**0.15**	76.84	12.54	10.35	0.25	0.86	0.99	0.92	**87.19**
GI4E	0.10	70.05	8.32	16.19	5.42	0.89	0.93	0.91	86.24
GI4E	0.09	64.18	7.34	17.06	11.4	0.89	0.85	0.87	81.25
GI4E	0.08	55.63	6.24	18.86	19.25	0.90	0.74	0.81	74.50
GI4E	0.05	8.68	0.80	24.07	66.45	0.90	0.12	0.20	32.74

Table 4.1: Detection average precision (%) on GI4E database for different τ values. Evaluation of the best *(dis)*similarity threshold τ value on GI4E over detection accuracy tested on 1236 images.

Data	$\mathbb{S}_{\chi^2}$ (τ)	TP [%]	FP [%]	TN[%]	FN [%]	Prec	Rec	$F_1 Score$	Acc [%]
EYaleB	**0.15**	79.61	14.19	4.46	1.72	0.85	0.98	0.91	**84.07**

Table 4.2: Test of τ value on 5751 of extended Yale-B face database

detected image I_1 (observed frequency) and a second histogram of a reference image (I_2). Both images are encoded with an LBP descriptor. EyeLSD uses $LBP^{riu2}_{16,2}$ histogram each pair of images.

$$\mathbb{S}_{\chi^2}(H,T) = \sum_{i=1}^{n} \frac{(H(i) - T(i))^2}{H(i) + T(i)} \tag{4.5}$$

where H and T are discrete sample and model distribution, respectively. They correspond to the probability of bin i in a given sample and a model distribution, n is the number of bins in the distribution ($n = 18bins$). To construct the eye template (reference image), a small corpus of left and right eye images were chosen among different people in the databases. These images of 24×24 pixels, represent open-and-closed eyes. Template separates the 2-class using $\mathbb{S}_{\chi^2}$ distance. Our aim is to detect eyes whatever their state and reject most probable non-eye according to τ value.

To find the best threshold value incorporated in our EyeLSD algorithm for an accurate estimation of the position of the eye for a given image or video, an interpretation of the results obtained from tests carried out on three databases is presented in Table 4.1, Table 4.2, and Table 4.3. The final results are expressed through TP, FP, TN and FN. recognition accuracy, *precision* (prec), *recall* (rec) and F_1-*score*. It should be noted that $\mathbb{S}_{\chi^2}$ values vary under hard imaging conditions (e.g., variability in terms of scale change and uneven light). This may lead to ambiguous eye state detection.

More stringent evaluation tests are shown by examples of successful location of eyes on used databases Fig. 4.3, Fig. 4.4, and a video is shown in `https://www.youtube.com/watch?v=P2ICKE6ALWs`. The obtained eye center location (**green** circle) are validated within the pupil radius. The different bounding-box colors correspond to various matching objects, detection results cover the correct detection (eye position) and false alarms (missed positive); **the blue** rectangle is an eye, this detection is carried out from (χ^2) measure, which is used to verify the presence of eyes with $\tau = 0.15$. **The magenta** rectangle is the classifier's false positive, rejected by the (χ^2) measure (considered an insignificant information by the algorithm). It must be emphasized that distance thresholding step is also integrated in an algorithm of eye localization and head motion estimation, the EyeLHM algorithm [186].

Data	$\mathbb{S}_{\chi^2}(\tau)$	TP [%]	FP [%]	TN[%]	FN [%]	Prec	Rec	$F_1 Score$	Acc [%]
GI4E-HP user-08-video-12	**0.15**	97.35	0.0	0.0	2.64	1.0	0.97	0.98	**97.35**
GI4E-HP user-03-video-04	**0.15**	96.08	0.0	0.0	3.92	1.0	0.96	0, 98	96.08
GI4E-HP user-02-video-03	**0.15**	91.50	0.37	0.0	8.13	0.99	0.92	0.95	91.5
GI4E-HP user-01-video-01	**0.15**	95.33	0.0	0.0	4.66	1.0	0.95	0.97	95.33
GI4E-HP user-07-video-08	**0.15**	96.85	0.0	0.0	3.14	1.0	0.96	0.97	96.85

Table 4.3: Test of τ value on GI4E head-pose database. GI4E head pose video sequences are resized to 360×240 pixels, and have been acquired at 30 frames/s. Every video is 10 seconds long, containing 300 frames.

Figure 4.3: Results of the analysis of the eye location by the EyeLHM [186] system that validates threshold value for an accurate detection of the eye under difficult conditions of GI4E dataset: the blue rectangles represent the precise eye location, the magenta rectangles are the false positives rejected from similarity measure, the green points represent the center of the eye detection.

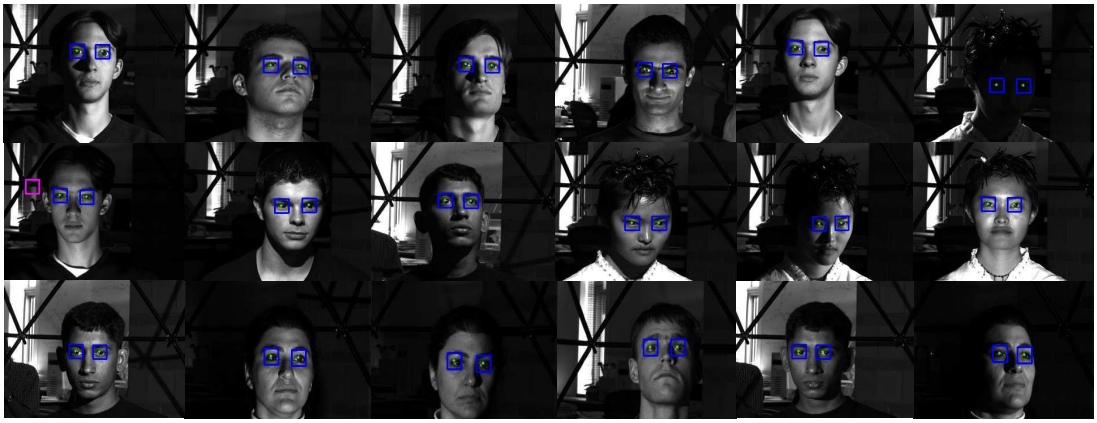


Figure 4.4: Results of the analysis of the eye location by the EyeLHM algorithm [186] to validate threshold value τ for an accurate eye localization under difficult conditions of Extended Yale-B dataset.

- **Classification**

4.1.5 Classification and Parameters Settings

For the classification phase, we implement a Support Vector Machine and a deep perceptron. These classifiers are favorable because, their implementation is open-source, modular, accessible and computationally efficient. EyeLSD formulates the eye localization step as a 2-class classification problem.

- SVM classifier is trained with two kernels (a linear and a nonlinear kernels) that help to find the hyper-plane, which maximizes separation gap between the two classes, while minimizing the number of errors for the training set. The hyper-parameters of the radial basis function (RBF) kernel are optimized by using grid search technique and the cross-validation group (CV). The best values over the CV group were used to build the learning model. The SVM implementation is done with LIBSVM 3.18 [87].

- We trained and tested the MLP network with several configurations, and changed the number of hidden neurons. We keep the number of maximum training epochs constant of 10000. Sigmoid functions were selected as transfer functions. The MLP classifier is fully connected and designed with a number of input neurons, equal to the length of each feature descriptor, i.e., 1440, 3776, 59, 40 ,90 ,236 and 390 input neurons for Gabor, LTP, LBPH^{u2}, eLBPH^{riu2}, ePLBPH^{riu2}, eLBPH^{u2} and ePLBPH*, respectively. BioID dataset and CAS-PEAL-R1 dataset validate the eye detector. MLP has a single hidden layer for each of Gabor, LTP, LBPH^{u2}, eLBPH^{riu2}, ePLBPH^{riu2}, eLBPH^{u2} and ePLBPH*, embedded with 120, 200, 20, 15, 15, 50, and 22 hidden units, respectively. The output vector for positive samples is $\mathbf{Y}_i = (1,0)^T$ and output vector for negative samples is $\mathbf{Y}_i = (0,1)^T$. The *Softmax* function is employed in the output layer and the number of output units is equally related to the number of classes. Regularization terms are used to improve the convergence velocity and avoid settling down in an over-fitting problem. The connecting weights ω are randomly initialized in a range of

$(-0.1, 0.1)$, momentum and learning rate are assigned as α and ξ, respectively. During the experiments, the exposed neural architectures are optimal, which found to be a good compromise between classification error minimization and architecture complexity. These neural structures provide a good generalization performance for new data and are valid for the various feature sets.

4.1.6 Dataset Description

In order to test accuracy of the eye classification models, we measured classification performance on the three benchmark datasets (two public datasets and a self-made dataset) presented below:

- **BioID Face database** [187], the BioID dataset contains images in real scenarios with a various illumination, background, and face sizes with and without accessories. The dataset contains 1521 images ($384 \times 286 pixels$, gray level).

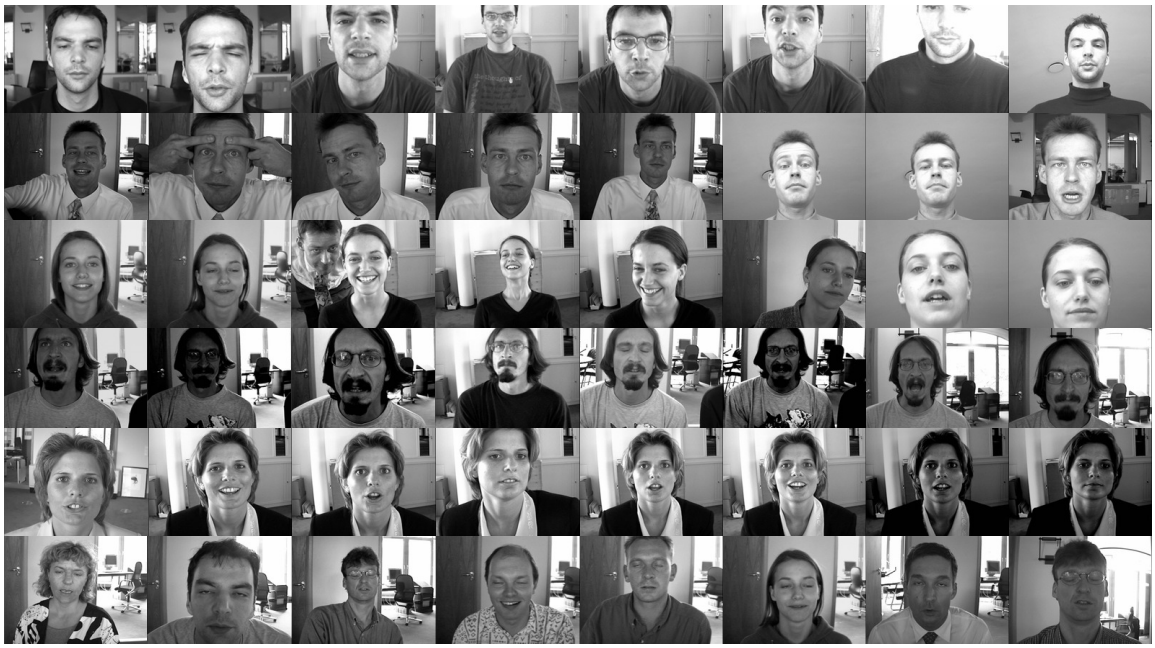


Figure 4.5: Sample images of BioID database.

- **CAS-PEAL-R1 database** [188], contains 30900 images ($360 \times 480 pixels$, gray level). This dataset has images acquired in realistic conditions. In our experiments, 1521 face images are selected, 464 subjects with open eyes: frontal slight rotated view; normal and expressions. 330 subjects with accessories, 101 subjects in different background, 302 subjects with eye closed and 324 subjects in different distances from the camera.

- To undertake a generalization tests of EyeLSD algorithm, we collected and annotated a small set of images of different subjects. The acquisition condition are diverse, which involves a low ambient lighting conditions, wide head rotation, open and closed eye states, facial expressions,

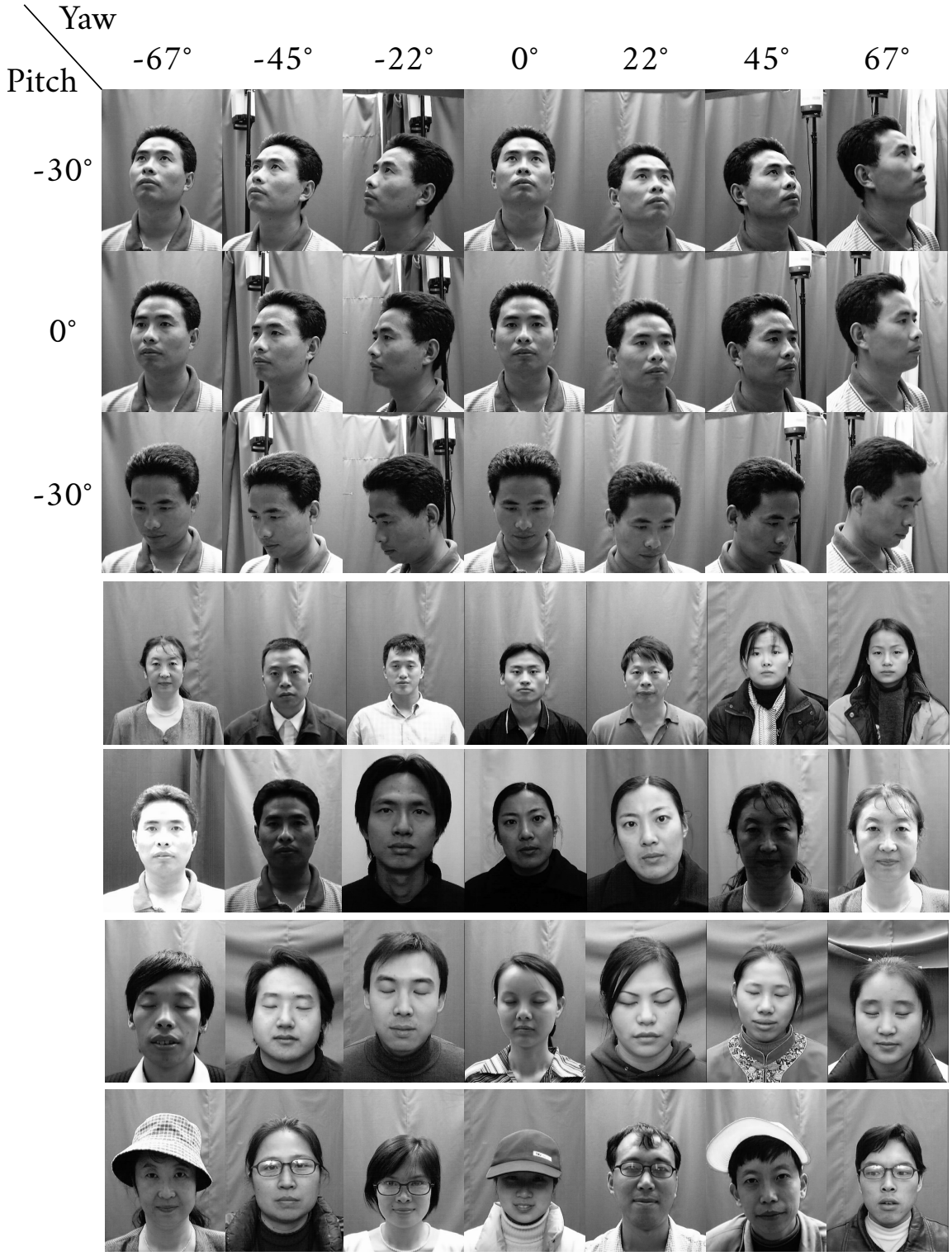


Figure 4.6: Sample images of CAS-PEAL database [188].The first row corresponds to the different head poses included in the database. The second row until the last one, correspond to the multiple variations introduced in the database, depth-of-field, lighting, facial expressions (e.g., eyes closed) and individuals wearing accessories (e.g., eyeglasses and hats)

change of perspective and depth-of-field. All these images are taken with simple web-camera at very low cost, providing images of resolution 640×480 pixels and resized during tests at 384×240 pixels. We present below some characteristics of our annotated image data:

- Eight subjects are presented in our database: all are male.
- The subjects present a facial hair (beard) of different intensities, head pose, facial expression (broad smile, eye closed), different head poses and depth-of-field.
- The images of the database are acquired during the afternoon under very low lighting conditions, i.e., indoor with artificial lighting and a web-camera, without any additional light source.

We have built datasets composed of images from BioID and CAS-PEAL-R1 datasets, each of which has 3042 eye images and 3419 non-eye images that are manually cropped. 4523 eye and not-eye images are rearranged into a training subset, and the remainder 1938 images with positive and negative samples are confounded and equally divided between test and validation subsets. These images are geometrically normalized into a size of 24×24 pixels.

4.1.7 Experimental Results

This subsection concerns the test phase of the experimentation realized to determine the optimal model configuration for eye localization. The BioID dataset and CAS-PEAL dataset are during evaluation. To prove the EyeLSD generalization capability, a new dataset was acquired with people from our laboratory with a normal laptop web-camera. This allows to assess generalization capability of our detector, especially with unseen images (images of people not present in the training datasets).

4.1.7.1 Feature descriptor classification performance

This subsection compares the performance of different image descriptors of eye features, results are presented in Tables 4.4 - 4.5. The objective is to show that the enhanced pyramid LBP with high number of bins improves the discriminative power for eye representation. The first experiment **Experiment # 1** compares performance between eLBPH^{riu2} and ePLBPH^{riu2} to classify the eye presence. The second experiment **Experiment # 2** compares the performance between eLBPH^{u2} and the ePLBPH*. In the third experiment **Experiment # 3** the categorization capability of the ePLBPH^{riu2} is compared against the proposed ePLBPH*. The last experiment **Experiment # 4** analyzes the benefits of ePLBPH* by comparing its performance against those realized by LBPH^{u2}, LTP, and Gabor feature descriptors.

1. **Experiment # 1**: comparison between eLBPH^{riu2} and ePLBPH^{riu2}. This experiment intends to highlight performance gained if more than a single resolution of descriptors are used in the pyramid image generation. Table 4.4 shows the descriptors performance on BioID dataset. The best classification accuracies generated by eLBPH^{riu2} are 83.78% and 87.29% for SVM(RBF) and MLP classifiers, respectively. By adapting an extension of LBP^{riu2} in a pyramid transform domain(ePLBPH^{riu2}), the approach realized improvements of 8.16% and 6.52% with SVM(RBF) and MLP, respectively. Table 4.5 shows the performance of

Method	TP (%)	FP (%)	TN (%)	FN (%)	Prec	Rec	F_1Score	Acc (%)	AUC (%)
SVM(Linear)									
LTP	45.25	3.04	49.84	1.86	0, 93	0.96	0.95	95.10	99.00
LBPH^{u2}	45.20	11.30	41.60	1.90	0.80	0.96	0.87	86.79	94.00
Gabor	44.37	2.94	49.95	2.73	0.94	0.94	0.94	94.32	98.26
ePLBPH*	45.49	2.62	50.28	1.59	0.95	0.97	0.95	95.78	99.09
eLBPH^{u2}	45.73	3.79	49.10	1.36	0.92	0.97	0.95	94.84	98.91
eLBPH^{riu2}	39.02	9.61	43.29	8.06	0.80	0.83	0.81	82.32	90.53
ePLBPH^{riu2}	41.42	6.14	46.76	5.68	0.87	0.88	0.87	88.18	94.95
SVM(Poly)									
Gabor	43.91	4.54	48.35	3.20	0.91	0.93	0.92	92.26	97.00
ePLBPH*	27.39	1.45	51.40	19.74	0.95	0.58	0.72	78.80	95.81
eLBPH^{u2}	7.17	0.42	52.53	39.86	0.94	0.15	0.26	59.71	94.91
eLBPH^{riu2}	30.48	11.25	41.65	16.55	0.73	0.65	0.68	72.14	81.42
ePLBPH^{riu2}	39.35	10.18	42.73	7.74	0.79	0.83	0.81	82.08	90.03
SVM(RBF)									
LTP	45.30	2.89	50.00	1.80	0.94	0.96	0.95	95.30	99.00
LBPH^{u2}	43.75	3.82	49.07	3.35	0.92	0.92	0.92	92.83	98.00
Gabor	42.10	0.31	52.58	5.00	0.99	0.89	0.94	94.69	99.00
ePLBPH*	45.59	1.78	51.12	1.50	0.96	0.97	0.96	96.72	99.58
eLBPH^{u2}	46.15	1.59	51.31	0.93	0.96	0.98	0.97	97.47	99.81
eLBPH^{riu2}	44.69	8.25	44.69	7.97	0.84	0.85	0.85	83.77	91.84
ePLBPH^{riu2}	43.43	4.45	48.5	3.66	0.90	0.92	0.91	91.93	97.76
MLP									
LTP	45.61	1.18	51.75	1.44	0.97	0.97	0.97	97.37	99.00
LBPH^{u2}	44.27	3.50	49.43	2.78	0.92	0.94	0.93	93.70	98.00
Gabor	46.18	1.23	51.70	0.98	0.97	0.98	0.97	97.78	99.00
ePLBPH*	46.38	1.12	51.73	0.75	0.97	0.98	0.98	98.12	99.06
eLBPH^{u2}	45, 87	1.17	51.78	1.17	0.97	0.97	0.97	97.65	99.31
eLBPH^{riu2}	41.42	7.08	45.87	5.63	0.85	0.88	0.86	87.29	93.33
ePLBPH^{riu2}	45.17	4.27	48.64	1.92	0.91	0.96	0.93	93.81	97.64

Table 4.4: Eye detection: statistical results on BioID database

Method	TP (%)	FP (%)	TN (%)	FN (%)	Prec	Rec	F_1Score	Acc (%)	AUC (%)
SVM(Linear)									
LTP	44.53	3.61	42.27	2.57	0.92	0.94	0.93	96.49	98.49
$LBPH^{u2}$	42.67	21.31	31.58	4.43	0.67	0.91	0.77	74.25	79.03
Gabor	45.51	2.47	50.41	1.60	0.95	0.96	0.96	95.92	99.20
$ePLBPH^{*}$	45.26	2.20	50.70	1.82	0.95	0.96	0.96	95.97	99.19
$eLBPH^{u2}$	44.55	5.15	47.74	2.53	0.90	0.95	0.92	92.31	99.51
$eLBPH^{riu2}$	34.00	15.71	37.24	13.04	0.68	0.72	0.70	71.24	78.60
$ePLBPH^{riu2}$	43.62	6.29	46.62	3.47	0.87	0.93	0.90	90.24	96.94
SVM(Poly)									
Gabor	44.42	3.15	49.74	2.68	0.93	0.94	0.94	94.17	98.76
$ePLBPH^{*}$	17.68	1.82	51.07	29.31	0.91	0.38	0.53	68.76	91.44
$eLBPH^{u2}$	0.93	0.79	52.11	46.15	0.54	0.019	0.04	53.05	83.20
$eLBPH^{riu2}$	21.95	13.27	39.63	25.14	0.62	0.47	0.53	61.58	71.27
$ePLBPH^{riu2}$	42.26	12.43	40.48	4.83	0.77	0.9	0.83	82.74	90.96
SVM(RBF)									
LTP	45.66	2.11	50.77	1.44	0.95	0.96	0.96	96.44	99.51
$LBPH^{u2}$	42.05	14.18	38.69	5.05	0.74	0.89	0.81	80.75	88.92
Gabor	42.20	0.41	52.47	4.90	0.99	0.89	0.94	94.68	99.56
$ePLBPH^{*}$	46.06	1.17	51.73	1.03	0.98	0.98	0.98	97.80	99.70
$eLBPH^{u2}$	45.82	2.48	50.42	1,26	0,95	0,97	0,96	96.25	99.39
$eLBPH^{riu2}$	39.63	9.95	42.96	7.45	0.80	0.84	0.82	82.60	91.17
$ePLBPH^{riu2}$	44.28	2.86	50.09	2.76	0.94	0.94	0.94	94.37	98.53
MLP									
LTP	45.04	2.22	50.72	2.01	0.95	0.96	0.95	96.65	97.65
$LBPH^{u2}$	42.31	5.46	47.47	4.74	0.88	0.89	0.89	89.78	95.55
Gabor	46.38	0.82	52.06	0.72	0.98	0.98	0.98	98.45	99.37
$ePLBPH^{*}$	45.77	1.31	51.64	1.26	0.97	0.97	0.97	97.42	97.97
$eLBPH^{u2}$	44.93	2.48	50.46	2.11	0.95	0.95	0.95	95.40	99.39
$eLBPH^{riu2}$	38.98	8.2	44.74	8.06	0.83	0.83	0.83	83.72	90.65
$ePLBPH^{riu2}$	42.26	2.62	50.28	1.83	0.95	0.96	0.95	95.54	98.11

Table 4.5: Eye detection: statistical results on CAS-PEAL-R1 database

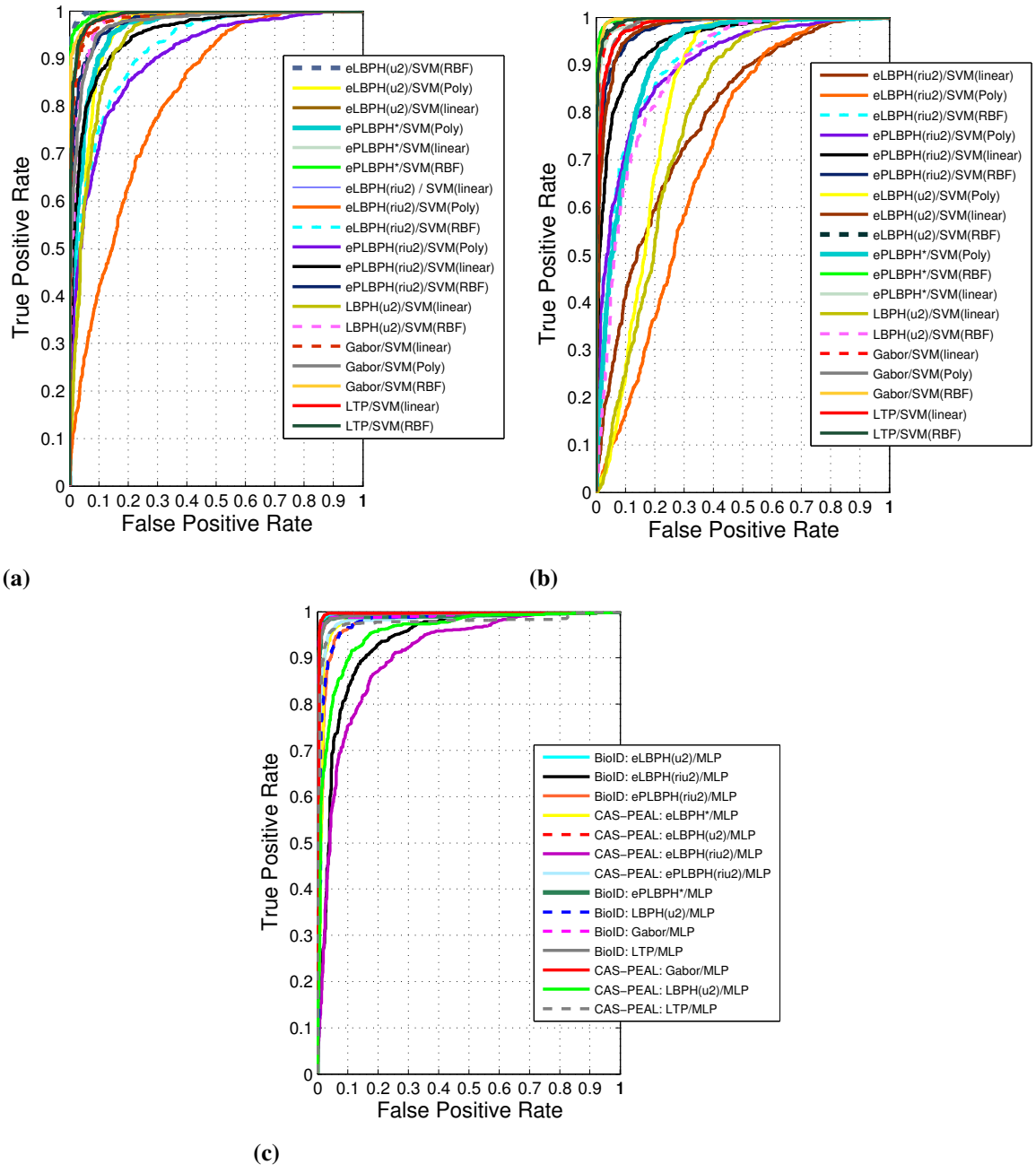


Figure 4.7: ROC curves of various features using the SVM classifier; (a) BioID database, (b) CAS-PEAL database. ROC curves of various features using the MLP classifier; (c) BioID and CAS-PEAL databases

ePLBPH^{riu2} and eLBPH^{riu2} assessed on CAS-PEAL-R1 image set. The best classification accuracy of eLBPH^{riu2} are 82.6% and 83.72% with SVM solver(RBF) and MLP, respectively. The ePLBPH^{riu2} realized improvements of 11.77% and 11.82% with SVM(RBF) and MLP classifiers, respectively.

From the Tables 4.4 and 4.5, we conclude that the performance gain of ePLBPH is better than those obtained with eLBPH, by considering descriptors uniquely based on LBPH^{riu2}. The number of ePLBPH^{riu2} pyramid levels is set to 3. The histogram dimension of ePLBPH^{riu2} is $(10bins \times 4) + (10bins \times 4) + 10bins = 90bins$, that is more than 2 times of the eLBPH^{riu2} of size $40bins$ ($10bins \times 4$). The ePLBPH accounts the eye in 3 different resolutions, which increase the information content extracted and the classification rate. Although, the eLBPH describes micro- (edges, corners, spots, etc) and macro-textures (global shape) of the eye pattern, but only at a single resolution.

2. **Experiment # 2**: This experiment compares discriminative performance of eLBPH^{u2} and ePLBPH*(proposed) Table 4.4 shows the performance comparison of eLBPH^{u2} and ePLBPH* on BioID dataset. The ePLBPH* is built according to the Eq.(4.4), ePLBPH* = { eLBPH*, eLBPH*, LBP*}, $* \in \{u2, ri, riu2\}$. The average performance assessment of eLBPH^{u2} features, yields a best scores of 97.46% with SVM(RBF) and 97.65% with MLP classifiers. The performance gained by adopting several LBP variants in spatial pyramid domain (ePLBPH*) are 0.469% with MLP classifier. Table 4.5 shows the statistical assessment of eLBPH^{u2} and ePLBPH* on CAS-PEAL-R1 dataset. The eLBPH^{u2} achieves a best scores of 96.24% and 95.40% with the SVM(RBF)and MLP, respectively. Among these two configurations, a performance improvement of ePLBPH* over eLBPH^{u2} are 1.54% and 2.01%, with SVM(RBF) and MLP, respectively. As shown in Tables 4.4 and 4.5, the ePLBPH* performance increases comparing to those of eLBPH^{u2}. So, there shows that the major discriminant properties of ePLBPH* are got from the image into the pyramid basis. This level of the pyramid is preprocessed in the same way as eLBPH^{u2} descriptor. It is noteworthy that applying region division method to form eLBPH is somewhat arbitrary. The division approach spatially enhances the LBP histogram, but also causes both aliasing effect due to the direct sampling and loss of resolution information. The ePLBPH solves these drawbacks, by applying the LPF on the images before pre-processing and LBP histogram calculation, in the 0^{th} and the 1^{st} pyramid image levels. Results show that the proposed method (ePLBPH) achieves a good generalization performance on unseen image set. Hence, the LBP features of a 3-level image pyramid are efficient.

3. **Experiment # 3**: comparison of the pyramid descriptor performance, between ePLBPH^{riu2} and ePLBPH*. This experience intends to verify whether performance is further improved, in case of more than an unique LBP mapping scheme are used in ePLBPH. Tables 4.4 and 4.5 show the performance comparison of ePLBPH^{riu2} and ePLBPH*, both are realized with 3-level of image pyramid. Table 4.4 shows improvement realized by ePLBPH* over those of ePLBPH^{riu2} on BioID dataset, which are about 4.78% and $4,31\%$ by using SVM(RBF) and MLP, respectively. In Table 4.5 the classification performance of ePLBPH^{riu2} assessed on CAS-PEAL-R1 dataset, are enhanced by ePLBPH* about 1.87% and 1.875% with SVM(RBF) and MLP, respectively.

The ePLBPH* feature vector length is $(59bins \times 4) + (36bins \times 4) + 10bins = 390bins$, that is more than 4 times of the ePLBP^{riu2} histogram of $(10bins \times 4) + (10bins \times 4) + 10bins = 90bins$. The multi-mapping pyramid image ePLBPH*, achieves a higher enhancement and outperforms the three descriptors of comparison; eLBPH^{riu2}, eLBPH^{u2} and ePLBPH^{riu2}. This improvement the ePLBPH* is statistically significant and observed through the present and the precedent experiment. The ePLBPH* feature sets compensate the information losses during the spatial down-sampling. Also, down-sampling process does not affect the discriminative performance of the descriptor very much. Even so, the enhanced multiple mapping scheme improves significantly the discriminative power of the descriptor, by comparing to that of ePLBPH^{riu2}. This is shown through the experience # 1.

4. **Experiment # 4**: comparison performance ePLBPH* among Gabor wavelets, LTP and LBPH (LBPH^{u2}) feature sets . We extend the performance evaluation of ePLBPH*, by introducing a series of comparison with other feature sets, that describe the local shape, the global shape, and the local texture information under difficult conditions. Current feature sets offer quite good performance under illumination variations and many other variations of the real world.

Tables 4.4 and 4.5 give the results of LBPH, LTP and Gabor feature sets on BioID and CAS-PEAL-R1 datasets. Please note that the SVM (polynomial) tends to overfit by using LBPH and LTP. For this reason we omitted these results.

In BioID dataset, we can observe that LTP realized a best accuracies of 95.3% and 97.37% with SVM(RBF) and MLP with highest AUC value, while Gabor has a slightly worst accuracy of 94.69% and a competitive performance of 97.78% with the same settings (dataset and classifiers). The same observations are valid with experiments conducted on CAS-PEAL-R1 dataset.

In this study, we reproduced the LTP code implementation realized by [36]. The threshold value of the LTP code is set to 5 computed from eye and non-eye images of size 24×24 pixels, which are divided into 3×6 sub-region and each sub-region is represented in an LTP histogram of $59bins$. The resulting LTP feature vector has a 3776-dimensional $(32 \times 59 \times 2)$. The LTP feature set improves generalization of LBP features, and has a good discriminative capability, while being tolerant to lighting changes and less sensitive to noise in uniform regions. Gabor filter bank realized competitive results with those of the ePLBPH*, by using polynomial kernel of the SVMs, Gabor outperforms ePLBPH* in terms of classification accuracy. MLP performs well with Gabor parameterization. We observe that in CAS-PEAL-R1 dataset, MLP classifier performs better than SVM(RBF), where 98.45% of correct classification was obtained for Gabor parameterization, whereas for BioID, Gabor features achieve 96.72% and 97.78% with the SVM(RBF) and MLP network classifier, respectively.

This classification enhancement of Gabor features, can be made clear by that CAS-PEAL-R1 dataset contains variation in illumination and pose, but not blurred images, which enhances significantly classification results, since Gabor is a powerful feature descriptor especially for non-rigid texture such face [10] and eyes, whereas BioID databset presents addition variations than previously mentioned, images are blurred. This parameter decreases the efficiency of Gabor method in BioID dataset, but not of our method. The proposed ePLBPH* handles well resolution variations and blurred images. This is proven by the best accuracy achieved in BioID dataset of 98.12% with MPL network.

Gabor features is implemented with 40 filters [10, 36] (8 orientations and 5 scales) applied on 24×24 pixels eye and non-eye patches, then down-sampling the resulting vector by 16. So, instead of $(5 \times 8 \times 24 \times 24)$ that yields 23040-dimensional, it is reduced to a 1440-dimensional vector.

The LBPH^{u2} obtains the lowest accuracies of 93.7% and 89.78% with MLP classifier and AUC values of 98% and 95.55% on BioID and CAS-PEAL-R1 datasets. LBPH is not quite appropriate for ocular region description, LBP features are more effective when the eye patterns are pre-processed upstream (sub-region division method). LBPH is sensitive to noise and can slightly tolerate texture rotation but not invariant to that. So, a holistic representation of LBPs can not preserve image local structures in presence of noise, blur, and extreme rotation because the small pixel differences of the eye patterns, make the descriptor vulnerable to noise.

4.1.8 Detection results

In this subsection, we present some visual results of the eye localization, where EyeLSD method is applied on the precedent datasets. Despite excellent results listed in table 4.4 and table 4.5, demonstrating the strength of EyeLSD framework for accurate eye localization on the BioID and the CAS-PEAL datasets. A more stringent evaluation tests are further required, to show some successful location of eyes on current datasets Fig. 4.8, 4.10, and 4.12.

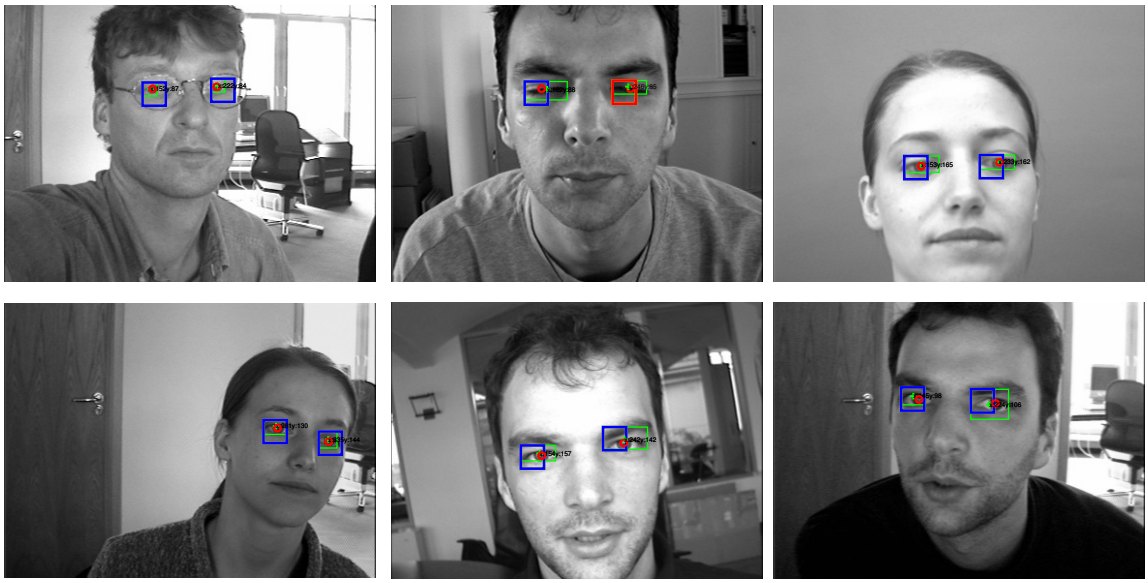


Figure 4.8: Example of some successful eye localization of our method on BioID Face images, including variations of pose, expressions, and even subjects wearing glasses of myopia.

Figure 4.9: Snapshots illustrate some successful detection on pictures captured within the laboratory.

In the present figures, the location of detected eye is represented by a green cross and the ground-truth of original eye location is represented by red circle, inside pupil radius (given in BioID image data and annotated manually in CAS-PEAL images). Different bounding-box colors correspond to diverse found elements. Detection results encompass desirable location of eyes, and false alarms are the misplaced detection; blue rectangle is an open eye and red rectangle means a closed eye. These are derived by applying $\mathbb{S}_{\chi^2}$ test, which is effective in bringing more accuracy to eye detection, but fails to detect whether they are open or closed. The yellow squares are the classifier false positive, denied by $\mathbb{S}_{\chi^2}$ test (considered a meaningless information by the algorithm).

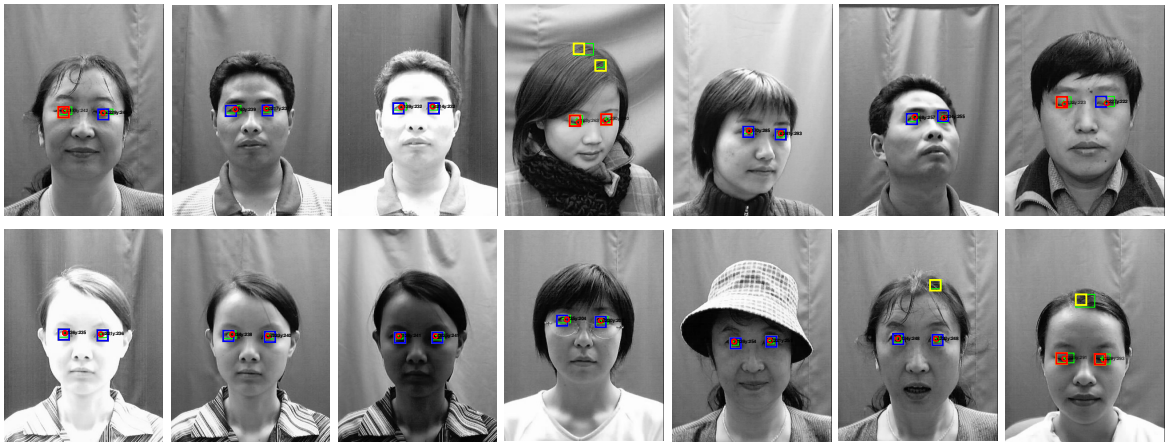


Figure 4.10: Locating eyes on images of the CAS-PEAL Face Database: the green cross corresponds to the output of our system and the red circularly form is the ground truth of the real eye coordinates, this needs to be marked manually in CAS-PEAL-R1 database.

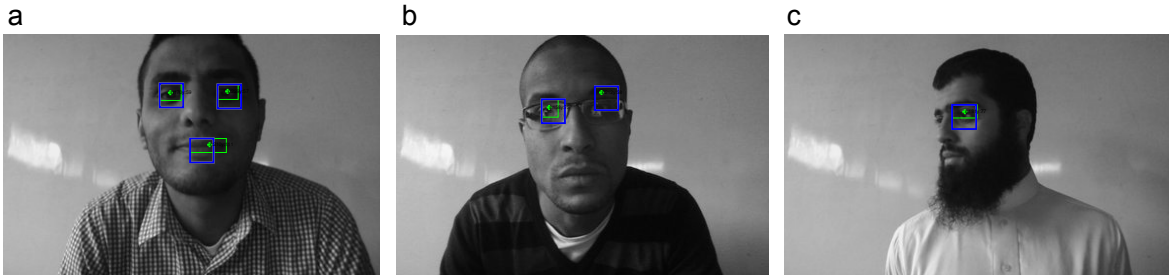


Figure 4.11: Snapshots from captured pictures illustrating some typical failures of our method: individuals with different head rotation angles, facial expressions and with accessories (myopia glasses, beard, expressions).

Figure 4.12: Some eye localization of EyeLSD [140] in challenging cases with hard variations in pose and facial expression: from top to bottom, the first three rows, represent the location of the eyes of different subjects with various head pose on CAS-PEAL-R1 database (e.g., from right to left, the first estimation of the eye location is made for a subject with a head pose of a yaw-angle of $-67°$ and a pitch-angle of $-30°$). The last row represents eye localization results of subjects variant facial expression on BioID Face Database.

4.1.9 Comparison with other methods

Table 4.6 shows most recent eye localization methods we are aware, beside of the proposed one. Withal, corresponding experimental settings are presented such as, image dataset, beside of its characteristics (number of test images with their corresponding size) and the final performance realized by each algorithm. The overall performance of the proposed eye location estimation scheme is similar, and in some cases better than algorithms of comparison. The common points between our eye detector and the listed methods (Table 4.6) is that the presented approaches depend on the appearance and image patch based methods, either by using supervised (SVM and MLP) or unsupervised learning methods (Boltzmann machine model and Independent Component Analysis (ICA)). These methods are using the same benchmark image gallery.

In [190], authors investigate the problem of eye localization for subjects wearing glasses under different constraints. This method is based on variance Filter (VF) that measures the gray intensity change of the eyes and ICA applies to recognize the correct eye location. The eye center is searched within detected ocular regions, by calculating the entropy of intensity change. [190] method realized a detection rate of 97.1% with 600 test images of BioID dataset. Nonetheless, the detection fails if the light reflection on glasses is too strong and occlusions occult the facial region.

In contrast, the proposed framework surpasses aforementioned methods, realized excellent results and capable of discriminating the ocular region under real challenging conditions. In [191], authors proposed a deep-learning algorithm for detecting the eyes under uncontrolled conditions. The robustness of the detection scheme is tested on different datasets and in different conditions (facial expression, low-resolution, pose, and illumination). The potential of [191] approach to handle the resolution variations, is assessed on BioID dataset. The eye images are evaluated with their original size and with images down-sampled to 50%. So, two resolutions of eye image patches are generated and used to show ability of the deep features trained model, to recognize the right eye location despite low-resolution images.

In [193], they are proposing an eye detection framework able to estimate the eye region location with precision. The input image is pre-processed to highlight the eye structure, that is used after the localization step. So, the eye pair is extracted by using binary template matching and SVM classifier. The next step consists to accurately detect the eyes by using VF. This algorithm is trained with 800 images collected from BioID database, and yielded a detection rate of 95.6% . However, the detector fails under hard illumination conditions, strong light reflection and closed eyes. This occurs mostly because the template matching step fails to find the right eye pair location.

Boltzmann-deep features learner is tested on 956 images of BioID dataset, and achieves a competitive results to those of the LBP features with Viola-Jones eye detector. The main advantage of Viola-Jones approach is its computational effectiveness. However, the performance of that method depends on the amount and diversity of the training data. So, it may not give a right eye location. Meanwhile, building a classifier that learns the variability of eyes might meet with problems, and even if a large set of training image is used. By using a descriptor window enlarged to $36 \times 36 pixels$, we are able to surpass performance realized by LBP [191] and Viola-Jones [192] methods. Intuitively, a larger eye patch of $36 \times 36 pixels$ has more discriminative information, and thus, reduces the false positive rate but it will be at the cost of losing generalization ability for locating eyes, and the extracted features will be less likely to be good representative of eyes [98]. Furthermore, our method shows robustness against the rotation of face area and extreme pose, while that most errors occurred

Method	data	challenge	**patch** size (pixels)	test images	**Acc.** (%)
EyeLSD with ePLBPH*(Proposed)+ MLP [140, 189]	BioID	Variation in distance (multi-resolution) Illumination condition, occlusion, gender, aging, expression, hard rotation.	**24 × 24**	969	**98.12**
EyeLSD with ePLBPH*+ SVM(RBF)	BioID	Variation in distance (multi-resolution) Illumination condition, occlusion, gender, aging, expression, hard rotation.	**36 × 36**	969	**98.86**
VF+ ICA [190]	BioID	Variations in views, lighting conditions, occlusions aging, expressions	60 × 30	600	97.1
Learning the Boltzmann [191] Machine model.	BioID	Expression, illumination, pose, low resolution	**36 × 36**	956	**98.12**
LBP [191], Viola-Jones [192]				956	**98.64**
Learning the Boltzmann [191] Machine model	BioID	Expression, illumination, pose, low resolution	18 × 18	956	98.12
LBP [191], Viola-Jones [192]				956	98.01
VF + SVM [193]	BioID	Dynamic background, moderate rotation, glasses wearing and face occlusions	25 × 8		95.6
SIFT features+ SRC [103]	BioID	Slight variation in pose illumination changes various background face sizes image rotations expressions	60 × 60	1000	91.5

Table 4.6: Comparison of eye location step of the algorithm EyeLSD with existing methods.

in the Viola Jones eye detector, are related to the facial textures rotations and head pose variations.

From Table 4.6, the results of the proposed approach are comparable with those obtained by Boltzmann machine model and LBP features with Viola-Jones methods for eye localization.

In [97] authors proposed a learning method for eye localization in arbitrary rotation settings. They examine the feasibility of localizing eyes without prior face detecting. A pyramid-like eye locating strategy is used for coding local-features, ensured with SIFT descriptor and SRC method, classifies the image patches through input image. Then, a searching map called (Heat-Map) is generated from the adjusted classifier's outputs, and the potential positions of eyes are highlighted. (The Heat-Map highlights eye centers by superposing the adjusted classifier's output values through Pyramid-like method). To locate the eye center, while reducing noise effect and influence of complex backgrounds, the skin color algorithm is applied in HSV color space, that improves skin detecting and isolates the facial region from the background. The false positives detected around the real center of eye positions are rejected by using similarity function and the center of eyes are retained if a maximal similarity score is reached. Their method are assessed on several datasets. For a fair comparison with our approach, we consider only tests performed on BioID database, which achieve an accuracy of 91.5% on 1000 image patches of size 60×60. We observe that the accuracy of the proposed method is better than the method of [194], tested on 969 images of same dataset.

Among all the the listed methods, our ePLBPH* achieves the highest classification score on BioID dataset. This enhancement is attributed to the combination of multiple LBP-mapping schemes and the application of spatial enhanced pyramid-like image decomposition strategy. The ePLBPH* tolerates several changes, illumination, image blur, perspective and rotation. So, the proposed eye detector is robust against the multiple variations, that the presented works shown their distinctiveness performance limitations, Fig. 4.8, 4.10,and 4.12.

4.2 Conclusion

In this chapter, we present EyeLSD algorithm for eye localization, using three LBP descriptors, preceded by preprocessing phase, to extract the most relevant information of the eye textures. EyeLSD includes two classifiers (SVM and MLP classifiers) for the binary classification phase between the eyes and non-eyes. These steps fit into the overall mechanism of the EyeLSD algorithm. We performed series of tests on the public databases BioID and CAS-PEAL-R1, to determine the optimal parameters of the two previous steps, namely the extraction of features and their classification. Subsequently, we compared our approach with the most recent work using the same database. EyeLSD outperforms these approaches and also our previous eye detection method presented in Chapter 3. These results prove that EyeLSD is very robust for eye localization, even when the environment of the acquisition is not controlled. This is highly favorable to the application for the next step of the EyeLSD algorithm, namely the recognition of the eye states.

CHAPTER 5

Estimation of eye states (open and closed)

Contents

5.1 Introduction

Understanding the eye states (open and closed) is a fundamental issue to a vast range of face-applied research work. In driver monitoring system, detecting eye states is a challenging task because the appearing of eyes may be unique for each face. There are many ambient factors that may modify the appearance of the eyes (see Chapter 3). Ineffective eye localization may present a considerable obstacle for this task.

The study of the eye states served as a trigger for a series of studies of the affective and physiological states of the human being [195, 196]. However, many of these studies focus on approximative detection of the ocular region, which is generally treated as a simple pre-processing step in the overall structure of the detection algorithm. Since it has been already mentioned a coarse detection of the eyes generates a high rate of false alarm when detecting their states. In order to solve this problem, the previous chapter dealt with the precise localization of the eyes and in the present chapter, we present final step of the EyeLSD algorithm, which detects the eye states (open or closed) within a given face image.

5.2 Related work of eye states detection

Fengyi Song et al. [197] classified methods for detecting eye states into two classes: feature-based methods and appearance-based methods.

5.2.1 Feature-based methods

Feature-based methods, generally use geometric characteristics of the eye, such as the visible iris and the elliptical shape of the eyelids [195, 198, 199, 200, 201]. Extraction of these components serves to differentiate a closed eye from an open one. Other material may also be used, such as variations of intensity distribution between an eye open and a closed one. This variation is produced by the presence and absence of iris and white region of the eye image. Accumulation of gray intensity through horizontal or vertical projection on a roughly detected eye, makes projection curves that show different shapes between closed and open eyes [202, 203, 204, 205]. These curves reflect the global intensity distribution and is vulnerable to inaccurate location and to various environmental changes.

5.2.2 Appearance-based methods

This type of method comprises two main steps: extraction of useful visual features from the photometric aspect of the eyes and their classification. Such methods are advantageous over other types of detection methods by their ability to provide richer, more reliable information for the subsequent classification step. Comparing to other methods, appearance-based methods have the ability to process low quality images and also handle more variations that eye pattern may undergo.

González-Ortega et al. [20] proposed a real-time visual method to find the eyes and recognizes their states (eye open or eye closed). Their eye state detector is based on a hybrid approach, which combines appearance and shape features of the eye. The algorithm works well in different conditions. However, it may fail under poor imaging conditions (low resolution, blur, and uneven light) that lead to ambiguous appearance of the eye. Cui Xu et al. [206] considered the detection of eye states as a binary classification problem, this means that eyes are classified into one of the two categories: closed or open. The eye image is first scanned with a series of scalable sub-windows, where is extracted LBP histograms. Then, for each sub-window, an optimal reference template is trained. Based on reference templates, the bin-wise statistical distances between extracted histograms and the corresponding templates are calculated to build a training feature set. This set and AdaBoost-based classifier are used to locate the eyes and to recognize their states. In theory these feature-descriptors can tolerate slight texture rotation. However, they are not invariant to high rotations. Moreover, the length of their LBP feature sets are relatively long, which may increase the computational cost. Fengyi Song et al. [197] proposed an eye closeness detection approach in still face images. In their work, face portion is firstly detected and cropped, then enhanced pictorial structural model [114] is adopted to find the eye locations. To define the eye state, they combine local and global structural appearance of eyes to build their state model. This state model uses a multi-scale histogram of principal oriented gradients (MultiHPOG) features. Their algorithm was validated on challenging eye datasets. Hashem Kalbkhani1 et al. [31], proposed an algorithm that estimates open and closed status of eyes in colored images. Their framework consists to crop the facial region first, and then

retaining only 60% of its upper area, which will be pre-processed after. The eyes are detected in this predefined region, using an improved version of EyeMap algorithm [103]. The eye is set as open if the number of white pixels inside the iris circle in a binary subspace is more than the number of black ones. Otherwise, the eyes are closed. Their algorithm achieves high recognition rate, and does not require training data. This framework is not applicable if two eyes are not completely visible in case of extreme face rotation for example. Ralph Oyini MBouna et al. [17] presented visual analysis of eye state and head pose for continuous monitoring of the driver alertness and determine the driver drowsiness or distraction level. Their scheme uses visual features such as eye states, pupil activity, and head-pose to dangerous behavior of the driver. SVM classifies the previous visual features in a sequence of video segments, for establishing whether the subject is alarmed or not, under realistic driving conditions.

The main advantages of detecting the consecutive eye closure in a video stream to determine the drowsiness of the driver are the simplicity of the technique and its adaptation to the realistic scenario. From the work presented above, we can affirm that a reliable detection of the ocular states requires precise positioning of the eye. This problem was solved in the previous chapter. In the following, the task of detecting ocular conditions is treated as a problem based on the change in the photometric appearance of the eye.

5.3 Estimation of the eye states

5.3.1 Patch-based LBP methods

The traditional forms of the LBP method and its derivatives are criticized, for coding only local information (micro-structure) and not being able to capture global information (macro-structure) of the image textures, which is sometimes dominant and highly necessary. LBP patch-based variants aim to handle this issue by describing larger areas of the image. There are, several mechanisms for introducing non-localities, including LBPF [136], LBP with Three Patches (TPLBP) [133], LBP with Four Patches (FPLBP) [133]. These methods have in common the geometry of patches used (rectangular, square or pixel arcs), the pre-processing step such as the filtering process (raw pixels or filtered values) and similarity calculation between each patch around a central one. if a single or multiple patch rings are used and directional or gradient information is captured.

In this section, a family of patch-based descriptor is adopted to encode further types of micro- and macro-textures of the eye images. The proposed descriptor is a Three-Patch LBP increasing in resolutions, preceded by a *Gaussian filtering*. Thus, enhances the discriminative power of the original TPLBP, which basically encodes the similarities between pixels neighboring patches of the image in different resolutions, and hence captures a complementary information to that of pixel-based descriptor.

5.3.2 Feature description using growing multi-resolution TPLBP combined with Gaussian filtering (Multi-TPLBP)

The Multi-TPLBP extends the TPLBP descriptor [133] (see Fig. 5.1) by calculating over different scales (multi-resolutions) of an image. The TPLBP of a pixel is obtained by comparing the values of three patches to provide a single bit value in the code assigned to the pixel. TPLBP for each pixel

is computed by taking a window $\omega \times \omega$ of region centered on the pixel and considering m sampling points in a perimeter of radius r pixels.

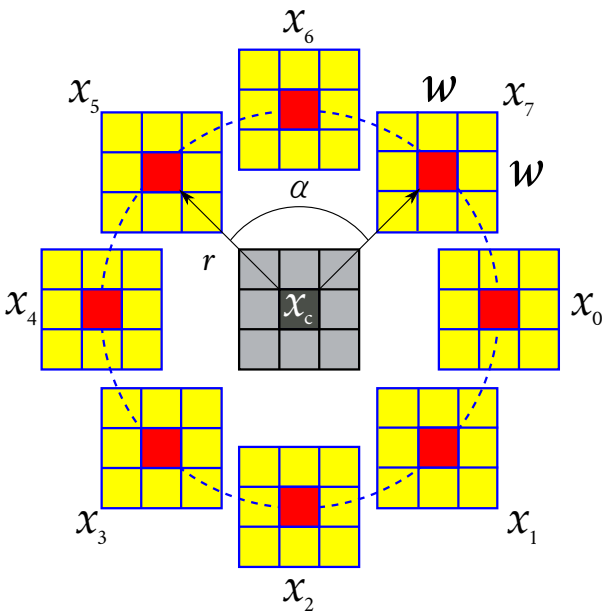


Figure 5.1: The Three-Patch LBP (TPLBP) descriptor [133, 143].

The TPLBP takes m patches around m pixels in the neighborhood, distributed uniformly on every side of the center patch. The inter-patch comparison in TPLBP is made by comparing the value of the center patch with a pair of patches that are α patches apart along the neighborhood circle. The value of a unique bit is set according to the similarity $d(.,.)$, of the two patches with the center patch. The function $d(.,.)$ is any similarity distance function between two patches (L_2 *norm* in our case). The resulting code has m bits per pixel and denoted as $\text{TPLBP}_{\mathbf{R},m,\omega,\alpha}$. Please refer to [133, 130] for more details about TPLBP operator.

The multi-resolution representation provides robustness to the original TPLBP, by collecting intensity information from a larger area, rather than the original single pixel. However, it might be noise sensitivity as sampling is made at single pixel locations, without preprocessing. The standard multi-scale mechanism counts some shortcomings and does not describe well the image textures due to following reasons:

1. The sampling is done at a single pixel location, rather than considering the effective region [136, 178].

2. The sparse sampling used by TPLBP in a large perimeter (radius) may not result in an adequate representation of the two-dimensional image signal, which may create an aliasing effect [136, 178].

3. The TPLBP is less stable by increasing the neighborhood radius, due to minimal correlation of the sampling points with the center pixel.

To solve this issues, an exponentially growing multi-resolution (Multi-TPLBP) is built by using low-pass filtering TPLBP (TPLBPF). The sampling positions are joint to a filtering process, to cover the neighborhood as well as possible and minimizing redundant information that may be collected by the operator. The Multi-TPLBP extracts both micro- and macro-structures of the eye pattern, which is a real need for efficient information retrieval and contributes positively to the description of the eye open and closed status.

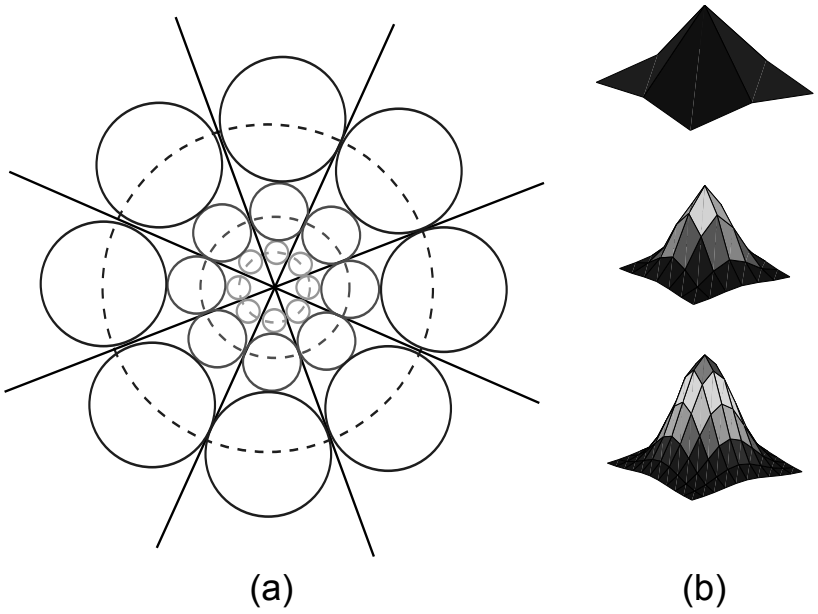


Figure 5.2: (a)The effective areas of TPLBPF and LBPF of filtered eye images in an $8-bit$ multi-resolution LBP operator. The dashed circles are the radius of the TPLBP rings. Sampling points P_n equally spaced circles with radius r_n (Eq. 5.1) and centered on the dashed circles with a radius $\mathbf{R}_n$, which are related to the effective region of each the image pixels. (b) Different Gaussian filter resolutions that can be used in the 1^{st}, 2^{nd} and 3^{rd} scales of the image [135, 136].

The eye image is preprocessed with a low pass filter (LPF), hence, the intensity information of a sample can be captured from a large area than the original single pixel, which is drawn with a solid circles in Fig. 5.2.

In LBPF, the m_n circles are with equal sizes and tangency [136, 178]. m_n circles are ensured to be tangency, if their radius is expressed as:

$$r_n = r_{n-1}(\frac{2}{1 - sin(\pi/m_n)} - 1), n \in \{2, \ldots, N\} \tag{5.1}$$

where N is the number of scales and m_n is the number of neighborhood samples at scale n. The low-pass filtering is useful only with radius larger than one for $m_n = 8$, r_1 is set to 1.5, which is the shortest distance between the center and the border of a 3×3 window.

The choice of the TPLBP radius is not randomly made, but according to the rule that the effective areas touch each other [135, 136] (see Fig. 5.2(a)). Therefore, the neighborhood radius at scale $n(n \geq 2)$ illustrated with dashed circles in the Fig. 5.2(a) is determined as follows:

$$\mathbf{R}_n = \frac{r_n + r_{n-1}}{2} \tag{5.2}$$

The effective area is realized with LPFs designed, so, that 95% of their mass lies within the solid circle [135, 136] (see Fig. 5.2(a)). The spatial size (width and height) of the Gaussian filter at scale n is calculated as follow:

$$w_n = 2\lceil \frac{r_n - r_{n-1}}{2} \rceil + 1 \tag{5.3}$$

where Eq. (5.3) is the symmetric weighting function with $(2K + 1)$ taps, that gives the Gaussian kernel f_G in terms of the rules of separable and symmetric. In this case Eq.(5.3) approximates the Gaussian function. Therefore, Eq.(4.1) can be reformulated as follows:

$$G_l = \sum_{m=-K}^{K} \sum_{n=-K}^{K} w(m,n) G_{l-1}(R_x x + m, R_y y + n) \tag{5.4}$$

where $R_x = R_y = 1$, which means no down-sampling used during the multi-scale image generation. The standard deviation of the Gaussian filter at scale n can be calculated from

$$\sigma_n = \frac{r_n - r_{n-1}}{2\sqrt{-2ln(1-\rho)}}, \rho \in [0, 1] \tag{5.5}$$

The effective areas in Multi-TPLBP, are realized with LPFs, where ρ in Eq.(5.5) is the probability that the mass of the distribution lies inside the solid circles of radius r (usually, ρ is set to 0.95).

To summarize, the procedure used for building the Multi-TPLBP is very similar to that used by [144]. The only difference lies with the neighborhood samples with radii greater than one are obtained via low-pass filtering. Furthermore, neighborhood radii are chosen following the rules presented before. The final Multi-TPLBP signature is obtained by concatenating the extracted TPLBP histograms at each scale. The Multi-TPLBP maps the eye image into $\mathbb{R}^{N \times d}$ representation, where d is the length of a single TPLBP code histogram at a scale n.

5.4 Classifiers

Classifier is an important component in the proposed architecture of the appearance-based eye states detection system. In this chapter, we use the SVM [86] and MLP [207] as our classifiers. The SVM method is trained with three kernels, namely linear and RBF kernels. Regarding, the neural configurations used to build the eye state models by using ZJU eyeblink image gallery [130], each neural model has 1-hidden layer with 12, 25, 100, and 160 hidden units, for Multi-LBP^{riu2} [136], Multi-LBP^{u2} [136], TPLBP [133] and Multi-TPLBP [140], respectively.

5.4.1 Dataset Description

To analyze the eye state detection accuracy of our proposed approaches, a dataset was selected:

- ZJU Eyeblink dataset [208], contains 80 video clips in the blinking video record of 20 individuals, four clips per individual, one clip in frontal view without glasses, one clip with frontal view and wearing myopia glasses, one clip in frontal view and black frame glasses, and the last

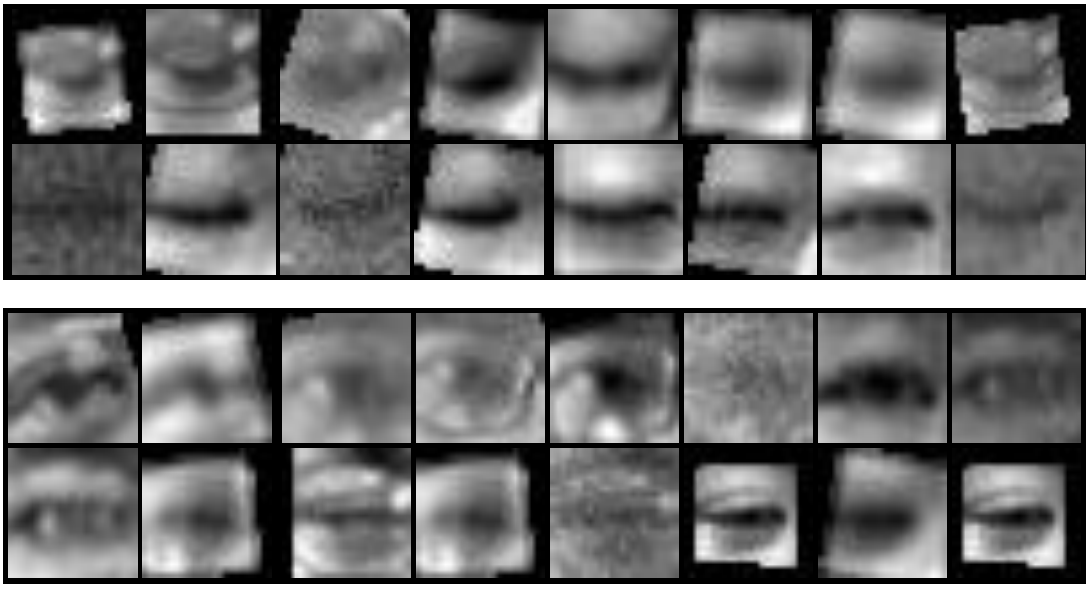


Figure 5.3: The pre-processed ZJU eye open and closed image gallery: patches in the top row are images of closed eyes, and patches in the bottom row are images of open eyes [130, 208].

clip with an upward view without glasses. The used dataset in our experiments is rearranged by [130], which expands the variety of the image samples, by adding various transformations, such as rotation, blurring, contrast, and Gaussian white noise.

The ZJU Eyeblink dataset [130], contains in the training subset 1574 closed eyes and 5770 open eyes, and in the testing subset 410 closed eyes and 1230 open eyes. Illustration of eye open and closed images in this dataset can be seen in Fig. 5.3.

5.5 Experimental Results

In this section, we first introduce a real-world dataset for algorithm verification. The performance of different feature descriptors on previous datasets. To verify the effectiveness of the proposed multi-scale extension of TPLBP, we describe and compare most recent methods with the proposed one. In this part, we consider two types of feature descriptors (patch-based and pixel-based LBP features), which can capture local and global texture information even under challenging conditions. In particular, we use the proposed extension of TPLBP histograms (Multi-TPLBP) described in Section 5.3.2 and multi-scale extension of two pixel-based LBP feature sets. The obtained results are given in Table 5.1.

In this experiment uniform pixel-based LBPs are extended to the multi-resolution representation and compared against patch-based LBP descriptors. In the proposed TPLBP extended to the hierarchical multi-scale sampling (Multi-TPLBP), the radius of the TPLBP rings is enlarged twice as well as joint to the LPF. During the multi-resolution image generation ($R_x = R_y = 1$), that means no down-sampling is made. Several constraints should be respected during descriptor construction

as described in Section 5.3.2. The 3-scale patch- and pixel-based descriptors are built with eight sampling points, and a radius values of $\mathbf{R} \in \{1.0, 2.4, 5.4\}$.

Table 5.1 gives a comprehensive classification performance comparison of precedent feature sets in terms of detection accuracy (Acc), TP(resp. TN), FP(resp. FN) and AUC. In this experience, TP (resp. TN) is the percentage of instances of eye open class (resp. eye closed class) well classified, while FP(resp. FN) is the percentage of instances of eye open class (resp. eye closed class) miss-classified. Several observations are made from this table.

Method	TP (%)	FP (%)	TN (%)	FN (%)	Prec	Rec	F_1Score	Acc (%)	AUC (%)
SVM(Linear)									
TPLBP	96.50	51.70	48.29	3.49	0.85	0.96	0.90	84.45	85.92
Multi-TPLBP	96.09	8.29	91.70	3.90	0.97	0.96	0.97	95.00	97.85
Multi-LBP^{u2}	96.09	18.29	81.70	3.90	0.94	0.96	0.95	92.50	97.45
Multi-LBP^{riu2}	96.74	60.48	39.51	3.25	0.83	0.97	0.89	82.43	86.94
SVM(RBF)									
TPLBP	96.01	49.51	50.48	3.98	0.85	0.96	0.90	84.63	86.82
Multi-TPLBP	96.34	8.29	91.70	3.65	0.97	0.96	0.97	95.18	97.83
Multi-LBP^{u2}	95.04	10.97	89.02	4.95	0.96	0.95	0.96	93.54	97.73
Multi-LBP^{riu2}	95.44	55.36	44.63	4.55	0.84	0.95	0.89	82.74	87.00
MLP									
TPLBP	96.01	47.80	52.19	3.98	0.86	0.96	0.91	85.06	87.63
Multi-TPLBP	96.17	8.53	91.46	3.82	0.97	0.96	0.97	95.00	98.12
Multi-LBP^{u2}	94.30	11.95	88.04	5.69	0.96	0.94	0.95	92.74	97.40
Multi-LBP^{riu2}	94.47	51.70	48.29	5.52	0.85	0.94	0.89	82.92	86.59

Table 5.1: Eye state: statistical results on ZJU database

1. we can see that the Multi-LBP^{u2} realized an enhancement over the performance of Multi-LBP^{riu2}, which is about 10.07%, 10.8% and 9.82% with SMV(linear), SVM(RBF) and MLP, respectively. At the first sight, a reasonable improvement of Multi-LBP^{u2} is realized, compared to the Multi-scale LBP^{riu2}. One difference can, however, arise between these two pixel-based feature sets, that is attributed to the feature vector length. Also, despite being invariant to rotation (the operator is tolerant to the texture rotation), Multi-LBP^{riu2} captures the uniform texture information even if rotation happens, and supports the major part of the texture information. However, the descriptor is too short and maybe the eye texture can be not reliably represented.

   The 3-scale LBP^{riu2} (Multi-LBP^{riu2}) generates a histogram of $10bins \times 3 = 30bins$, which is less than six times the length of the Multi-LBP^{u2} histogram of $177bins$. In addition, uniform LBP can represent the most local structures of the eye image which are represented by uniform codes, while the noise patterns most likely fall into the non-uniform codes. The extension of this descriptor in the multi-resolution domain allows the descriptor to capture additional information than the basic representation. In that way Multi-LBP^{u2} extracts the local information about the eye state, the global shape of the eyes, and the deformation of their textures. How-

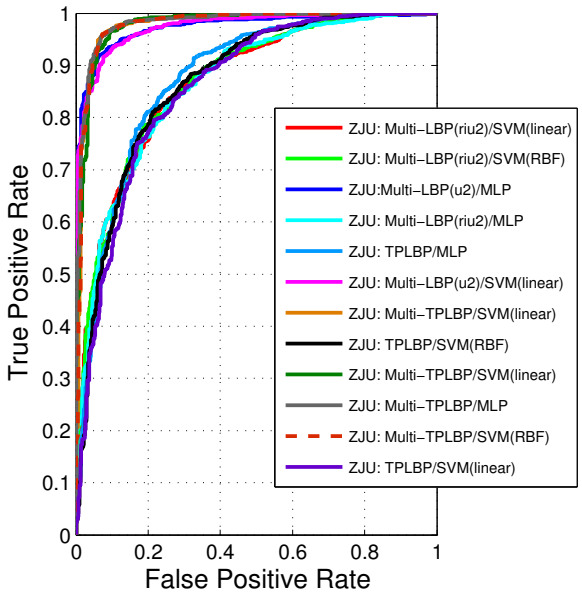


Figure 5.4: ROC curves of various features using the SVM and MLP classifiers on ZJU Eyeblink dataset.

ever, some image patterns such as lines are not captured in uniform codes [72]. These line patterns may appear less frequently than uniform codes, but they represent a set of important local primitives for pattern recognition.

2. The Multi-TPLBP generates a performances gained over those of TPLBP of $10,55\%$, $10,55\%$ and $9,94\%$ with SMV(linear), SVM(RBF) and MLP, respectively. The poor imaging conditions, such as low-resolution, blur, Gaussian noise, and uneven light are leading to ambiguous appearance of the eyes and specifically difficult to differentiate an eye state from another, as shown in Fig. 5.3. The obtained performance proves that TPLBP cannot handle well all these variations. So, TPLBP realized a low accuracy of 85.06%. TPLBP describes well the eyes in open state with 96.01% of open eyes correctly classified, but only 52.19% of closed eyes are classified well. One possible explanation, when the eyes are screwed up, it is difficult to TPLBP to describe the appearance of closed eyes with a coarse account of the global shape information.

3. The Multi-LBP^{u2} outperforms the TPLBP descriptor with classification score 93.54%. LBP^{u2} shows an improved performance for the eye state description and the 3-resolution fusion approach compared to the TPLBP. One reasoning can be, the pixel-based computation of LBP captures texture variations minimally when compared with the TPLBP descriptor.

4. Over the precedent comparison, we can observe that the proposed Multi-TPLBP improves the performance upon its original version. In ZJU dataset, the Multi-TPLBP clearly outperforms precedent descriptors, with a best accuracy of 95.18% with SVM(RBF) and an AUC value of 97.83%. Figure 5.4 gives the ROC curves of those features. It can be seen that the Multi-TPLBP feature realized best performance in terms of AUC values with SVM and MLP

models, followed by Multi-LBP^{u2}, which further verifies that our patch-based LBP extensions is beneficial to eye state detection task.

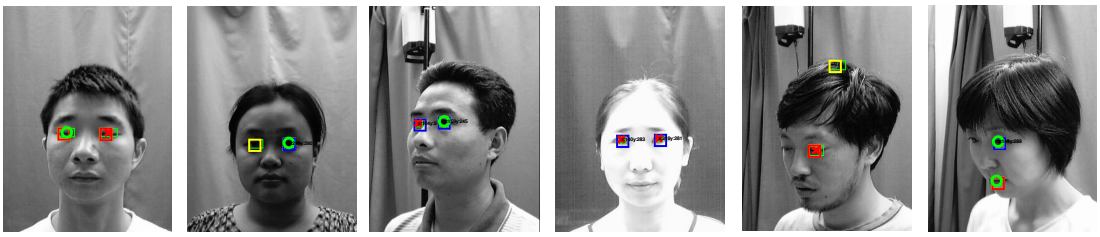


Figure 5.5: Snapshots illustrating some typical failures of our eye state detection method: it includes individuals with different head rotation angles, illumination variation.

Figure 5.5, illustrates different failure of the proposed approach. It can be noticed that Multi-TPLBP fails on extracting the eye features under hard angles of head rotation, in case of the eye is not detected at all, or the perspective changes even when the eye is well localized.

Figures 5.4 and 5.5 show the eye state estimation results and false alarms generated by the algorithm. It can be noticed that the change in facial expression results in degradation of the recognition performance of patch-based LBP descriptor. In the real-world applications such as the detection of the driver drowsiness and the weak attention paid to the road, facial images are not always captured in frontal view. This introduces a pose variation with the ocular region variations and occlusions depending of the head-pose. We can observe, that by lowering eyebrows while exhibiting expression changes the ocular region. In Fig. 5.4, it is noticed that there is no performance degradation of the eye state estimation for smiling scenario tests. In Fig. 5.4 and 5.5 it can be seen that large variations in pose significantly affect performance of the eye state estimation.

Figure 5.4 clearly shows that the Multi-TPLBP descriptor handles well perspective variations and correctly recognizes the eye states, without employing any face frontalization algorithm [209]. From Fig. 5.4 we can notice that the algorithm detects well the location of the eyes, but the state of the eye is not correctly estimated under different lighting conditions. This is due to the image regions captured in different scenarios, with a very different pixel intensity values. The description of these regions implies mismatches of the similarities between neighboring patches of pixels (self-similarity calculation to generate the TPLBP code), and hence our algorithm detects an eye closed instead of open.

5.5.1 Comparison with other works

Table 5.2 compares the proposed EyeLSD and recent works we are aware tested on ZJU dataset, with corresponding experimental settings (number of eye open (+) and eye closed (-), the number and the size of the eye images used during tests, and challenges faced), and the performance realized by each approach is listed beside ours.

We have to note that one of the main reasons why most current work on eye-state detection can not be directly compared is due to the lack of common data sets used for assessment.

Pan et al. present in [208] an appearance-based eye blink detection application, the ZJU Eye blink dataset is used to carry on tests. The performance realized by their framework is 93.3%. The proposed Multi-TPLBP outperforms this approach with an accuracy of 95.18%.

Multi-TPLBP captures more texture information complementary to those of the LBP-based pixels. This is reflected through the performance gain. In addition, it is invariant to scales and that forms a great need to build a robust eye state model that resists well to the real-world constraints. In such scenarios the eyes may undergo various poses and scales and not really obvious for some local shape descriptors, to neutralize the effects involved by those variations.

Fengyi Song et al. present in [130] two methods to extract the eye features under real-world conditions. Their eye state approach is tested with a first method called Histograms of principal Oriented Gradients (HPOG) and a second one called Multi-scale Histograms of principal Oriented Gradients (MultiHPOG). Their approaches were assessed on ZJU eye blink database, under different variations of facial expression, lighting, individual identity, and image noise. Their system architecture includes geometric alignment, which is considered as a key point of the algorithm.

Method	Data	challenge	**patch size** (pixel)	**# Test** (+) open, (-)closed	**Acc** (%)
Multi-TPLBP(Proposed) + MLP [140]	ZJU	Varying in distance, Illumination condition, occlusion, gender, aging, expression, hard rotations. varying pose, lighting, accessory.	24×24	$1230(+), 410(-)$	95.18
LBP + SVM [208]	ZJU	Variations in pose, lighting,accessory.	0.74×0.37 (ratio to eyes distance)	Rear clips (clip #3, clip#4)	90.37 84.37
HPOG (without alignment) +SVM [130]	ZJU		24×24	$1230(+), 410(-)$	94.04
HPOG (alignment) +SVM [130]					95.91
MultiHPOG +SVM [130]					95.60
MultiHPOG/LTP/Gabor + SVM [130]					96.83

Table 5.2: Comparison of the eye state model with existing methods

In Table 5.2 the performance realized by the Histograms of HPOG without alignment are shown, including geometric normalization of the eye image patches. In case of no alignment used, HPOG

achieved a recognition accuracy of 94.04%. However, when the alignment is used, recognition performance is enhanced with 1.87%. The performance enhancement realized by Multi-TPLBP, is partly owed to the additional local information captured from the enlarged feature extraction area of the eye pattern and the texture perimeter that the descriptor can reach with its enlarged TPLBP radius. In addition, applying Gaussian low-pass filters to construct the multi-resolution descriptor, attenuates the image noise effect and increases the contextual information amount captured. EyeLSD outperforms the HPOG (without alignment) approach. Fengyi Song et al [130] investigate the enhancement of feature fusion to describe images under uncontrolled conditions. They combine Haar-like feature approach, Multi-HPOG, LTP, and Gabor wavelets to extract salient eye feature map. The multi-scale HPOG is able to represent the eye image patch in varying scales. Thus, captures the eye appearance at different scales and further information that is normally missed by local descriptors.

The fusion of feature sets can enhance the overall system accuracy, that realized 96.83% on ZJU database. These results are slightly better than those realized by Multi-TPLBP, tested on the same image gallery. Nonetheless, fusion of feature sets involves a high calculation complexity with a small improvement realized compared to use of Multi-HPOG only. By combining into the ePLBPH structure the strength of pixel-based LBP approaches (for local description) with that of histogram concatenation (global information encoding), and multi-level pyramidal architecture (multi-scale information captured under various depth), the performance of the eye detection algorithm is significantly enhanced.

5.5.2 Runtime performance evaluation

The overall run-time of our MATLAB implementation is performed on an Intel Core i7-4790 Processor with 3.6 GHz and 8.0 GB Ram. We run the EyeLSD on an image of $360 \times 480 pixels$. The average computation costs of the principal EyeLSD stages are listed in Table 5.3, that reports the average elapsed time at each processing step. These steps must run sequentially in each key-point of the pre-processed image.

Pre-processing (ms)	Eye Localization (ePLBPH*)(ms)	Eye Localization (Prediction)(ms)	$\mathbb{S}_{\chi^2}$(ms)	Eye state (Multi-TPLBP)(ms)	Eye state (Prediction) (ms)	Total (ms)
112.89	14.9	9.20	0.41	20.7	3.0	161.1

Table 5.3: Computation time of each step of EyeLSD approach

From Table 5.3, we can observe that the pre-processing step represents an embarrassingly parallel workload, since each key-point and neighborhood pixels are scanned, which span the entire local minimum regions of the input image. The table shows that the feature extraction and classification steps take about 30% of the total time, while 70% of the time is due to the pre-processing step. In practice, the pre-processing step can be replaced with a facial landmark localization algorithm [105], that precisely locates different facial traits and highlights them as a landmarks (e.g., the nose tip, mouth corners, eye centers) in the input image. So, instead of searching for an eye in the entire image over key-points. we can only scan the facial landmarks to locate the eye positions and recognize whether open or closed. This process can save the time spent on the steps of pre-processing,

and exploits more the potential of the proposed feature extractors. Moreover, by using other low-level programming language such as C++ and beside of code optimization strategies, the algorithm computation time can be further enhanced.

5.6 Conclusion

In this chapter, we test the ability of EyeLSD to perceive eye conditions using unconstrained face images. We conducted a series of thorough analysis on the performance of LBP-like descriptors. We have proposed the multi-scale LBP framework (Multi-TPLBP), which is a variant of the three-point LBP. The multi-TPLBP descriptor has so far achieved the highest eye state recognition score on the ZJU Eyeblink real-world database. These findings are due to the fact that Multi-TPLBP captures more texture information complementary to that of LBP-based pixels, and is invariant to scale. We also examine several typical feature descriptors to understand their respective of distinguishing closed eyes from open ones, and find that the Multi-TPLBP descriptor is efficient even when the quality of eye images is decreasing. Experiments indicate that large variations of the head posture do not affect the performance of the system. This is due to the invariant nature of the LBP operator and the algorithm ability for the eye localization phase, which provide additional hints by delimiting the exact area of action (i.e., where exactly the state detector verifies whether the eye is closed or open).

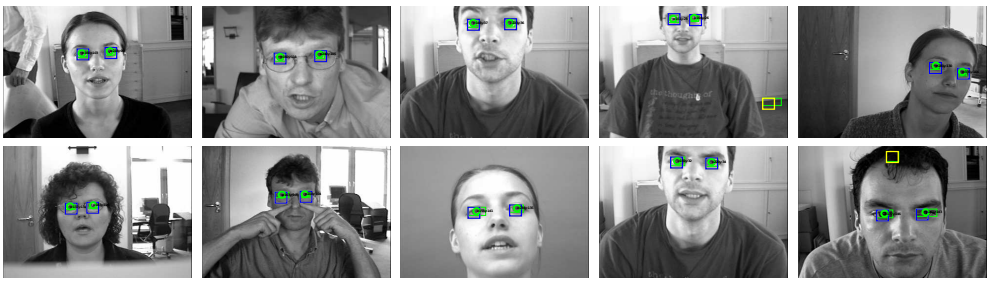
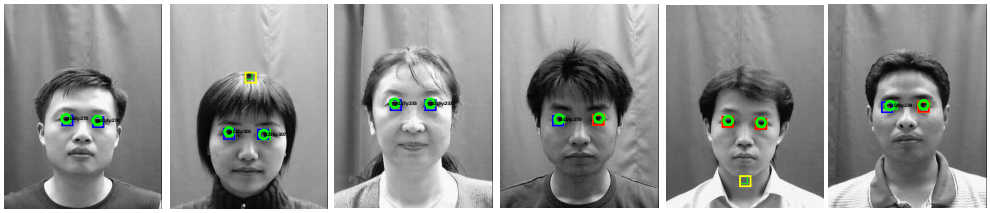
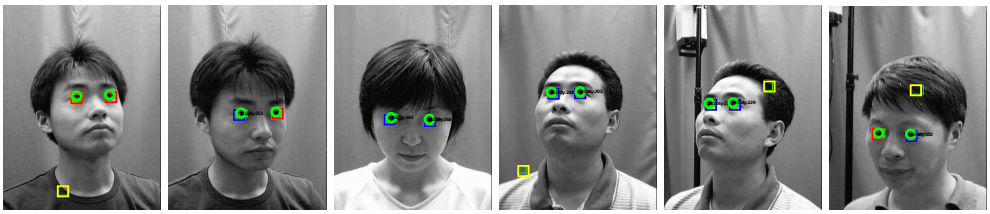
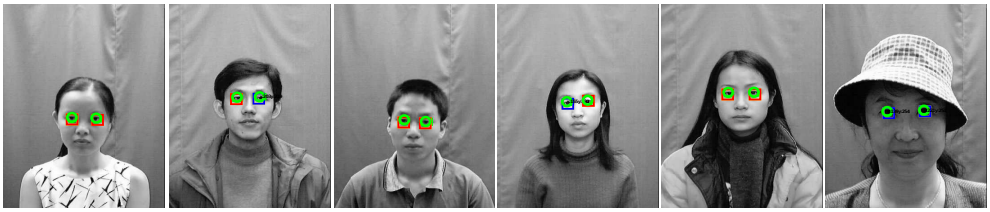
Challenge	Detection results
Eye closed CAS-PEAL	
Eye open BioID	
Eye open CAS-PEAL	
Pose	
Lighting	
Resolution accessory	

Table 5.4: Examples of some typical success our eye open/closed approach, tested on different conditions (Pose, lighting, resolution, facial expression, occlusion)

CHAPTER 6

Conclusion and perspectives

6.1 General conclusion

Many researchers working on the next generation of autonomous vehicles and advanced driver assistance systems (ADAS) use computer vision techniques. Motivated by the unresolved problems in difficult lighting conditions, the lack of precision or the challenge represented by the texture analysis, this thesis provides methods used in the development of imaging systems for driver assistance and drowsiness detection in real world conditions.

This thesis aimed to develop an non-intrusive-vision-based system to monitor driver's drowsy behavior. It also demonstrates the ultimate goal of the automated vision-based systems is to help anticipating potential hazard situations, by analyzing facial expressions and thus reducing the number of accidents that may lead to a substantial saving in lives and money. The proposed systems were able to analyze facial expressions by recognizing the state of the eyes.

Driver monitoring is a complex task that involves many parameters of behavior and physiology. Analyzing facial expressions and eye states (open and closed) by computer vision techniques can produce exact estimation of the state of the driver. Three main levels constitutes these techniques. The first level is face detection, the second is eye localization, and the third is eye state detection (open and closed). This thesis has concentrated on the latter, using appearance-based methods to characterize the ocular region. The focus was in three aspects: a) an appearance-based eye localization systems without face detection; b) an appearance-based eye state recognition system for eyes open-and-closed detection; and c) an in-depth analysis of the proposals, on training data sets, including various degrees of variation that correlates with those of the real driving.

Three models have been proposed and tested on datasets of static images and video sequences. The datasets were acquired under various realistic unconstrained situations, e.g., face is rotated right and left resulting in pose variations and lighting variations. The most accurate among the proposals was a novel algorithm EyeLSD for Eye Localization and State Detection. Two new LBP-like descriptors have been proposed to be used in EyeLSD and have improved both the reliability and the representation capacity of the ocular region. An appearance model built from a training set has been also integrated. Coupled with a searching strategy, the robustness of the model to real world variations, occlusions and head poses improved. The main interest of conducted tests have been to show whether the models could work in different situations, and comply with the requirements of production system, namely robustness to head pose and partial occlusions, facial expressions and varying illumination, and users accessories (e.g., glasses, hat, beard, among others). EyeLSD algorithm achieved a much better performance than conventional Viola-Jones with a detection rate of 98.86% on BioID dataset.

In the implementation developed for this thesis, EyeLSD is effective to locate the eyes and recognize whether they are open or closed. EyeLSD algorithm realized great performance and has been

tested on a series of unconstrained realistic conditions, including head poses, occlusions, illumination variations, facial expression and resolution changes. The obtained results were published in two peer-reviewed papers, four international conferences and as two chapters in two separate lecture notes books.

6.2 Summary

The first chapter provided the basic knowledge that we needed in order to start research.

In Chapter 1, we conducted a literature review on the existing research in the field of drowsiness detection and driver assistance systems, to identify the current research challenges. Based on our motivations, we narrowed down our research to the particular topic of driver behavior by computer vision techniques for facial expressions analysis. We also highlight the main shortcomings such as the lack of reference data sets, the limited studies on driver monitoring with algorithms based on monocular web cameras. Based on existing hardware and software platforms at the time of the research, the thesis intends to develop systems with monocular cameras, which can achieve high accuracy with modest computation time. These systems consist of three levels:(1) the detection of the face area;(2) the accurate detection of the facial features (the eyes); (3) the analysis of the eye states. We also claimed possibly accuracy of above 98% for the proposed algorithms under various weather and challenging lighting conditions.

In Chapter 2, we provided some basic computer-vision methods that we needed to use and refer to in the next chapters. We also introduced some initial definitions, notations, equations, and terminologies to ensure consistency in the rest of the thesis; however, if necessary, some concepts have been included in later chapters to ensure self-contained discussion in the given chapter (e.g., the concept of Gabor wavelets in Chapters 3 and 4).

In Chapter 3, we focused on the overall difficulties, challenges in real-life scenarios and present a detailed review of prominent approaches, flexible and efficient statistical eye models. In addition, we organized the discussion of the global aspects of eye localization in uncontrolled environments, towards the development of a robust eye localization system. We proposed a novel technique of long-short term contextual information discrimination of the ocular region, by *Long-Short Term Memory Recurrent Network* and *enhanced Local Binary Patterns* to maintain speed efficiency and higher detection-accuracy of the ocular region. The algorithm was successfully implemented and tested under extremity low light, noisy conditions, cluttered background and challenging lighting conditions.

In Chapter 4, we developed a robust system to lighting-adaptive, head pose and expression changes. The system is able accurately detect the individual's eyes under these situations. We proposed a novel features descriptor enhanced Pyramidal Local Binary Pattern for ocular region description, a significant enhancement compared to the previous state-of-the-art by introducing the methods of *Local Binary Patterns* and other features descriptors. The proposed eye localization approach achieved higher detection-accuracy than that of other eye detectors, including conventional Viola-Jones eye detector.

Focusing on driver drowsiness detection, in Chapter 5 discussed about eye-states detection methods by proposing our approach. Extending the previously developed method of eye localization (from Chapter 4) and by integrating a novel features descriptor *Multi-resolution Three-Patch Local Binary Pattern*.

Experiments were conducted in real-world database that includes rotation, blurring, contrast, and Gaussian white noise. Extending tests were made on images with eye closed and open, head rotations and pose variations, different lighting conditions and depth-of-field variations.

6.3 Future work

The thesis presents an application-based research that concentrates on development of an ADAS based on computer vision techniques. However, most of developed approaches, such as face detection, facial features analysis are applicable in many other domains and are not limited to road safety. After analyzing the proposed methods in this thesis, we present the perspectives of our research. The field to explore and applications are numerous and we may cite, for instance, the following future work:

- **Extension of the Dataset**. Collect and annotate a database encompassing the different states of the driver by considering realistic scenarios. The availability of a wide and diversified database publicly available is of great importance in the field of driver safety, as it will provide the scientific community with ready-to-use data to facilitated future comparisons between different approaches. There are few databases in this area and most of them are not very wide.
- **A deep learning-based approach for driver's somnolence and distraction detection**. An approach focuses on driver's eye states and head pose monitoring in low-resolution video stream. The approach must be effective to localize the driver's eyes, recognizes their states and detect the different head-pose, to estimate the cognitive distraction of the driver.
- **Extend EyeLSD for real-time driver monitoring system**. The next step is to build a driver monitoring embedded system designed to integrate the EyeLSD algorithm, with additional devices, which analyze the eye states and extract the relevant information (PERCLOS, the period of opening and closing of the eyes, the flashing frequency). These measures may reflect drowsiness, and set estimation algorithms. The final step is consists to install the complete system in a production vehicle.

Bibliography

[1] Volvo., "European accident research and safety report, volvo trucks." http://pnt.volvo.com/pntclient/loadAttachment.aspx?id=27116, 2013.

[2] CARRS-Q., "State of the road, a fact sheet of carrs-q (centre of accident research and road safety-queensland." www.carrsq.qut.edu.au/publications/corporate/hooning_fs.pdf, 2011.

[3] ASFA., "La sécurité, asfa (association professionnelle autoroutes et ouvrages routiers)." www.autoroutes.fr/FCKeditor/UserFiles/File/B-la_securite.pdf, 2010.

[4] NHTSA *et al.*, "Traffic safety facts 2009: a compilation of motor vehicle crash data from the fatality analysis reporting system and the general estimates system. early edition. washington, dc: Us department of transportation, national highway traffic safety administration; 2010," *National Center for Statistics and Analysis, US Department of Transportation*, p. 20590, 2011.

[5] K. Hartman and J. Strasser, "Saving lives through advanced vehicle safety technology: Intelligent vehicle initiative final report," *Federal Highway Administration, FHWA-JPO-05-057*, 2005.

[6] M. A. Carskadon and W. C. Dement, "Daytime sleepiness: quantification of a behavioral state," *Neuroscience & Biobehavioral Reviews*, vol. 11, no. 3, pp. 307–317, 1987.

[7] J. Horne and L. Reyner, "Driver sleepiness," *Journal of sleep research*, vol. 4, no. s2, pp. 23–29, 1995.

[8] C. François, T. Hoyoux, T. Langohr, J. Wertz, and J. G. Verly, "Tests of a new drowsiness characterization and monitoring system based on ocular parameters," *International journal of environmental research and public health*, vol. 13, no. 2, p. 174, 2016.

[9] A. A. Borb and P. Achermann, "Sleep homeostasis and models of sleep regulation," *Journal of biological rhythms*, vol. 14, no. 6, pp. 559–570, 1999.

[10] C. A. Czeisler and J. Gooley, "Sleep and circadian rhythms in humans," in *Cold Spring Harbor symposia on quantitative biology*, vol. 72. Cold Spring Harbor Laboratory Press, 2007, pp. 579–597.

[11] T. Åkerstedt, J. Connor, A. Gray, and G. Kecklund, "Predicting road crashes from a mathematical model of alertness regulationâthe sleep/wake predictor," *Accident Analysis & Prevention*, vol. 40, no. 4, pp. 1480–1485, 2008.

[12] W. C. Dement, L. E. Miles, and M. A. Carskadon, "âwhite paperâ on sleep and aging," *Journal of the American Geriatrics Society*, vol. 30, no. 1, pp. 25–50, 1982.

[13] M. Härmä and M. Sallinen, *Hyvä uni-hyvä työ*. Työterveyslaitos, 2004.

[14] A. Anund, C. Fors, G. Kecklund, W. v. Leeuwen, and T. Åkerstedt, *Countermeasures for fatigue in transportation: a review of existing methods for drivers on road, rail, sea and in aviation*. Statens väg-och transportforskningsinstitut, 2015.

[15] M. A. Regan. (2010) Distraction du conducteur : définition, mécanismes, effets et facteurs modérateurs.

[16] J. N. Chiquero, "Face tracking with active models for a driver monitoring application," Ph.D. dissertation, Universidad de Alcalá, 2009.

[17] R. O. Mbouna, S. G. Kong, and M.-G. Chun, "Visual analysis of eye state and head pose for driver alertness monitoring," *IEEE transactions on intelligent transportation systems*, vol. 14, no. 3, pp. 1462–1469, 2013.

[18] D. E. Benrachou, B. Boulebtateche, and S. Bensaoula, "Gabor/pca/svm-based face detection for drivers monitoring," *Journal of Automation and Control Engineering*, vol. 1, pp. 115–118, 2013.

[19] M. J. Flores, J. M. Armingol, and A. de la Escalera, "Real-time warning system for driver drowsiness detection using visual information," *Journal of Intelligent & Robotic Systems*, vol. 59, no. 2, pp. 103–125, 2010.

[20] D. González-Ortega, F. Díaz-Pernas, M. Antón-Rodríguez, M. Martínez-Zarzuela, and J. Díez-Higuera, "Real-time vision-based eye state detection for driver alertness monitoring," *Pattern Analysis and Applications*, vol. 16, no. 3, pp. 285–306, 2013.

[21] D. F. Dinges and R. Grace, "Perclos: A valid psychophysiological measure of alertness as assessed by psychomotor vigilance," *US Department of Transportation, Federal Highway Administration, Publication Number FHWA-MCRT-98-006*, 1998.

[22] Y. Saito, M. Itoh, and T. Inagaki, "Driver assistance system with a dual control scheme: Effectiveness of identifying driver drowsiness and preventing lane departure accidents," *IEEE Transactions on Human-Machine Systems*, vol. 46, no. 5, pp. 660–671, 2016.

[23] X. Zhu, W.-L. Zheng, B.-L. Lu, X. Chen, S. Chen, and C. Wang, "Eog-based drowsiness detection using convolutional neural networks." in *IJCNN*, 2014, pp. 128–134.

[24] K. Šušmáková, "Human sleep and sleep eeg," *Measurement science review*, vol. 4, no. 2, pp. 59–74, 2004.

[25] Z. Zhang, D. Luo, Y. Rasim, Y. Li, G. Meng, J. Xu, and C. Wang, "A vehicle active safety model: vehicle speed control based on driver vigilance detection using wearable eeg and sparse representation," *Sensors*, vol. 16, no. 2, p. 242, 2016.

[26] L.-C. Shi and B.-L. Lu, "Dynamic clustering for vigilance analysis based on eeg," in *Engineering in Medicine and Biology Society, 2008. EMBS 2008. 30th Annual International Conference of the IEEE*. IEEE, 2008, pp. 54–57.

[27] L.-C. Shi and B.-L. Lu, "Eeg-based vigilance estimation using extreme learning machines" , *Neurocomputing*, vol. 102, pp. 135–143, 2013.

[28] L.-C. Shi, H. Yu, and B.-L. Lu, "Semi-supervised clustering for vigilance analysis based on eeg," in *Neural Networks*, 2007. IJCNN 2007. International Joint Conference on . IEEE, 2007, pp. 1518–1523.

[29] L.-C. Shi and B.-L. Lu, "Off-line and on-line vigilance estimation based on linear dynamical system and manifold learning," in *Engineering in Medicine and Biology Society (EMBC)*, *2010 Annual International Conference of the IEEE* . IEEE, 2010, pp. 6587–6590.

[30] H. Yu, H. Lu, T. Ouyang, H. Liu, and B.-L. Lu, "Vigilance detection based on sparse representation of eeg," in *Engineering in Medicine and Biology Society (EMBC)*, *2010 Annual International Conference of the IEEE*. IEEE, 2010, pp. 2439–2442.

[31] H. Kalbkhani, M. G. Shayesteh, and S. Mohsen Mousavi, "Efficient algorithms for detection of face, eye and eye state," *Computer Vision, IET*, vol. 7, no. 3, pp. 184–200, 2013.

[32] S. K. Lal and A. Craig, "A critical review of the psychophysiology of driver fatigue," Biological psychology, vol. 55, no. 3, pp. 173–194, 2001.

[33] J.-X. Ma, L.-C. Shi, and B.-L. Lu, "Vigilance estimation by using electrooculographic features," in *Engineering in Medicine and Biology Society (EMBC)*, 2010 Annual International Conference of the IEEE. IEEE, 2010, pp. 6591–6594.

[34] R. Schleicher, N. Galley, S. Briest, and L. Galley, "Blinks and saccades as indicators of fatigue in sleepiness warnings: looking tired?" *Ergonomics*, vol. 51, no. 7, pp. 982–1010, 2008.

[35] P. P. Caffier, U. Erdmann, and P. Ullsperger, "Experimental evaluation of eye-blink parameters as a drowsiness measure," *European journal of applied physiology*, vol. 89, no. 3-4, pp. 319–325, 2003.

[36] S.-J. Jung, H.-S. Shin, and W.-Y. Chung, "Driver fatigue and drowsiness monitoring system with embedded electrocardiogram sensor on steering wheel," *IET Intelligent Transport Systems*, vol. 8, no. 1, pp. 43–50, 2014.

[37] DAC. (2008) Volvo driver alert control (accessed on 06.2014).

[38] R. P. Loce, E. A. Bernal, W. Wu, and R. Bala, "Computer vision in roadway transportation systems: a survey," *Journal of Electronic Imaging*, vol. 22, no. 4, pp. 041 121–041 121, 2013.

[39] P. Angkititrakul, R. Terashima, and T. Wakita, "On the use of stochastic driver behavior model in lane departure warning," *IEEE Transactions on intelligent transportation systems*, vol. 12, no. 1, pp. 174–183, 2011.

[40] S. Mammar, S. Glaser, and M. Netto, "Time to line crossing for lane departure avoidance: A theoretical study and an experimental setting," *IEEE Transactions on Intelligent Transportation Systems*, vol. 7, no. 2, pp. 226–241, 2006.

[41] G. Cario, A. Casavola, G. Franze, M. Lupia, G. Brasili, and I. it SpA, "Predictive time-to-lane-crossing estimation for lane departure warning systems," in *Proceedings of the 21st International Technical Conference on the Enhanced Safety of Vehicles (ESV)*, 2009.

[42] M. Rezaei and R. Klette, "Computer vision for driver assistance."

[43] W. Wang, D. Zhao, J. Xi, and W. Han, "A learning-based approach for lane departure warning systems with a personalized driver model," *arXiv preprint arXiv:1702.01228*, 2017.

[44] T. Baltrušaitis, P. Robinson, and L.-P. Morency, "Openface: an open source facial behavior analysis toolkit," in *Applications of Computer Vision (WACV), 2016 IEEE Winter Conference on*. IEEE, 2016, pp. 1–10.

[45] HealthyRoad., http://healthyroad.pt/, 2016.

[46] P. Viola and M. J. Jones, "Robust real-time face detection," *International journal of computer vision*, vol. 57, no. 2, pp. 137–154, 2004.

[47] E. Murphy-Chutorian and M. M. Trivedi, "Head pose estimation in computer vision: A survey," *IEEE transactions on pattern analysis and machine intelligence*, vol. 31, no. 4, pp. 607–626, 2009.

[48] E. Ohn-Bar and M. M. Trivedi, "Beyond just keeping hands on the wheel: Towards visual interpretation of driver hand motion patterns," in *Intelligent Transportation Systems (ITSC), 2014 IEEE 17th International Conference on*. IEEE, 2014, pp. 1245–1250.

[49] C. Tran, A. Doshi, and M. M. Trivedi, "Modeling and prediction of driver behavior by foot gesture analysis," *Computer Vision and Image Understanding*, vol. 116, no. 3, pp. 435–445, 2012.

[50] U. Trutschel, B. Sirois, D. Sommer, M. Golz, and D. Edwards, "Perclos: An alertness measure of the past," in *Proceedings of the Sixth International Driving Symposium on Human Factors in Driver Assessment, Training and Vehicle Design*, 2011, pp. 172–179.

[51] M. Golz, D. Sommer, M. Holzbrecher, and T. Schnupp, "Detection and prediction of driver's microsleep events," in *Proc 14th Int Conf Road Safety on Four Continents*, vol. 11, 2007.

[52] T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham, "Active shape models-their training and application," *Computer vision and image understanding*, vol. 61, no. 1, pp. 38–59, 1995.

[53] B. Kégl, "The return of adaboost. mh: multi-class hamming trees," *arXiv preprint arXiv:1312.6086*, 2013.

[54] K. Hollingsworth, S. Clark, J. Thompson, P. J. Flynn, and K. W. Bowyer, "Eyebrow segmentation using active shape models," in *SPIE Defense, Security, and Sensing*. International Society for Optics and Photonics, 2013, pp. 871 208–871 208.

[55] X. Sun, L. Xu, and J. Yang, "Driver fatigue alarm based on eye detection and gaze estimation," in *Proc. SPIE*, vol. 6786, no. 1, 2007, pp. 678 612–678 612.

[56] A. M. Malla, P. R. Davidson, P. J. Bones, R. Green, and R. D. Jones, "Automated video-based measurement of eye closure for detecting behavioral microsleep," in *Engineering in Medicine and Biology Society (EMBC), 2010 Annual International Conference of the IEEE*. IEEE, 2010, pp. 6741–6744.

[57] J. He, S. Roberson, B. Fields, J. Peng, S. Cielocha, and J. Coltea, "Fatigue detection using smartphones," *Journal of Ergonomics*, vol. 3, no. 03, pp. 1–7, 2013.

[58] R. Dinges, D. et Grace, *PERCLOS: a valid psychophysiological measure of alertness as assessed by psychomotor vigilance*. Indianapolis. In: Federal Highway Administration, Office of Motor Carriers, Tech. Rep. MCRT-98-006, 1998.

[59] T. Danisman, I. M. Bilasco, C. Djeraba, and N. Ihaddadene, "Drowsy driver detection system using eye blink patterns," in *Machine and Web Intelligence (ICMWI), 2010 International Conference on*. IEEE, 2010, pp. 230–233.

[60] J. Jo, S. J. Lee, K. R. Park, I.-J. Kim, and J. Kim, "Detecting driver drowsiness using feature-level fusion and user-specific classification," *Expert Systems with Applications*, vol. 41, no. 4, pp. 1139–1152, 2014.

[61] S. Abtahi, B. Hariri, and S. Shirmohammadi, "Driver drowsiness monitoring based on yawning detection," in *Instrumentation and Measurement Technology Conference (I2MTC), 2011 IEEE*. IEEE, 2011, pp. 1–4.

[62] N. Alioua, A. Amine, and M. Rziza, "Driverś fatigue detection based on yawning extraction," *International journal of vehicular technology*, vol. 2014, 2014.

[63] X. Fan, B.-C. Yin, and Y.-F. Sun, "Yawning detection for monitoring driver fatigue," in *Machine Learning and Cybernetics, 2007 International Conference on*, vol. 2. IEEE, 2007, pp. 664–668.

[64] S. Abtahi, S. Shirmohammadi, B. Hariri, D. Laroche, and L. Martel, "A yawning measurement method using embedded smart cameras," in *Instrumentation and Measurement Technology Conference (I2MTC), 2013 IEEE International*. IEEE, 2013, pp. 1605–1608.

[65] K. Barry, "Yawn if you dare. your car is watching you," *Wired Mag., Autopia Section*, 2009.

[66] P. Navarrete and J. Ruiz-del Solar, "Analysis and comparison of eigenspace-based face recognition approaches," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 16, no. 07, pp. 817–830, 2002.

[67] M.-H. Yang, D. J. Kriegman, and N. Ahuja, "Detecting faces in images: A survey," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 24, no. 1, pp. 34–58, 2002.

[68] C. Zhang and Z. Zhang, "A survey of recent advances in face detection," 2010.

[69] O. Jesorsky, K. J. Kirchberg, and R. W. Frischholz, "Robust face detection using the hausdorff distance," in *International Conference on Audio-and Video-Based Biometric Person Authentication*. Springer, 2001, pp. 90–95.

[70] R.-L. Hsu, M. Abdel-Mottaleb, and A. K. Jain, "Face detection in color images," *IEEE transactions on pattern analysis and machine intelligence*, vol. 24, no. 5, pp. 696–706, 2002.

[71] Y. Ban, S.-K. Kim, S. Kim, K.-A. Toh, and S. Lee, "Face detection based on skin color likelihood," *Pattern Recognition*, vol. 47, no. 4, pp. 1573–1585, 2014.

[72] C. M. Vong, K. I. Tai, C. M. Pun, and P. K. Wong, "Fast and accurate face detection by sparse bayesian extreme learning machine," *Neural Computing and Applications*, vol. 26, no. 5, pp. 1149–1156, 2015.

[73] L. Xiaohua, K.-M. Lam, S. Lansun, and Z. Jiliu, "Face detection using simplified gabor features and hierarchical regions in a cascade of classifiers," *Pattern Recognition Letters*, vol. 30, no. 8, pp. 717–728, 2009.

[74] R. Lienhart, A. Kuranov, and V. Pisarevsky, "Empirical analysis of detection cascades of boosted classifiers for rapid object detection," *Pattern Recognition*, pp. 297–304, 2003.

[75] R. Lienhart and J. Maydt, "An extended set of haar-like features for rapid object detection," in *Image Processing. 2002. Proceedings. 2002 International Conference on*, vol. 1. IEEE, 2002, pp. I–I.

[76] C. Huang, H. Ai, Y. Li, and S. Lao, "High-performance rotation invariant multiview face detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 4, pp. 671–686, 2007.

[77] S.-K. Pavani, D. Delgado, and A. F. Frangi, "Haar-like features with optimally weighted rectangles for rapid object detection," *Pattern Recognition*, vol. 43, no. 1, pp. 160–172, 2010.

[78] R. Hoogenboom and M. Lew, "Face detection using local maxima," in *Automatic Face and Gesture Recognition, 1996., Proceedings of the Second International Conference on*. IEEE, 1996, pp. 334–339.

[79] A. Tharwat, H. Mahdi, A. El Hennawy, and A. E. Hassanien, "Face sketch synthesis and recognition based on linear regression transformation and multi-classifier technique," in *The 1st International Conference on Advanced Intelligent System and Informatics (AISI2015), November 28-30, 2015, Beni Suef, Egypt*. Springer, 2016, pp. 183–193.

[80] R. C. Gonzalez and R. E. Woods, "Digital image processing (2nd ed)." Prentice-Hall Englewood Cliffs, NJ, 2002, pp. 523–532.

[81] S. Lin-Lin and J. Zhen, "Gabor wavelet selection and svm classification for object recognition," *Acta Automatica Sinica*, vol. 35, no. 4, pp. 350–355, 2009.

[82] M. Haghighat, S. Zonouz, and M. Abdel-Mottaleb, "Cloudid: Trustworthy cloud-based and cross-enterprise biometric identification," *Expert Systems with Applications*, vol. 42, no. 21, pp. 7905–7916, 2015.

[83] Q. Wang, X. Zhang, M. Li, X. Dong, Q. Zhou, and Y. Yin, "Adaboost and multi-orientation 2d gabor-based noisy iris recognition," *Pattern Recognition Letters*, vol. 33, no. 8, pp. 978–983, 2012.

[84] C. K. Chui, "An introduction to wavelets, wavelet analysis and its application vol. 1," *1, Academic Press, Boston*, 1992.

[85] Y. W. Hen, M. Khalid, and R. Yusof, "Face verification with gabor representation and support vector machines," in *Modelling & Simulation, 2007. AMS'07. First Asia International Conference on*. IEEE, 2007, pp. 451–459.

[86] V. N. Vapnik and V. Vapnik, *Statistical learning theory*. Wiley New York, 1998, vol. 1.

[87] C.-C. Chang and C.-J. Lin, "Libsvm: A library for support vector machines," *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 3, pp. 27:1–27:27, May 2011. [Online]. Available: http://doi.acm.org/10.1145/1961189.1961199

[88] F. S. Samaria and A. C. Harter, "Parameterisation of a stochastic model for human face identification," in *Applications of Computer Vision, 1994., Proceedings of the Second IEEE Workshop on*. IEEE, 1994, pp. 138–142.

[89] "The ORL Database of Faces." http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html.

[90] R. Gross, I. Matthews, J. Cohn, T. Kanade, and S. Baker, "Multi-pie," *Image Vision Comput.*, vol. 28, no. 5, pp. 807–813, May 2010. [Online]. Available: http://dx.doi.org/10.1016/j.imavis.2009.08.002

[91] "The CMU Multi-PIE Face Database." http://www.cs.cmu.edu/afs/cs/project/PIE/MultiPie/Multi-Pie/Home.html.

[92] "The CMU Frontal Face Dataset." http://vasc.ri.cmu.edu//idb/html/face/frontal_images/index.html.

[93] S. Chen, T. Shan, and B. C. Lovell, "Robust face recognition in rotated eigen space," in *The Twenty-second International Image and Vision Computing New Zealand Conference*, 2007.

[94] A. Jalil, I. Qureshi, A. Manzar, R. Zahoor, and M. Jinnah, "Rotation-invariant features for texture image classification," in *Engineering of Intelligent Systems, 2006 IEEE International Conference on*. IEEE, 2006.

[95] P. Moreno, A. Bernardino, and J. Santos-Victor, "Gabor parameter selection for local feature detection," in *Iberian Conference on Pattern Recognition and Image Analysis*. Springer, 2005, pp. 11–19.

[96] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller, "Labeled faces in the wild: A database for studying face recognition in unconstrained environments," Technical Report 07-49, University of Massachusetts, Amherst, Tech. Rep., 2007.

[97] Y. Ren, S. Wang, B. Hou, and J. Ma, "A novel eye localization method with rotation invariance," *Image Processing, IEEE Transactions on*, vol. 23, no. 1, pp. 226–239, 2014.

[98] F. Song, X. Tan, S. Chen, and Z.-H. Zhou, "A literature survey on robust and efficient eye localization in real-life scenarios," *Pattern Recognition*, vol. 46, no. 12, pp. 3157–3173, 2013.

[99] J. Naruniec, "A survey on facial features detection," *International Journal of Electronics and Telecommunications*, vol. 56, no. 3, pp. 267–272, 2010.

[100] F. Alonso-Fernandez and J. Bigun, "A survey on periocular biometrics research," *Pattern Recognition Letters*, vol. 82, pp. 92–105, 2016.

[101] K. Seshadri and M. Savvides, "Towards a unified framework for pose, expression, and occlusion tolerant automatic facial alignment," *IEEE transactions on pattern analysis and machine intelligence*, vol. 38, no. 10, pp. 2110–2122, 2016.

[102] A. L. Yuille, P. W. Hallinan, and D. S. Cohen, "Feature extraction from faces using deformable templates," *International journal of computer vision*, vol. 8, no. 2, pp. 99–111, 1992.

[103] S. Sirohey, A. Rosenfeld, and Z. Duric, "A method of detecting and tracking irises and eyelids in video," *Pattern Recognition*, vol. 35, no. 6, pp. 1389–1401, 2002.

[104] J. Wu and L. Mei, "A face recognition algorithm based on asm and gabor features of key points," in *2012 International Conference on Graphic and Image Processing*. International Society for Optics and Photonics, 2013, pp. 87 686L–87 686L.

[105] Y. Shi, Z. Yan, H. Ge, and L. Mei, "Visual objects location based on hand eye coordination," in *Future Information Technology*. Springer, 2014, pp. 403–408.

[106] M. Nixon, "Eye spacing measurement for facial recognition," in *29th Annual Technical Symposium*. International Society for Optics and Photonics, 1985, pp. 279–285.

[107] Y. Zheng and Z. Wang, "Robust and precise eye detection based on locally selective projection," in *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on*. IEEE, 2008, pp. 1–4.

[108] Z. Zhu and Q. Ji, "Robust real-time eye detection and tracking under variable lighting conditions and various face orientations," *Computer Vision and Image Understanding*, vol. 98, no. 1, pp. 124–154, 2005.

[109] Z. Zhu, K. Fujimura, and Q. Ji, "Real-time eye detection and tracking under various light conditions," in *Proceedings of the 2002 symposium on Eye tracking research & applications*. ACM, 2002, pp. 139–144.

[110] J. Jo, S. J. Lee, H. G. Jung, K. R. Park, and J. Kim, "Vision-based method for detecting driver drowsiness and distraction in driver monitoring system," *Optical Engineering*, vol. 50, no. 12, pp. 127 202–127 202, 2011.

[111] P. F. Felzenszwalb and D. P. Huttenlocher, "Pictorial structures for object recognition," *International journal of computer vision*, vol. 61, no. 1, pp. 55–79, 2005.

[112] T. F. Cootes, G. J. Edwards, and C. J. Taylor, "Active appearance models," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 23, no. 6, pp. 681–685, 2001.

[113] D. Cristinacce and T. F. Cootes, "Feature detection and tracking with constrained local models." in *BMVC*, vol. 1, no. 2, 2006, p. 3.

[114] X. Tan, F. Song, Z.-H. Zhou, and S. Chen, "Enhanced pictorial structures for precise eye localization under incontrolled conditions," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. IEEE, 2009, pp. 1621–1628.

[115] A. Lehmann, B. Leibe, and L. Van Gool, "Fast prism: Branch and bound hough transform for object class detection," *International journal of computer vision*, vol. 94, no. 2, pp. 175–197, 2011.

[116] B. Leibe, A. Leonardis, and B. Schiele, "Robust object detection with interleaved categorization and segmentation," *International journal of computer vision*, vol. 77, no. 1-3, pp. 259–289, 2008.

[117] D. Yi, Z. Lei, and S. Z. Li, "A robust eye localization method for low quality face images," in *Biometrics (IJCB), 2011 International Joint Conference on*. IEEE, 2011, pp. 1–6.

[118] Y. Ito, W. Ohyama, T. Wakabayashi, and F. Kimura, "Detection of eyes by circular hough transform and histogram of gradient," in *Pattern Recognition (ICPR), 2012 21st International Conference on*. IEEE, 2012, pp. 1795–1798.

[119] D. Monzo, A. Albiol, J. Sastre, and A. Albiol, "Precise eye localization using hog descriptors," *Machine Vision and Applications*, vol. 22, no. 3, pp. 471–480, 2011.

[120] S. Ge, R. Yang, H. Wen, S. Chen, and L. Sun, "Eye localization based on correlation filter bank," in *Multimedia and Expo (ICME), 2014 IEEE International Conference on*. IEEE, 2014, pp. 1–5.

[121] M. Zhou, X. Wang, H. Wang, J. Heo, and D. Nam, "Precise eye localization with improved sdm," in *Image Processing (ICIP), 2015 IEEE International Conference on*. IEEE, 2015, pp. 4466–4470.

[122] M. Leo, D. Cazzato, T. De Marco, and C. Distante, "Unsupervised eye pupil localization through differential geometry and local self-similarity matching," *PloS one*, vol. 9, no. 8, p. e102829, 2014.

[123] H. Kim, J. Jo, K.-A. Toh, and J. Kim, "Eye detection in a facial image under pose variation based on multi-scale iris shape feature," *Image and Vision Computing*, vol. 57, pp. 147–164, 2017.

[124] K. Rajakumar and S. Muttan, "Medical image retrieval using modified dct," *Procedia Computer Science*, vol. 2, pp. 298–302, 2010.

[125] S. Fischer, F. Šroubek, L. Perrinet, R. Redondo, and G. Cristóbal, "Self-invertible 2d log-gabor wavelets," *International Journal of Computer Vision*, vol. 75, no. 2, pp. 231–246, 2007.

[126] T. Ojala, M. Pietikäinen, and D. Harwood, "A comparative study of texture measures with classification based on featured distributions," *Pattern recognition*, vol. 29, no. 1, pp. 51–59, 1996.

[127] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.

[128] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 1. IEEE, 2005, pp. 886–893.

[129] X. Tan and B. Triggs, "Enhanced local texture feature sets for face recognition under difficult lighting conditions," *IEEE transactions on image processing*, vol. 19, no. 6, pp. 1635–1650, 2010.

[130] F. Song, X. Tan, X. Liu, and S. Chen, "Eyes closeness detection from still images with multi-scale histograms of principal oriented gradients," *Pattern Recognition*, vol. 47, no. 9, pp. 2825–2838, 2014.

[131] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, vol. 2. IEEE, 2006, pp. 2169–2178.

[132] G. Mahalingam and K. Ricanek, "Lbp-based periocular recognition on challenging face datasets," *EURASIP Journal on Image and Video processing*, vol. 2013, no. 1, p. 36, 2013.

[133] L. Wolf, T. Hassner, and Y. Taigman, "Descriptor based methods in the wild," in *Workshop on Faces in'Real-Life'Images: Detection, Alignment, and Recognition*, 2008.

[134] G. D. Furman, A. Baharav, C. Cahan, and S. Akselrod, "Early detection of falling asleep at the wheel: A heart rate variability approach," in *Computers in Cardiology, 2008*. IEEE, 2008, pp. 1109–1112.

[135] X. Qian, X.-S. Hua, P. Chen, and L. Ke, "Plbp: An effective local binary patterns texture descriptor with pyramid representation," *Pattern Recognition*, vol. 44, no. 10, pp. 2502–2515, 2011.

[136] T. Mäenpää and M. Pietikäinen, "Multi-scale binary patterns for texture analysis," *Image analysis*, pp. 267–275, 2003.

[137] C. M. Bishop, *Pattern recognition and machine learning*. springer, 2006.

[138] M. Everingham and A. Zisserman, "Regression and classification approaches to eye localization in face images," in *Automatic Face and Gesture Recognition, 2006. FGR 2006. 7th International Conference on*. IEEE, 2006, pp. 441–446.

[139] Z. Niu, S. Shan, S. Yan, X. Chen, and W. Gao, "2d cascaded adaboost for eye localization," in *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on*, vol. 2. IEEE, 2006, pp. 1216–1219.

[140] D. E. Benrachou, F. N. dos Santos, B. Boulebtateche, and S. Bensaoula, "Eyelsd a robust approach for eye localization and state detection," *Journal of Signal Processing Systems*, pp. 1–27, 2017.

[141] N. Markuš, M. Frljak, I. S. Pandžić, J. Ahlberg, and R. Forchheimer, "Eye pupil localization with an ensemble of randomized trees," *Pattern recognition*, vol. 47, no. 2, pp. 578–587, 2014.

[142] S. R. Eddy, "Hidden markov models," *Current opinion in structural biology*, vol. 6, no. 3, pp. 361–365, 1996.

[143] L. Liu, P. Fieguth, Y. Guo, X. Wang, and M. Pietikäinen, "Local binary features for texture classification: Taxonomy and experimental study," *Pattern Recognition*, vol. 62, pp. 135–160, 2017.

[144] T. Ojala, M. Pietikainen, and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 24, no. 7, pp. 971–987, 2002.

[145] T. Ahonen, A. Hadid, and M. Pietikainen, "Face description with local binary patterns: Application to face recognition," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 28, no. 12, pp. 2037–2041, 2006.

[146] W. Jian and Z. Honglian, "Eye detection based on multi-angle template matching," in *Image Analysis and Signal Processing, 2009. IASP 2009. International Conference on*. IEEE, 2009, pp. 241–244.

[147] D. E. Benrachou, F. N. dos Santos, B. Boulebtateche, and S. Bensaoula, "Online vision-based eye detection: Lbp/svm vs lbp/lstm-rnn," in *CONTROLO'2014–Proceedings of the 11th Portuguese Conference on Automatic Control*. Springer, 2015, pp. 659–668.

[148] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning internal representations by error propagation," California Univ San Diego La Jolla Inst for Cognitive Science, Tech. Rep., 1985.

[149] P. J. Werbos, "Generalization of backpropagation with application to a recurrent gas market model," *Neural networks*, vol. 1, no. 4, pp. 339–356, 1988.

[150] C. M. Bishop, *Neural networks for pattern recognition*. Oxford university press, 1995.

[151] J. S. Bridle, "Probabilistic interpretation of feedforward classification network outputs, with relationships to statistical pattern recognition," in *Neurocomputing*. Springer, 1990, pp. 227–236.

[152] A. Graves *et al.*, *Supervised sequence labelling with recurrent neural networks*. Springer, 2012, vol. 385.

[153] S. Leglaive, R. Hennequin, and R. Badeau, "Singing voice detection with deep recurrent neural networks," in *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*. IEEE, 2015, pp. 121–125.

[154] S. Hochreiter, Y. Bengio, P. Frasconi, J. Schmidhuber *et al.*, "Gradient flow in recurrent nets: the difficulty of learning long-term dependencies," 2001.

[155] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[156] R. J. Williams and D. Zipser, "Gradient-based learning algorithms for recurrent networks and their computational complexity," *Backpropagation: Theory, architectures, and applications*, vol. 1, pp. 433–486, 1995.

[157] S. Hochreiter, "Untersuchungen zu dynamischen neuronalen netzen," *Diploma, Technische Universität München*, vol. 91, 1991.

[158] T. Schaul, J. Bayer, D. Wierstra, Y. Sun, M. Felder, F. Sehnke, T. Rückstieß, and J. Schmidhuber, "Pybrain," *J. Mach. Learn. Res.*, vol. 11, pp. 743–746, Mar. 2010. [Online]. Available: http://dl.acm.org/citation.cfm?id=1756006.1756030

[159] Itseez, "Open source computer vision library," https://github.com/itseez/opencv, 2015.

[160] B. Yang and S. Chen, "A comparative study on local binary pattern (lbp) based face recognition: Lbp histogram versus lbp image," *Neurocomputing*, vol. 120, pp. 365–379, 2013.

[161] L. Nanni, S. Brahnam, and A. Lumini, "A simple method for improving local binary patterns by considering non-uniform patterns," *Pattern Recognition*, vol. 45, no. 10, pp. 3844–3852, 2012.

[162] M. Pietikäinen, T. Ojala, and Z. Xu, "Rotation-invariant texture classification using feature distributions," *Pattern Recognition*, vol. 33, no. 1, pp. 43–52, 2000.

[163] X. Huang, S. Z. Li, and Y. Wang, "Shape localization based on statistical method using extended local binary pattern," in *Image and Graphics (ICIG'04), Third International Conference on*. IEEE, 2004, pp. 184–187.

[164] T. Mäenpää, T. Ojala, M. Pietikäinen, and M. Soriano, "Robust texture classification by subsets of local binary patterns," in *Proc. 15th International Conference on Pattern Recognition*, vol. 3. in: Proc. 15th International Conference on Pattern Recognition, Barcelona, Spain, 3: 947-950., 2000, pp. 947–950.

[165] T. Ojala, T. Maenpaa, M. Pietikainen, J. Viertola, J. Kyllonen, and S. Huovinen, "Outex-new framework for empirical evaluation of texture analysis algorithms," in *Pattern Recognition, 2002. Proceedings. 16th International Conference on*, vol. 1. IEEE, 2002, pp. 701–706.

[166] K. I. Laws, "Texture energy measures," in *Proc. Image understanding workshop*, 1979, pp. 47–51.

[167] F. A. Gers, J. A. Schmidhuber, and F. A. Cummins, "Learning to forget: Continual prediction with lstm," *Neural Comput.*, vol. 12, no. 10, pp. 2451–2471, Oct. 2000. [Online]. Available: http://dx.doi.org/10.1162/089976600300015015

[168] F. A. Gers, N. N. Schraudolph, and J. Schmidhuber, "Learning precise timing with lstm recurrent networks," *J. Mach. Learn. Res.*, vol. 3, pp. 115–143, Mar. 2003. [Online]. Available: http://dx.doi.org/10.1162/153244303768966139

[169] L. A. Alexandre and J. P. M. de Sá, "Error entropy minimization for lstm training," in *Proceedings of the 16th International Conference on Artificial Neural Networks - Volume Part I*, ser. ICANN'06. Berlin, Heidelberg: Springer-Verlag, 2006, pp. 244–253. [Online]. Available: http://dx.doi.org/10.1007/11840817_26

[170] W. Zhang, S. Shan, W. Gao, X. Chen, and H. Zhang, "Local gabor binary pattern histogram sequence (lgbphs): A novel non-statistical model for face representation and recognition," in *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, vol. 1. IEEE, 2005, pp. 786–791.

[171] Z. Li, G. Liu, Y. Yang, and J. You, "Scale-and rotation-invariant local binary pattern using scale-adaptive texton and subuniform-based circular shift," *IEEE Transactions on Image Processing*, vol. 21, no. 4, pp. 2130–2140, 2012.

[172] R. Davarzani, S. Mozaffari, and K. Yaghmaie, "Scale-and rotation-invariant texture description with improved local binary pattern features," *Signal Processing*, vol. 111, pp. 274–293, 2015.

[173] Z. Guo, L. Zhang, and D. Zhang, "A completed modeling of local binary pattern operator for texture classification," *IEEE Transactions on Image Processing*, vol. 19, no. 6, pp. 1657–1663, 2010.

[174] L. Liu, Y. Long, P. W. Fieguth, S. Lao, and G. Zhao, "Brint: binary rotation invariant and noise tolerant texture classification," *IEEE Transactions on Image Processing*, vol. 23, no. 7, pp. 3071–3084, 2014.

[175] L. Liu, S. Lao, P. W. Fieguth, Y. Guo, X. Wang, and M. Pietikäinen, "Median robust extended local binary pattern for texture classification," *IEEE Transactions on Image Processing*, vol. 25, no. 3, pp. 1368–1381, 2016.

[176] Z. Guo, L. Zhang, and D. Zhang, "Rotation invariant texture classification using lbp variance (lbpv) with global matching," *Pattern recognition*, vol. 43, no. 3, pp. 706–719, 2010.

[177] T. Ahonen, J. Matas, C. He, and M. Pietikäinen, "Rotation invariant image description with local binary pattern histogram fourier features," *Image analysis*, pp. 61–70, 2009.

[178] T. Mäenpää, *The Local binary pattern approach to texture analysis: Extensions and applications*. Oulun yliopisto, 2003.

[179] T. Ojala, M. Pietikäinen, and T. Mäenpää, "Gray scale and rotation invariant texture classification with local binary patterns," in *Computer Vision-ECCV 2000*. Springer, 2000, pp. 404–420.

[180] G. Zhao, T. Ahonen, J. Matas, and M. Pietikainen, "Rotation-invariant image and video description with local binary pattern features," *Image Processing, IEEE Transactions on*, vol. 21, no. 4, pp. 1465–1477, 2012.

[181] Gi4e face database, [online]. available: http://gi4e.unavarra.es/databases/gi4e/.

[182] A. Georghiades, P. Belhumeur, and D. Kriegman, "From few to many: Illumination cone models for face recognition under variable lighting and pose," *IEEE Trans. Pattern Anal. Mach. Intelligence*, vol. 23, no. 6, pp. 643–660, 2001.

[183] Gi4e head-pose video database, [online]. available: http://gi4e.unavarra.es/databases/hpdb/.

[184] Extended yale-b database, [online]. available: http://vision.ucsd.edu/~leekc/extyaledatabase/extyaleb.html.

[185] D. Harwood, T. Ojala, M. Pietikäinen, S. Kelman, and L. Davis, "Texture classification by center-symmetric auto-correlation, using kullback discrimination of distributions," *Pattern Recognition Letters*, vol. 16, no. 1, pp. 1–10, 1995.

[186] D. E. Benrachou, F. N. dos Santos, B. Boulebtateche, and S. Bensaoula, "Eyelhm: Real-time vision-based approach for eye localization and head motion estimation," in *Autonomous Robot Systems and Competitions (ICARSC), 2016 International Conference on*. IEEE, 2016, pp. 305–310.

[187] "The BioID Face Database." http://www.bioid.com/downloads/software/bioid-face-database.html.

[188] W. Gao, B. Cao, S. Shan, X. Chen, D. Zhou, X. Zhang, and D. Zhao, "The cas-peal large-scale chinese face database and baseline evaluations," *IEEE TRANSACTIONS ON SYSTEMS, MAN AND CYBERNETICS PART A SYSTEMS AND humans"*, vol. 38, no. 1, p. 149, 2008.

[189] D. E. Benrachou, F. N. dos Santos, B. Boulebtateche, and S. Bensaoula, "Automatic eye localization; multi-block lbp vs. pyramidal lbp three-levels image decomposition for eye visual appearance description," in *Iberian Conference on Pattern Recognition and Image Analysis*. Springer, 2015, pp. 718–726.

[190] M. Hassaballah, T. Kanazawa, and S. Ido, "Efficient eye detection method based on grey intensity variance and independent components analysis," *IET Computer Vision*, vol. 4, no. 4, pp. 261–271, 2010.

[191] Y. Wu and Q. Ji, "Learning the deep features for eye detection in uncontrolled conditions," in *Pattern Recognition (ICPR), 2014 22nd International Conference on*. IEEE, 2014, pp. 455–459.

[192] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, vol. 1. IEEE, 2001, pp. I–I.

[193] Q. Wang and J. Yang, "Eye detection in facial images with unconstrained background," *Journal of Pattern Recognition Research*, vol. 1, no. 1, pp. 55–62, 2006.

[194] S. Sirohey, A. Rosenfeld, and Z. Duric, "A method of detecting and tracking irises and eyelids in video," *Pattern Recognition*, vol. 35, no. 6, pp. 1389–1401, 2002.

[195] J. Orozco, F. X. Roca, and J. Gonzàlez, "Real-time gaze tracking with appearance-based models," *Machine Vision and Applications*, vol. 20, no. 6, pp. 353–364, 2009.

[196] K. Arai and R. Mardiyanto, "Real time blinking detection based on gabor filter," *International Journal of Human Computer Interaction*, vol. 1, no. 3, pp. 33–45, 2010.

[197] F. Song, X. Tan, X. Liu, and S. Chen, "Eyes closeness detection from still images with multi-scale histograms of principal oriented gradients," *Pattern Recognition*, vol. 47, no. 9, pp. 2825–2838, 2014.

[198] W. O. Lee, E. C. Lee, and K. R. Park, "Blink detection robust to various facial poses," *Journal of neuroscience methods*, vol. 193, no. 2, pp. 356–372, 2010.

[199] L. Yunqi, Y. Meiling, S. Xiaobing, L. Xiuxia, and O. Jiangfan, "Recognition of eye states in real time video," in *Computer Engineering and Technology, 2009. ICCET'09. International Conference on*, vol. 1. IEEE, 2009, pp. 554–559.

[200] F. Yutian, H. Dexuan, and N. Pingqiang, "A combined eye states identification method for detection of driver fatigue," 2009.

[201] A. Liu, Z. Li, L. Wang, and Y. Zhao, "A practical driver fatigue detection algorithm based on eye state," in *Microelectronics and Electronics (PrimeAsia), 2010 Asia Pacific Conference on Postgraduate Research in*. IEEE, 2010, pp. 235–238.

[202] M. Eriksson and N. P. Papanikotopoulos, "Eye-tracking for detection of driver fatigue," in *Intelligent Transportation System, 1997. ITSC'97., IEEE Conference on*. IEEE, 1997, pp. 314–319.

[203] M. S. Devi and P. R. Bajaj, "Driver fatigue detection based on eye tracking," in *Emerging Trends in Engineering and Technology, 2008. ICETET'08. First International Conference on*. IEEE, 2008, pp. 649–652.

[204] M. Dehnavi and M. Eshghi, "Design and implementation of a real time and train less eye state recognition system," *EURASIP Journal on Advances in Signal Processing*, vol. 2012, no. 1, p. 30, 2012.

[205] L. Lu, X. Ning, M. Qian, and Y. Zhao, "Close eye detected based on synthesized gray projection," in *Advances in Multimedia, Software Engineering and Computing Vol. 2*. Springer, 2011, pp. 345–351.

[206] C. Xu, Y. Zheng, and Z. Wang, "Eye states detection by boosting local binary pattern histogram features," in *Image Processing, 2008. ICIP 2008. 15th IEEE International Conference on*. IEEE, 2008, pp. 1480–1483.

[207] C. Xiang, S. Q. Ding, and T. H. Lee, "Geometrical interpretation and architecture selection of mlp," *IEEE Transactions on Neural Networks*, vol. 16, no. 1, pp. 84–96, 2005.

[208] G. Pan, L. Sun, Z. Wu, and S. Lao, "Eyeblink-based anti-spoofing in face recognition from a generic webcamera," in *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*. IEEE, 2007, pp. 1–8.

[209] L. Goldmann, U. J. Monich, and T. Sikora, "Components and their topology for robust face detection in the presence of partial occlusions," *IEEE Transactions on Information Forensics and Security*, vol. 2, no. 3, pp. 559–569, 2007.