Livrable WP1 - L4 : Détection des mains et du visage du locuteur

Brigitte Bigi

▶ **To cite this version:**

Brigitte Bigi. Livrable WP1 - L4 : Détection des mains et du visage du locuteur. WP1-L4, FIRAH. 2024. hal-04518632

HAL Id: hal-04518632
https://hal.science/hal-04518632

Submitted on 24 Mar 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Automatic Cued Speech / LfPC automatisée

La réalité augmentée au service des personnes sourdes

Livrable WP1 - L4 : Détection des mains et du visage du locuteur

Contexte et objectifs

Description du corpus

CLeLfPC - Corpus de Lecture en LfPC, contient des enregistrements audio/vidéo de lecture à voix haute en codant en Langue française Parlée Complétée. Le corpus a été enregistré en août 2021 à l'occasion du stage organisé par l'ALPC (https://alpc.asso.fr).

Le corpus est constitué des enregistrements de 25 thèmes par 23 participants. Une série de 10 thèmes de lecture avait été établie, elle peut être consultée à cette adresse : https://sppas.org/LFPC/.

Chacun des 10 thèmes se compose de 4 sessions distinctes :

1. enregistrement audio/vidéo de 32 syllabes "CV" isolées (1 seule clé produite pour chaque syllabe),
2. enregistrement audio/vidéo de 32 mots ou expressions,
3. enregistrement audio/vidéo de phrases isolées,
4. enregistrement audio/vidéo d'un texte.

Objectif

Le corpus doit être enrichi d'annotations pour pouvoir être exploité dans le cadre de ce projet. Ce livrable concerne les **annotations de la main codeuse et du visage**. Pour chaque image de chaque vidéo, nous cherchons à déterminer les coordonnées de points spécifiques, à savoir 21 sur la main et 68 sur le visage. L'analyse de ces coordonnées permettra de modéliser la trajectoire suivie par la main et son inclinaison durant le codage, ainsi que la localisation des voyelles autour du visage.

Il est à noter que dans le cadre de ce projet, nous n'analyserons pas les mouvements des doigts, ni la vitesse de déplacement de la main.

> **Résultat majeur : tous les locuteurs du corpus CLeLfPC ont été annotés automatiquement avec SPPAS.**

Outils et méthode d'annotation

Généralités

Nous avons apporté des améliorations au système implémenté dans SPPAS pour générer automatiquement les annotations. Ce système fonctionne en différentes étapes pour optimiser la qualité du résultat, même si c'est au détriment de la rapidité. Sur un ordinateur de bureau, chaque vidéo d'environ 3 minutes a nécessité environ 30 minutes de temps de traitement, soit 10 fois temps réel. Ces traitements ont pour but de générer des coordonnées pour la main codeuse et pour le visage.

Voici les étapes implémentées pour détecter les points du visage :

1. utilisation d'un système existant de détection automatique de visage, ou, utilisation de plusieurs systèmes et fusion de leurs résultats,
2. assignation d'une identité aux différents visages détectés sur les images.

Voici les étapes implémentées pour détecter les points de la main codeuse :

1. utilisation d'un système existant de détection automatique du corps humain,
2. utilisation d'un système existant pour détecter la main et ses points sur les régions sélectionnées pour la main droite et pour la main gauche.

Les différents traitements automatiques sont opérés soit par la bibliothèque OpenCV, soit par Mediapipe.

La bibliothèque OpenCV s'est imposée comme un standard dans le domaine de la recherche parce qu'elle propose un nombre important d'outils issus de l'état de l'art en dans le domaine de la "vision par ordinateurs". En outre, il existe des modèles de visages sous licence libres disponibles sur le web pour cette bibliothèque.

Mediapipe est une bibliothèque qui permet de réaliser différentes tâches de détections automatiques, également dans le domaine de la vision. Contrairement à OpenCV, Mediapipe est une solution toute-en-un, peu paramétrable, et qui inclue les modèles des systèmes proposés.

- En savoir plus sur OpenCV
- En savoir plus sur Mediapipe
- Voir les démos Mediapipe

Détection des points du visage du locuteur

PROBLÉMATIQUE

Dans le domaine de la *vision par ordinateur* la détection d'objets consiste à détecter la présence et la localisation précise d'un ou plusieurs objects dans une image donnée. La détection de visages humains est donc un cas particulier de la détection d'objets, mais la tâche est rendue plus difficile à cause de la forte variabilité intra-classe (couleur de peau, présence de lunettes, géométrie des visages, orientation, ...). Lorsqu'il s'agit d'une vidéo, cette détection s'effectue indépendamment sur chacune des images de la vidéo.

De nombreux problèmes surviennent si on veut suivre le visage d'une personne dans une vidéo. Dans la mesure où la détection des visages s'effectue indépendamment d'une image à l'autre, un "visage" peut apparaître ou disparaître sur une image au milieu d'une séquence. Effectivement, le système ne garantit pas de détecter uniquement le visage de la seule personne présente dans nos vidéos : sur certaines images, le visage ne sera pas détecté, sur d'autres, plusieurs résultats peuvent être proposés. Par ailleurs, rien ne relie le visage d'une personne sur une image à son visage dans l'image suivante. Ainsi, il n'y a pas de cohérence dans la taille de l'objet détecté sur des images consécutives. Enfin, contrairement à ceux des images d'une photo, les visages des images d'une vidéo sont flous dès lors qu'ils sont en mouvement.

Pour le corpus CLeLfPC, le résultat de la détection est assez bon, car le locuteur est assis (il ne bouge relativement pas puisqu'il est en train de lire), la vidéo est filmée à 60 images par secondes, et l'éclairage est excellent. Malgré ces conditions, la détection du visage de quelques locuteurs n'a pas donné les résultats attendus lorsqu'une seule méthode de détection était utilisée. Nous avons donc opté pour la méthode de fusion de résultats de SPPAS.

ILLUSTRATIONS

> **Dans ce document, nous illustrons les résultats obtenus sur la vidéo de démo de SPPAS, car la licence de cette vidéo permet toute forme d'utilisation. Elle est de qualité relativement proche de celle du corpus CLeLfPC (même micro, même caméra, mêmes distances...). Elle dure 10.5 secondes, filmée à 30 images par secondes (313 images en tout).**

OpenCV + Haar Cascade Classifier

Le "Haar Cascade Classifier" est une méthode de détection d'objets proposée par Paul Viola et Michael Jones dans leur article, "*Rapid Object Detection using a Boosted Cascade of Simple Features*" en 2001. C'est une approche basée sur de l'apprentissage automatique où une fonction en cascade est créée à partir de nombreuses images positives et négatives. Il est ensuite utilisé pour détecter des objets dans de nouvelles images. En savoir plus...

▶ Vidéo du résultat de détection de visage avec un HaarCascade Classifier

▼ **Voir la ligne de commande**

Pour obtenir cette vidéo, il faut lancer la ligne de commande suivante :

```
.sppaspyenv~/bin/python ./sppas/bin/facedetection.py -i demo/demo.webm
 -r resources/faces/haarcascade_frontalface_alt.xml --tag=true -o demo_haarcascade
```

Temps d'exécution :

- MacBook Pro, Sonoma 14.2, processeur Apple M2 Max : 16 secondes
- PC-Desktop, Windows 10, processeur Intel i7-6700k : 33 secondes

Versions utilisées : SPPAS 4.18, Python 3.11, OpenCV 4.9.0.

OpenCV + Artificial Neural Network (modèle Caffe)

En savoir plus...

▶ Vidéo du résultat de détection de visage avec un réseau de neurones profond

▾ **Voir la ligne de commande**

Pour obtenir cette vidéo, il faut lancer la ligne de commande suivante :

```
.sppaspyenv~/bin/python ./sppas/bin/facedetection.py -i demo/demo.webm
 -r resources/faces/res10_300x300_ssd_iter_140000_fp16.caffemodel --tag=true -o demo_dnncaffe
```

Temps d'exécution :

- MacBook Pro, Sonoma 14.2, processeur Apple M2 Max : 11 secondes
- PC-Desktop, Windows 10, processeur Intel i7-6700k : 15 secondes

Versions utilisées : SPPAS 4.18, Python 3.11, OpenCV 4.9.0.

OpenCV + Artificial Neural Network (modèle TensorFlow)

En savoir plus...

► Vidéo du résultat de détection de visage avec un réseau de neurones profond

▾ **Voir la ligne de commande**

Pour obtenir cette vidéo, il faut lancer la ligne de commande suivante :

```
.sppaspyenv~/bin/python ./sppas/bin/facedetection.py -i demo/demo.webm
 -r resources/faces/opencv_face_detector_uint8.pb --tag=true -o demo_opencv
```

Temps d'exécution :

- MacBook Pro, Sonoma 14.2, processeur Apple M2 Max : 15 secondes
- PC-Desktop, Windows 10, processeur Intel i7-6700k : 16 secondes

Versions utilisées : SPPAS 4.18, Python 3.11, OpenCV 4.9.0.

Détection avec Mediapipe

► Vidéo du résultat de détection de visage avec Mediapipe

▾ **Voir la ligne de commande**

Pour obtenir cette vidéo, il faut lancer la ligne de commande suivante :

```
.sppaspyenv~/bin/python ./sppas/bin/facedetection.py -i demo/demo.webm
 -r mediapipe --tag=true -o demo_mediapipe
```

Temps d'exécution :

- MacBook Pro, Sonoma 14.2, processeur Apple M2 Max : 11 secondes
- PC-Desktop, Windows 10, processeur Intel i7-6700k : 12 secondes

Versions utilisées : SPPAS 4.18, Python 3.11, Mediapipe 0.10.9.

SOLUTION APPORTÉE

Détection des visages

Fusion des résultats de plusieurs modèles. Selon les locuteurs du corpus, nous avons activé différents modèles afin d'obtenir le meilleur résultat possible. Notamment, nous avons activé le HaarCascade lorsque les systèmes à base de réseaux de neurones ne détectaient pas suffisamment souvent le visage du locuteur.

► Vidéo du résultat de détection de visage avec SPPAS

▾ **Voir la ligne de commande**

Pour obtenir cette vidéo, il faut lancer la ligne de commande suivante :

```
.sppaspyenv~/bin/python ./sppas/bin/facedetection.py -I demo/demo.webm
 --model:opencv_face_detector_uint8.pb=true
 --model:haarcascade_frontalface_alt.xml=true
 --model:res10_300x300_ssd_iter_140000_fp16.caffemodel=false
 --model:mediapipe=true
```

Temps d'exécution :

- MacBook Pro, Sonoma 14.2, processeur Apple M2 Max : 54 secondes
- PC-Desktop, Windows 10, processeur Intel i7-6700k : 168 secondes

Versions utilisées : SPPAS 4.18, Python 3.11, OpenCV 4.9.0, Mediapipe 0.10.9.

Identification des personnes

Cette annotation automatique attribue une identité de personne aux visages détectés d'une vidéo. Elle prend en entrée une vidéo et un fichier annoté avec les coordonnées des visages détectés, et elle produit un fichier annoté avec les coordonnées des visages identifiés. Contrairement à la reconnaissance faciale, cette annotation ne requiert pas d'avoir appris *a priori* un modèle des personnes présentes. Cette annotation permet également d'effectuer un "lissage" des coordonnées du visage dans les séquences d'images de la vidéo.

Pour notre corpus, deux vidéos sont générées : la vidéo initiale avec le visage de la personne identifiée, une vidéo au format selfie de la personne identifiée.

► Vidéo d'identification de la personne -- SPPAS (personne entière)

► Vidéo d'identification de la personne -- SPPAS (selfie)

▾ **Voir la ligne de commande**

Pour obtenir cette vidéo, il faut lancer la ligne de commande suivante :

```
.sppaspyenv~/bin/python ./sppas/bin/faceidentity
```

Temps d'exécution :

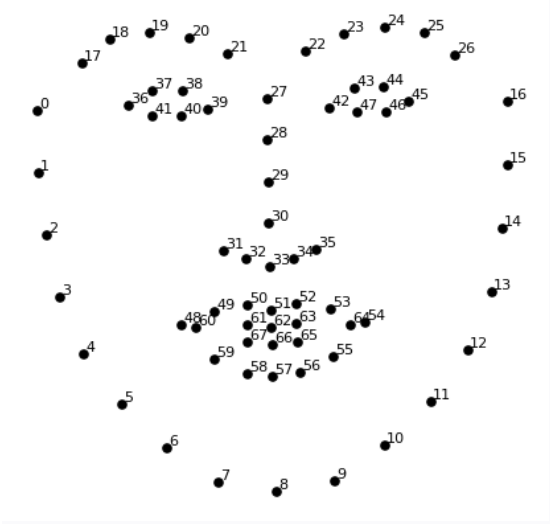
- MacBook Pro, Sonoma 14.2, processeur Apple M2 Max : 36 secondes
- PC-Desktop, Windows 10, processeur Intel i7-6700k : 64 secondes

Versions utilisées : SPPAS 4.18, Python 3.11, OpenCV 4.9.0, Mediapipe 0.10.9.

Détection des points sur le visage

La détection de points sur le visage, appelée "Face Landmark", est opérée à la fois par Mediapipe et par OpenCV avec un modèle libre obtenu sur le web. Leurs résultats sont ensuite fusionnés par SPPAS pour obtenir un résultat unique.

Il est d'usage que ces méthodes déterminent 68 coordonnées de points sur le visage. En revanche, Mediapipe est un "Face Mesh", c'est-à-dire qu'il détermine 435 points que SPPAS réduit aux 68 qui nous intéressent, comme illustré dans l'image ci-après:



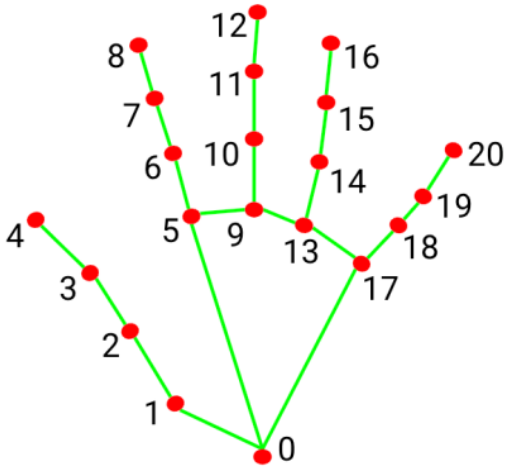
Le résultat est visible sur la vidéo ci-dessous.

Détection des points de la main du locuteur

PROBLÉMATIQUE

La problématique ici et de déterminer où se trouve la même dans chacune des images de la vidéo. Comme pour la détection de visage la difficulté réside dans le fait que le système n'a pas de connaissance *a priori*. Il considère ainsi qu'il est possible qu'il y ait plusieurs mains dans chaque image, et il n'effectue aucun suivi d'une image à l'autre.

Pour réaliser cette tâche, nous avons utilisé l'outil de détection Hand Landmark Detection, inclue également dans *Mediapipe*. Il permet en effet de détecter les mains, à la manière d'une détection d'objet. Nous avons été contraint d'adapter cette solution afin qu'elle ne détecte que deux mains par image : une seule main droite et une seule main gauche.



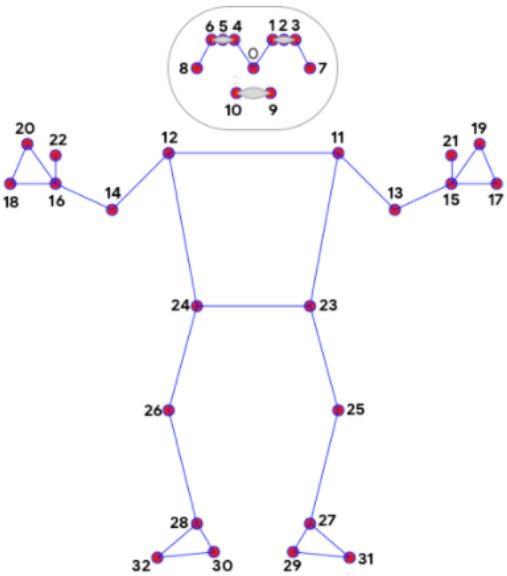
0. WRIST
1. THUMB_CMC
2. THUMB_MCP
3. THUMB_IP
4. THUMB_TIP
5. INDEX_FINGER_MCP
6. INDEX_FINGER_PIP
7. INDEX_FINGER_DIP
8. INDEX_FINGER_TIP
9. MIDDLE_FINGER_MCP
10. MIDDLE_FINGER_PIP
11. MIDDLE_FINGER_DIP
12. MIDDLE_FINGER_TIP
13. RING_FINGER_MCP
14. RING_FINGER_PIP
15. RING_FINGER_DIP
16. RING_FINGER_TIP
17. PINKY_MCP
18. PINKY_PIP
19. PINKY_DIP
20. PINKY_TIP

SOLUTION APPORTÉE

Détection des membres

La solution que nous avons apportée commence par la détection des membres. Pour se faire, nous avons utilisé l'outil "Pose Landmark Detection" proposé également par dans *mediapipe*. Pour chaque image de la vidéo, il détecte les corps entiers des personnes avec des coordonnées associées à chacun de leurs membres.

Lorsqu'il n'y a pas d'erreur de détection (une et une seule personne est détectée), nous pouvons déterminer les régions de l'image dans lesquelles se situent chacune des deux mains. Avec cette annotation, nous disposons d'ores et déjà de quatre coordonnées de points spécifiques sur chaque main.



0. nose
1. left_eye_inner
2. left_eye
3. left_eye_outer
4. right_eye_inner
5. right_eye
6. right_eye_outer
7. left_ear
8. right_ear
9. mouth_left
10. mouth_right
11. left_shoulder
12. right_shoulder
13. left_elbow
14. right_elbow
15. left_wrist
16. right_wrist
17. left_pinky
18. right_pinky
19. left_index
20. right_index
21. left_thumb
22. right_thumb
23. left_hip
24. right_hip
25. left_knee
26. right_knee
27. left_ankle
28. right_ankle
29. left_heel
30. right_heel
31. left_foot_index
32. right_foot_index

Détection des mains

À ce stade, nous avons sélectionné deux régions spécifiques de l'image d'origine afin d'effectuer la détection de chaque main. Pour chacune de ces deux régions, nous appliquons l'outil de détection de mains, de manière classique. Dans cette situation, il y a deux possibilités :

1. soit l'outil de détection a détecté une main et une seule dans la région de l'image, donc nous disposons du résultat attendu, à savoir les coordonnées de 21 points.
2. soit le système n'a pas détecté la main, et nous utilisons les quatre coordonnées déterminées par l'annotation précédente.

Exemples du résultat obtenu

Vidéos générées automatiquement avec le logiciel SPPAS - version 4.18 :

▸ Transcription de la vidéo
▸ Audio de la vidéo
▸ Transcription de la vidéo
▸ Audio de la vidéo

Licences

Vidéos et annotations

Les fichiers vidéos et les annotations sont déposés sous les termes des deux licences suivantes :

- Licence Creative Commons - Attribution 4.0 International, CC-By-NC-4.0
- Licence avec obligation de partage à l'identique ODbL-1.0 : ODC Open Database License (ODbL) version 1.0, conformément à la réglementation française (loi pour une république numérique, 2016).

Ils peuvent être téléchargés à partir de la version 8 du dépôt https://www.ortolang.fr par tout membre d'un Etablissement Supérieur de la Recherche. Pour toute autre demande, envoyer un e-mail à brigitte.bigi[at]cnrs.fr.

Tout usage non prévu ne sera pas autorisé.

Logiciels

SPPAS est déposé sous les termes de la licence publique GNU GPL, v3.

OpenCV (pour Open Computer Vision) est une bibliothèque graphique libre, initialement développée par Intel, spécialisée dans le traitement d'images en temps réel. La société de robotique Willow Garage et la société ItSeez se sont succédé au support de cette bibliothèque. Depuis 2016 et le rachat de ItSeez par Intel, le support est de nouveau assuré par Intel. Cette bibliothèque est distribuée sous les termes de la licence libre BSD (Berkeley Software Distribution License). *Wikipédia*

Mediapipe est distribué par Google sous les termes de la licence libre Apache version 2.0.

Contributeurs

Annotation du corpus : Brigitte Bigi

Dernière mise à jour : 23 mars 2024

Licence de ce document : GNU documentation libre - FDL 1.3

Projet financé par la FIRAH (2023-2026)

Copyright (C) LPL 2023-2024

Ce site respecte votre vie privée.

Nous ne collectons aucune information et n'utilisons pas de cookies.