CNN 3D pour la reconnaissance des émotions faciales dans des vidéos

Jad Haddad, Olivier Lézoray, Philippe Hamel

CNN 3D pour la reconnaissance des émotions faciales dans des vidéos *

Jad HADDAD[1,2], Olivier LÉZORAY[1], Philippe HAMEL[2]

[1]Normandie Univ, UNICAEN, ENSICAEN, CNRS, GREYC, 14000 Caen, France

[2]Zero To One Technology, Campus Effiscience, Colombelles, France

{jad.haddad,philippe.hamel}@zto-technology.com,olivier.lezoray@unicaen.fr

Résumé – Dans cet article, nous considérons les réseaux de neurones convolutifs (CNN) 3D pour la prédiction des émotions faciales dans des vidéos. Nous optimisons l'architecture du CNN 3D par la recherche d'hyper-paramètres, et prouvons que cela a une très forte influence sur les résultats, même si le réglage de l'architecture des CNN 3D n'a pas été beaucoup abordé dans la littérature. Les expériences montrent un avantage par rapport à l'état de l'art.

Abstract – In this paper we consider 3D convolutional neural networks (CNN) for predicting facial emotions in videos. We optimize the 3D-CNN architecture through hyper-parameters search, and prove that this has a very strong influence on the results, even if architecture tuning of 3D CNN has not been much addressed in the literature. Experiments show a benefit over the state of the art.

1 Introduction

La reconnaissance des émotions faciales a suscité beaucoup d'intérêt au cours des dernières décennies, avec des applications en sciences cognitives et en informatique affective. De nombreuses approches ont été proposées pour la reconnaissance automatique des expressions faciales [1]. Récemment, avec l'essor de l'apprentissage profond, des résultats prometteurs ont été rapportés [2–4] en utilisant des caractéristiques géométriques et photométriques apprises. Pour les vidéos, de nombreux travaux ont combiné les CNN 2D classiques avec des réseaux récurrents [5, 6]. Dans ces approches, un RNN (ou LSTM) prend en entrée les caractéristiques extraites par un CNN sur des frames individuelles et encode la dynamique temporelle. Peu de travaux ont cependant été menés sur l'utilisation des CNN 3D [7–9] par rapport à l'usage d'une combinaison CNN-LSTM. Dans cet article nous proposons d'exploiter des CNN 3D dont l'architecture est optimisée pour la reconnaissance des émotions dans les videos [2, 10, 11]. La section 2 présente notre approche pour optimiser une architecture de CNN 3D. La section 3 présente des résultats sur des bases de données de référence. La dernière section conclut.

2 Une architecture CNN 3D Optimisée

Nous considérons des CNN 3D pour la reconnaissance des émotions dans des vidéos. Nous basons nos travaux sur ceux de Tran *et al*. [7]. Premièrement nous modifions leur architecture en régularisant la partie extraction de caractéristiques du réseau avec une normalisation par lot. Deuxièmement nous explorons comment optimiser la structure et les paramètres du réseau pour obtenir de meilleures performances. Notre architecture complète à optimiser est présentée dans la Figure 1. Les composants en blanc sont fixés et les composants en rouge ont leur hyper-paramètres à optimiser. Explorer de manière échantillonnée l'espace de tous les hyper-paramètres peut s'avérer très difficile car soumis à une explosion combinatoire, surtout étant donné le nombre d'hyper-paramètres que nous souhaitons ajuster (ceux associés aux composants en rouge dans la Figure 1). En apprentissage automatique, l'optimisation des hyper-paramètres consiste à trouver les hyper-paramètres du modèle qui minimisent une fonction objectif (généralement le taux d'erreur). Le problème est que l'évaluation de la fonction objectif peut être très coûteuse car elle nécessite de ré-entraîner le modèle pour chaque combinaison d'hyper-paramètres, ce qui est irréalisable. Pour y remédier, au lieu d'utiliser une recherche aléatoire, il est préférable de recourir à l'optimisation Bayésienne. Ces approches Bayésiennes conservent la trace des résultats d'évaluation antérieurs qu'elles utilisent pour former un modèle probabiliste associant les hyper-paramètres à la probabilité d'un score sur la fonction objectif. Le modèle est beaucoup plus facile à optimiser que la fonction objectif réelle. Les méthodes d'optimisation séquentielle basées sur un modèle (SMBO) sont une formalisation de l'optimisation Bayésienne. L'aspect séquentiel fait référence à l'exécution d'essais les uns après les autres, en essayant à chaque fois de meilleurs hyper-paramètres en appliquant un raisonnement Bayésien et en mettant à jour un modèle de probabilité. L'intérêt de SMBO est de réduire le nombre de fois où la fonction objectif doit être exécutée en choisissant uniquement l'ensemble d'hyper-paramètres le plus prometteur à évaluer sur la base des appels précédents à la fonction d'évaluation. L'ensemble suivant d'hyper-paramètres est sélectionné sur la base d'un modèle de la fonction objectif. Le pseudo-code de l'optimisation séquentielle générique basée sur un modèle (SMBO) est présenté dans l'Algorithme 1, avec

*Ce travail a bénéficié d'un financement de l'ANRT dans le cadre d'une thèse CIFRE.

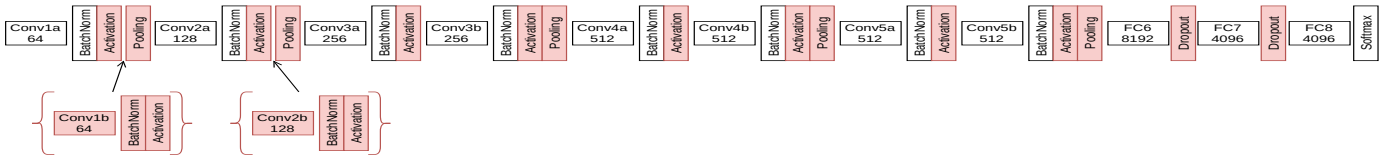

FIGURE 1 – Notre architecture de 3C-CNN dont les composants en rouge sont à optimiser.

f la vraie fonction d'évaluation d'un modèle $f : X \rightarrow \mathbb{R}$, qui est couteuse à évaluer (une mesure d'erreur). La boucle interne d'un algorithme SMBO consiste en l'optimisation d'un modèle associé à la fonction f. Le point x^* qui minimise le modèle devient la proposition pour évaluer la vraie fonction f (dont le nombre d'évaluations doit être contrôlé car couteux). $\mathcal{H}$ est l'historique des observations, M est un modèle ajusté à $\mathcal{H}$, T est le nombre maximum d'essais à effectuer dans la boucle SMBO, et S est la fonction qui sélectionne une nouvelle configuration x à explorer.

Algorithm 1 Le pseudo-code de SMBO.

procedure SMBO(f, M_0, T, S)
  $\mathcal{H} \leftarrow \theta$
  for $t \leftarrow 1, T$ **do**
    $x^* \leftarrow \arg\min_x S(x, M_{t-1})$
    Évaluer $f(x^*)$ ▷ Étape couteuse
    $\mathcal{H} \leftarrow \mathcal{H} \bigcup (x^*, f(x^*))$
    Adapter un nouveau modèle M_t à $\mathcal{H}$
  end for
  return $\mathcal{H}$
end procedure

Il y a plusieurs choix possibles pour le modèle, et nous avons choisi d'utiliser le cadre introduit par Optuna [1] [12] qui considère les estimateurs d'arbres de Parzen comme introduit par Bergstra et al. [13]. Lors de l'exploration de l'espace des hyper-paramètres, TPE ajuste un modèle de mélange Gaussien (GMM) $l(x)$ pour chaque hyper-paramètre à partir de l'ensemble des valeurs de celui-ci qui ont conduit aux meilleures performances et un autre GMM $g(x)$ pour les autres valeurs de l'hyper-paramètre. Ensuite, il choisit la valeur x qui maximise le rapport $l(x)/g(x)$. Nous sommes intéressés par explorer les hyper-paramètres décrits dans le tableau 1 à l'aide d'Optuna.

3 Expériences

Nous évaluons l'intérêt de disposer d'un CNN 3D sur les bases de référence CK+ [14] et Oulu-CASIA [15]. Nous utilisons la technique de validation croisée LOSO (Leave-One-Subject-Out) comme métrique pour structurer et optimiser l'architecture de notre réseau. Cependant, pour comparer nos résultats avec la littérature, nous utiliserons une validation croisée en 10-fold pour le modèle optimal. Nous traitons les 10 dernières images de chaque séquence dont nous extrayons le visage et les redimensionnons en images 112×112. Nous effectuons une augmentation de données en appliquant des transformations

1. http://optuna.org/

Hyper-paramètre	Valeurs possibles
Optimisation	Lookahead+Adam/Adam
CLAHE sur les images	oui/non
Normalisation des images	oui/non
Fonction d'activation	ReLU/ELU/pReLU/ leaky ReLU/Mish
Poids des classes dans la loss CE	oui/non
Taille convolution temporelle	1/3/5/7/9
Initialisation	Xavier uniform/Xavier normal
Couche de pooling	AvgPooling/MaxPooling
Ajout de bloc de convolution	oui/non
Dropout entre les couches FC	$[0, 1]$

TABLE 1 – Valeurs des hyper-paramètres à explorer.

géométriques (miroir, rotation) et photométrique (contraste linéaire). Ceci permet en outre de balancer les quantités de données de chaque classe. Enfin, nous dupliquons la dernière image pour les séquences vidéo de moins de 10 images. Dans SMBO, nous élaguons les essais qui donnent une précision LOSO inférieure à 96,5%, ce qui nous permet d'itérer plus rapidement sur les différentes combinaisons générées par l'estimateur sans trop pénaliser la diversité des solutions explorées.

3.1 Optimisation des hyper-paramètres

Nous présentons tout d'abord le résultat de l'optimisation des hyper-paramètres sur CK+. Plus de 800 configurations différentes ont été explorées. Nous présentons dans la Figure 2 celles dont la LOSO est au dessus du seuil d'élagage. Les meilleurs résultats

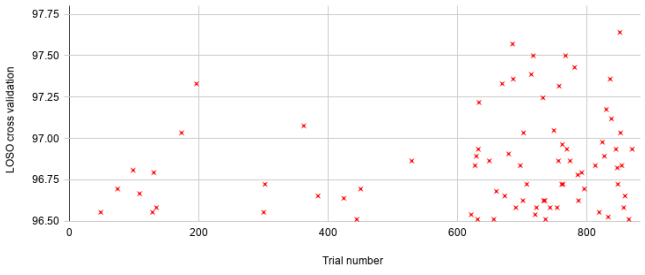

FIGURE 2 – Recherche des meilleurs hyper-paramètres sur CK+. Chaque croix rouge représente une configuration différente.

apparaissent dans la partie haute-droite de la figure. En étudiant les hyper-paramètres des configurations, on peut s'apercevoir que certains hyper-paramètres contribuent fortement à la performance du modèle (il s'agit de l'égalisation CLAHE, l'ini-

tialisation Xavier uniform, une convolution temporelle de taille 3) et que d'autres y nuisent (normalisation, convolution temporelle autre que 3, dropout parmi $[0.5, 1]$, etc.). La meilleure configuration LOSO d'hyper-paramètres sur CK+ donne un taux de **97.56%**.

3.2 Résultats

La configuration finale obtenue pour CK+ est donnée dans la Figure 3 et les hyper-paramètres sont détaillés dans le tableau 2. Une fois l'architecture fixée, nous avons évalué ses performances en validation croisée 10-fold pour nous comparer à l'état de l'art. Nous obtenons un taux de **100%**, meilleur taux jamais obtenu sur ce dataset. Le tableau 3 présente le placement de notre approche avec la littérature. Notre optimisation d'architecture permet à elle seule de devancer des approches plus complexes combinant 3D CNN et LSTM [16], montrant l'importance de la recherche des hyper-paramètres.

Hyper-paramètre	**Valeur**
Optimisation	Adam
CLAHE sur les images	oui
Normalisation des images	non
Fonction d'activation	ReLU
Poids des classes dans la loss CE	non
Taille convolution temporelle	3
Initialisation	Xavier uniform
Couche de pooling	AvgPooling
Ajout de bloc de convolution	non
Dropout	0.2511

TABLE 2 – Hyper-paramètres de l'architecture finale obtenue sur CK+.

Approche	**Précision** (%)
LBP/Gabor + SRC [17]	98.09
DBN + MLP [18]	98.57
CNN [19]	98.62
FAN [20]	99.69
STC-NLSTM [16]	99.8
Notre approche	**100**

TABLE 3 – Résultats en validation croisée 10-fold sur CK+.

Pour Oulu-CASIA, la Figure 4 présente la configuration finale de l'architecture et les hyper-paramètres sont détaillés dans le tableau 4. De même que pour CK+, une fois la meilleure configuration LOSO d'hyper-paramètres obtenue, nous avons évalué ses performances en validation croisée 10-fold pour nous comparer à l'état de l'art. Nous obtenons un taux de **84.25%**. Son placement vis à vis de l'état de l'art est présenté dans le tableau 5. Si notre approche est très bien placée vis à vis des autres approches de CNN 3D, elle est en retrait par rapport à [16] : l'ajout d'une prise en compte plus fine de la dimension temporelle par des LSTM permet un gain significatif. Néanmoins, nous pouvons espérer un gain similaire en modifiant notre architecture avec un LSTM.

Hyper-paramètre	**Valeur**
Optimisation	Lookahead + Adam
CLAHE sur les images	non
Normalisation des images	oui
Fonction d'activation	Leaky ReLU
Poids des classes dans la loss CE	non
Taille convolution temporelle	3
Initialisation	Xavier normal
Couche de pooling	MaxPooling
Ajout de bloc de convolution	non
Dropout	0.4233

TABLE 4 – Hyper-paramètres de l'architecture finale obtenue sur OULU-CASIA.

Approche	**Accuracy** (%)
FLT [2]	74.17
C3D [2]	74.38
FLT+C3D [2]	81.49
Our approach	84.25
STC-NLSTM [16]	**93.45**

TABLE 5 – Résultats en validation croisée 10-fold sur Oulu-CASIA.

Pour illustrer l'importance d'avoir une architecture optimisée pour une base de donnée particulière, nous étudions cela relativement à un transfert d'apprentissage d'un réseau vers un autre. Pour cela, nous évaluons l'architecture optimisée de CK+ sur Oulu-CASIA et vice-versa. Ceci permet de quantifier l'importance d'avoir une architecture qui est optimisée pour le jeu de données en question par opposition à avoir une architecture commune pour les deux jeux de données, même en effectuant un fine tuning de celle-ci. Les tests sont faits en validation croisée 10-fold et nous présentons le taux moyen dans le tableau 6. On peut constater que l'architecture optimisée sur une base de données fournit toujours de meilleurs résultats qu'une autre architecture optimisée sur un autre jeu de données. Un simple transfer learning n'est donc pas suffisant pour obtenir de bonnes performances. Un fine-tuning ne permet pas non plus d'améliorer les résultats, ce qui confirme l'importance d'avoir une architecture optimisée pour chaque base de données : même si les architectures diffèrent peu, leur hyper-paramètres respectifs peuvent amener à des résultats très différents. Ces résultats étant effectués en validation croisée, on peut mesurer à l'aide d'un test statistique de la valeur de p si les différences obtenues sont significatives, ce qui s'avère être le cas.

4 Conclusion

Nous avons proposé des CNN 3D pour la reconnaissance des expressions faciales dans des vidéos. Les CNN 3D peuvent extraire des caractéristiques temporelles très subtiles qui per-

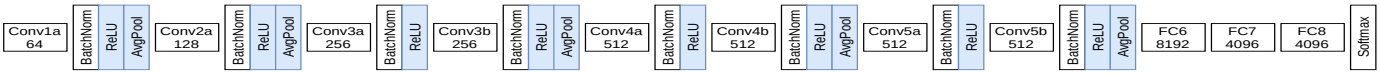

FIGURE 3 – Architecture finale pour CK+, le nombre de filtres est précisé dans chaque boite de convolution.

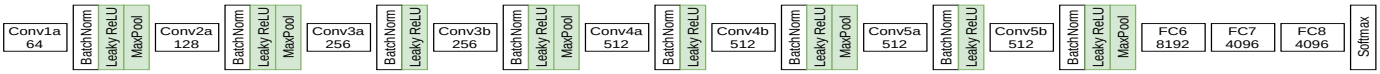

FIGURE 4 – Architecture finale pour OULU-CASIA, le nombre de filtres est précisé dans chaque boite de convolution

Architecture	Base d'évaluation	Post-traitement	Taux moyen en 10-fold CV
Architecture optimisée sur CK+	CK+	Aucun	100
Architecture optimisée sur OULU-CASIA	CK+	Aucun	90.76
Architecture optimisée sur OULU-CASIA	CK+	Fine-tuning sur CK+	91.14
Architecture optimisée sur OULU-CASIA	OULU-CASIA	Aucun	84.25
Architecture optimisée sur CK+	OULU-CASIA	Aucun	77.08
Architecture optimisée sur CK+	OULU-CASIA	Fine-tuning sur OULU-CASIA	79.75

TABLE 6 – Résultats de transferts de modèles optimisés entre bases de données (en validation croisée 10-fold).

mettent d'aller au-delà des CNN 2D. Cependant, leur conception peut être délicate et nous avons proposé d'utiliser une recherche efficace d'hyper-paramètres pour résoudre ce problème. Les expériences ont confirmé l'intérêt de notre approche. Les résultats sur CK+ montrent que notre réseau surpasse les résultats actuels de l'état de l'art des CNN 3D sur CK+ et OULU-CASIA (sans LSTM pour cette dernière). Nous avons en outre montré que disposer d'une architecture optimisée peut s'avérer plus intéressant que du transfer learning avec ou sans fine-tuning.

Références

[1] Y. Huang, F. Chen, S. Lv, and X. Wang, "Facial expression recognition : A survey," *Symmetry*, vol. 11, no. 10, 2019.

[2] H. Jung, S. Lee, J. Yim, S. Park, and J. Kim, "Joint fine-tuning in deep neural networks for facial expression recognition," *IEEE ICCV*, vol. 2015 Inter, pp. 2983–2991, 2015.

[3] A. Mollahosseini, D. Chan, and M. H. Mahoor, "Going deeper in facial expression recognition using deep neural networks," *IEEE WACV*, 2016.

[4] G. Sharma, L. Singh, and S. Gautam, "Automatic Facial Expression Recognition Using Combined Geometric Features," *3D Research*, vol. 10, no. 2, 2019.

[5] T. H. S. Li, P. H. Kuo, T. N. Tsai, and P. C. Luan, "CNN and LSTM Based Facial Expression Analysis Model for a Humanoid Robot," *IEEE Access*, vol. 7, pp. 93 998–94 011, 2019.

[6] N. Jain, S. Kumar, A. Kumar, P. Shamsolmoali, and M. Zareapoor, "Hybrid deep neural networks for face emotion recognition," *Pattern Recognition Letters*, vol. 115, pp. 101–106, 2018.

[7] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3D convolutional networks," *IEEE ICCV*, pp. 4489–4497, 2015.

[8] B. Hasani and M. H. Mahoor, "Facial Expression Recognition Using Enhanced Deep 3D Convolutional Neural Networks," in *IEEE CVPR*, may 2017, pp. 2278–2288.

[9] J. Zhao, X. Mao, and J. Zhang, "Learning deep facial expression features from image and optical flow sequences using 3D CNN," *Visual Computer*, vol. 34, no. 10, pp. 1461–1475, 2018.

[10] J. Haddad, O. Lézoray, and P. Hamel, "3d-cnn for facial emotion recognition in videos," in *International Symposium on Visual Computing (ISVC)*, vol. LNCS 12510, 2020, pp. 298–309.

[11] S. P. Teja Reddy, S. Teja Karri, S. R. Dubey, and S. Mukherjee, "Spontaneous Facial Micro-Expression Recognition using 3D Spatiotemporal Convolutional Neural Networks," in *JCNN*, 2019, pp. 1–8.

[12] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, "Optuna : A Next-generation Hyperparameter Optimization Framework," *ACM SIGKDD*, pp. 2623–2631, 2019.

[13] J. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl, "Algorithms for hyper-parameter optimization," *NIPS*, pp. 1–9, 2011.

[14] P. Lucey, J. F. Cohn, T. Kanade, J. Saragih, Z. Ambadar, and I. Matthews, "The Extended Cohn-Kanade Dataset (CK+) : A complete dataset for action unit and emotion-specified expression," in *IEEE CVPR*, 2010, pp. 94–101.

[15] G. Zhao, X. Huang, M. Taini, S. Z. Li, and M. Pietikäinen, "Facial expression recognition from near-infrared videos," *Image and Vision Computing*, vol. 29, no. 9, pp. 607–619, aug 2011.

[16] Z. Yu, G. Liu, Q. Liu, and J. Deng, "Spatio-temporal convolutional features with nested LSTM for facial expression recognition," *Neurocomputing*, vol. 317, pp. 50–57, 2018.

[17] S. Zhang, X. Zhao, and B. Lei, "Facial expression recognition using sparse representation," *WSEAS Transactions on Systems*, vol. 11, no. 8, pp. 440–452, 2012.

[18] X. Zhao, X. Shi, and S. Zhang, "Facial expression recognition via deep learning," *IETE Technical Review (Institution of Electronics and Telecommunication Engineers, India)*, vol. 32, no. 5, pp. 347–355, 2015.

[19] R. Breuer and R. Kimmel, "A Deep Learning Perspective on the Origin of Facial Expressions," pp. 1–16, 2017.

[20] D. Meng, X. Peng, K. Wang, and Y. Qiao, "Frame Attention Networks for Facial Expression Recognition in Videos," *Proceedings - International Conference on Image Processing, ICIP*, vol. 2019-Septe, no. September, pp. 3866–3870, 2019.