DNN - Privileged Attribution Constrained Deep Networks for Facial Expression Recognition

Guillaume CARRIERE, Romain GREGOIRE, Corentin PION, Alex POIRON, Tom THIL

1 Introduction et présentation du sujet

Dans ce rapport nous allons présenter le travail que nous avons réalisé dans le cadre du cours de Deep Neural Networks (DNN). Dans un premier temps, nous ferons une présentation de notre sujet ainsi qu'un résumé de l'article scientifique qui nous a été attribué. Nous verrons ensuite comment nous avons réalisé notre implémentation suivant les indications du papier de recherche. Enfin nous discuterons des différents résultats et des performances obtenus.

Notre projet a pour sujet **la reconnaissance d'expression faciale**. Une expression faciale correspond à un signe visible sur le visage qui indique ce que ressent une personne. Cette expression du visage peut ainsi montrer différentes émotions comme la joie, la tristesse ou encore la douleur. Le domaine de recherche formé autour de ces expressions est très important et permet de mieux comprendre le comportement humain. L'objectif que nous nous sommes fixés est de déceler, à partir d'une image de visage, l'expression qui correspond à son expression faciale. Nous traiterons ici 7 émotions diférentes : tristesse, colère, bonheur, surprise, dégoût, peur ainsi que neutre. On pourrait donc ramener cette problématique à un modèle de classification à 7 classes différentes.

Afin d'essayer d'améliorer les résultats obtenus à l'état de l'art sur ce problème de classification, Bonnard et al. ont rédigé un article présentant une nouvelle fonction de perte. En effet, celle-ci se focalise sur les zones importantes du visage et donc va permettre d'avoir une attribution dite **privilégiée** sur ces mêmes zones essentielles à la détermination de l'expression faciale. Nous avons repris ces travaux et nous vous présenterons notre implémentations ainsi que les résultats obtenus. Avant cela, nous allons expliquer en détail l'article afin d'avoir un contexte théorique du sujet.

2 Résumé du papier

Cet article s'intitule **Privileged Attribution Constrained Deep Networks for Facial Expression Recognition [Bonnard et al. 2022]** et traite comme son nom l'indique de la reconnaissance d'expression faciale (*FER*). Le problème des méthodes de FER, c'est que les jeux de données sont en général assez petits et contiennent des données bruitées. On va alors parfois avoir besoin de spécifier dans les images présentes dans le jeu de données des **repères**, aussi appelés (*landmarks*). Ces repères pourront ainsi guider le modèle afin que celui-ci se focalise plus particulièrement dessus. On parle ici de zones faciales spécifiques comme les yeux, la bouche ou les sourcils. Ce papier introduit différentes solutions afin de permettre un apprentissage exploitant ces *landmarks* :

- Une fonction de perte : **Privileged Attribution Loss (PAL)**, qui va comme expliqué précédemment suggérer modèle au modèle de donner de l'importance à ces zones. Cette *loss* va maximiser la *cross-correlation* entre des cartes d'attribution (*attribution maps*) ainsi qu'une carte de chaleur formée par les repères faciaux (*heatmap*).
- Des stratégies au niveau des canaux (*channel strategies*) afin de laisser une certaine liberté au modèle quant à ses prédictions.

2.1 Introduction à la PAL

Dans un premier temps, concentrons nous sur la *PAL* et voyons comme celle-ci est formulée :

$$L_{PAL}(\Theta) = -\sum_{i,j,c} \frac{a^l_{i,j,c} - \mu(a^l)}{\sigma(a^l)} a^*_{i,j,c}$$

On retrouve donc le terme $a^l_{i,j,c}$ qui correspond à notre valeur d'attribution pour le pixel (i,j) du c-ième canal de l'image à la couche l. Cette valeur d'attribution correspond à l'influence de notre pixel sur le résultat. Toutes ces valeurs calculées sur une seule image d'entrée vont permettre de guider le modèle pendant son apprentissage. Dans l'article, il nous est présenté deux méthodes afin d'obtenir cette valeur :

- **Grad**: $a^l_{i,j,c}(I) = |\frac{\delta \sum f_o}{\delta f^l_{i,j,c}}(I)|$
- **Grad*Input**: $a^l_{i,j,c}(I) = |\frac{\delta \sum f_o}{\delta f^l_{i,j,c}}(I)|.f^l_{i,j,c}(I)$

La différence dans ces deux formules est que **Grad** reflète à quel point un changement minime au sein de l'image d'entrée peut impacter la valeur en sortie du réseau. Tandis que **Grad*Input** reflète l'attribution totale d'une *feature* sur la valeur en sortie du réseau.

Désormais, passons au terme $a^*_{i,j,c}$. La carte a^* est celle qui va mettre en évidence certains *landmarks* dans l'image d'entrée. Ces régions seront représentées par des pixels blancs et donc à l'inverse, les zones ayant une importance moindre seront représentée en noir. La valeur $a^*_{i,j,c}$ correspond quant à elle, à la valeur de la *map* a^* au pixel (i,j) et peut s'écrire de la manière suivante :

$$a^*_{i,j,c} = \mathbb{1}^L_{i,j}$$

On applique par la suite un **filtre gaussien** sur a^* avec un écart-type $\sigma = 3$. On obtient alors la valeur suivante qui est utilisée dans la fonction **PAL** :

$$a^{*filtered}_{i,j} = \sum_{k=1}^{n} \frac{1}{\sqrt{2\pi\sigma^2}} \exp(-\frac{(i - y_{k,1})^2 + (j - y_{k,2})^2}{2\sigma^2})$$

2.2 Channel strategies

L'article énonce le fait que par rapport à ce qui a été défini, nous avons une attribution $a^l_{i,j,c}$ qui est contrainte à correspondre à une certaine carte a^* sur tous ses *channels*. Cependant, forcer à ce que tous les *channels* soient similaire à cette *heatmap* constitue une contrainte trop importante. Afin de résoudre ce problème, deux stratégies ont été mises en place dans l'article. La première est la **Mean Strategy** qui consiste à forcer la moyenne des attributions sur l'ensemble des canaux, à ressembler à a^*. La valeur de l'attribution pour le pixel (i,j) à la couche l devient alors:

$$a^l_{i,j} = \frac{1}{C} \sum_{c=1}^{C} a^l_{i,j,c}$$

Cependant cette technique est potentiellement toujours trop contraignante. Afin d'autoriser à certains canaux de rester libre et d'apprendre d'autres motifs qui peuvent être important pour les prédictions, on peut réécrire cette formule avec une valeur C_1 telle que $C_1 < C$. Cette autre stratégie s'appelle la **Half Mean Strategy** puisque dans le papier, $C_1 = \frac{C}{2}$. On en conclut donc :

$$a^l_{i,j} = \frac{1}{C_1} \sum_{c=1}^{C_1} a^l_{i,j,c}$$

2.3 Expérimentations et résultats

Dans la dernière partie de l'article nous retrouvons donc l'explication des résultats et des experimentations. Ils ont utilisés deux jeux de données différents : RAF-DB et AffecNet. Les deux sont des *datasets* d'images qui ont été annotés manuellement et contenant 7 émotions basiques, commme cité plus haut. L'implémentation a été réalisé à l'aide d'un **backbone VGG16 pré-entraîné sur VGGFace [Omkar M. Parkhi and Zisserman 2015]**. Pour améliorer les résultats ils ont également utilisé un **backbone ResNet50 pré-entraîné sur VGGFace [Omkar M. Parkhi and Zisserman 2015]**. Nous expliquerons par la suite en détail les paramètres utilisés durant l'implémentation.

En termes de résultats, toutes les combinaisons des différentes méthodes présentées ont été testées. Les résultats obtenus, sont très encourageant et démontrent une nette amélioration sur cette problématique tout en n'ayant besoin d'aucune infomations supplémentaires. On donnera ici deux tableaux comparatifs tirés directement de l'article visibles en fig. 1.

TABLE I
ABLATION STUDY ON RAF-DB DATASET COMPARING DIFFERENT ATTRIBUTION METHODS AND CHANNEL STRATEGIES.

Method	*Attribution*	*Channels*	*Acc*
$VGG16$	$---$	$---$	85.4 ± 0.2
$VGG16 + PAL$	$Grad$	All Channels	85.31 ± 0.23
$VGG16 + PAL$	$Grad$	Mean	86.21 ± 0.12
$VGG16 + PAL$	$Grad * Input$	Mean	86.38 ± 0.17
$VGG16 + PAL$	$Grad$	Mean of half	86.46 ± 0.13
$VGG16 + PAL$	$Grad * Input$	Mean of half	86.82 ± 0.1

TABLE II
ABLATION STUDY ON RAF-DB DATASET COMPARING OF DIFFERENT VALUES FOR C_1 ON THE CHANNEL STRATEGY.

Method	C_1	*Acc*
$VGG16$	$---$	85.4 ± 0.2
$VGG16 + PAL$	$C/4$	86.55 ± 0.56
$VGG16 + PAL$	$C/2$	86.82 ± 0.1
$VGG16 + PAL$	$3C/4$	86.32 ± 0.25
$VGG16 + PAL$	C	86.38 ± 0.17

Figure 1: Résultats obtenus par les auteurs originaux

3 Implémentation

Pour ce qui est de notre propre implémentation et expérimentation, nous avons voulu reproduire deux modèles présentés dans l'ablation study du papier à savoir: la baseline avec laquelle est comparée la méthode proposée et le modèle le plus performant obtenu par les auteurs sur RAF-DB. Ainsi notre baseline sera le modèle VGG16 pré-entrainé sur VGGFace, puis réentrainé à classifier RAF-DB. Pour le second modèle nous avons entrainé la même base VGG16 sur RAF-DB mais cette fois ci en appliquant la **PAL** sur la couche 15 de VGG16, avec la méthode d'attribution *Grad*Input* et la channel strategy "Half mean" (C/2). Cette configuration semble être celle qui a conduit aux meilleurs résultat dans le papier original. Ainsi, nous pourrons comparer ces deux modèles afin de voir clairement l'impact apporté par la PAL. Pour implémenter ces modèles, nous avons choisi la librairie pytorch.

3.1 Construction du dataset

Pour le dataset, nous avons choisi RAF-DB, l'un des deux datasets utilisés par les auteurs originaux, en raison de sa taille moins importante adaptée à nos moyens. Cependant, ce dataset ne dispose pas des *landmarks* nécessaires à l'application de la PAL. Malheureusement, nous n'avons pas trouvé d'implémentation open-source de la méthode standard utilisée dans le papier pour obtenir ces landmarks [Arnaud, Dapogny, and Bailly 2019]. Pour obtenir nos landmarks sur RAF-DB, nous avons donc utilisé la librairie dlib, qui implémente une méthode standard de détection de facial landmarls [Kazemi and Sullivan 2014]. En donnant des données labéllisées, un ensemble de *regression tree* est entrainé à détécter la position des *landmarks* directement à partir de l'intensité du pixel (on utilise l'image en niveaux de gris), pas besoin de feature extraction. Du coup, à partir d'un image montrant un visage, on obtient un autre image avec les landmarks en points blancs sur fond noir. Une fois cela fait, comme dans l'article, nous appliquons un filtre gaussien avec une écart-type de 3, cela permet d'améliorer les performances de la reconnaissance d'exprission faciale.

3.2 Pré-entrainement sur VGGFace

Le protocole expérimental du papier d'origine part d'un modèle VGG16 pré-entrainé sur VGGFace. Nous avons donc choisi d'opérer de la même manière, ce qui nous a demandé de trouver des poids pré-entrainés sur VGGFace utilisable sur notre implémentation pytorch. Ces derniers, qui formeront donc la base de nos modèles, ont été obtenus sur la page internet publique de Samuel Albanie (https://samuelalbanie.com/). Nous avons décidé de n'utiliser que les poids pré-entrainés des couches convolutionnelles et non pas des couches *fully connected*, afin de n'utiliser que le *backbone* pré-entrainé de features, conformément au papier.

3.3 Paramètres d'entraînement utilisés

Nous avons choisi de suivre le protocole expérimental du papier et avons donc reproduit les détails d'implémentation des entraînements effectués par les auteurs. Ainsi, nous avons utilisé l'optimiseur ADAM avec un learning rate de 5e-5 sur 75 epochs. Conformément au papier, le learning rate a suivi un *polynomial decay*. Les détails de ce *decay* n'étant pas mentionnés dans le papier, nous avons donc opté pour un *polynomial decay* sur les 75 epochs, avec une valeur de puissance à 2. Ensuite, nous avons appliqué le même pre-processing sur notre dataset, à commencer par un redimensionnement par interpolation à 224x224 (dimensions d'entrée de VGG16). Ce redimensionnement est suivi par une augmentation consistant en une rotation aléatoire de $-10/10°$ ainsi qu'un retournement horizontal aléatoire. Enfin, nous avons sauvegardé le meilleur modèle durant notre entraînement en se basant sur la précision calculée par rapport au dataset de validation. Ce dernier a été obtenu à partir d'un split de 50% stratifié sur le dataset de test. Les proportions finales des données sur les datasets de training/test/validation sont donc respectivement de 80/10/10%.

4 Résultats & performances

Le modèle baseline entraîné obtient une précision de 86.4% sur le dataset de test, ce qui une amélioration de 1% comparé aux résultats obtenus dans le papier. Cela peut être du aux poids de pré-entraînement de VGG différents, ou bien à la méthode utilisée pour obtenir les *landmarks* sur RAF-DB. Pour ce qui est du second modèle utilisant la PAL sur la couche 15, avec Grad*Input comme méthode d'attribution et la channel strategy half-mean, la précision obtenue sur le dataset de test est de 83.3%. Ces résultats sont encourageants, mais moins bon que notre baseline ainsi que les résultats du papier. Cela peut être du également aux poids de pré-entrainements trop stricts pour permettre une adaptation exploitant les facial landmarks, ou potentiellement aux positions de ces derniers. L'évolution de la loss et de la précision durant l'entraînement de nos modèles est visible fig. 2.

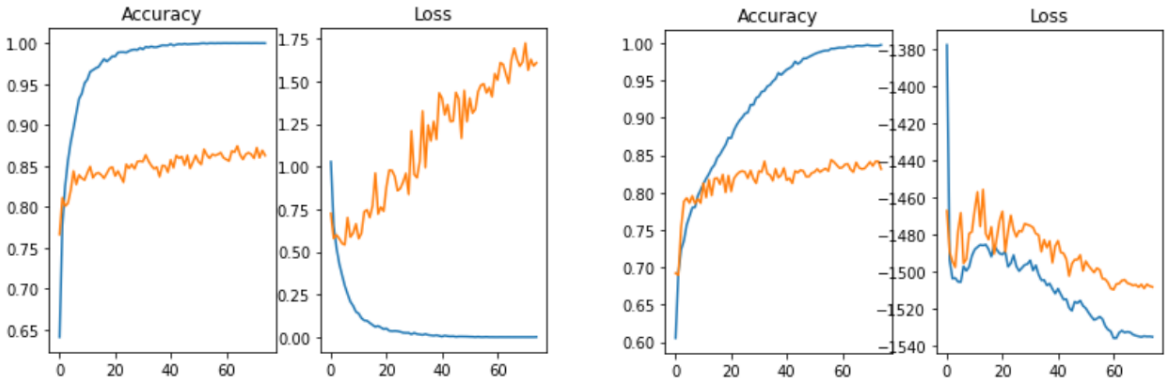

Figure 2: Evolution de la précision et de la loss durant l'entraînement, pour le modèle baseline (gauche) et utilisant la PAL (droite)

5 Conclusion

Pour conclure ce rapport, nous avons reproduit les differentes implémentations du papier et obtenus par consequent des résultats significatifs. Concernant notre baseline nous avons augmenté notre performance mais l'ajout de la méthode spécifique au papier ne nous a pas permis de les améliorer. Cependant notre implémentation de la PAL semble correcte par rapport à ses résultats encourageants. Ainsi, il serait intéressant de tester d'autres configurations et paramètres pour déterminer lesquels correspondent le mieux à notre stratégie d'implémentation afin d'améliorer nos résultats.

References

Arnaud, Estèphe, Arnaud Dapogny, and Kevin Bailly (Oct. 2019). "Tree-Gated Deep Mixture-of-Experts for Pose-Robust Face Alignment". In: 2, pp. 122–132.

Bonnard, Jules et al. (Mar. 2022). *Privileged Attribution Constrained Deep Networks for Facial Expression Recognition.*

Kazemi, Vahid and Josephine Sullivan (2014). "One millisecond face alignment with an ensemble of regression trees". In: *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1867–1874.

Omkar M. Parkhi, Andrea Vedaldi and Andrew Zisserman (Sept. 2015). *Deep Face Recognition.*