



Deep Neural Network:

Privileged Attribution Constrained Deep Networks for Facial Expression Recognition

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



Sommaire

- Introduction et présentation du sujet
- L' article
 - PAL (Privileged Attribution Loss)
 - Channel strategies
 - Expérimentations
- Implémentation
 - Objectifs
 - Construction du dataset
 - Pré-entraînement sur VGGFace
 - Paramètres d'entraînements utilisés
- Résultats & performances
- Conclusion

Introduction & Présentation du sujet

- Reconnaissance d'**expression faciale**
- Problème de classification à **7 classes**
- Etat de l'art
- Notre article



L' article - PAL

- Explication de la **Loss**
- Méthodes d'attributions : **Grad** et **Grad*Input**
- Le **filtre Gaussien**



Les formules: Loss & filtre Gaussien

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

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

Les formules: Attribution methods

- **Grad** : $a_{i,j,c}^l(I) = \left| \frac{\partial \Sigma f_o}{\partial f_{i,j,c}^l}(I) \right|$
- **Grad*Input** : $a_{i,j,c}^l(I) = \left| \frac{\partial \Sigma f_o}{\partial f_{i,j,c}^l}(I) \right| \cdot f_{i,j,c}^l(I)$

L' article - Channel strategies

Les formules :

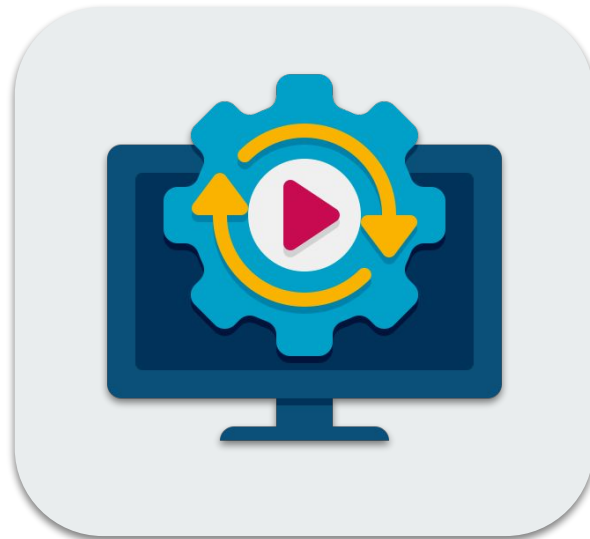
- **Mean Strategy:**
$$a_{i,j}^l = \frac{1}{C} \sum_{c=1}^C a_{i,j,c}^l$$
- **Half Mean Strategy:**
$$a_{i,j}^l = \frac{1}{C_1} \sum_{c=1}^{C_1} a_{i,j,c}^l$$

avec $C_1 = \frac{C}{2}$



L' article - Expérimentations

- AffecNet & **RAF-DB**
- ResNet50 & **VGG16**
- Ablation study





Implémentation - Objectifs

- **Baseline** : VGG16 (pretrained on VGGFace), RAF-DB
- **Best model** : Baseline + PAL on layer 15, Grad*Input, Half of mean strategy

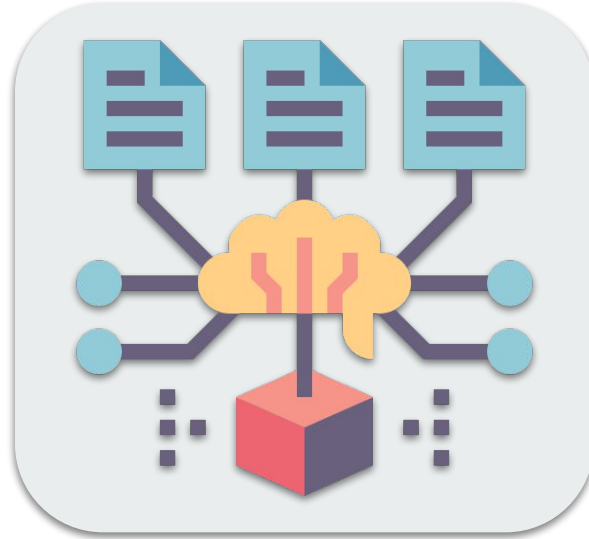
Implémentation - Construction du dataset

- **RAF-DB**
- **Landmarks**
- **Avantages**
- **Inconvénients**



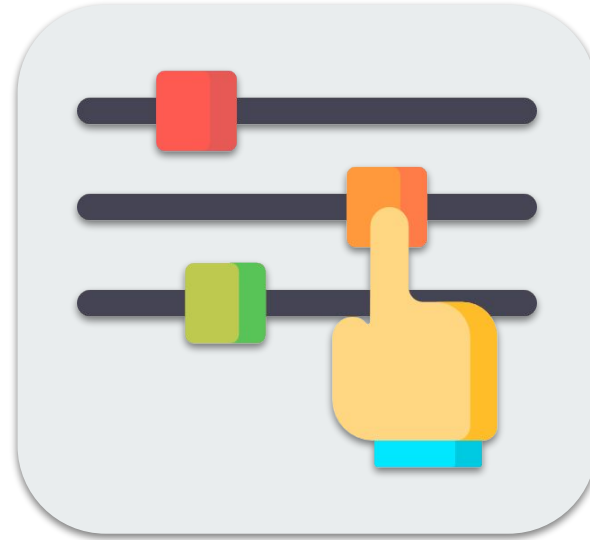
Implémentation - Pré-entraînement sur VGGFace

- VGG16
- VGGFACE
- <https://samuelalbanie.com/>
- Poids Couches Convolutionnelles



Implémentation - Paramètres d'entraînements utilisés

- Optimiseur : ADAM, 75 epochs
- **Learning rate** : $5e-5$, Polynomial decay
- **Pre-processing** et **augmentation** du dataset :
 - Random rotation ($-10/+10^\circ$)
 - Random horizontal flip





Résultats & Performances

Résultats du papier :

- Baseline : 85.4 %
- Best : 86.82%

Nos résultats :

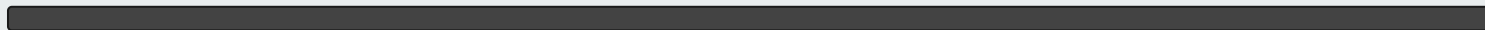
- Baseline : 86.4 %
- Best : 83.3%



Conclusion



Merci de votre attention



Avez-vous des questions ?