Analyse de l'empathie faciale pour la préservation historique par apprentissage profond

Réalisée par Emma Falkiewitz

UV : TZ (6 crédits PSF IAD)

Semestre : A24

Enseignant : SETITRA Insaf et LOURDEAUX Domitile

Introduction

Ce travail de recherche est consacré à l'interprétabilité des algorithmes d'apprentissage et à l'évaluation d'un certain nombre des algorithmes utilisés pour la reconnaissance émotionnelle. Il convient donc de définir l'interprétabilité et à quoi elle sert.

L'interprétabilité des algorithmes d'apprentissage désigne la capacité à comprendre et expliquer les décisions ou prédictions qu'ils produisent de manière compréhensible pour des humains. Si les modèles simples (arbres de décision, régressions linéaires) présentent une transparence intrinsèque, les modèles complexes comme les réseaux de neurones profonds ou les forêts d'arbres aléatoires sont souvent considérés comme des boîtes noires.

Cette opacité pose pourtant certains problèmes de confiance, de diagnostic d'erreur, qui affectent l'adoption de l'IA et de l'apprentissage dans des secteurs sensibles (santé, finance, droit). Par exemple, dans le cas d'une application médicale, il est indispensable de comprendre pourquoi l'algorithme a recommandé tel ou tel traitement afin de pouvoir la croire. De plus, l'interprétabilité est nécessaire pour assurer l'équité des modèles et la conformité aux règles régissant leur utilisation.

On note d'ailleurs que plusieurs types d'approches ont été conçues dans le but de répondre à ce besoin, l'interprétation intrinsèque (lorsque qu'elle est présente) ou encore les méthodes post hoc avec LIME et SHAP qui produisent des interprétations locales (ou globales) d'un modèle opaque. On s'efforcera alors d'apporter une meilleure précision d'exécution et d'efficacité tout en préservant l'interprétabilité pour une utilisation éclairée de l'IA et de l'apprentissage.

Dans cette étude, nous allons explorer l'état de l'art sur l'interprétabilité des classifications mais aussi l'état de l'art sur la reconnaissance des émotions. Ensuite, seront présentés un éventail d'algorithmes capables de classer des expressions faciales. De ce fait, plusieurs méthodes d'apprentissage automatique seront détaillées, testées et leurs résultats seront analysés. Enfin, une partie sur l'explicabilité de ces algorithmes sera développée.

Table des matières

État de l’art sur l’interprétabilité de la classification

Plusieurs approches ont été développées pour expliquer les différents algorithmes de machine learning comme Lime et Shap. *LIME* (Local Interpretable Model-agnostic Explanations) a été introduit en 2016 dans un article publié par Marco Tulio Ribeiro, Sameer Singh et Carlos Guestrin [1]. *SHAP* (SHapley Additive exPlanations) , basé sur la théorie des jeux, a été développé par Scott Lundberg et Su-In Lee et présenté en 2017 [2]. Ces deux méthodes sont devenues des outils majeurs pour l'interprétabilité des modèles complexes en intelligence artificielle.

LIME

LIME est une méthode d’explication des modèles d’apprentissage automatique qui vise à rendre les prédictions des modèles complexes interprétables de manière locale. Contrairement à des techniques d’explication globale, LIME se concentre sur une instance spécifique et génère un modèle simplifié qui approxime le comportement du modèle complexe dans son voisinage immédiat.

Pour expliquer son fonctionnement, LIME estime localement un modèle complexe $f : \mathbb{R}^d \rightarrow \mathbb{R}$ avec un modèle explicable g, généralement une régression linéaire ou un arbre de décision. L’objectif est de minimiser une fonction de perte qui mesure la fidélité de l’explication *g* par rapport à *f* dans le voisinage de l’instance *x* :

$$\xi(x) = \arg\min_{g \in G} L(f, g, \pi_x) + \Omega(g)$$

Équation 1: Minimisation de l'erreur locale pour Lime

Où $L(f, g, \pi_x)$ est une fonction de fidélité qui évalue l’écart entre *f* et *g*, et *Ω(g)* représente la complexité de g, régularisant l'explication pour la rendre interprétable.

Ensuite, LIME génère des perturbations de l'instance d’entrée *x* sous forme de nouvelles instances *z*, en modifiant les caractéristiques de *x :*

- Pour des données textuelles, l’utilisation de vecteurs binaires est utilisé suivant la présence ou non de chaque mot dans le texte.
- Pour les images, chaque région de l’image peut être activée ou désactivée suivant un découpage en « super-pixels ».

Le modèle *g* utilise cette représentation interprétable pour produire des explications. Les perturbations sont une manière de générer des exemples artificiels similaires à une instance *x* pour explorer le comportement local du modèle complexe *f*.

Ces perturbations sont pondérées en fonction de leur proximité avec *x* via une fonction de similarité $\pi_x(z)$, ce qui permet de construire un modèle localement fidèle. Le modèle interprétable *g* est ensuite ajusté pour minimiser l’erreur dans ce voisinage. Le modèle *g* devient une approximation locale de *f* expliquant la contribution de chaque caractéristique à la prédiction pour l’instance *x*. Ces perturbations permettent d'analyser la sensibilité du modèle à des variations mineures des données, aidant à interpréter les mécanismes de décision locaux de *f*. Les

caractéristiques avec des coefficients élevés dans g sont les plus influentes et les contributions positives ou négatives des caractéristiques sont mises en évidence.

LIME est principalement utilisé pour expliquer des modèles de classification dans des domaines tels que la vision par ordinateur et le traitement du langage naturel. Par exemple, dans le cas de la classification d'images, LIME peut identifier les régions importantes de l'image qui influencent la prédiction. Pour les données textuelles, LIME peut déterminer les mots ou expressions qui ont le plus d'impact sur la classification d'un document.

Néanmoins, bien que l'algorithme de LIME soit efficace, il présente plusieurs limitations. La méthode repose sur l'idée de linéarité locale, ce qui peut entraîner des explications peu fidèles si le modèle complexe n'est pas assez linéaire, même localement. De plus, le processus de génération de perturbations et d'ajustement de modèles peut être coûteux en temps de calcul, particulièrement pour de grands ensembles de données ou des modèles complexes.

En conclusion, LIME offre une approche puissante pour rendre les modèles de classification plus transparents et interprétables, notamment en permettant aux utilisateurs de mieux comprendre pourquoi un modèle fait une prédiction donnée. Cependant, ses performances peuvent être limitées par la complexité des modèles sous-jacents et le coût computationnel associé aux perturbations.

SHAP

SHAP (SHapley Additive exPlanations) est une méthode d'interprétabilité basée sur la théorie des jeux coopératifs. Elle utilise les valeurs de Shapley pour attribuer une importance équitable à chaque caractéristique dans une prédiction donnée.

SHAP part du principe que les caractéristiques d'une instance sont traitées comme des *joueurs* dans un jeu coopératif. Le "gain" du jeu est la prédiction *f(x)*. La valeur de Shapley attribue à chaque caractéristique *i* une contribution moyenne sur toutes les combinaisons possibles d'autres caractéristiques. Pour une caractéristique *i*, la valeur de Shapley ϕ_i est donnée par :

$$\phi_i = \sum_{S \subseteq N \setminus i} \frac{|S|!(|N| - |S| - 1)!}{|N|!}[f(S \cup i) - f(S)]$$

Équation 2: Formule des valeurs de Shapley

Avec :

- N : Ensemble des caractéristiques.
- S : Sous-ensemble de N qui ne contient pas i.
- $f(S)$: Prédiction du modèle en ne considérant que les caractéristiques dans S.
- $f(S \cup i) - f(S)$: Contribution marginale de i dans S.
- Les poids sont proportionnels au nombre de permutations possibles des caractéristiques.

Les valeurs de Shapley satisfont des propriétés mathématiques importantes :

- **Symétrie** : Deux caractéristiques ayant le même impact reçoivent la même valeur.
- **Efficacité** : La somme des valeurs de Shapley est égale à la prédiction $f(x)$.
- **Nullité** : Si i n'influence pas f, alors $\phi_i = 0$.

Pour calculer efficacement les valeurs de Shapley, SHAP utilise des approximations adaptées au modèle (e.g., méthodes basées sur des arbres ou des simulations de Monte Carlo). Les contributions calculées sont présentées comme une décomposition additive :

$$f(x) = \phi_0 + \sum_{i=1}^{n} \phi_i$$

Équation 3: Approximations de SHAP

Avec :

- ϕ_0 est la prédiction moyenne sur l'ensemble des données (base value).
- ϕ_i est la contribution attribuée à chaque caractéristique i.

En conclusion, SHAP est largement utilisé pour expliquer des prédictions locales (pour une instance donnée) ou globales (moyenne sur toutes les instances). Il offre une approche rigoureuse pour analyser les influences des caractéristiques dans des modèles complexes.

Dans le reste de notre approche, nous avons préféré Lime à Shap. Ce choix s'est fait dans la contrainte où le code de Lime a été à notre disposition.

État de l’art sur la reconnaissance d’émotion

La reconnaissance des émotions désigne la capacité à identifier et prédire les états émotionnels humains en utilisant une variété de techniques et de modalités. Cela inclut l'analyse de questionnaires, ainsi que des signaux physiques et physiologiques. L'aspect physique englobe l'expression verbale (discours) et la reconnaissance faciale, tandis que l'aspect physiologique s'intéresse aux enregistrements électrophysiologiques tels que l'électroencéphalogramme (EEG), l'électrocardiogramme (ECG), la réponse galvanique de la peau, les stimuli émotionnels et les systèmes automatisés de détection des émotions.

D’après l’article de Khare, S. K., Blanes-Vidal, V., Nadimi, E. S., & Acharya, U. R. [3] qui passe en revue les différentes bases de données utilisées pour la reconnaissance des émotions, la base de données CK+ figure parmi celles mentionnées pour la reconnaissance faciale.

En 2022, plusieurs approches ont été utilisées pour analyser les émotions à partir de CK+, avec des résultats variés :

- L’étude [122] a utilisé deux modèles distincts, SVM et RF, avec des caractéristiques extraites via FLC (Feature Level Combination). Bien que le SVM ait obtenu une précision modeste de 85 %, l'utilisation de Random Forest a significativement amélioré les performances à 97,86 %.
- L’étude [132] a utilisé un modèle LSTM associé à une méthode d'extraction basée sur l’aligned face crop, permettant une excellente précision de 97,2 %.

Ces deux approches montrent que l'alignement des visages et des modèles sophistiqués comme LSTM ou RF peuvent atteindre des performances élevées sur CK+.

En 2021, la combinaison de CNN et LSTM a été explorée dans l'étude [233]. L'extraction de caractéristiques, utilisant des méthodes avancées comme GM-WLBP, GLCM et GLRM, a permis d’obtenir une précision de 91,42 %. Bien que cette performance soit inférieure à celles des modèles plus récents comme RF ou CNN, elle reste robuste, démontrant la pertinence de l'utilisation des caractéristiques texturales combinées à des architectures hybrides.

En 2020 il y a eu une diversité dans les approches utilisées :

- Dans l’étude [138], une combinaison de normalisation, mise à l’échelle et augmentation des données, avec un modèle CNN a atteint une précision remarquable de 99,36 %, l'une des meilleures performances enregistrées sur CK+.
- L’étude [125] a utilisé un modèle DAGSVM et des caractéristiques génériques et texturales. Cette méthode a atteint une précision de 91,11 %. Bien que solide, cette performance est inférieure à celle des méthodes basées sur des réseaux neuronaux plus avancés.
- L’étude [129] a combiné un modèle CNN avec une attention et des techniques de recadrage et d’extraction faciale, une précision de 98,9 % a été obtenue, soulignant l'importance des mécanismes d'attention dans la classification des émotions.

Ces résultats montrent que les approches CNN et leur combinaison avec des techniques d’augmentation des données et d’attention surpassent généralement les modèles classiques comme DAGSVM.

En 2019, une étude majeure ([240]) a introduit le modèle MSWGT-SVM. Grâce à cette méthode, une précision élevée de 98,9 % a été obtenue. Bien que cette performance rivalise avec les

résultats des modèles basés sur CNN, elle s’appuie sur des techniques classiques, démontrant que des algorithmes bien optimisés peuvent encore produire d’excellents résultats.

Comparaison générale

En comparant les méthodes au fil des années, on observe une nette évolution dans l'utilisation des techniques d'apprentissage. Les études plus anciennes utilisaient principalement des modèles traditionnels comme SVM (par exemple, MSWGT-SVM en 2019) ou DAGSVM en 2020, avec des performances comprises entre 91,11 % et 98,9 %. Cependant, les approches récentes favorisent les réseaux neuronaux profonds comme CNN et LSTM, associés à des techniques avancées d'extraction et de traitement des caractéristiques (aligned face crop, attention, augmentation des données). Ces méthodes ont permis d'atteindre des précisions supérieures, souvent au-delà de 97 %, avec un sommet à 99,36 % en 2020 grâce à une combinaison de CNN et d'augmentation des données.

Ainsi, la tendance actuelle privilégie les architectures complexes et hybrides, qui offrent des performances optimales sur CK+, surpassant les techniques plus classiques.

Critique explicabilité

Bien que les travaux récents utilisant CK+ aient démontré des performances impressionnantes, notamment avec des modèles complexes comme CNN, LSTM, ou des architectures hybrides (par exemple, CNN avec attention ou MTCNN), une limite majeure réside dans la non-prise en compte de l’explicabilité. En effet, ces modèles dits « boîtes noires » offrent peu de visibilité sur leurs processus décisionnels.

Les réseaux neuronaux profonds, bien qu'efficaces, sont souvent opaques, rendant difficile l'interprétation de leurs décisions. Cela peut poser des problèmes dans des domaines sensibles comme la reconnaissance des émotions, où la compréhension du pourquoi et du comment des prédictions est essentielle pour éviter les biais et garantir une adoption éthique et responsable. Par exemple, les modèles comme CNN avec attention (Réf. [129]) ou LSTM (Réf. [132]) atteignent des précisions élevées mais sacrifient l’explicabilité au profit de la performance brute.

Notre approche

Pour pallier ces limites, dans notre approche, nous avons choisi d’utiliser des modèles simples et interprétables, en faisant le compromis d’une éventuelle légère baisse de prédiction, tels que SVM et KNN. Ces algorithmes classiques permettent de mieux comprendre les mécanismes sous-jacents au processus de classification. En combinant ces modèles avec des méthodes d’extraction de caractéristiques bien établies, comme :

- HOG (Histogram of Oriented Gradients),
- Descripteurs basés sur les pixels avec et sans normalisation,
- MobileNet et VGG (réseaux convolutifs pré-entraînés).

Nous sommes donc en mesure de décomposer les décisions du classifieur et d’analyser l’influence des différentes caractéristiques sur la prédiction finale. Cette approche favorise une meilleure transparence et une compréhension approfondie du fonctionnement du modèle.

Approche de la classification

Dans notre approche, nous avons exploré différentes méthodes d'extraction de caractéristiques, combinées avec des classifieurs simples (SVM et KNN), pour mieux comprendre l'impact des caractéristiques sur la prédiction des émotions. Voici une présentation détaillée des cinq stratégies d'extraction que nous avons mises en œuvre :

Augmentation des données

Pour chaque cas de notre apprentissage et prédiction, nous avons utilisé l'augmentation de données afin d'avoir une potentielle amélioration (partie Résultats). Pour cela nous avons utilisé plusieurs techniques qui ont été choisies aléatoirement au moment de l'augmentation des données. Les différentes augmentations possibles:

1. Le flou gaussien qui est une technique utilisée pour lisser une image.
2. La transformation affine qui est la combinaison de translations, rotations, mises à l'échelle et cisaillements (distorsions en biais).
3. La transformation totale qui transforme des quadrilatères en d'autres quadrilatères quelconques.
4. La transformation euclidienne qui inclut seulement les rotations et les translations.
5. La modification de contraste qui étire ou compresse les niveaux de gris (ou de couleur).
6. Le retournement de l'image (flip) , renversement horizontal ou vertical.

Extraction des caractéristiques

1. <u>Extraction sans prétraitement (Raw Pixels)</u>

Cette méthode repose sur l'utilisation brute des pixels de l'image sans appliquer de traitement ou de normalisation préalable.

Les images en niveaux de gris sont chargées et converties en vecteurs de pixels unidimensionnels. Chaque vecteur représente les intensités des pixels de l'image.

Étapes clés :

- Les images sont directement extraites en tant que vecteurs à partir des valeurs de pixels.
- Aucun prétraitement ni normalisation n'est appliqué.
- Les données d'entraînement et de test sont générées à partir des intensités des pixels.

Cette méthode permet d'observer les performances du classifieur sans intervention extérieure ni modification des données, offrant une base de référence. Cependant l'absence de normalisation ou de réduction de bruit peut limiter la performance sur des images non homogènes.

2. <u>Extraction via les pixels (avec normalisation)</u>

Cette méthode repose sur l'extraction des pixels des images, mais elle inclut une étape de normalisation pour uniformiser les valeurs des caractéristiques.

Les images sont converties en vecteurs de pixels, et une normalisation est appliquée pour standardiser les données.

Étapes clés :

- Les images en niveaux de gris sont chargées et aplaties en vecteurs unidimensionnels.
- Les valeurs des pixels sont normalisées à l'aide de l'algorithme StandardScaler.
- Les données normalisées sont ensuite utilisées pour entraîner les classifieurs.

La normalisation améliore les performances des modèles, en particulier pour des classifieurs sensibles à l'échelle des données comme SVM. Cependant l'ajout d'une étape de normalisation peut complexifier légèrement le pipeline par rapport à l'approche brute.

3. Extraction via HOG (Histogram of Oriented Gradients)

La troisième méthode utilise le descripteur HOG, qui extrait des caractéristiques basées sur les contours et la structure des images.

HOG capture les informations directionnelles des gradients de l'image, permettant une description robuste des formes et textures.

Étapes clés :

- Les images en niveaux de gris sont pré-traitées et divisées en petites cellules.
- Pour chaque cellule, les gradients sont calculés et les orientations sont regroupées en histogrammes.
- Les histogrammes sont ensuite concaténés pour obtenir un vecteur de caractéristiques final.

HOG est efficace pour capturer des informations discriminantes dans les images, même en présence de variations légères. Néanmoins, la méthode peut être sensible à des variations importantes dans la pose ou l’éclairage.

4. Extraction via MobileNet

La quatrième approche utilise MobileNet, un modèle de réseau neuronal convolutif pré-entraîné, comme méthode d'extraction de caractéristiques.

MobileNet, conçu pour être léger et efficace, est utilisé comme un extracteur de caractéristiques en supprimant les couches de classification finales.

Étapes clés :

- Les images sont redimensionnées pour correspondre aux dimensions d'entrée de MobileNet.
- Le réseau pré-entraîné sur ImageNet extrait des caractéristiques profondes.
- Ces caractéristiques sont ensuite utilisées pour entraîner un classifieur comme SVM ou KNN.

MobileNet offre une extraction rapide et précise des caractéristiques grâce à son architecture optimisée pour les appareils légers. Par contre, les caractéristiques extraites sont abstraites, rendant leur interprétation difficile.

5. Extraction via VGG

Enfin, nous avons utilisé le réseau VGG comme méthode d'extraction de caractéristiques. VGG est un réseau convolutif plus profond que MobileNet, avec des couches empilées qui extraient des caractéristiques hiérarchiques complexes.

VGG est utilisé comme un extracteur de caractéristiques profondes, fournissant des représentations riches des images.

Étapes clés :

- Les images sont redimensionnées pour correspondre aux dimensions d'entrée de VGG.
- Les couches finales de classification sont supprimées, et les sorties intermédiaires sont utilisées comme vecteurs de caractéristiques.
- Ces vecteurs alimentent les classifieurs SVM ou KNN.

VGG capture des caractéristiques riches et discriminantes, idéales pour la classification fine des émotions. Cependant, les calculs sont plus coûteux en termes de temps et de mémoire par rapport à MobileNet.

Modèles utilisés

Support Vector Machine (SVM)

SVM est un modèle de classification qui sépare les classes en trouvant une ligne (ou un hyperplan) qui divise les données de manière optimale. Il maximise la marge entre les points des deux classes les plus proches.

L' hyperplan optimal est une surface (ligne en 2D, plan en 3D, etc.) qui sépare les classes avec la plus grande marge possible. Les vecteurs supports sont les points les plus proches de l'hyperplan. Ils influencent directement sa position et son orientation. Quand les données ne peuvent pas être séparées par une ligne simple, SVM utilise des fonctions noyaux (kernels) pour projeter les données dans un espace de plus grande dimension, où une séparation devient possible.

Les types de noyaux couramment utilisés sont les suivants :

1. Linear Kernel :
    - Utilisé lorsque les données sont linéairement séparables.
    - Fonction : $K(x_i, x_j) = x_i \cdot x_j$.
2. Polynomial Kernel :
    - Gère des relations complexes en créant des séparations de type polynomiale.
    - Fonction : $K(x_i, x_j) = (\gamma x_i \cdot x_j + r)^d$, où d est le degré du polynôme.
3. RBF (Radial Basis Function) Kernel :
    - Très populaire pour des données complexes et non linéaires.
    - Fonction : $K(x_i, x_j) = \exp(-\gamma \| x_i - x_j \|^2)$, où γ contrôle la largeur de la zone d'influence des points.

En résumé, SVM est puissant et flexible grâce aux noyaux, ce qui le rend adapté à une variété de problèmes.

K-Nearest Neighbors (KNN)

KNN est un algorithme simple qui classe un point en fonction des classes de ses K voisins les plus proches dans l'espace des caractéristiques.

Lorsqu'un point doit être classé, on calcule la distance (souvent euclidienne) entre ce point et tous les points d'entraînement. Les K points les plus proches sont sélectionnés, et la classe majoritaire parmi eux est attribuée au point.

K est un hyperparamètre qui influence directement la performance de l'algorithme : un K trop faible (par exemple, K=1) rend le modèle sensible au bruit. Et au contraire, un K trop grand peut ignorer les structures locales et mélanger les classes.

Pour choisir K, on utilise la validation croisée :

- On divise les données en plusieurs sous-ensembles (folds).

- On entraîne KNN sur une partie des données et évalue sa précision sur les données restantes, pour différentes valeurs de K.
- On sélectionne le K qui maximise les performances moyennes sur tous les folds.

En résumé, KNN est intuitif et facile à mettre en œuvre, mais son efficacité dépend fortement du choix de K et de la distance utilisée.

Explicabilité de la classification

Dans le cadre de l'analyse des modèles de machine learning (ici SVM ou KNN), l'explicabilité est une composante essentielle pour comprendre, justifier, et améliorer leurs décisions. Pour répondre à ce besoin, nous avons adopté LIME (Local Interpretable Model-agnostic Explanations) comme méthode principale pour expliquer les prédictions de nos modèles. Trois éléments principaux jouent un rôle crucial dans cette approche : les superpixels, les perturbations, et la normalisation.

1. Superpixels : Décomposer l'image

Les superpixels représentent des régions cohérentes dans une image en termes de couleur et de texture. Ils permettent de regrouper des pixels adjacents en unités interprétables, simplifiant ainsi l'analyse.

Les superpixels sont générés en appliquant l'algorithme de segmentation *Quickshift* à partir d'une image convertie en format RGB. Chaque superpixel est identifié par un entier unique, permettant d'activer ou de désactiver ces régions lors des perturbations.

2. Perturbations : Générer des variations locales

Les perturbations consistent à générer des variantes d'une image en activant ou désactivant aléatoirement des superpixels. Cela permet d'évaluer l'influence de chaque région sur la prédiction.

Une matrice de perturbation aléatoire (binaire) est générée :

- Chaque ligne de la matrice correspond à une version perturbée de l'image, où certains superpixels sont activés (valeur 1) et d'autres désactivés (valeur 0).
- Les images perturbées sont redimensionnées et leurs caractéristiques (dépendamment de l'extracteur choisi) sont extraites avant d'être passées au modèle.

3. Normalisation : Pondérer les contributions locales

La normalisation joue un rôle essentiel pour relier les perturbations au comportement global du modèle de manière significative. Elle intervient principalement dans deux étapes :

(a) Calcul des distances :

Les distances entre l'image originale et les perturbations sont calculées à l'aide de la distance cosinus. Cela permet d'évaluer la similitude entre les différentes versions de l'image :

```
distances = sklearn.metrics.pairwise_distances(
            perturbations, original_image, metric="cosine").ravel()
```

(b) Application d'une fonction noyau :

Une fonction gaussienne est utilisée pour transformer les distances en poids compris entre 0 et 1. Ces poids reflètent l'importance relative de chaque perturbation dans l'explication locale :

```
weights = np.sqrt(np.exp(-(distances**2) / kernel_width**2))
```

4. Régression Linéaire : Générer des Explications Interprétables

Une fois les images perturbées et leurs prédictions obtenues, un modèle de régression linéaire est ajusté pour relier les activations des superpixels à la prédiction. Les coefficients de ce modèle indiquent l'importance de chaque superpixel pour la prédiction. Les superpixels ayant les coefficients les plus élevés sont considérés comme critiques pour la prédiction.

Au final, un affichage des régions de l'image les plus importantes pour le classifier est effectué. Les résultats incluent :

- Les images perturbées générées au cours des 10 répétitions.
- Une moyenne des 10 images perturbées pour chaque image d'entrée.
- Les superpixels les plus influents sur la décision du modèle.

Les images perturbées et leur moyenne sont visualisées pour une meilleure compréhension des zones critiques.

Exemple pour une image sélectionnée Figure 1 et Figure 2:

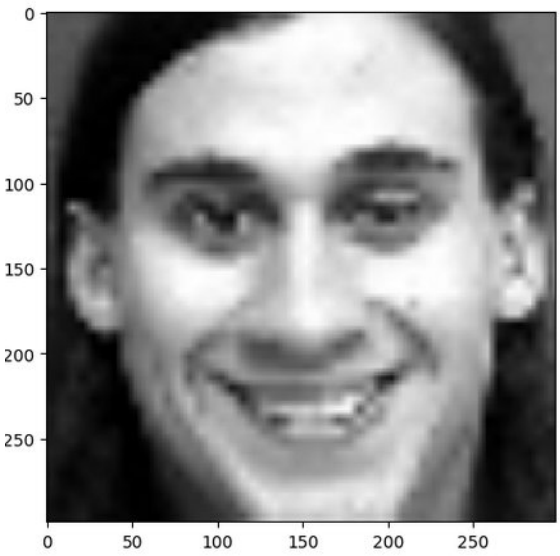


Figure 1: Image avant Lime

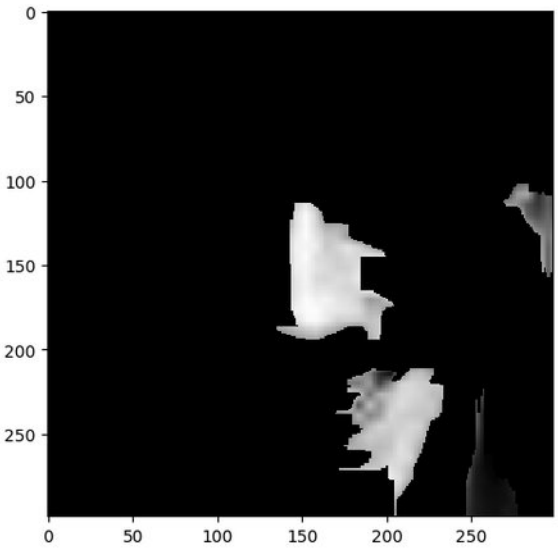


Figure 2: Régions importantes selon Lime pour le classifier donné

Résultats

Classification

Taux de classification obtenus

Les taux de classification suivant les deux types de modèles avec ou sans augmentation sont résumés dans les tableaux suivants.

Résultats pour le modèle SVM avec et sans augmentation :

	Sans augmentation			Avec augmentation		
Extracteurs /kernel	Polynomial	RBF	Linear	Polynomial	RBF	Linear
Pixels sans normalisation	81 %	75.8%	83%	79 %	81 %	83 %
Pixels avec normalisation	67%	73%	83%	74%	78%	75%
HOG	67%	80%	**84.95%**	76%	**88.17%**	83%
VGG	63 %	66 %	78 %			
MobileNet	63,97 %	67,2%	76,88 %	62 %	70 %	78 %

Tableau 1: Résultats pour le modèle SVM avec et sans augmentation de données

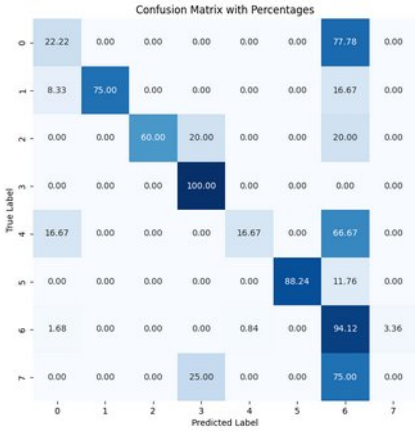
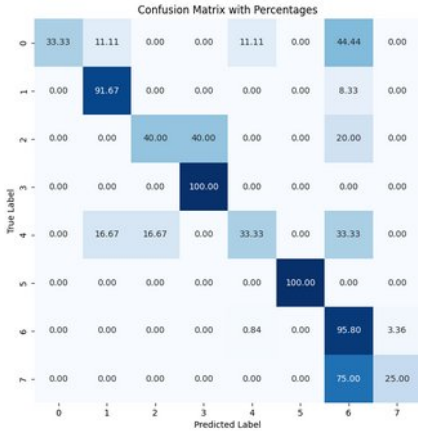
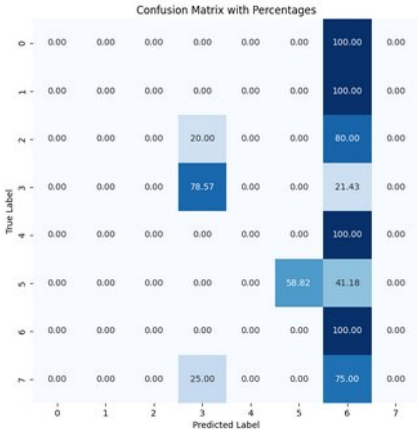
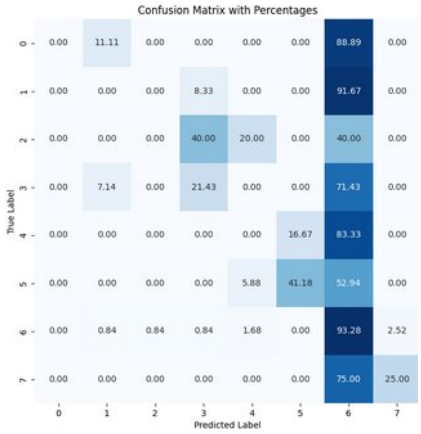
Résultats pour le modèle KNN avec et sans augmentation :

	Sans augmentation	Avec augmentation
Pixels sans normalisation	59.67 %, k=8 voisins	61.8 %, k=1 voisin
Pixels avec normalisation	63 %, k= 6 voisins	58,6 %, k=1 voisin
HOG	**75 %, k=13 voisins**	**66 %, k=1 voisin**
VGG	65 %, k= 5 voisins	
MobileNet	64 %, k=6 voisins	

Tableau 2: Résultats pour le modèle KNN avec et sans augmentation de données

Matrices de confusion

Les matrices de confusion des meilleurs modèles (scores mis en gras) sont résumés dans le tableau suivant.

Modèle / Augmentation	Sans augmentation	Avec augmentation
SVM, extracteur = HOG	 *Kernel = linear, 84.95 % d'accuracy*	 *Kernel = rbf, 88 % d'accuracy*
KNN, extracteur = HOG	 75 % accuracy, k=13 voisins	 66 % d'accuracy, k=1 voisin

Parmi les meilleurs résultats obtenus, la méthode SVM se distingue par une meilleure qualité de classification par rapport au KNN. Bien que le KNN affiche une précision relativement élevée, il a une forte tendance à classer les images dans la classe 6, qui correspond à «neutre». Cela suggère que le KNN éprouve des difficultés à distinguer correctement les autres classes, malgré une accuracy globale plutôt bonne. En revanche, l'approche SVM, avec un score proche de 90 %, montre une classification plus précise, illustrée par une diagonale bien définie dans la matrice de confusion. Toutefois, des erreurs persistent, notamment avec la classe 6 «neutre». Cela peut s'expliquer par le fait que, lors de l'entraînement, la classe 6 est la classe majoritaire, et bien que des techniques d'augmentation de données aient été appliquées pour équilibrer les classes, celles-ci ne génèrent que des transformations des images existantes et non de nouvelles images. Par conséquent, le classifieur reste confronté aux mêmes échantillons pendant l'entraînement, ce qui n'est pas suffisant pour améliorer la classification des images autre que la classe 6.

Explicabilité

Afin de pouvoir expliquer ce que font nos algorithmes de classification, nous avons utilisé Lime pour expliquer localement chaque première image bien classée de chaque classe/émotion.

Chaque première image bien classée est passée 10 fois à l'algorithme de Lime afin d'avoir une moyenne des prédictions et par conséquent d'avoir une moyenne des explications locales de chaque image par le classifieur.

Sur chaque figure suivante sont montrées les parties utilisées par le classifieur correspondant pour bien classer l'image dans sa catégorie.

SVM avec HOG, kernel linear sans augmentation de données – 84.95 % d'accuracy

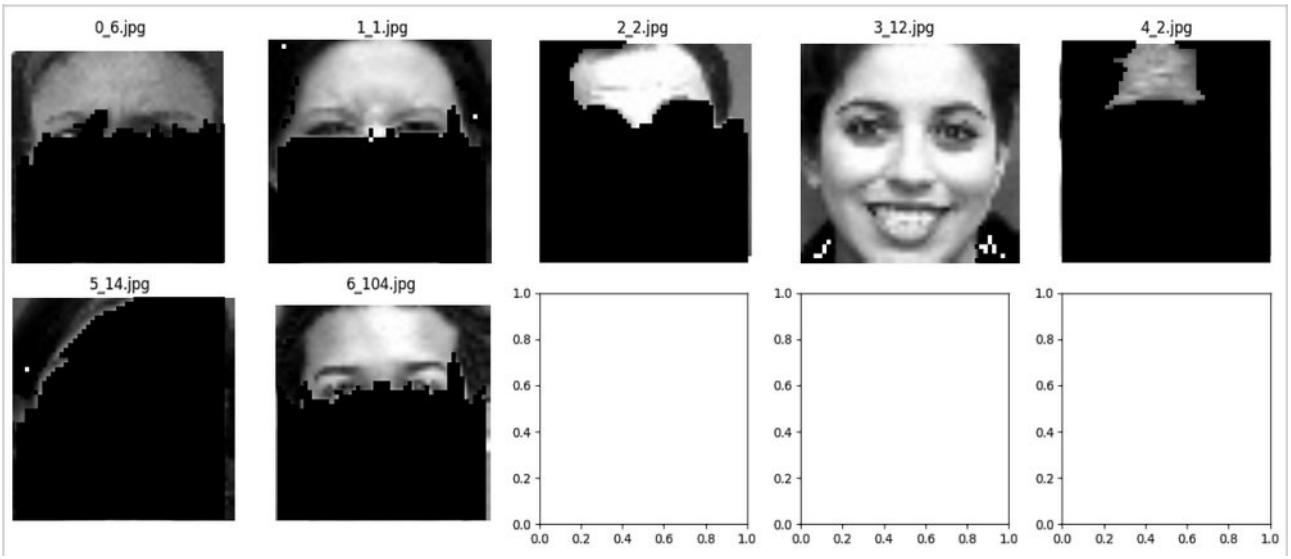


Figure 3: Résultats de l'algorithme Lime sur le modèle SVM, HOG kernel linear sans augmentation de données

SVM avec HOG, kernel rbf avec augmentation de données – 88 % d'accuracy

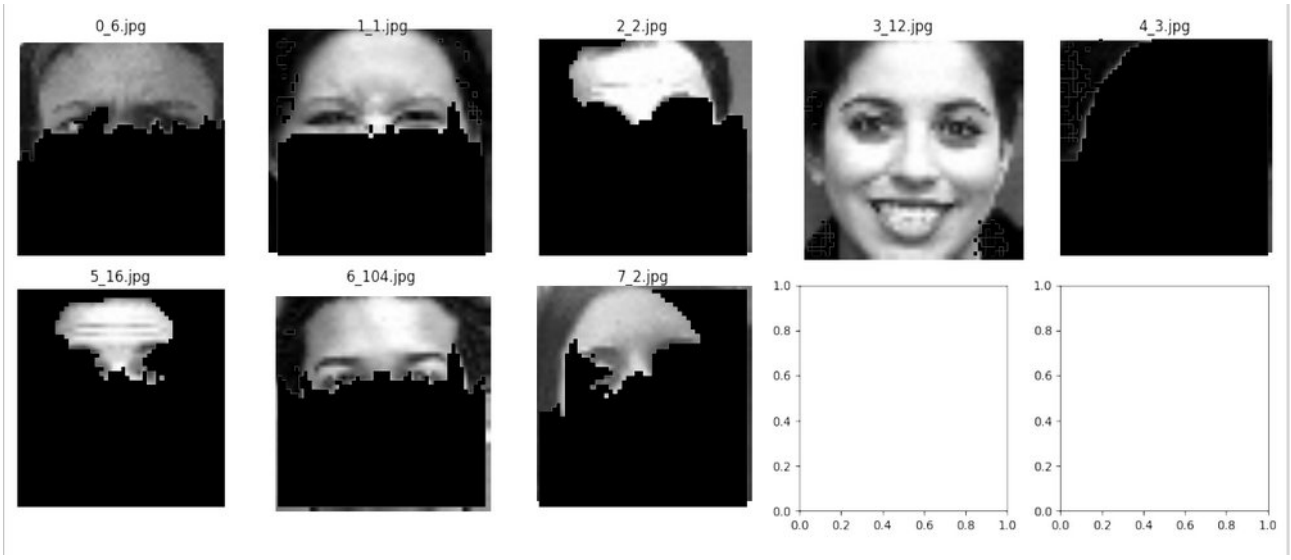


Figure 4: Résultats de Lime sur le modèle SVM avec HOG, kernel rbf avec augmentation de données

KNN avec HOG sans augmentation de données – 75 % d’accuracy

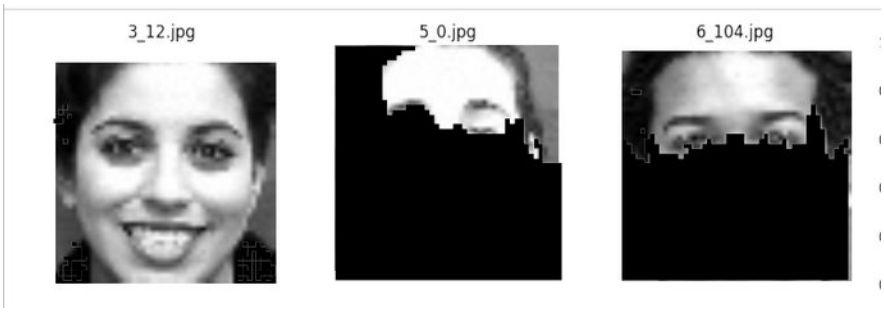


Figure 5: Résultat de Lime sur le modèle KNN avec HOG sans augmentation de données

KNN avec HOG et augmentation de données – 66 % d’accuracy

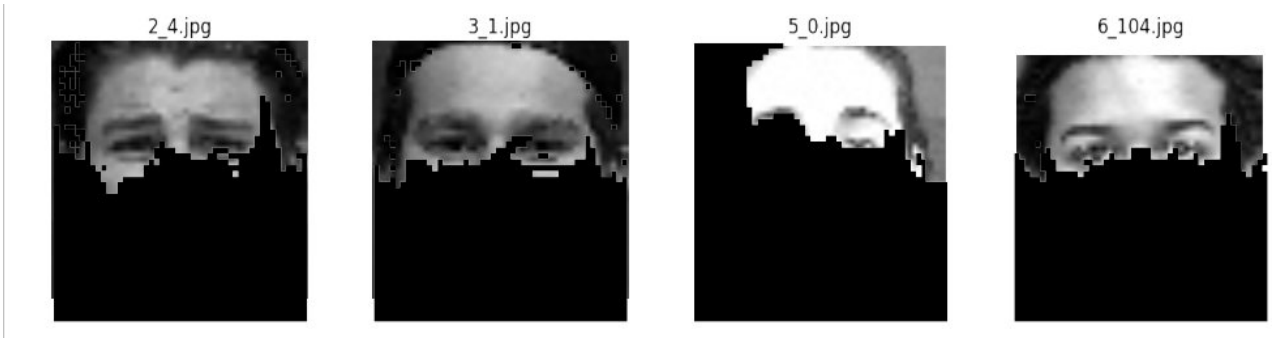


Figure 6: Résultat de Lime sur le modèle KNN avec HOG et augmentation de données

Nous pouvons remarquer que dans la plupart des cas, le haut du visage est l’endroit le plus utilisé par le classifieur pour classifier les émotions. Cela est plutôt cohérent puisque le haut du visage est porteur de beaucoup d’informations. En effet, la région située au-dessus des yeux, incluant le front et les sourcils, joue un rôle crucial dans la communication des émotions. Les mouvements des sourcils, par exemple, sont très révélateurs d’états émotionnels comme la surprise, la colère ou la peur. Le front et les sourcils sont en effet des zones du visage qui, par leurs mouvements, peuvent exprimer une large gamme d’émotions, souvent de manière très rapide et expressive. Néanmoins, la région autour de la bouche apporte également beaucoup d’informations mais dans ces tests, les classifieurs ne semblent pas en tenir compte.

Conclusion

Ce rapport a permis d'étudier l'enjeu de la classification des émotions au moyen d'algorithmes de machine learning avec un accent particulier sur l'explicabilité des choix opérés par ces modèles. Dans un premier temps, un état de l'art sur l'interprétabilité des classifieurs a été présenté et plus spécifiquement l'état des techniques LIME et SHAP. Il en ressort que comprendre le fonctionnement interne des modèles de classification est primordial pour en assurer la fiabilité et l'acceptabilité sociétale, notamment dans le cadre d'applications sensorielles à risques comme la reconnaissance des émotions humaines.

L'analyse des modèles a permis de sélectionner le Support Vector Machine (SVM) et le K-Nearest Neighbors (KNN) et de comparer les performances et l'interprétabilité. Le SVM autorise un bon niveau de classification, mais reste encore déficitaire quand il s'agit de traiter des classes dominantes comme la classe « neutre. » Le KNN montre avec du recul des résultats intéressants tant sur le plan de la simplicité que sur celui de la précision dans certains cas, hélas la tendance à plus souvent classifier comme neutre a montré ses limites dans des contextes dans lesquels nous souhaitons parfois la nuance de l'émotion.

Pour l'explicabilité LIME a été plus convaincante puisqu'elle a vraiment permis de visualiser les régions de l'image identifiées comme clés dans le choix de classification des émotions. Dans certains contextes elle est perfectible notamment par l'adjonction de l'approche de SHAP qui, du fait de la finesse de ses valeurs de Shapley, pourrait mieux décrire les contributions de chacune des caractéristiques à la classification de l'image, ce qui serait certainement favorable à la transparence du processus de décision.

Enfin, concernant la question de choix du modèle, si SVM et KNN semblent être de bons candidats, le recours à des profondeurs comme celle des réseaux de neurones complexes tels que les CNN semble prometteuse pour une modalité de classification d'émotions plus précise et plus robuste. Les CNN en raison de leur aptitude à extraire automatiquement des caractéristiques pertinentes et à manipuler des données plus complexes que d'autres, comme l'image, devraient donner de meilleurs résultats que les méthodes classiques, particulièrement lorsqu'ils sont associés à des techniques de traitement avancées notamment d'augmentation de données et d'explicabilité, comme SHAP pour son interprétabilité.

D'ores et déjà, l'utilisation simultanée de méthodes avancées comme SHAP et de modèles plus puissants comme les CNN ouvre des perspectives intéressantes pour rendre d'une part le système de classification des émotions plus performant et d'autre part, l'interprétation de la décision des modèles d'émotions plus visibles pour les utilisateurs finaux. L'exploration plus en avant de ces modèles n'est donc pas sans intérêt pour à la fois augmenter la précision des prédictions, mais également faire en sorte que la décision des modèles soit plus compréhensible et explicable.

Références

1: Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin, "Why Should I Trust You?": Explaining the Predictions of Any Classifier, 2016, https://doi.org/10.1145/2939672.2939778

2: Lundberg, Scott M and Lee, Su-In, Advances in Neural Information Processing Systems. A Unified Approach to Interpreting Model Predictions, 2017, https://proceedings.neurips.cc/paper_files/paper/2017/file/8a20a8621978632d76c43dfd28b67767-Paper.pdf%20

3: Khare, S. K., Blanes-Vidal, V., Nadimi, E. S., & Acharya, U. R. , Emotion recognition and artificial intelligence: A systematic review (2014–2023) and research recommendations, 2024, https://doi.org/10.1016/j.inffus.2023.102019