



Rapport

Projet Deep Learning

Reconnaissance d'émotions sur un visage humain

Amoros Louis - Barraud Nathan - Dabrowski Rémi - Gaud Nathan - Guidez Martin

Sommaire

- I. Introduction**
- II. Base de Données**
 - 1. Acquérir, annoter et partitionner les données
 - 2. Utiliser notre base de données
- III. Pronostics et résultats attendus**
- IV. Mise en oeuvre**
 - 1. Modèle du réseau de neurones convolutif
 - 2. Pré-traitement du dataset
 - 3. Entraînement du modèle
 - 4. Augmentation de données
 - 5. Transfer Learning
 - 6. Fine-tuning
- V. Analyse des résultats**
 - 1. Protocole d'analyse
 - 2. Modèle de base
 - 3. Ajout de l'augmentation de données
 - 4. Ajout du transfer learning
 - 5. Ajout du fine-tuning
- VI. Conclusion**

I. Introduction

Notre projet de reconnaissance d'émotions humaines à l'aide de l'apprentissage profond vise à développer un modèle permettant de reconnaître cinq émotions différentes à partir de photos de visages humains.

Nous avons choisi d'étudier les émotions suivantes:

- Peur
- Tristesse
- Joie
- Colère
- Surprise



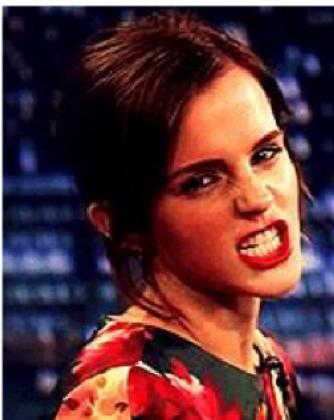
(a) Peur



(b) Tristesse



(c) Joie



(d) Colère



(e) Surprise

Figure 1: Emotions étudiées au cours du projet

Le but de cette étude est d'élaborer un modèle d'apprentissage profond le plus performant possible en utilisant différentes techniques notamment celles vues en TPs.

II. Base de données

1. Acquérir, annoter et partitionner les données

Afin de constituer notre base de données d'image, nous avons utilisé différents sites de banque d'images tels que Getty Images, ShutterStock et Google Images. Nous avons sélectionné des images avec un visage plutôt visible et une émotion clairement identifiable (le moins d'ambiguïté possible) que nous avons sauvegardées en centrant le visage sur l'image. Chaque membre du groupe s'est vu attribuer une émotion qu'il devait compléter.

Les images sont stockées sur un dépôt GitHub dans des dossiers séparés afin de déterminer facilement la classe de l'image. Vous pourrez observer notre base de données sous le lien suivant :

https://github.com/Learza7/deep_learning_project/tree/main/emotion_images

Pour chaque classe (une émotion = une classe), nous avons recueilli environ 200 images que nous avons réparties dans différents sous-dossiers pour chaque étape du projet en proportions établies: 80% pour l'entraînement, 10% pour la validation et 10% pour la phase de test.

2. Utiliser notre base de données

Pour pouvoir charger et utiliser les images de notre base de données, nous avons créé un script de chargement en nous inspirant du script donné. Les images sont associées au label correspondant au dossier depuis lequel elles sont chargées. Notre fonction de chargement retourne deux listes : la première correspondant aux images et la seconde à leur label.

III. Pronostics et résultats attendus

La reconnaissance de cinq émotions à partir d'images représente un défi complexe. Cependant, avec une base de données d'un peu plus de 1000 images de bonne qualité et l'utilisation d'un réseau de neurones convolutif, nous pouvons espérer obtenir des résultats satisfaisants.

Pour optimiser les performances de notre modèle, nous allons exploiter des techniques telles que le "transfer learning" et le "fine-tuning". Cette approche consiste à utiliser un réseau de neurones pré-entraîné sur une tâche similaire, puis à l'adapter à notre problème spécifique de reconnaissance des émotions. En tirant parti des connaissances déjà acquises par le réseau pré-entraîné, nous pourrons accélérer l'apprentissage et améliorer la précision de notre modèle.

De plus, nous allons appliquer des techniques de régularisation pour limiter le surapprentissage. L'une de ces techniques est l'augmentation de la base de données, où nous allons générer de nouvelles images à partir des images existantes en effectuant des transformations telles que des rotations, des zooms ou des translations. Cela permettra d'augmenter la diversité des exemples d'entraînement et d'améliorer la généralisation du modèle.

Nous visons un résultat d'au moins 80% de précision dans la reconnaissance des émotions. Cependant, il est important de noter que les performances du modèle dépendront de plusieurs facteurs. Tout d'abord, la qualité de la base de données utilisée. Des images de bonne qualité et représentatives des différentes émotions sont essentielles pour obtenir de bons résultats. De plus, la complexité de l'architecture du réseau de neurones et les hyperparamètres choisis, tels que le taux d'apprentissage et la taille du lot, auront un impact sur les performances. Enfin, le temps d'entraînement alloué au modèle jouera également un rôle dans sa capacité à apprendre et à généraliser correctement.

IV. Mise en oeuvre

Vous trouverez toute la mise en oeuvre de notre projet sur le document Google Colab suivant :

<https://colab.research.google.com/drive/1XVSWpFjXtUIIfTNIcrgTaLtLFCUhX5rb?usp=sharing>

1. Modèle du réseau de neurones convolutif

Pour concevoir notre réseau de neurones, nous sommes partis de ce qui avait été fait en TP, puis nous l'avons adapté et étendu afin de pouvoir répondre à notre problématique plus complexe.

Notre réseau de neurones est construit à l'aide de la bibliothèque Keras et est composé d'une répétition de plusieurs couches.

- Une couche de convolution chargée d'apprendre les caractéristiques locales des images. Elle utilise la fonction d'activation Rectified Linear Unit (ReLU). Cette fonction d'activation est largement utilisée car elle permet d'atténuer le problème de la disparition des gradients, qui se produit lorsque les gradients sont rétropropagés dans le réseau et deviennent très faibles.
- Une couche de pooling réduisant l'échelle de l'entrée dans ses dimensions spatiales (largeur et hauteur) en prenant la valeur maximale sur une fenêtre d'entrée (une fenêtre de 2x2 dans notre cas) pour chaque canal de l'entrée. Cela réduit la complexité de calcul pour les couches suivantes et fournit également une forme d'invariance de traduction dans les caractéristiques apprises.
- Une couche de normalisation permettant d'accélérer le processus de formation en normalisant les entrées de chaque couche, afin d'obtenir une distribution stable des valeurs d'activation tout au long de la formation. Elle a également un léger effet de régularisation.

Ce schéma est répété 4 fois avec des nombres de filtres croissants pour la couche de convolution (32, 64, 128, 256). Chaque filtre de ces couches apprendra à reconnaître une caractéristique différente des données d'entrée.

On utilise ensuite une couche flatten aplatisant le tenseur d'entrée en un tenseur (vecteur) unidimensionnel, afin qu'il puisse être introduit dans les couches denses. La première couche dense se compose de 128 neurones afin de capturer les relations non linéaires les plus complexes entre les caractéristiques. La deuxième et dernière couche est une couche dense avec

5 neurones, correspondant aux 5 émotions à reconnaître. Elle utilise la fonction d'activation ‘softmax’ pour prédire les probabilités d'appartenance à chaque classe.

Ce modèle est compilé avec l'optimiseur Adam avec un taux d'apprentissage de 1e-4, une fonction de perte de catégorisation croisée entière qui convient aux tâches de classification multi-classes où les classes sont mutuellement exclusives, et une métrique de précision catégorielle éparse, qui correspond à la fonction de perte choisie et la mieux adaptée à notre classification.

2. Pré-traitement du dataset

Tout d'abord, nous chargeons les images (associées aux labels) pour l'entraînement et la validation. Nous leur appliquons une procédure de traitement afin de fixer la taille standard à 128 par 128 pixels, taille que nous avons estimée suffisante pour contenir assez d'informations permettant l'identification de l'émotion et nécessaire pour accélérer le traitement des images dans le réseau de neurones. De plus, les valeurs des pixels sont normalisées (entre 0 et 1) et chaque label (chaîne de caractères) est associé à un entier entre 0 et 4 (5 classes au total).

3. Entraînement du modèle

On entraîne le modèle avec différents nombre d'époques afin de tester le fonctionnement du modèle. Comme l'entraînement est instable, on déclenche une sauvegarde du modèle dans notre répertoire Google Drive à chaque fois que la perte de validation atteint un nouveau minimum pour le réutiliser ultérieurement sans devoir reprendre l'entraînement à zéro.

4. Augmentation de données

Dans le but d'améliorer les performances de notre modèle et de le rendre plus robuste, nous avons mis en œuvre une technique d'augmentation de données (data augmentation).

Celle-ci consiste à générer de nouvelles images à partir des images existantes en appliquant des transformations aléatoires telles que des rotations, des zooms, des translations, des retournements horizontaux, etc...

L'objectif de l'augmentation de données est d'augmenter la diversité des exemples d'entraînement, ce qui aide le modèle à mieux généraliser et à être plus résistant aux variations mineures dans les images. Pour notre étude, nous avons appliqués différentes transformations à nos données :

- rotation (20°)
- déplacement horizontal (30%)
- déplacement vertical (30%)
- cisaillement (30%)
- zoom (30%)
- symétrie horizontale
- blanchiment
- variation de la saturation (20%)

Nous avons choisi ces paramètres d'augmentation de données de manière à créer une diversité maximale tout en préservant la cohérence avec le reste des images et le contexte de notre étude. Par exemple, nous avons évité les rotations excessives qui pourraient retourner les visages ou les déplacements qui feraient sortir les visages du cadre.



En visualisant les images générées par l'augmentation de données, nous avons pu constater que les émotions restaient reconnaissables et cohérentes avec nos attentes. Les transformations appliquées n'ont pas altéré la signification des expressions faciales, mais ont plutôt ajouté de la variabilité aux images d'entraînement. Cela a contribué à améliorer la capacité de notre modèle à généraliser et à reconnaître efficacement les différentes émotions dans de nouvelles images.

5. Transfer Learning

Le Transfer Learning, ou apprentissage par transfert, est une technique qui consiste à utiliser un modèle de référence pré-entraîné sur une tâche similaire, puis à l'adapter à notre problème spécifique. Son objectif est de capitaliser sur les connaissances déjà acquises par le modèle de référence pour accélérer l'apprentissage et améliorer les performances du modèle dans notre tâche de reconnaissance d'émotions humaines.

Nous avons utilisé un modèle de référence largement utilisé et performant, à savoir le modèle VGG16. C'est un réseau de neurones convolutif profond qui a été pré-entraîné sur l'ensemble de données ImageNet, qui comprend des millions d'images provenant de différentes catégories. Ce modèle a été choisi pour sa capacité à extraire des caractéristiques complexes à partir d'images et pour sa généralisation élevée.

Pour l'implémentation du Transfer Learning avec VGG16, nous avons gelé les poids des couches pré-entraînées, afin de conserver les connaissances qu'il a déjà acquises. Seules les couches supérieures du modèle ont été réinitialisées et adaptées à notre problème spécifique de reconnaissance des émotions.

Cela nous a permis de profiter des représentations de haut niveau apprises par VGG16 et d'éviter un apprentissage de zéro qui aurait nécessité un volume de données et de ressources beaucoup plus important. En utilisant le modèle VGG16 comme point de départ, nous avons pu obtenir des performances améliorées et une convergence plus rapide de notre modèle.

6. Fine-tuning

Le Fine-tuning, ou affinage, est une technique qui consiste à ajuster les poids des couches pré-entraînées d'un modèle déjà entraîné, en plus d'adapter les couches supérieures, pour améliorer les performances sur une tâche spécifique. L'objectif du Fine-tuning est d'optimiser davantage le modèle en le spécialisant pour notre problème de reconnaissance d'émotions humaines.

Pour appliquer le Fine-tuning, nous avons de nouveau utilisé le modèle VGG16 pré-entraîné, dont les couches supérieures ont déjà été adaptées à notre tâche. Au lieu de geler complètement les poids des couches pré-entraînées, nous avons décidé de les laisser être ajustés pendant l'entraînement avec un taux d'apprentissage très faible.

En affinant les poids des couches pré-entraînées, nous avons permis au modèle de mieux s'adapter aux caractéristiques spécifiques de notre problème de reconnaissance d'émotions. Cela lui a également permis de capturer des informations plus précises et discriminantes pour différencier les émotions dans les images.

V. Analyse des résultats

1. Protocole d'analyse

Plusieurs outils nous permettent d'étudier les résultats obtenus. Pour ce qui est de l'analyse des performances globales, nous avons utilisé les deux indicateurs suivants:

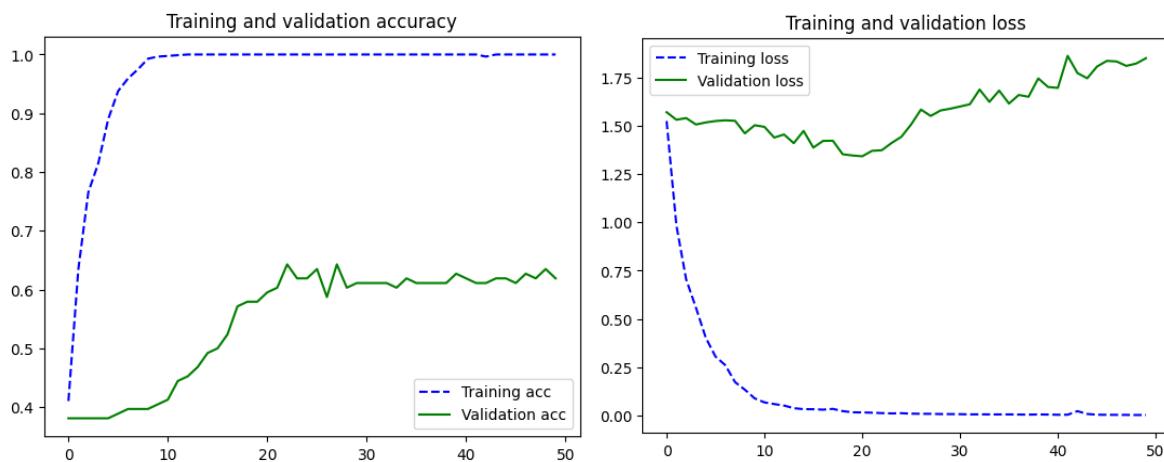
- **La précision du modèle (accuracy)** : Il est le nombre de prédictions où la valeur prédite est égale à la valeur réelle. Cette donnée est exprimée en pourcentage, et on l'analyse généralement à l'aide de graphes représentant la précision en fonction des époques. Le but est d'avoir une précision la plus élevée possible, ce qui traduit le fait que le modèle est capable de prédire correctement la plupart des échantillons. Cependant, il est important de prendre en compte le contexte du problème de classification et de considérer d'autres métriques pour une évaluation plus complète.
- **La perte, ou fonction de coût (loss)** : Elle permet de donner une vision plus nuancée du modèle que la précision. En effet la perte prend en compte l'incertitude d'une prédiction en fonction de l'écart entre la prédiction et la valeur réelle. Une perte élevée indique que le modèle fait des prédictions erronées avec une grande différence par rapport aux étiquettes réelles. Une perte faible, en revanche, indique que le modèle est capable de faire des prédictions plus précises. On cherche donc à minimiser la perte, tout en surveillant sur les ensembles de validation et de test pour éviter le surapprentissage.

Enfin, nous avons utilisé des matrices de confusion afin d'évaluer les performances de nos modèles, en comparant les prédictions avec les véritables étiquettes de classe. Chaque ligne de la matrice correspond à une classe réelle, tandis que chaque colonne représente une classe prédite par le modèle. L'objectif principal en analysant la matrice de confusion est de minimiser les erreurs de classification. On cherche à obtenir un nombre élevé d'échantillons correctement prédits comme positifs ou négatifs, et un nombre faible d'échantillons réels positifs qui ont été incorrectement prédits comme négatifs (et inversement). Une matrice de confusion idéale serait une matrice diagonale, avec tous les échantillons correctement classés.

2. Modèle

Pour commencer, nous avons fait un premier entraînement du modèle sur nos images dédiées à l'entraînement. Ce dernier n'a été réalisé que sur 50 epochs car nous avons déterminé qu'il n'était pas forcément nécessaire de faire durer l'entraînement plus longtemps (la précision et la perte tendant respectivement vers 1 et 0).

On peut voir d'après les graphes suivants que les performances obtenues ne sont pas satisfaisantes (60% de précision sur la validation).



Le premier graphe représente l'évolution de la précision en fonction des époques d'entraînement pour le set de données d'entraînement et de validation. Le second graphe représente l'évolution de la perte sur ces deux mêmes sets.

On peut voir qu'il y a du sur-apprentissage. En effet, le premier élément nous l'indiquant est le fait que la précision est très bonne pour les données d'entraînement, mais elle se dégrade significativement sur les données de validation. De plus, on peut voir qu'il y a un écart extrêmement important entre la perte d'entraînement et celle de validation. Ce constat est appuyé par les résultats finaux sont les suivant:

```
27/27 - 0s - loss: 0.0016 - sparse_categorical_accuracy: 1.0000 - 247ms/epoch - 9ms/step
Training accuracy: 100.00%
Training loss: 0.16%
4/4 - 0s - loss: 1.8507 - sparse_categorical_accuracy: 0.6190 - 57ms/epoch - 14ms/step
Evaluation accuracy: 61.90%
Evaluation loss: 185.07%
```

La matrice de confusion obtenue est la suivante:



On peut la lire comme suit :

- La première ligne indique les prédictions pour la classe "Angry". On observe que 88% des échantillons réellement "Angry" sont correctement prédits comme "Angry", tandis que 27% des échantillons "Angry" sont inexactement prédits comme "Sad", 4,8% comme "Fearful", 40% comme "Happy" et 17% comme "Surprised".
- La deuxième ligne concerne les prédictions pour la classe "Sad". Ici, on constate que 62% des échantillons "Sad" sont correctement prédits comme "Sad", 10% sont inexactement prédits comme "Angry", et ainsi de suite.
- Les lignes suivantes suivent le même schéma pour les autres classes : "Fearful", "Happy" et "Surprised".

De manière général, une bonne matrice de confusion devrait avoir des valeurs élevées sur la diagonale principale, ce qui indiquerait des prédictions correctes. Cependant, dans cette matrice, nous pouvons voir que certaines classes ont des valeurs élevées en dehors de la diagonale principale, ce qui signifie qu'il y a des confusions entre ces classes. Par exemple, le réseau confond souvent les émotions "Angry" et "Happy" (40% des échantillons "Angry" sont prédits comme "Happy"), ainsi que les émotions "Fearful" et "Surprised" (20% des échantillons "Fearful" sont prédits comme "Surprised").

En résumé, cette matrice de confusion suggère que le modèle a des difficultés à distinguer certaines émotions, ce qui peut indiquer des problèmes de performance ou de représentation des données.

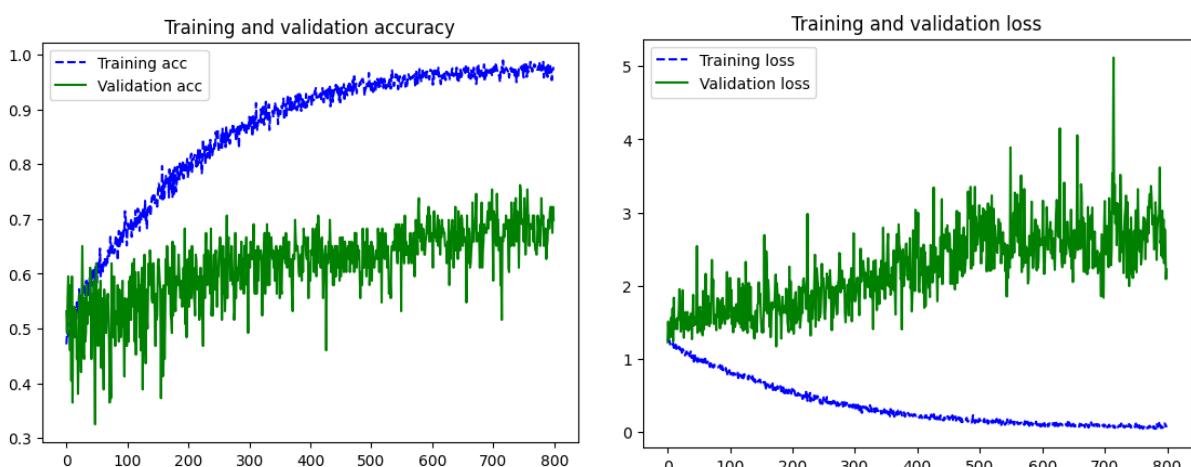
Voilà quelque exemples de mauvaise prédictions:



On peut voir que certaines émotions qui sont plutôt éloignées sont mal interprétées. Les performances des prédictions ne sont pas satisfaisantes.

3. Ajout de l'augmentation de données

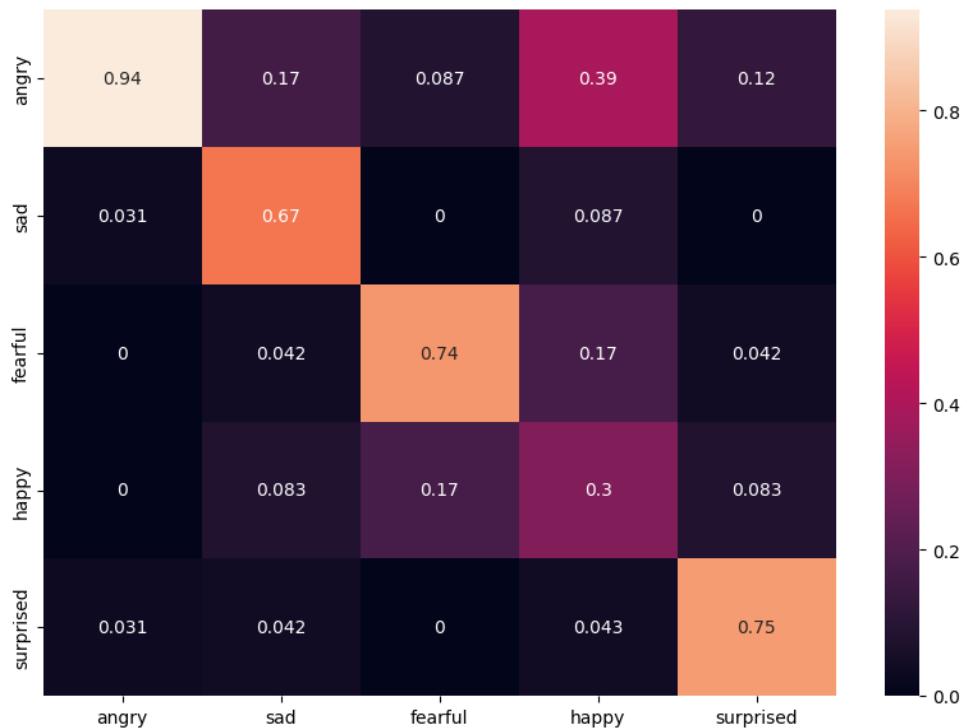
Afin d'améliorer nos résultats, et d'essayer d'éviter le sur apprentissage, nous avons ensuite ajouté de l'augmentation de données. Après un deuxième entraînement, cette fois-ci plus long, nous avons obtenu les résultats suivants, qui n'étaient pas encore tout à fait satisfaisants (70% de précision) :



On peut voir que malgré l'augmentation de données, il y a toujours du sur-apprentissage. En effet, comme pour l'entraînement précédent, la précision est bonne pour les données d'entraînement, mais elle se dégrade significativement sur les données de validation. On retrouve également un écart important entre la perte d'entraînement et de validation. Cependant, on peut voir que ces phénomènes sont bien moins marqués que pour l'entraînement précédent. De plus, les performances du modèle se sont améliorées :

```
27/27 - 0s - loss: 0.0058 - sparse_categorical_accuracy: 0.9988 - 243ms/epoch - 9ms/step
Training accuracy: 99.88%
Training loss: 0.58%
4/4 - 0s - loss: 2.2197 - sparse_categorical_accuracy: 0.6984 - 53ms/epoch - 13ms/step
Evaluation accuracy: 69.84%
Evaluation loss: 221.97%
```

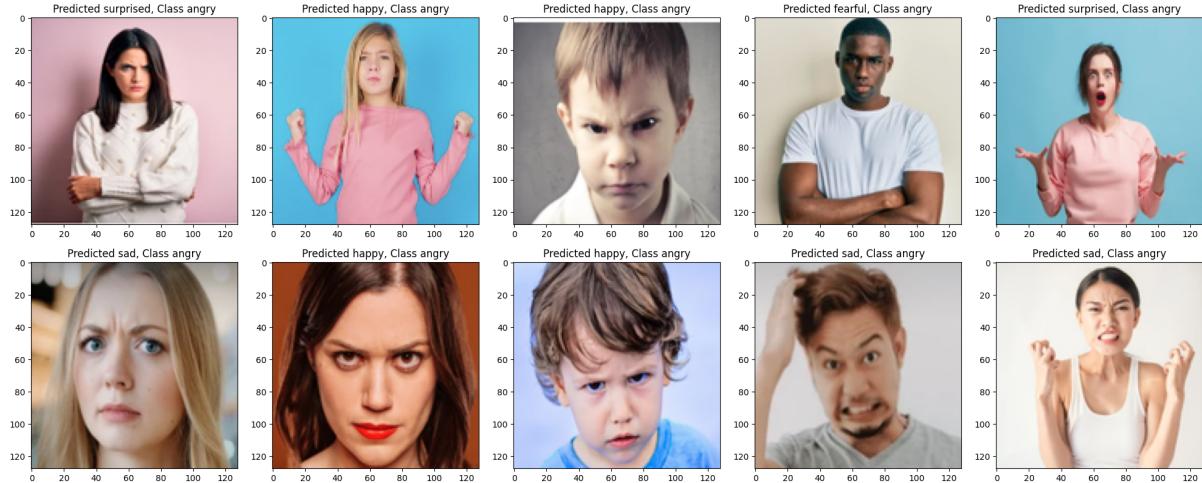
La matrice de confusion obtenue est la suivante :



Les valeurs de la diagonale sont plus élevées, et globalement on est plus centré sur cette dernière. Cependant, on voit que les prédictions confondent encore certaines émotions, notamment assez souvent "angry" et "happy" (39% des échantillons "angry" sont prédits comme "happy").

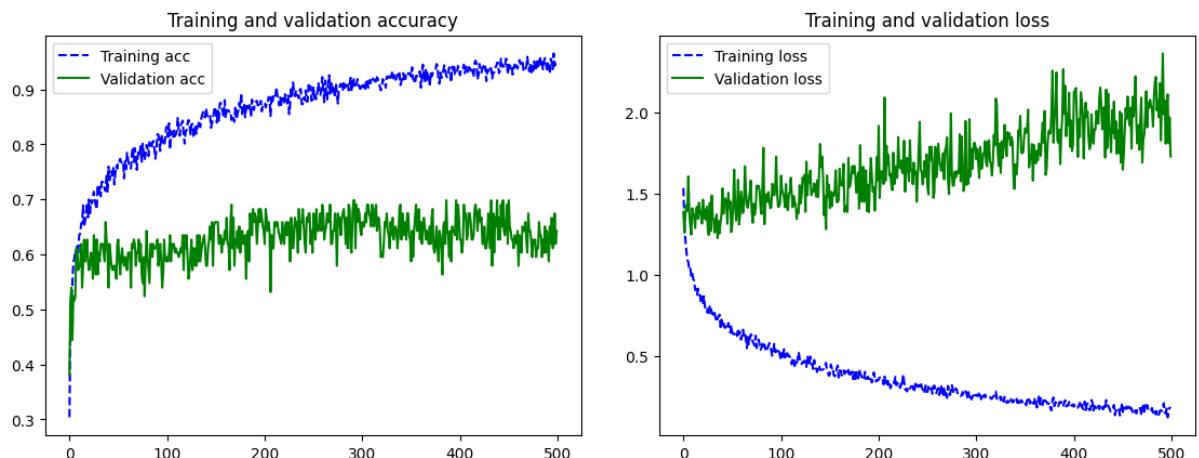
On peut voir sur les exemples suivants de mauvaises prédictions que "angry" semble effectivement être souvent confondu avec "happy". De plus, on peut conjecturer que certaines émotions sont trop proches. Un exemple peut être les sourcils froncés : c'est un trait du visage partagé par plusieurs

émotions (surprised, angry, sad). On peut supposer que ce trait commun explique certaines des erreurs présentées ci-dessous.



4. Ajout du transfer learning

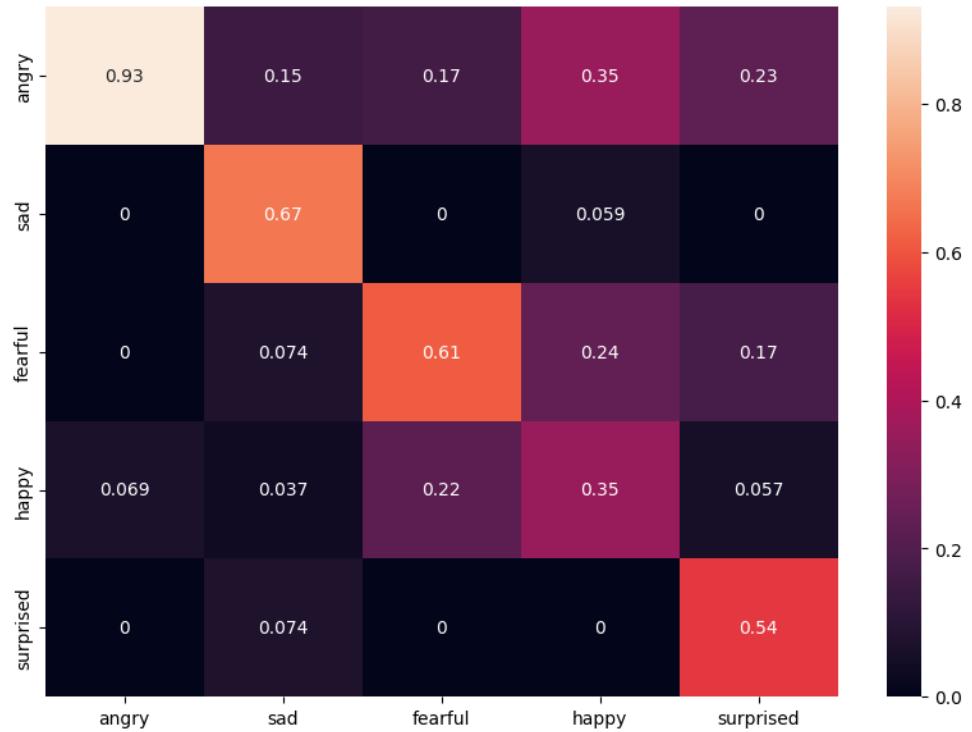
Toujours dans une optique d'améliorer nos résultats, nous avons utilisé le transfer learning (comme décrit plus haut). Après l'entraînement effectué, nous avons obtenu des performances moins satisfaisantes que précédemment (65 % de précision) :



On retrouve les mêmes allures que lors des entraînements précédents, traduisant un certain sur-apprentissage. On peut voir que les performances générales des prédictions se sont dégradées:

```
27/27 - 2s - loss: 0.0312 - sparse_categorical_accuracy: 0.9916 - 2s/epoch - 62ms/step
Training accuracy: 99.16%
Training loss: 3.12%
4/4 - 0s - loss: 1.7266 - sparse_categorical_accuracy: 0.6429 - 172ms/epoch - 43ms/step
Evaluation accuracy: 64.29%
Evaluation loss: 172.66%
```

La matrice de confusion obtenue est la suivante :



Les valeurs de la diagonale sont plutôt moins élevées, et globalement on est moins centré sur cette dernière. La probabilité de prédire “surprised” lors d'un vrai “surprised” a également beaucoup baissé (en passant de 75% à 54%).

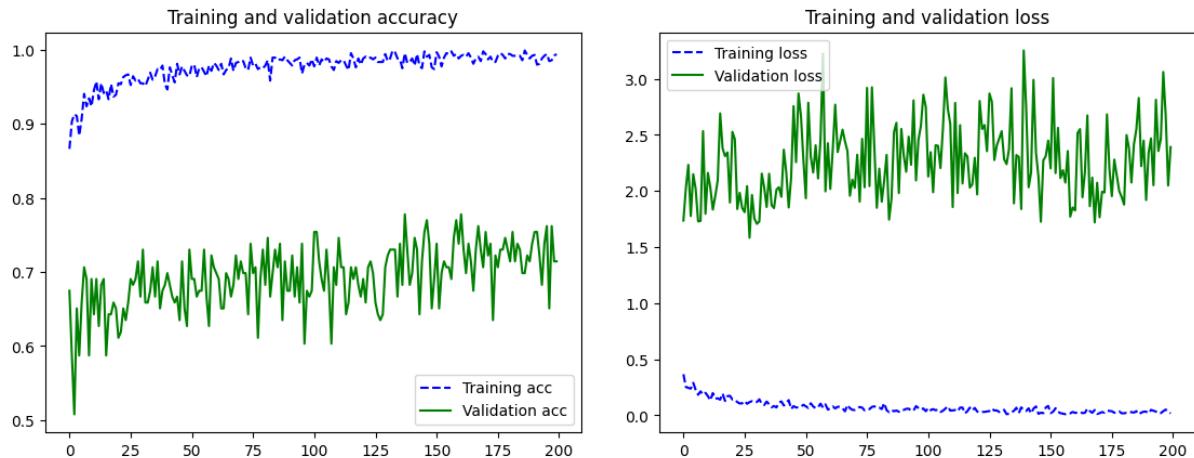
Cette perte de performances peut s'expliquer par le fait que le modèle n'est pas entraîné en profondeur pour notre tâche spécifique. Ainsi, il est naturel qu'il ne soit pas très performant avant que ses couches profondes n'aient été ajustées.

Voici des exemples de mauvaises prédition pour cette phase d'entraînement :



5. Ajout du fine-tuning

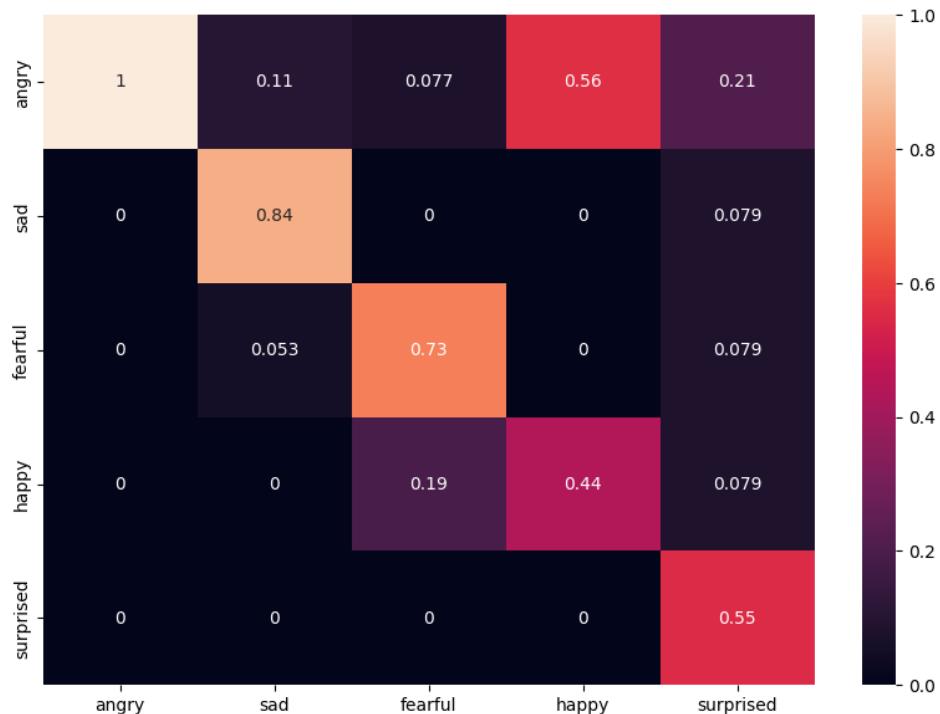
Pour finir, nous avons ajouté le fine-tuning. Cela nous a permis d'obtenir des résultats bien plus satisfaisants (75-80%), bien qu'il reste de grandes limitation :



On retrouve une fois de plus l'allure des entraînements précédents, à la différence que que les performances se sont améliorées, notamment si on regarde la précision:

```
27/27 - 1s - loss: 0.0018 - sparse_categorical_accuracy: 1.0000 - 1s/epoch - 40ms/step
Training accuracy: 100.00%
Training loss: 0.18%
4/4 - 0s - loss: 2.3912 - sparse_categorical_accuracy: 0.7143 - 165ms/epoch - 41ms/step
Evaluation accuracy: 71.43%
Evaluation loss: 239.12%
```

La matrice de confusion obtenue est la meilleure obtenue jusqu'à présent :



On peut voir sur cette version finale que l'on est beaucoup plus centré sur la diagonale que lors des étapes précédentes. On peut notamment voir que le modèle est par exemple très bon pour reconnaître l'émotion "Sad". Cependant, il se trompe très souvent sur les images "Angry", en prédisant 56% du temps qu'il s'agit de "Happy".

Par rapport à ce qui précède (transfer learning sans fine-tuning) on remarque que les prédictions sont bien plus performantes. La modification et l'ajustement des couches profondes ont permis au modèle d'être plus spécifique à notre problème et donc de mieux réaliser de meilleures prédictions.

Ainsi, ce modèle est le plus concluant que nous avons réussi à obtenir, même s'il présente encore des problèmes de confusion entre certaines émotions.

Voici des exemples de mauvaises prédictions pour ce modèle :



VI. Conclusion

En conclusion, notre projet de reconnaissance des émotions sur des visages humains a été une expérience enrichissante et un défi significatif. Nous avons réussi à développer un modèle capable de distinguer cinq émotions différentes : la colère, la peur, la joie, la tristesse et la surprise. Au cours de ce processus, nous avons utilisé un réseau de neurones convolutif, qui s'est avéré efficace pour ce type de tâche d'analyse d'image. Le choix de cette architecture a été motivé par sa capacité à capturer les caractéristiques spatiales des images, ce qui est crucial pour la reconnaissance des émotions.

Cependant, notre travail n'a pas été exempt de difficultés. L'une des principales difficultés rencontrées lors du développement de ce modèle a été la gestion des classes d'émotions qui peuvent être proches les unes des autres. Par exemple, certaines images peuvent présenter des caractéristiques visuelles communes à plusieurs émotions, rendant leur distinction plus délicate.

De plus, nous avons constaté que certaines images étaient un peu ambiguës, ce qui a pu affecter la performance de notre modèle. Ces situations où une image peut correspondre à plusieurs émotions soulignent la complexité de la reconnaissance des émotions humaines à partir d'images et soulèvent des défis supplémentaires pour la modélisation. Ainsi, l'idée de créer des classes plus complexes combinant deux émotions, telles que "joie surprise" ou "peur colère", aurait pu être explorée pour mieux représenter ces nuances émotionnelles.

En termes de perspectives d'amélioration, nous pensons qu'un travail supplémentaire sur notre base de données pourrait améliorer la performance de notre modèle. En particulier, augmenter le nombre d'images dans notre ensemble de données pourrait aider à rendre notre modèle plus robuste et à mieux couvrir les variations et les subtilités des émotions. De plus, l'utilisation de techniques plus avancées de prétraitement des images pourrait contribuer à améliorer la qualité de notre ensemble de données.

En fin de compte, malgré les défis rencontrés, nous sommes satisfaits des résultats obtenus. Ce projet nous a permis de mettre en pratique nos connaissances en apprentissage profond et de comprendre les défis associés à la reconnaissance des émotions à partir d'images. Il ouvre également la voie à de futures recherches pour affiner et améliorer la modélisation des émotions complexes et des nuances émotionnelles dans les images de visages humains.