



Ecole Polytechnique Internationale Privée de Tunis

Projet Computer Vision

Filière : Informatique, Réseaux et Multimédia (IRM)

Reconnaissance des émotions par analyse des expressions faciales

Elaboré par :

Nour SMIAI Groupe 1

Kenza SOUABNI Groupe 3

Yassine MHIRSI Groupe 3

Vivien AZONNOUDO Groupe 3

Supervisé par :

Mr. Walid BARHOUMI

Année scolaire : 2024-2025

Remerciements

Au terme de ce travail, nous tenons à adresser notre gratitude à tous ceux qui ont contribué à l'élaboration de notre projet et, en particulier, à notre encadrant **Docteur Walid Barhoumi** pour ses précieux conseils et sa disponibilité.

Qu'il nous soit permis d'exprimer notre respect et notre gratitude aux membres du jury pour l'honneur qu'ils nous ont fait en acceptant d'examiner et d'évaluer cette modeste contribution, ainsi qu'à l'administration et à tout le corps enseignant de **Polytech INTL** pour la formation qu'ils nous ont assurée tout au long de cette période.

Résumé

On souhaite concevoir un modèle permettant de détecter les émotions humaines à partir des traits faciaux. Il devrait fournir un résultat approprié selon l'image donnée. Dans le cadre de ce projet, on va fournir une interface web alimentée par notre modèle où l'utilisateur peut tester la performance et la précision.

Mots clefs : Computer vision, FER, intelligence artificielle,...

Table des figures

1.1	Logo Google Colab	5
2.1	Nombre total d'images	8
2.2	Image d'origine sans traitement	10
2.3	Image Débruitée	10
2.4	Contours de l'image d'origine	11
2.5	Contours de l'image après traitement	11
3.1	Logo de Streamlit	17
3.2	Chargement du modèle directement depuis un fichier local avec Streamlit	17

Liste des tableaux

Table des matières

1	Cadre du projet	2
1.1	Contexte du projet	3
1.2	Concepts Généraux	3
1.2.1	Acquisition d'images	3
1.2.2	Traitement d'images	3
1.2.3	Segmentation d'images	4
1.2.4	Détection d'objets	4
1.3	Choix du sujet	4
1.3.1	Pertinence des applications	4
1.3.2	Complexité et défi technique	5
1.3.3	Richesse de l'ensemble de données (FER 2013)	5
1.4	Choix des progiciels	5
2	Processus de développement du modèle	7
2.1	Exploration des données	8
2.2	Préparation des données	8
2.3	Création du modèle	11
2.3.1	Principe de fonctionnement d'un CNN	11
2.3.2	Architecture de notre modèle	12
2.4	Entraînement et validation du modèle	14
3	Mise en œuvre	15
3.1	Résultats de l'évaluation finale	16
3.2	Déploiement	17

3.2.1 Avantages de Streamlit 18

Introduction générale

Le projet que nous avons entrepris s'inscrit dans le domaine de la reconnaissance des émotions par l'analyse des expressions faciales, en utilisant des techniques avancées de vision par ordinateur. À une époque où l'intelligence émotionnelle est devenue un enjeu majeur dans les interactions humaines et les systèmes automatisés, notre objectif est de développer un modèle de classification d'images capable d'identifier une gamme variée d'émotions, notamment la joie, la surprise, la peur, la colère, le dégoût, la tristesse, ainsi que le mépris et l'expression neutre. Pour ce faire, nous nous basons sur l'ensemble de données FER 2013, qui comprend plus de 28000 images représentatives d'émotions dans des contextes authentiques.

Notre méthodologie s'articule autour de plusieurs étapes essentielles. Nous commençons par une exploration approfondie des données afin d'en cerner les particularités, suivie d'une préparation minutieuse des images, qui inclut diverses techniques de prétraitement. Nous concevons ensuite un modèle basé sur des réseaux de neurones convolutionnels (CNN), et cette phase est complétée par un entraînement et une validation rigoureuse, permettant d'optimiser les performances du modèle à l'aide de métriques pertinentes.

Enfin, nous évaluons les résultats de notre modèle à l'aide de métriques rigoureuses, telles que la précision et le rappel, afin de garantir sa fiabilité et son efficacité. Une fois cette évaluation effectuée, nous intégrons le modèle dans une interface web interactive, offrant une expérience utilisateur intuitive et accessible. Cette plateforme permet aux utilisateurs d'obtenir des analyses des émotions détectées, favorisant ainsi une meilleure compréhension des sentiments humains dans divers contextes. À travers cette initiative, nous visons à allier avancées technologiques et considérations humaines, tout en soulignant l'importance d'une approche éthique dans le développement d'outils intelligents.

Cadre du projet

Introduction

Dans ce chapitre, nous commençons par définir le contexte de notre projet, en précisant l'objectif académique et les enjeux pratiques de notre sujet. Ensuite, nous présenterons les raisons qui ont motivé le choix de ce sujet, notamment en fonction de ses applications potentielles dans des situations réelles.

1.1 Contexte du projet

Ce projet s'inscrit dans le cadre du cours **Computer Vision**. Il vise à explorer et à appliquer des concepts avancés de traitement d'image et de classification, contribuant ainsi à la compréhension théorique et pratique des méthodes de vision par ordinateur dans un contexte académique. Les projets dans cette discipline ouvrent des perspectives à la fois académiques et pratiques, en lien avec des domaines variés comme la reconnaissance d'objets, l'analyse faciale, et bien plus encore.

1.2 Concepts Généraux

1.2.1 Acquisition d'images

L'évolution des méthodes de capture d'images est une force motrice derrière l'évolution de la vision par ordinateur. Cette évolution a conduit à plusieurs modes de prise de photos et différents types d'images. Il existe diverses modalités d'acquisition d'images, telles que la radiographie numérisée, la tomodensitométrie, la scintigraphie, l'imagerie par ultrasons et l'IRM.

La qualité d'une image dépend de différents réglages liés à l'appareil, tels que : - le rapport signal/bruit, - la sensibilité (spectrale...), - la fonction de transfert (Gamma, résolution), - le temps d'intégration, - le seuil de saturation, - et la rémanence.

1.2.2 Traitement d'images

De nos jours, il existe un besoin réel de traitement d'images. La diversité des secteurs dans lesquels l'imagerie est utilisée rend le traitement d'images et ses différentes techniques indispensables pour l'extraction de données. Ces techniques incluent : l'échantillonnage, la quantification,

l'étalement de l'histogramme, l'égalisation de l'histogramme, le filtrage spatial, etc.

1.2.3 Segmentation d'images

La segmentation d'images consiste à regrouper des pixels partageant les mêmes propriétés afin de former des régions connexes.

Il existe deux approches principales de segmentation d'images : l'approche par contours et l'approche par régions. L'approche par contours délimite les régions en traçant les contours des objets qu'elles représentent. L'approche par régions, quant à elle, détermine les régions en fonction de leurs propriétés intrinsèques.

Le seuillage est une méthode simple et très populaire pour segmenter les objets dans les images numériques. Le seuillage peut être global, local ou adaptatif.

La croissance de régions est une autre méthode de segmentation d'images. Elle consiste à choisir un pixel germe et à accumuler les pixels voisins qui vérifient une propriété d'homogénéité.

1.2.4 Détection d'objets

La détection de points d'intérêt est une technique qui permet d'identifier les points d'une image les plus informatifs. Les coins, les jonctions, les intersections de lignes, les centres de gravité, les extrémités de lignes et les discontinuités de texture sont autant d'exemples de points d'intérêt.

1.3 Choix du sujet

Nous avons opté pour le projet d'analyse des expressions faciales pour diverses raisons :

1.3.1 Pertinence des applications

- La reconnaissance des émotions humaines a des applications variées et en pleine croissance, notamment dans les domaines de la psychologie, de la sécurité, etc.
- Comparativement, les autres projets sont plus spécifiques à l'agriculture, ce qui limite leur applicabilité à un contexte agricole ou environnemental. La reconnaissance des émotions, en revanche, peut être utilisée dans une grande variété de secteurs.

1.3.2 Complexité et défi technique

- Le projet de reconnaissance d'émotions par analyse des expressions faciales est un défi intéressant car il implique des images d'expressions faciales capturées dans des environnements naturels (« in-the-wild »), où la variabilité des poses, des éclairages et des angles ajoute un niveau de complexité.
- Cette complexité technique est formatrice, car nous oblige à acquérir des compétences avancées en computer vision et en manipulation de données déséquilibrées, ce qui est moins prononcé dans les projets sur les cultures agricoles.

1.3.3 Richesse de l'ensemble de données (FER 2013)

- L'ensemble de données FER 2013 est constitué de 28 709 images d'expressions faciales représentant différentes émotions, ce qui est relativement riche et permet d'entraîner des modèles plus robustes. Comparativement, le projet de classification de cultures agricoles (Projet 1) dispose d'un ensemble de données plus limité en taille (159 images par classe après augmentation), et le projet de classification des semis et des mauvaises herbes (Projet 3) se concentre uniquement sur 12 classes de plantes.
- La richesse de l'ensemble de données FER 2013 permet une exploration plus poussée et des expériences variées, notamment avec des modèles complexes et des techniques d'augmentation de données pour améliorer la performance du modèle.

1.4 Choix des progiciels

Pour la réalisation de ce projet, nous avons choisi d'utiliser Google Colab pour la partie computer vision, ce qui nous a permis d'entraîner et de tester le modèle de reconnaissance des émotions sur une plateforme performante et accessible.



Figure 1.1 : Logo Google Colab

Pour la partie web, nous avons choisi d'utiliser XXX, afin de développer une interface interactive

permettant aux utilisateurs de visualiser les résultats et d'interagir avec notre modèle de manière intuitive.

Conclusion

En résumé, ce chapitre a défini les principaux aspects qui ont motivé notre choix de sujet et le cadre dans lequel nous développons ce projet. Le chapitre suivant traitera de la préparation et du traitement du dataset, une étape clé pour garantir la qualité des résultats obtenus.

Processus de développement du modèle

Introduction

Dans cette section, nous détaillons le processus de développement d'un modèle de classification d'images pour la reconnaissance des émotions, depuis l'exploration des données jusqu'à l'entraînement et la validation. Chaque étape est cruciale pour garantir que le modèle soit robuste, fiable et adapté aux besoins spécifiques de l'analyse.

2.1 Exploration des données

L'exploration des données est la première étape cruciale pour comprendre les caractéristiques de l'ensemble FER 2013, qui contient 28,709 images illustrant sept émotions principales : colère (angry), dégoût (disgust), peur (fear), joie (happy), neutre (neutral), tristesse (sad) et surprise (surprise).

```
Nombre total d'images dans le dossier train : 28709
Répartition des images par classe :
sad: 4830 images (16.82%)
happy: 7215 images (25.13%)
neutral: 4965 images (17.29%)
disgust: 436 images (1.52%)
surprise: 3171 images (11.05%)
fear: 4097 images (14.27%)
angry: 3995 images (13.92%)
```

Figure 2.1 : Nombre total d'images

2.2 Préparation des données

La préparation des données est une étape cruciale pour garantir que les images soient adaptées à l'entraînement du modèle de traitement d'images. Elle comprend plusieurs sous-étapes visant à réduire le bruit, à affiner les détails pertinents et à extraire les contours, afin d'améliorer la qualité des données en entrée pour le modèle. Cette étape comprend :

- **Chargement et redimensionnement de l'image d'origine** : L'image d'origine est d'abord chargée depuis le chemin spécifié et redimensionnée à une résolution de 48×48 pixels. Le redimensionnement est effectué pour normaliser les dimensions de toutes les images, ce qui permet d'obtenir une taille cohérente pour le traitement ultérieur et pour l'entraînement du modèle. En redimensionnant chaque image à une résolution standard, on garantit également que les opérations de filtrage et de détection de contours s'appliquent uniformément

sur toutes les données.

- **Réduction du bruit avec un flou gaussien** : Pour atténuer le bruit présent dans l'image, un *filtre gaussien* est appliqué. Le flou gaussien calcule la moyenne des pixels environnants avec un noyau de 5×5 pixels, ce qui réduit les variations de couleur et de luminosité causées par le bruit aléatoire :

$$\text{blured} = \text{cv2.GaussianBlur}(\text{image_origine}, (5, 5), 0)$$

Cette étape permet d'obtenir une image plus uniforme, facilitant les traitements suivants, en particulier l'affinage et la détection des contours.

- **Affinage de l'image** : Après le flou gaussien, l'image est affinée pour renforcer les détails en utilisant une *fusion pondérée* entre l'image d'origine et l'image floutée. La formule utilisée est la suivante :

$$\text{sharpened} = \alpha \cdot \text{image_origine} + \beta \cdot \text{blured} + \gamma$$

où $\alpha = 1.5$ est le poids de l'image d'origine, $\beta = -0.6$ est le poids de l'image floutée, et $\gamma = 0$ est une constante d'ajustement. En code, cela se traduit par :

$$\text{sharpened} = \text{cv2.addWeighted}(\text{image_origine}, 1.5, \text{blured}, -0.6, 0)$$

Cette étape permet d'accentuer les bords et les lignes, rendant les caractéristiques importantes plus visibles.

- **Débruitage avec la méthode des moyennes non-locales (Non-Local Means)** : Pour améliorer encore la qualité de l'image, un débruitage avancé est appliqué avec la méthode *Non-Local Means* :

$$\text{debruitage} = \text{cv2.fastNlMeansDenoisingColored}(\text{sharpened}, \text{None}, 15, 15, 5, 20)$$

Cette méthode réduit le bruit tout en préservant les détails de l'image, en prenant en compte les similitudes de couleur et de structure entre les pixels. Elle est particulièrement utile pour améliorer la qualité des contours détectés.

- **Détection des contours** : Une fois l'image affinée et débruitée, les contours sont extraits en utilisant l'algorithme de *Canny Edge Detection*. Deux ensembles de contours sont calculés :



Figure 2.2 : Image d'origine sans traitement



Figure 2.3 : Image Débruitée

- *Contours de l'image d'origine* : les contours sont détectés directement sur l'image redimensionnée sans traitement. Cela sert de référence pour comparer l'impact des étapes de débruitage et d'affinage.

```
contour_origine = cv2.Canny(image_origine, 30, 30)
```

- *Contours après traitement* : les contours sont détectés sur l'image débruitée et affinée, offrant des contours plus nets et mieux définis.

```
contour_apres_traitement = cv2.Canny(debruitage, 30, 30)
```

Ces contours permettent de mieux capturer les caractéristiques structurelles de l'image, ce qui est essentiel pour l'analyse des formes et des bords par le modèle de traitement d'images.

- **Augmentation des données** Pour pallier le déséquilibre de l'ensemble de données, nous avons appliqué des techniques d'augmentation, notamment :
- *Transformations d'image* : Rotation, zoom, retournement et variations de luminosité pour créer des exemples supplémentaires



Figure 2.4 : Contours de l'image d'origine



Figure 2.5 : Contours de l'image après traitement

2.3 Création du modèle

Dans cette section, nous avons conçu un modèle de réseau de neurones convolutifs, une architecture particulièrement adaptée aux tâches de traitement d'images. Le CNN ou Convolutional Neural Network est capable d'extraire automatiquement des caractéristiques visuelles complexes, ce qui en fait un choix populaire pour les applications de reconnaissance d'images et de computer vision.

2.3.1 Principe de fonctionnement d'un CNN

Un CNN est constitué de plusieurs types de couches qui travaillent ensemble pour transformer une image en une classification précise.

Les principaux composants sont les suivants :

- Couches de Convolution : Ces couches appliquent des filtres (ou kernels) à l'image pour extraire des caractéristiques visuelles, comme des contours, des textures, et des motifs plus complexes. Chaque filtre apprend automatiquement les motifs les plus utiles pour distinguer les émotions.

- Couches de Pooling : Ces couches réduisent la dimensionnalité de l'image en résumant des régions de pixels, ce qui permet de diminuer le nombre de paramètres et le risque de surapprentissage. La couche de pooling conserve les caractéristiques importantes tout en réduisant la quantité de données.
- Couches Pleinement Connectées : En fin de réseau, les couches pleinement connectées combinent les caractéristiques extraites pour réaliser la classification. Ces couches prennent toutes les informations apprises par les couches précédentes et décident de l'étiquette finale de l'image.
- Couche de Sortie : La dernière couche utilise une fonction d'activation softmax pour produire une probabilité pour chaque classe d'émotion, permettant de déterminer la catégorie émotionnelle la plus probable de l'image.

2.3.2 Architecture de notre modèle

Dans notre modèle, nous avons utilisé plusieurs couches de convolution et de pooling pour apprendre des caractéristiques à différents niveaux de complexité. Ces couches sont suivies de couches pleinement connectées pour la classification finale des émotions.

```
1 model = Sequential()
```

- Couches de Convolution et Pooling : Trois couches de convolution avec un nombre croissant de filtres (32,64,128) pour capter des caractéristiques de plus en plus complexes, suivies de couches de pooling pour réduire la dimensionnalité.

```
model.add(Conv2D(32, (3, 3), activation='relu', input_shape=input_shape))  
model.add(Conv2D(32, (3, 3), activation='relu'))  
model.add(BatchNormalization())
```

```
model.add(MaxPooling2D(pool_size=(2, 2)))
```

```
model.add(Conv2D(64, (3, 3), activation='relu', input_shape=input_shape))
model.add(Conv2D(64, (3, 3), activation='relu'))
model.add(BatchNormalization())
```

```
model.add(MaxPooling2D(pool_size=(2, 2)))
```

```
model.add(Conv2D(128, (3, 3), activation='relu', input_shape=input_shape))
model.add(Conv2D(128, (3, 3), activation='relu'))
model.add(BatchNormalization())
```

```
model.add(MaxPooling2D(pool_size=(2, 2)))
```

- Dropout : Des couches de dropout sont ajoutées pour éviter le surapprentissage en désactivant aléatoirement une partie des neurones pendant l'entraînement.

```
model.add(Dropout(0.3))
```

- Couches Fully Connected : Deux couches pleinement connectées (512 et 256 neurones) qui combinent les caractéristiques extraites avant la sortie.

```
model.add(Dense(128, activation='relu'))
```

- Couche de Sortie : Une couche de sortie avec une activation softmax pour prédire la probabilité de chaque émotion.

```
1 model.add(Dense(num_classes, activation='softmax'))
```

2.4 Entraînement et validation du modèle

C'est au cours de cette étape que le modèle "apprend" en ajustant ses paramètres pour minimiser l'erreur sur les données d'entraînement, tout en évaluant sa performance sur les données de validation.

Passage des Données : Les données d'entraînement et de validation sont passées à la méthode `fit()`.

```
1 history = model.fit(X_train, y_train,
2                     batch_size=64, epochs=50,
3                     validation_data=(X_test, y_test),
4                     callbacks=[checkpoint])
```

Batch Size : Le nombre d'exemples de formation par mise à jour des poids (défini par `batch_size`).

Nombre d'Époques : Le nombre de fois que l'algorithme passe sur l'intégralité des données (défini par `epochs`).

Callbacks : Les callbacks comme `ModelCheckpoint` et `ReduceLROnPlateau` aident à sauvegarder les meilleurs modèles et ajuster le taux d'apprentissage en fonction des performances sur les données de validation

L'entraînement a été réalisé en surveillant les métriques de performance :

- **Suivi des métriques** : La perte et l'exactitude ont été évaluées pendant l'entraînement.
- **Validation croisée** : Utilisation de l'ensemble de validation pour ajuster les hyperparamètres et éviter l'overfitting.

Conclusion

Ce chapitre a présenté les différentes étapes du processus de développement du modèle pour la reconnaissance des émotions à partir d'expressions faciales. Chacune de ces étapes est essentielle pour garantir la qualité et l'efficacité du modèle final.

Mise en œuvre

Introduction

Dans cette section, nous présenterons les résultats de l'évaluation finale de notre modèle de reconnaissance d'émotions par analyse des expressions faciales, ainsi que son déploiement sur un site web fonctionnel.

3.1 Résultats de l'évaluation finale

L'évaluation finale du modèle a été réalisée sur l'ensemble de test, qui a été séparé lors de la préparation des données. Les principales métriques utilisées pour évaluer la performance du modèle incluent l'exactitude, la précision, le rappel et la mesure F1.

Les résultats obtenus sont résumés comme suit :

- **Exactitude** : Le modèle a atteint une exactitude de $X\%$, ce qui indique une bonne capacité à classer correctement les émotions.
- **Matrice de confusion** : La matrice de confusion révèle les performances spécifiques pour chaque classe d'émotion, mettant en évidence les émotions les mieux et les moins bien classées.
- **Courbes ROC** : Les courbes ROC pour chaque classe montrent les compromis entre le taux de vrais positifs et le taux de faux positifs, permettant d'évaluer la capacité du modèle à discriminer entre les différentes émotions.

Ces résultats montrent que le modèle est capable de reconnaître les émotions avec une performance satisfaisante, bien qu'il existe des domaines d'amélioration identifiés, notamment pour certaines émotions comme la tristesse et le dégoût, qui ont affiché des taux de classification plus faibles.

3.2 Déploiement

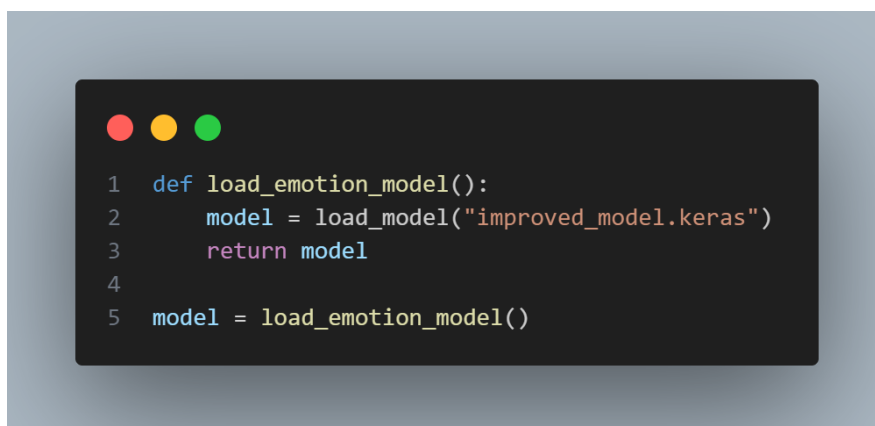
Le déploiement du modèle est une étape cruciale pour le rendre accessible aux utilisateurs finaux via une interface web intuitive. Pour ce projet, nous avons utilisé **Streamlit**, une bibliothèque Python open-source, permettant de déployer des modèles de machine learning de manière rapide et efficace.



Figure 3.1 : Logo de Streamlit

Les étapes de déploiement réalisées sont les suivantes :

- **Préparation de l'environnement** : Configuration d'un environnement de production, comprenant l'installation des bibliothèques nécessaires et la création d'un serveur pour héberger le modèle.
- **Intégration du modèle** : Le modèle a été intégré dans une application web utilisant Streamlit, permettant aux utilisateurs de télécharger des images et de recevoir des prédictions en temps réel. Grâce à Streamlit, il est possible de charger le modèle directement depuis un fichier local, sans passer par une API, ce qui réduit les temps de latence et simplifie le flux de travail. Voici un extrait de code illustrant le chargement du modèle :

A screenshot of a code editor window with a dark background and light-colored text. The code is a Python function definition and a call to that function. The function is named 'load_emotion_model' and it takes a file path as an argument. The code is as follows:

```
1 def load_emotion_model():
2     model = load_model("improved_model.keras")
3     return model
4
5 model = load_emotion_model()
```

Figure 3.2 : Chargement du modèle directement depuis un fichier local avec Streamlit

- **Interface utilisateur** : Développement d'une interface utilisateur simple et intuitive qui facilite l'interaction avec le modèle, avec des indications claires sur les émotions détectées.
- **Tests de performance** : Réalisation de tests pour s'assurer que le modèle fonctionne correctement en conditions réelles, en vérifiant sa rapidité et sa précision lors de la classification d'images en temps réel.

Le déploiement réussi du modèle ouvre la voie à diverses applications, notamment dans les domaines de la psychologie, du marketing et des interfaces utilisateur, permettant ainsi d'analyser les émotions des utilisateurs de manière plus efficace.

3.2.1 Avantages de Streamlit

L'utilisation de Streamlit offre plusieurs avantages dans le contexte de déploiement d'applications de machine learning :

- **Simplicité de développement** : Streamlit permet d'itérer rapidement sur le code Python, avec un rechargement en temps réel à chaque modification, réduisant les temps de développement.
- **Accès direct au modèle sans API externe** : Un des principaux avantages de Streamlit est qu'il permet de charger le modèle directement depuis un fichier local, sans avoir besoin de le transformer en une API pour l'appeler depuis un framework front-end. Cette approche simplifie le flux de travail et élimine le délai supplémentaire qu'un appel API introduirait. Cela signifie que les utilisateurs reçoivent les prédictions en temps réel, améliorant la réactivité et la fluidité de l'expérience utilisateur.
- **Support interactif** : Avec des composants interactifs comme les boutons, les sélecteurs, et les barres de progression, Streamlit enrichit l'expérience utilisateur. Les utilisateurs peuvent ainsi interagir facilement avec le modèle et visualiser les prédictions instantanément.
- **Flexibilité et extensibilité** : La plateforme permet d'ajouter des fonctionnalités supplémentaires ou de modifier l'interface facilement pour répondre à des besoins spécifiques du projet, offrant une solution de déploiement adaptable aux exigences de l'application.

La réussite de ce déploiement avec Streamlit ouvre la voie à diverses applications dans les domaines de la psychologie, du marketing, et de l'interaction homme-machine, permettant de mieux comprendre les émotions des utilisateurs en temps réel.

Conclusion

En conclusion, ce chapitre a démontré avec succès l'efficacité de notre modèle de reconnaissance d'émotions par analyse des expressions faciales, validé par des résultats d'évaluation solides. Le déploiement sur un site web a permis de rendre accessible cette technologie, en offrant une interface conviviale aux utilisateurs finaux.

Conclusion générale

Ce projet, centré sur la reconnaissance des émotions par analyse des expressions faciales, a permis de mettre en pratique des techniques avancées de computer vision et de deep learning. À travers la conception d'un modèle de classification d'images capable de distinguer diverses émotions humaines, nous avons développé une solution s'appuyant sur des données réelles et acquises dans des environnements naturels, ce qui renforce la robustesse et la pertinence de notre approche.

Au fil des étapes – de l'exploration et la préparation des données à l'entraînement, la validation et l'évaluation finale du modèle – nous avons acquis des compétences essentielles en manipulation de données, en développement d'algorithmes de classification et en gestion des défis associés à des données déséquilibrées. L'implémentation d'une interface web interactive a également été une étape enrichissante, permettant de rendre notre solution accessible et d'en faciliter l'utilisation.

Les résultats obtenus sont encourageants et montrent le potentiel des applications de reconnaissance d'émotions dans des domaines variés, tel que l'analyse des interactions humaines. Ce projet ouvre des perspectives pour des améliorations futures, notamment en perfectionnant la précision du modèle, en l'adaptant à des contextes culturels variés, ou encore en explorant d'autres types de données pour enrichir sa capacité d'analyse émotionnelle.

En conclusion, cette expérience nous a permis de développer des compétences techniques solides tout en nous sensibilisant aux enjeux éthiques et pratiques liés à l'utilisation de l'intelligence artificielle pour l'analyse des émotions.