

# Rapport de projet : Classification d'émotions à partir d'images

Iris Pouth Nadia

Ourdani Salima

Sadi Manal

---

## - Dataset initial

Lien du dataset : <https://www.kaggle.com/datasets/saharnazyaghoobpoor/happy-and-sad-image>

## - Notebook Colab (.ipynb) :

Lien google colab : <https://colab.research.google.com/drive/1LfFK3-OvbnIHa3l7izpz5FEKWEWoQFbw#scrollTo=FEJGdjj3ESo4>

## Description du Dataset et de la Tâche

La tâche consiste en une classification binaire d'images pour détecter deux émotions humaines : la joie et la tristesse.

- **Source** : Le dataset a été téléchargé depuis Kaggle via kagglehub (saharnazyaghoobpoor/happy-and-sad-image).
- **Composition** : Le jeu de données comprend un total de 157 images, réparties comme suit :
  - Happy : 89 images.
  - Sad : 68 images.
- **Prétraitement** : Les images sont redimensionnées en 128x128 pixels et converties en niveaux de gris (1 canal) pour réduire la complexité et se concentrer sur les traits faciaux.

## Modèle de Base et Résultats

Le modèle utilisé, nommé BetterCNN, est un réseau de neurones convolutif personnalisé.

- **Architecture** : Extracteur de caractéristiques (Features) : Trois blocs de convolution (Conv2d) avec normalisation par lots (BatchNorm2d), activation ReLU et MaxPool2d (réduisant la dimension de 128x128 à 16x16).
- **Classificateur** : Une couche de Dropout (0.5) pour limiter le surapprentissage, suivie de deux couches linéaires.
- **Résultats initiaux** : Lors de la première époque, le modèle commence avec une perte de 0.7143 et une précision sur le test de 45.71%.

## Tableau Récapitulatif des Expériences

Expérience	Description du Modèle	Principales Modification	Résultat(Test Accuracy)
<b>Modèle de Base</b>	CNN Standard (Conv + ReLU)	Architecture simple sans régularisation.	~45.7% (Époque 1)
<b>Modèle Amélioré</b>	BetterCNN + BatchNorm	Ajout de la normalisation par lots pour la stabilité.	~74.3% (Époque 5)
<b>Modèle Finale</b>	BetterCNN + Optimisations	Dropout (0.5), Label Smoothing, et Data Augmentation.	88.57% (Époque 19)

### **Modifications et Fine-tuning**

Plusieurs stratégies ont été implémentées pour optimiser le modèle face au faible volume de données :

➤ **Augmentation de Données (Data Augmentation) :**

**Modifications** : Ajout de rotations aléatoires ( $20^\circ$ ), de zooms/déformations (RandomAffine) et de retournements horizontaux.

**Pourquoi** : Pour simuler des têtes penchées et varier les angles de vue, compensant ainsi la petite taille du dataset et améliorant la robustesse.

➤ **Régularisation :**

**Modifications** : Utilisation du Label Smoothing (0.1) dans la fonction de perte et d'un Weight Decay ( $1e-4$ ).

**Pourquoi** : Le label smoothing empêche le modèle de devenir trop confiant dans ses prédictions, ce qui réduit le surapprentissage.

➤ **Planificateur de Taux d'Apprentissage (Scheduler) :**

**Modifications** : ReduceLROnPlateau avec un facteur de 0.1 et une patience de 2 époques.

**Pourquoi** : Permet de réduire dynamiquement le taux d'apprentissage quand la perte stagne, afin d'affiner la convergence vers un minimum global.

### **Conclusions Principales**

**Performance** : Le modèle atteint une excellente précision maximale de 88.57% sur l'ensemble de test (Époque 19).

**Généralisation** : Malgré un dataset très réduit (< 200 images), l'utilisation combinée de l'augmentation de données et du dropout a permis d'obtenir un modèle capable de prédire correctement des images externes (testé avec succès sur une URL d'image web).

**Stabilité** : L'ajout de la BatchNorm2d et du scheduler a stabilisé l'entraînement, bien qu'une légère fluctuation des performances soit visible en fin d'entraînement (époque 20), justifiant la sauvegarde automatique du "meilleur modèle".