

Real-Time Facial Emotion Recognition System With Improved Preprocessing and Feature Extraction

Dr Ansamma John, Abhishek MC, Ananthu S Ajayan, Sanoop S and Vishnu R Kumar



Présenté par : Philippe SAVARD - Gaspard PETITCLAIR

INF8801A- Application multimédia

• Plan de présentation

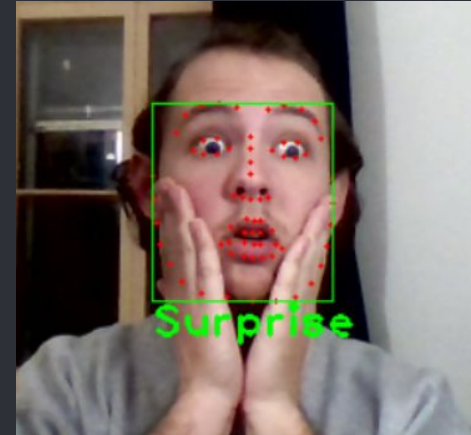
- I. Contexte applicatif
- II. Objectifs du projet
- III. Méthodologie
- IV. Configuration
- V. Analyse et résultats
- VI. Conclusion



Contexte applicatif

- Contexte applicatif

- Détection en temps réel des émotions
- Amélioration des méthodes traditionnelles par CNN
 - Points clés du visage
 - Histogramme des orientations du gradient





● Objectifs du projet

● Objectifs du projet

- Implémenter la méthodologie proposée
 - Implémenter et entraîner des réseaux CNN
 - Tester différents prétraitements
- Comparer les résultats obtenus avec ceux de l'article afin d'en faire l'analyse
- Évaluer la précision pour différents prétraitements
 - CNN uniquement
 - CNN + points clés
 - CNN + points clés + HOG



● Méthodologie



● Méthodologie

1. Hors ligne

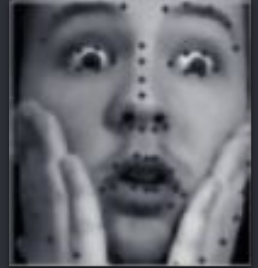
- a. Détection de la région d'intérêt du visage
- b. Extraction et sauvegarde des features
- c. Entraînement et enregistrement des réseaux

2. En ligne

- a. Démarrage de la caméra
- b. Image par image :
 - i. Détection de la région d'intérêt du visage
 - ii. Extraction des features
 - iii. Classification par CNN

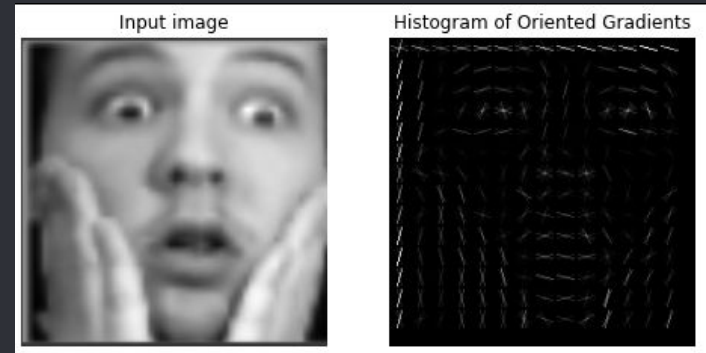
- Détection du visage et prétraitement

- Conversion de l'image en tons de gris avec OpenCV



- Recadrage et détection des points clés avec la librairie DLIB

- Extraction du HOG

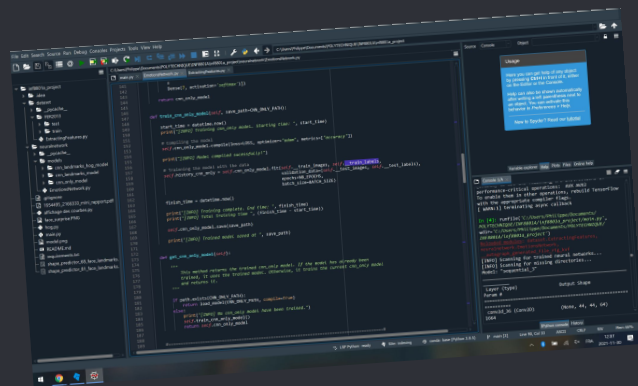


```
self.train_cnn_landmarks_hog_model()
File "C:\Anaconda3\envs\keras\lib\site-packages\keras\models.py", line 100, in
train_cnn_landmarks_hog_model
self.history_cnn_landmarks_hog = self.cnn_landmarks_hog_model.fit(
File "C:\Anaconda3\envs\keras\lib\site-packages\keras\models.py", line 47, in
error_handler
raise e.with_traceback(filtered_tb) from None
File "C:\Anaconda3\lib\site-packages\keras\engine\data_adapter.py", line 1097, in
_check_data_cardinality
raise ValueError(msg)
ValueError: Data cardinality is ambiguous:
  x sizes: 7178, 7178, 28709
  y sizes: 7178
Make sure all arrays contain the same number of samples.
```

In [4]:



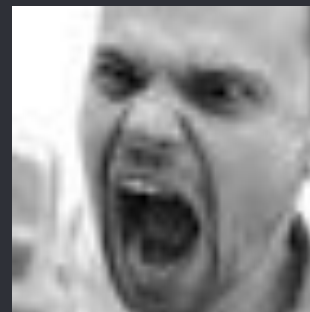
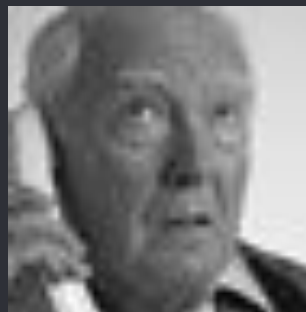
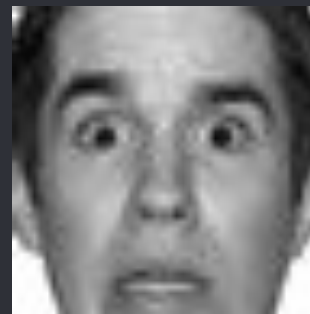
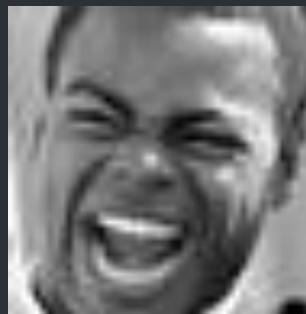
Configuration



• Dataset FER2013

Banque d'images étiquetées avec 7 émotions :

1. Colère
2. Dégoût
3. Peur
4. Joie
5. Tristesse
6. Neutre
7. Surprise



- Extraction des éléments clés avec dlib

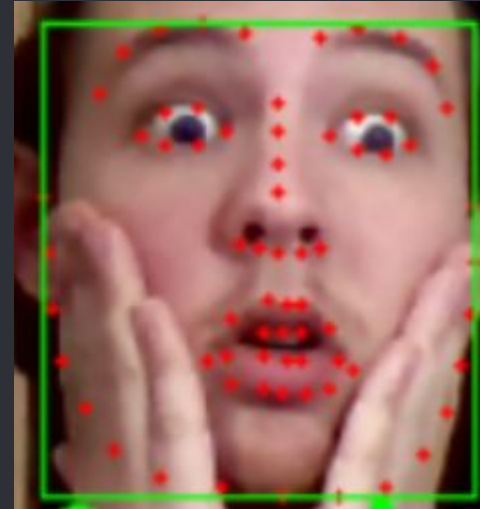
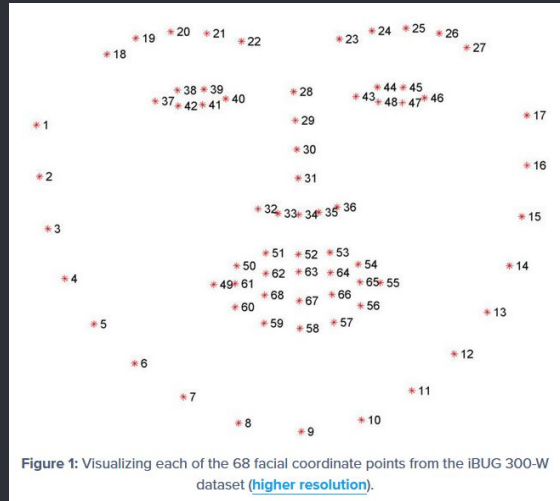
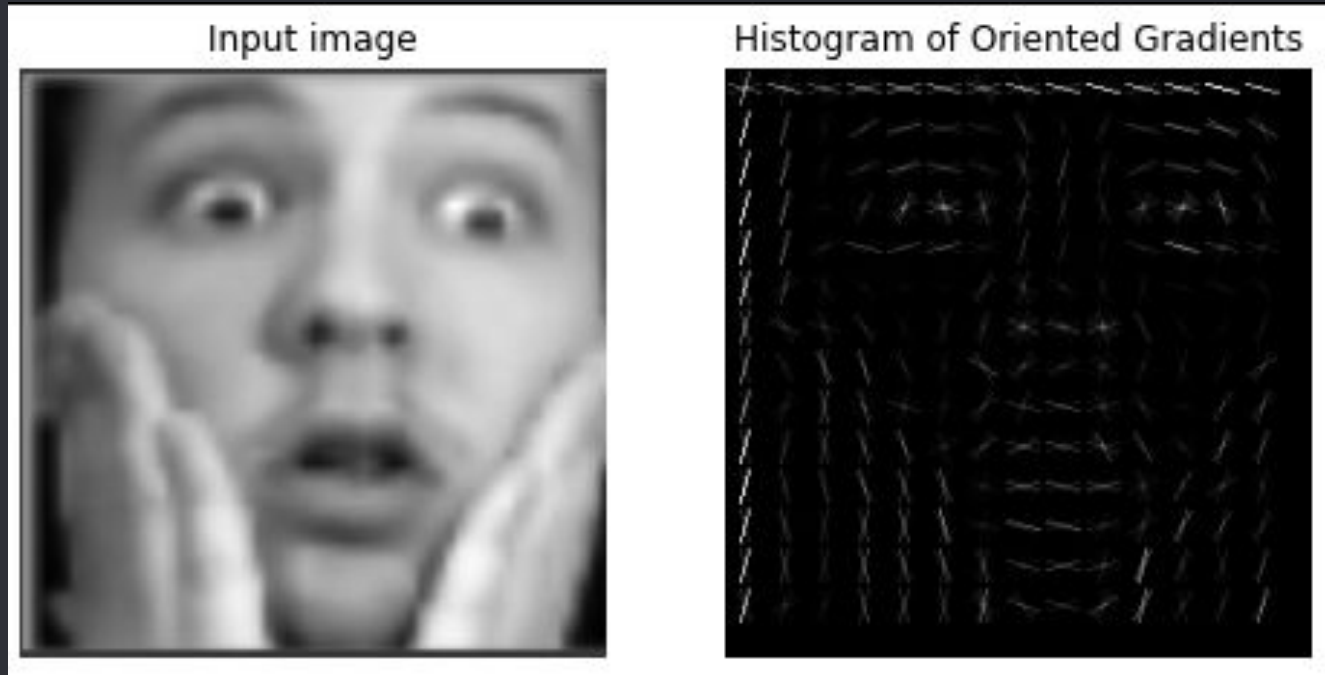


Figure 1. - Points clés du visage extraits par dlib [1]

- Extraction du HOG avec skimage



• Structure des réseaux de neurones utilisés

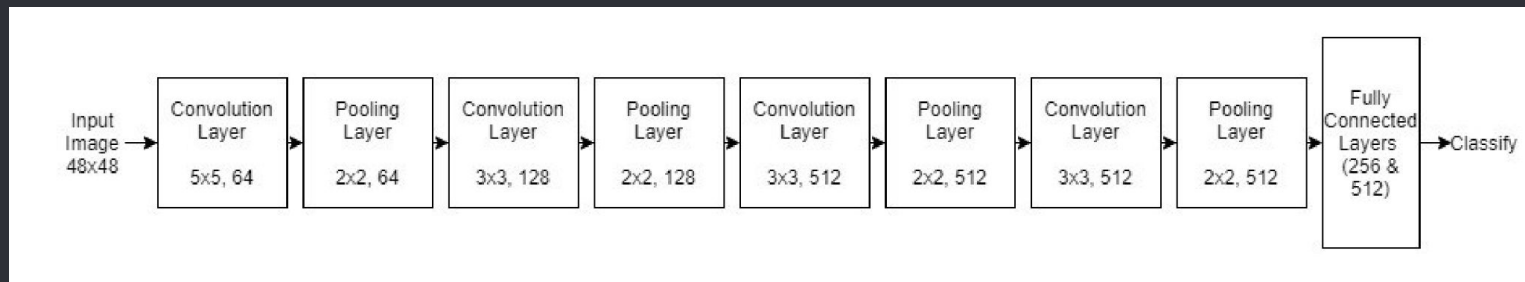
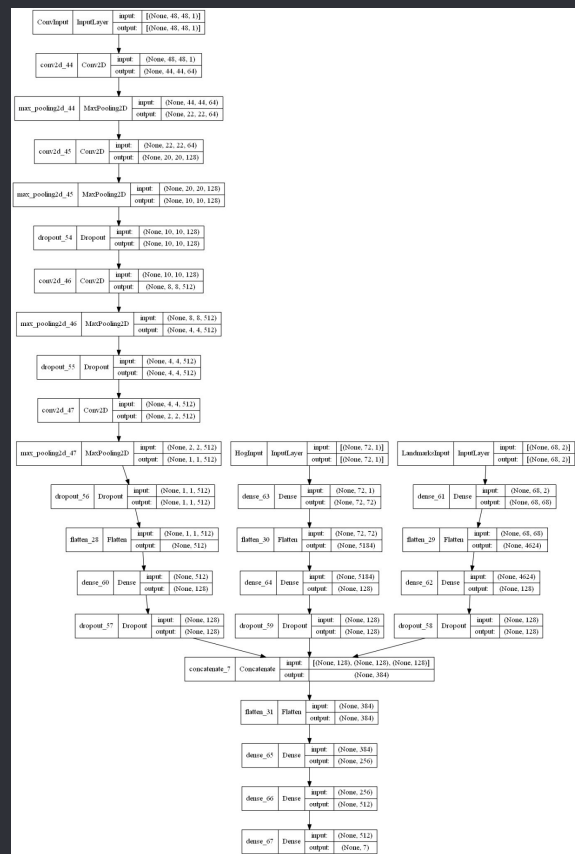
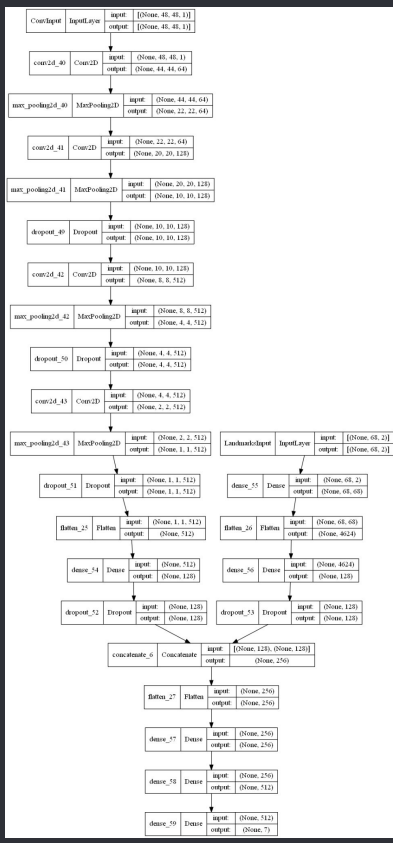
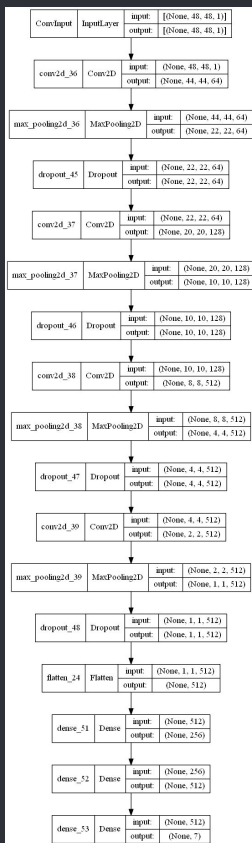


Figure 2. - Architecture du CNN proposée par l'article [2]

Structure des réseaux de neurones utilisés





Analyse et résultats

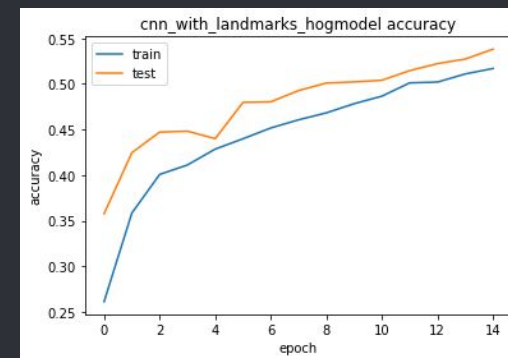
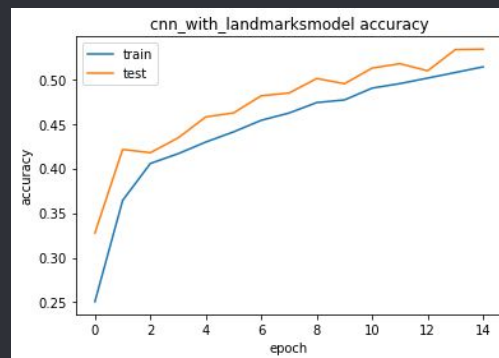
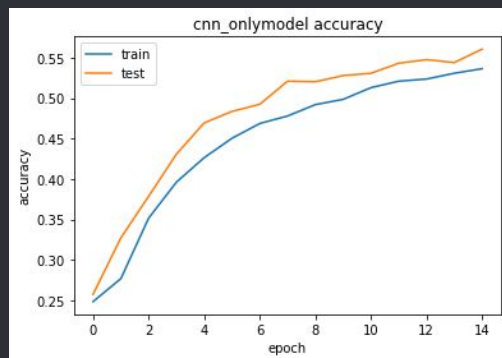
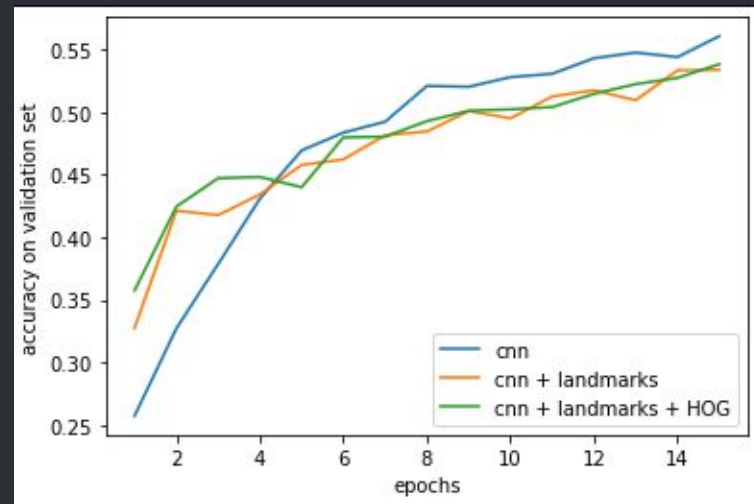
Analyse des performances

Précision atteinte après 15 epochs:

CNN : 56,05%

CNN + Landmarks : 53,37%

CNN + Landmarks + HOG : 53,82%



ACCURACY WHEN FEATURE DESCRIPTORS ARE INCLUDED

Experiments	Accuracy	
	JAFFE	FER2013
CNN (on raw data)	90.698%	73.5%
CNN + Facial landmarks	91.2%	74.4%
CNN + Facial landmarks + HOG	90.698%	73.2%

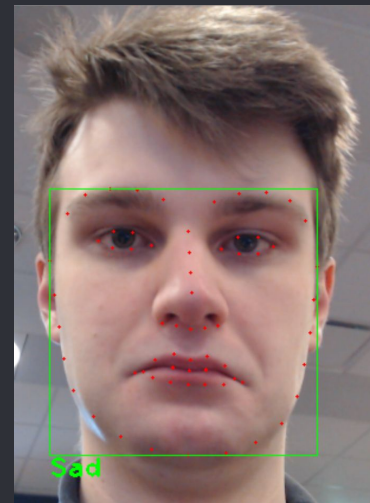
Figure 3. - Résultats de l'article [2]



Conclusion

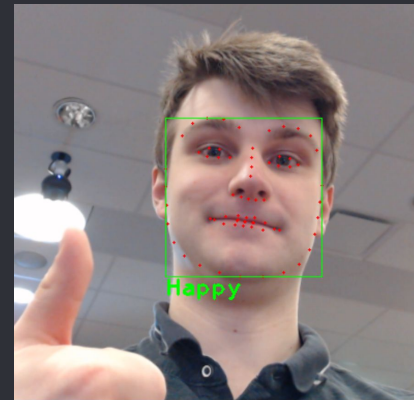
Limitations

- Détails de l'implémentation des réseaux CNN insuffisants pour reproduire les résultats intégralement
- Base de données plutôt subjective
- JAFFE non disponible pour le public
- Temps d'entraînement très grand pour un obtenir les résultats présentés dans l'article



- Apprentissages

- Apprendre comment entraîner un réseau de neurones avec Tensorflow
- Comprendre les impacts des paramètres sur la performance du réseau
- Utiliser un réseau CNN dans une application multimédia concrète





Merci de votre attention !!

Références

— — —

[1] Adrian Rosebrock. (Avril 2017). *Detect eyes, nose, lips, and jaw with dlib, OpenCV, and Python*. pyimagesearch.
<https://www.pyimagesearch.com/2017/04/10/detect-eyes-nose-lips-jaw-dlib-opencv-python/>

[2] A. John, A. MC, A. S. Ajayan, S. Sanoop and V. R. Kumar, "Real-Time Facial Emotion Recognition System With Improved Preprocessing and Feature Extraction," 2020 Third International Conference on Smart Systems and Inventive Technology (ICSSIT), 2020, pp. 1328-1333, doi: 10.1109/ICSSIT48917.2020.9214207.