République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur
et de la Recherche Scientifique

ECOLE SUPÉRIEURE EN INFORMATIQUE
8 Mai 1945 - Sidi-Bel-Abbès

الجمهورية الجزائرية الديمقراطية الشعبية
وزارة التعليم العالي والبحث العلمي

المدرسة العليا للإعلام الآلي
8 ماي 1945 - سيدي بلعباس

Rapport De Projet

Titre

REAL-TIME FACE RECOGNITION AND EMOTION ANALYSIS

Group members:

- DJELLOULI Ahmed.
- SAIDOUNE Houssem.
- BELKHODJA Sabrina.
- YOUSFI Madani Assia.

Supervisors:

Pr. RAHMOUN Abdelatif
Dr. BENSENANE Hamdan

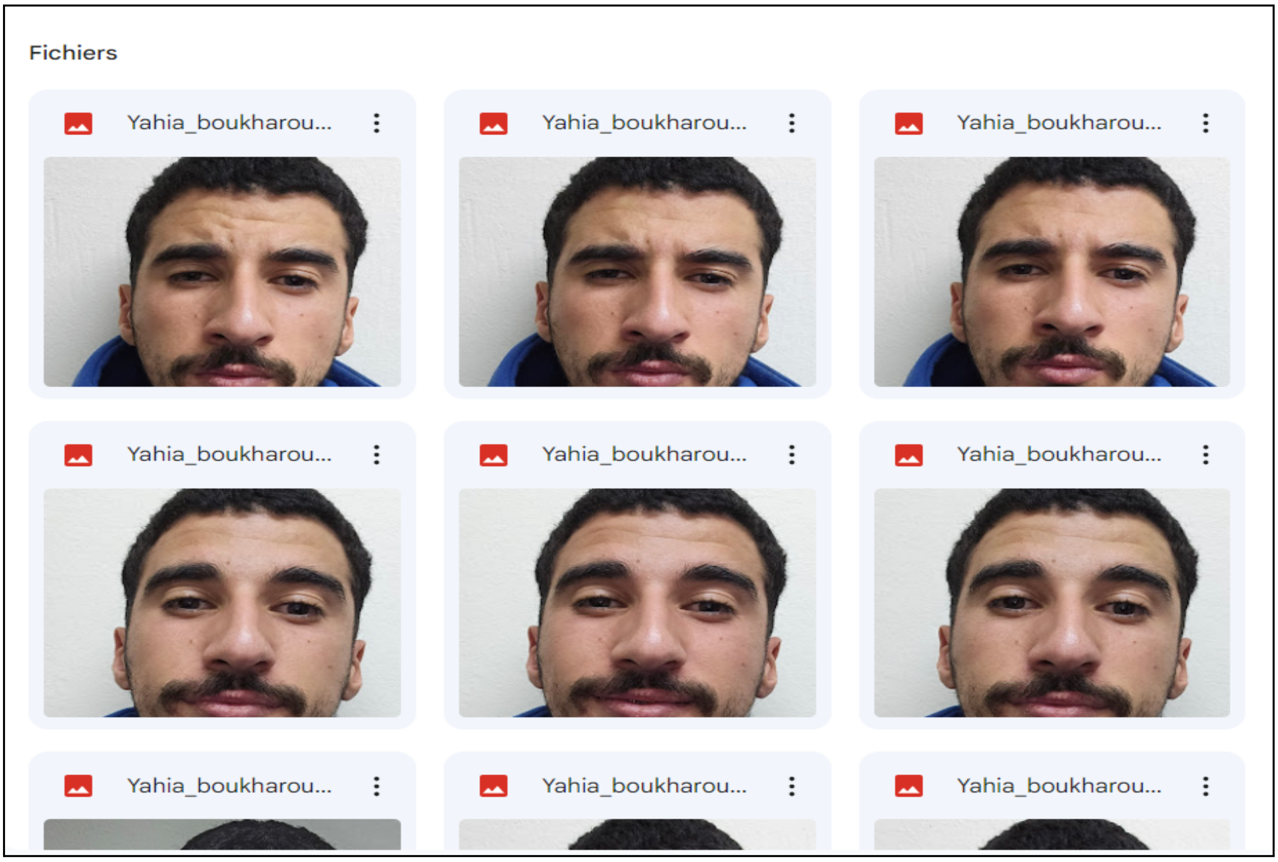
Table of contents :

1. Introduction:

This report presents a comprehensive overview of a project focused on real-time face recognition and emotion analysis. The aim of the project was to develop an intelligent system capable of detecting and recognizing faces in real-time, while also analyzing the emotional state of the individuals. The project utilized state-of-the-art deep learning techniques and computer vision algorithms to achieve accurate face recognition and emotion analysis results.

2. Collecting the data:

Collecting data for the real-time face recognition and emotion analysis project is a crucial step in building a robust and accurate system.To ensure a comprehensive dataset, we designed a data collection process that involved participants taking photos of their faces displaying different emotions. We were able to capture a wide range of facial expressions, including **happiness, sadness, tiredness and neutral expression** , from different sides of 19 person.



3. Data augmentation :

We applied several image augmentation techniques to increase the diversity and size of our dataset. These techniques allow us to create new variations of the original images, enhancing the model's ability to generalize and improve its performance.

- **Flip Image:** flip_image() function was used to horizontally flip the images, effectively generating mirror images. This augmentation helps the model to learn from different perspectives and orientations.
- **Rotate Image:** rotate_image() function was employed to rotate the images by a specified angle. By rotating the images, we introduced variations in their orientation, making the model more robust to different object alignments.
- **Zoom Image:** zoom_image() function was used to randomly zoom in or out on the images. This augmentation alters the scale of objects within the image, providing the model with additional information about different object sizes and distances.
- **Translate Image:** translate_image() function was utilized to shift the images horizontally and vertically within a certain range. This

augmentation helps the model to learn the importance of object placement and position in the image.

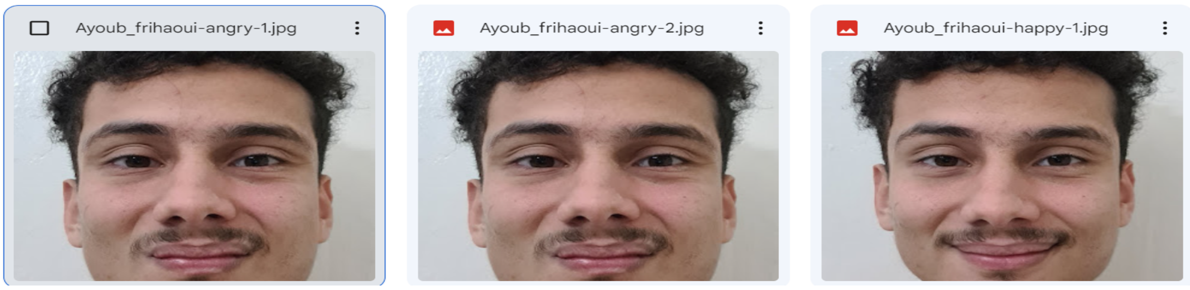
4. Face Detection:

To identify and localize only faces in our images we used **MTCNN** which is a deep learning-based algorithm commonly used for face detection and facial feature alignment.MTCNN is designed to detect and align faces in images with varying poses, scales, and lighting conditions in three stages :

- **Face Detection :** The network scans the image at multiple scales and aspect ratios to detect faces.
- **Facial Landmark Localization:** Once potential face regions are detected, the second stage of MTCNN is employed to locate facial landmarks such as the eyes, nose, and mouth. This stage uses another CNN to regress the coordinates of these landmark.
- **Bounding Box Refinement:** The final stage of MTCNN refines the bounding box coordinates of the detected faces and adjusts them to tightly fit around the face regions.

5. Cropping faces :

For each detected face, extract the corresponding region from the original image by using the coordinates of the bounding box. **img[y:y+h, x:x+w]**, where (x, y) represents the top-left corner of the bounding box, and (w, h) represents the width and height of the bounding box
and then we Save the cropped face region as a separate image file.



6. Face recognition :

a. VGG-Face :

Vgg_face is a convolutional neural network (CNN) specifically designed for face recognition. The VGG-Face model is based on the VGG16 architecture and has been trained on a large dataset of face images to learn discriminative features for face recognition. Here's a breakdown of the architecture:

Layer (type)	Output Shape	Param #
input_7 (InputLayer)	[(None, 224, 224, 3)]	0
block1_conv1 (Conv2D)	(None, 224, 224, 64)	1792
block1_conv2 (Conv2D)	(None, 224, 224, 64)	36928
block1_pool (MaxPooling2D)	(None, 112, 112, 64)	0
block2_conv1 (Conv2D)	(None, 112, 112, 128)	73856
block2_conv2 (Conv2D)	(None, 112, 112, 128)	147584
block2_pool (MaxPooling2D)	(None, 56, 56, 128)	0
block3_conv1 (Conv2D)	(None, 56, 56, 256)	295168
block3_conv2 (Conv2D)	(None, 56, 56, 256)	590080
block3_conv3 (Conv2D)	(None, 56, 56, 256)	590080
block3_pool (MaxPooling2D)	(None, 28, 28, 256)	0
block4_conv1 (Conv2D)	(None, 28, 28, 512)	1180160
block4_conv2 (Conv2D)	(None, 28, 28, 512)	2359808
block4_conv3 (Conv2D)	(None, 28, 28, 512)	2359808
block4_pool (MaxPooling2D)	(None, 14, 14, 512)	0
block5_conv1 (Conv2D)	(None, 14, 14, 512)	2359808
block5_conv2 (Conv2D)	(None, 14, 14, 512)	2359808
block5_conv3 (Conv2D)	(None, 14, 14, 512)	2359808
block5_pool (MaxPooling2D)	(None, 7, 7, 512)	0
flatten (Flatten)	(None, 25088)	0
fc1 (Dense)	(None, 4096)	102764544

Total params: 117,479,232

- **ZeroPadding2D:** Adds zero-padding to the input images to maintain spatial dimensions.
- **Convolution2D:** Performs 2D convolution on the input images, extracting features using small filters.
- **MaxPooling2D:** Performs max pooling to reduce the spatial dimensions and capture the most important features.
- **Dropout:** Helps prevent overfitting by randomly dropping out a fraction of the neurons during training.

- **Flatten:** Flattens the multi-dimensional feature maps into one-dimensional vector.
- **Activation:** apply softmax function to the last layer to obtain class probabilities.

We used this architecture with pre-trained weights to obtain numerical vectors that capture the unique characteristics of a face called “embeddings” from our images.

b. classifier network :

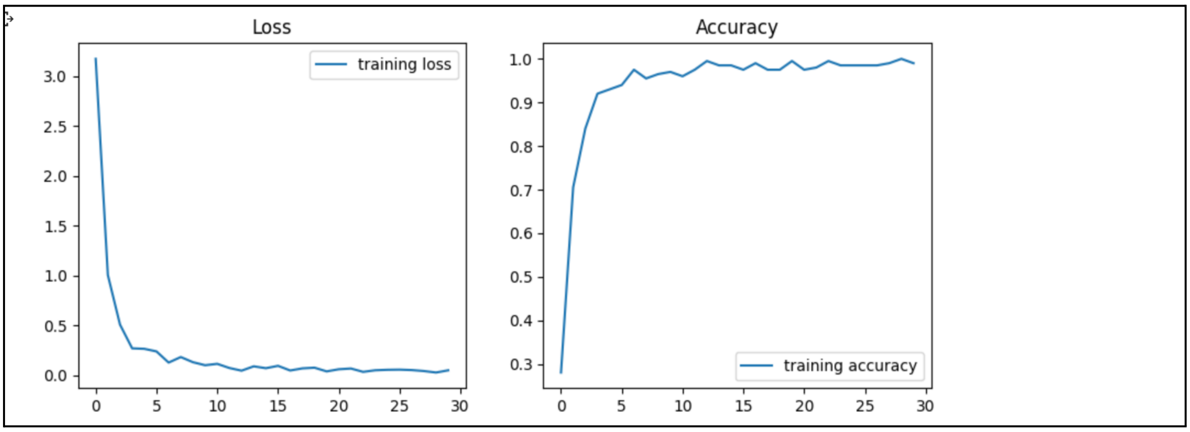
to recognize the identity of the person we passed the resulted embeddings of vgg-face model , this is the architecture of our classifier :

```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 flatten_1 (Flatten)         (None, 2622)              0

 dense (Dense)               (None, 128)               335744

 dense_1 (Dense)             (None, 18)                2322

=================================================================
Total params: 338,066
Trainable params: 338,066
Non-trainable params: 0
_________________________________________________________________
```

The Optimizer and loss function:

We have chosen the **"Adam"** optimization algorithm and categorical cross-entropy loss from the keras library.
Adam controls the rate of gradient descent in such a way that there is minimum oscillation when it reaches the global minimum while taking big enough steps (step-size) so as to pass the local minima hurdles along the way.

The evaluation of Accuracy and loss during the training :



Model Performance Analysis on Test Dataset :

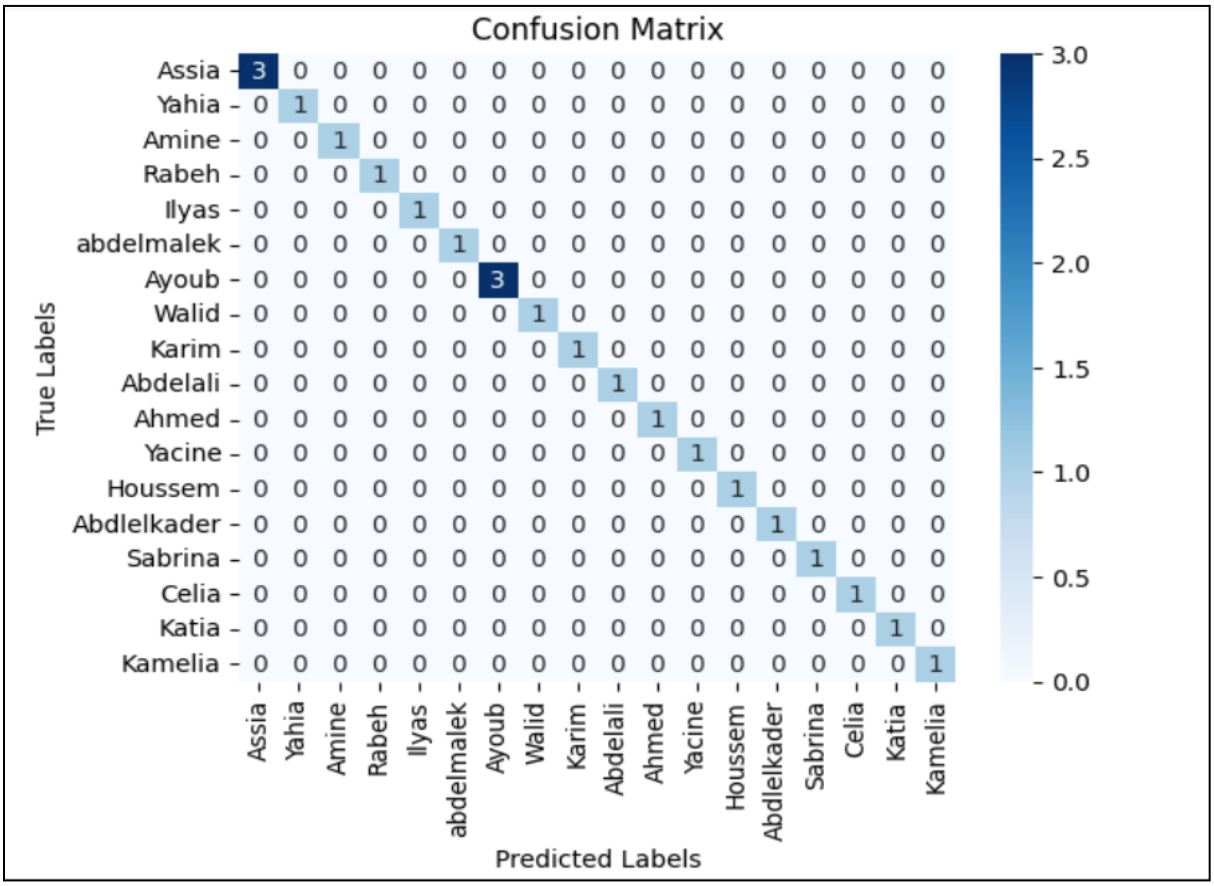
The results on the test data indicate that the model performed remarkably well, showcasing superior accuracy and robustness :

```
1/1 [==============================] - 0s 94ms/step
the real person: Yahia_boukharouba-happy-2.jpg
the predicted person: Yahia
________________________________

1/1 [==============================] - 0s 23ms/step
the real person: Yahia_test1.jpg
the predicted person: Yahia
________________________________

1/1 [==============================] - 0s 31ms/step
the real person: Yahia_test2.jpg
the predicted person: Yahia
________________________________

1/1 [==============================] - 0s 32ms/step
the real person: Karim_cherrab-neutral-2.jpg
the predicted person: Karim
________________________________

1/1 [==============================] - 0s 34ms/step
the real person: Celia_Lazili-neutral-3.jpg
the predicted person: Celia
________________________________

1/1 [==============================] - 0s 36ms/step
the real person: Ahmed_Djellouli-neutral-1.jpg
the predicted person: Ahmed
________________________________

1/1 [==============================] - 0s 22ms/step
the real person: Rabeh_Aouar-happy-1.jpg
the predicted person: Rabeh
________________________________

1/1 [==============================] - 0s 30ms/step
the real person: Assia_Madani-happy-1.jpg
the predicted person: Assia
________________________________
```

- **Classification report :**

	precision	recall	f1-score	support
Assia	1.00	1.00	1.00	3
Yahia	1.00	1.00	1.00	1
Amine	1.00	1.00	1.00	1
Rabeh	1.00	1.00	1.00	1
Ilyas	1.00	1.00	1.00	1
abdelmalek	1.00	1.00	1.00	1
Ayoub	1.00	1.00	1.00	3
Walid	1.00	1.00	1.00	1
Karim	1.00	1.00	1.00	1
Abdelali	1.00	1.00	1.00	1
Ahmed	1.00	1.00	1.00	1
Yacine	1.00	1.00	1.00	1
Houssem	1.00	1.00	1.00	1
Abdlelkader	1.00	1.00	1.00	1
Sabrina	1.00	1.00	1.00	1
Celia	1.00	1.00	1.00	1
Katia	1.00	1.00	1.00	1
Kamelia	1.00	1.00	1.00	1
accuracy			1.00	22
macro avg	1.00	1.00	1.00	22
weighted avg	1.00	1.00	1.00	22

- **Confusion matrix :**



7. Emotion recognition :

a. Data:

We first started by splitting the original dataset into a new one based on emotions, in order to create a dataset with 5 emotions : angry, happy, neutral, sad and finally tired.

b. Classifier network :

We pass the images through our classifier, which takes as input an image of size (224,224,1) with one channel (gray images). We apply convolutional layers for feature extraction, followed by a flatten layer, and finally a fully connected layer.

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 222, 222, 32)      320

 max_pooling2d (MaxPooling2D  (None, 111, 111, 32)     0
 )

 conv2d_1 (Conv2D)           (None, 109, 109, 64)      18496

 max_pooling2d_1 (MaxPooling  (None, 54, 54, 64)       0
 2D)

 batch_normalization (BatchN  (None, 54, 54, 64)       256
 ormalization)

 conv2d_2 (Conv2D)           (None, 52, 52, 128)       73856

 max_pooling2d_2 (MaxPooling  (None, 26, 26, 128)      0
 2D)

 conv2d_3 (Conv2D)           (None, 24, 24, 256)       295168

 max_pooling2d_3 (MaxPooling  (None, 12, 12, 256)      0
 2D)

 batch_normalization_1 (Batc  (None, 12, 12, 256)      1024
 hNormalization)

 conv2d_4 (Conv2D)           (None, 10, 10, 512)       1180160

 max_pooling2d_4 (MaxPooling  (None, 5, 5, 512)        0
 2D)

 conv2d_5 (Conv2D)           (None, 3, 3, 512)         2359808

 max_pooling2d_5 (MaxPooling  (None, 1, 1, 512)        0
 2D)

 batch_normalization_2 (Batc  (None, 1, 1, 512)        2048
 hNormalization)

 flatten (Flatten)           (None, 512)               0

 dropout (Dropout)           (None, 512)               0

 dense (Dense)               (None, 128)               65664

 dropout_1 (Dropout)         (None, 128)               0

 dense_1 (Dense)             (None, 64)                8256

 dropout_2 (Dropout)         (None, 64)                0

 dense_2 (Dense)             (None, 5)                 325

=================================================================
Total params: 4,005,381
Trainable params: 4,003,717
```

- **Convolution2D:** Performs 2D convolution on the input images, extracting features using small filters (3*3).
- **Batch normalization:** Normalize the activations of a neural network layer by adjusting and stabilizing their mean and variance to improve training speed and performance.
- **MaxPooling2D:** Performs max pooling to reduce the spatial dimensions and capture the most important features.
- **Dropout:** Helps prevent overfitting by randomly dropping out a fraction of the neurons during training.
- **Flatten:** Flattens the multi-dimensional feature maps into one-dimensional vector.
- **Activation:** apply Relu activation function after each batch normalization to introduce a non-linearity, enabling it to learn more complex representations.
- **Dense:** Allowing complex relationships and learning of non-linear patterns in the data.
- **Output:** Apply softmax activation function to obtain class probabilities.

The Optimizer and loss function:

We used the Adam optimizer and categorical_crossentropy as the loss function with a learning rate equal to 0.001.We also used a regularization with L2 norm to prevent overfitting and improve the generalization performance of the model.

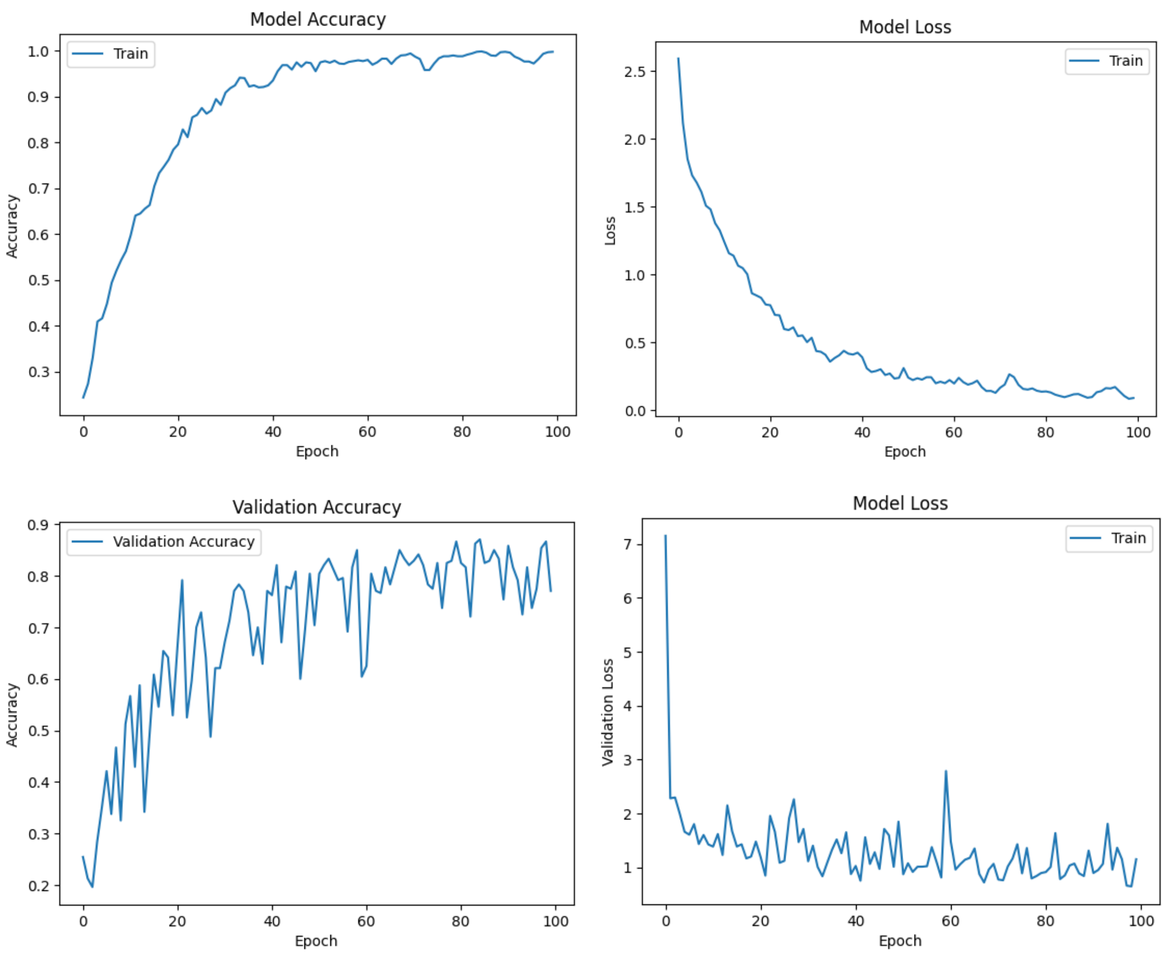
Training :

During training, we used the ModelCheckPoint from keras.callbacks to save the best model weights based on validation accuracy(30 epochs).

```
from tensorflow.keras.callbacks import ModelCheckpoint

# Create a ModelCheckpoint callback
checkpoint = ModelCheckpoint('best_weights.h5',
                             monitor='val_accuracy',
                             save_best_only=True,
                             save_weights_only=True,
                             mode='max',
                             verbose=1)
```

```
epochs = 100
history = new_model.fit(
    train_set,
    batch_size=batch_size,  # Number of samples per gradient update
    epochs=epochs,  # Number of times to iterate over the entire dataset
    validation_data=train_val,
    shuffle=True,# Shuffle the training data before each epoch
    callbacks=[checkpoint]
)
```

The evaluation of Accuracy and loss during the training :



Model Performance Analysis on Test Dataset :

Predicted Emotion: happy True Emotion: happy

Predicted Emotion: sad True Emotion: sad

Predicted Emotion: tired True Emotion: tired

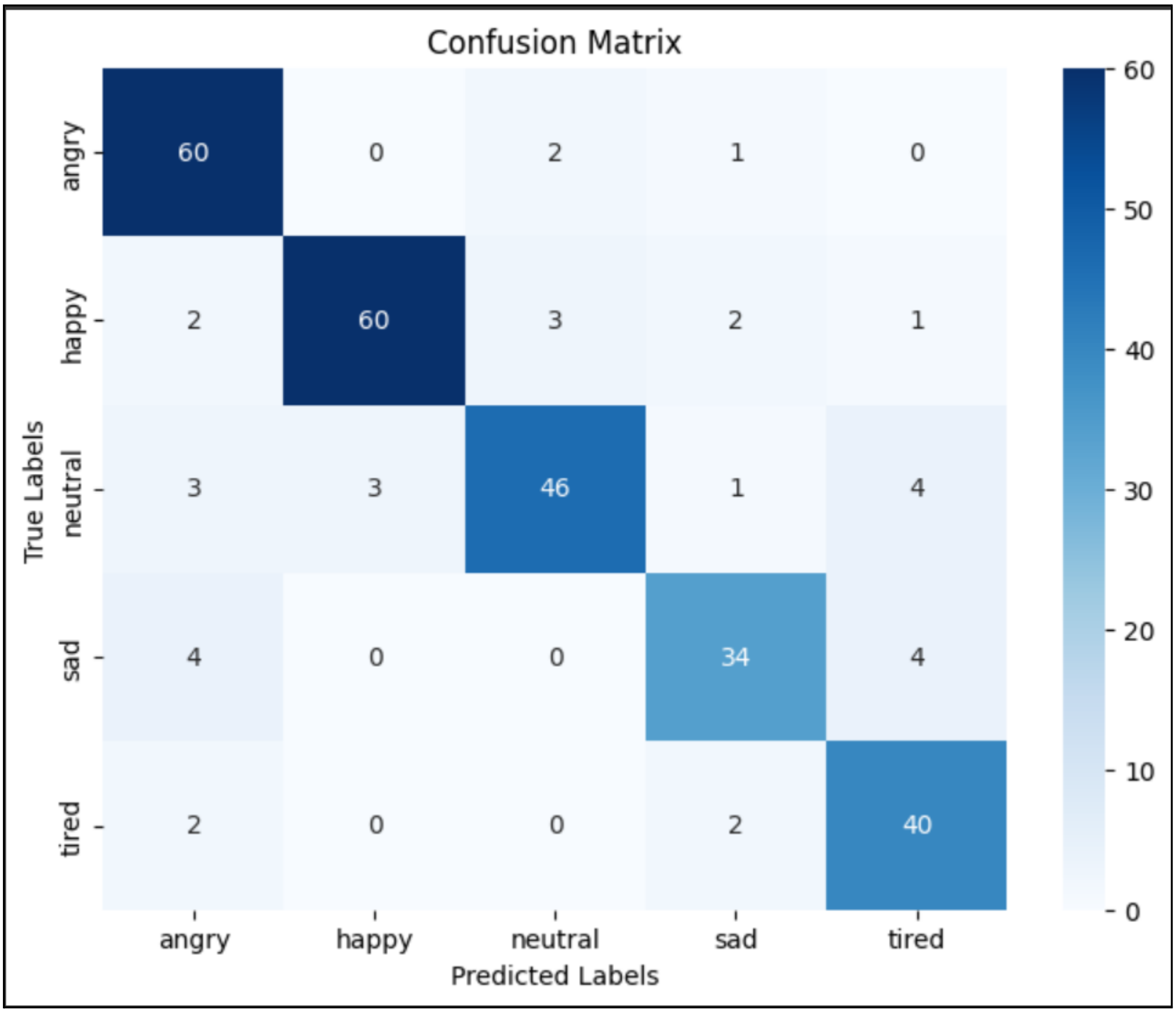
Predicted Emotion: angry True Emotion: angry

- **Classification report :**

```
from sklearn.metrics import classification_report
report = classification_report(y_true, y_predicted)
```

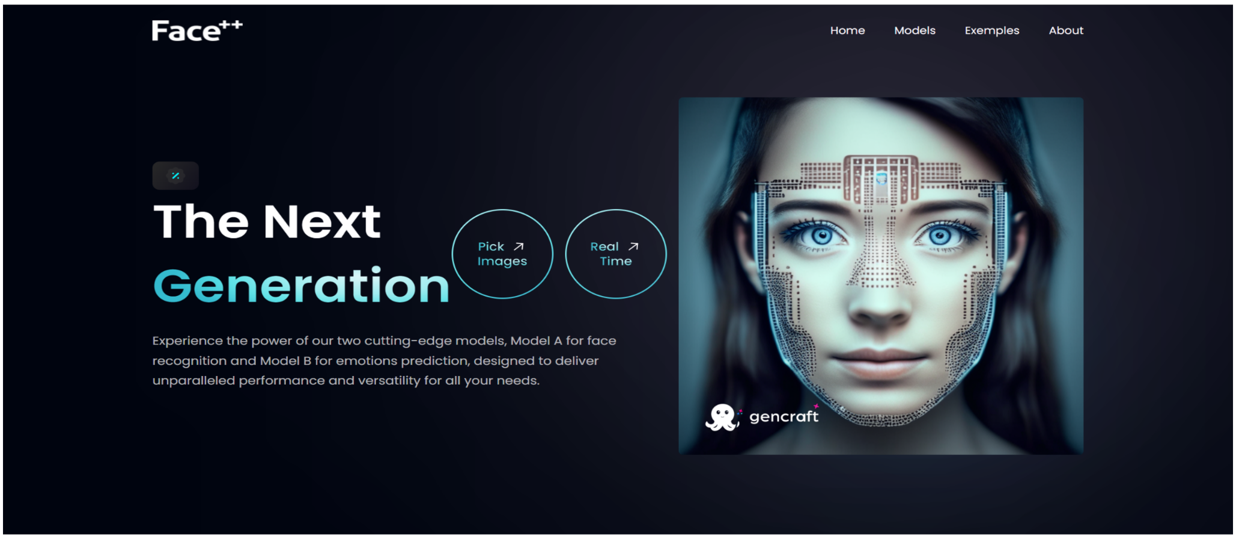
```
[18] print(report)

              precision    recall  f1-score   support

       angry       0.85      0.95      0.90        63
       happy       0.95      0.88      0.92        68
     neutral       0.90      0.81      0.85        57
         sad       0.85      0.81      0.83        42
       tired       0.82      0.91      0.86        44

    accuracy                           0.88       274
   macro avg       0.87      0.87      0.87       274
weighted avg       0.88      0.88      0.88       274
```

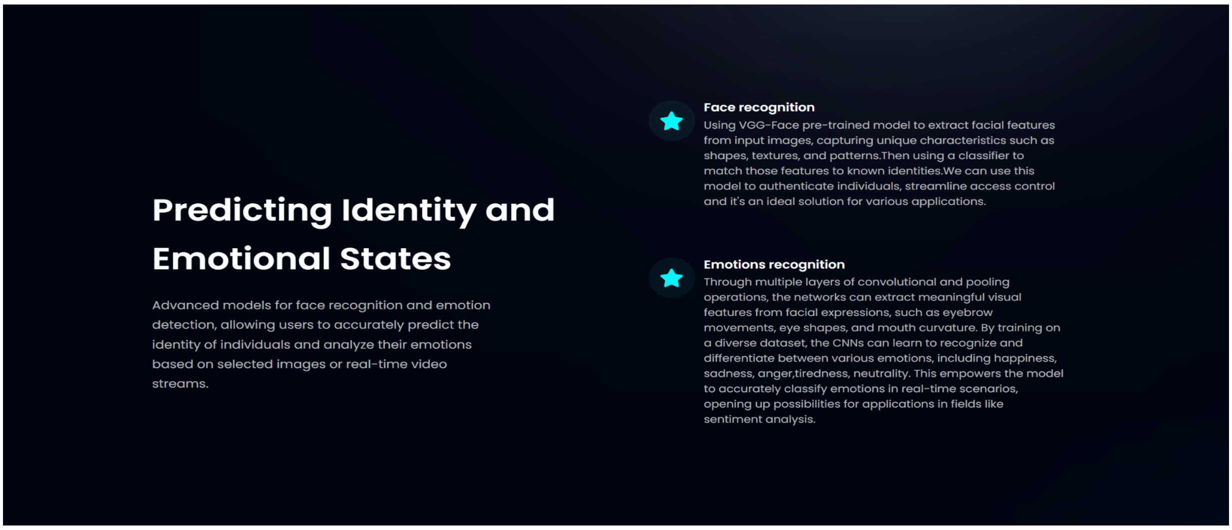
- **Confusion matrix :**



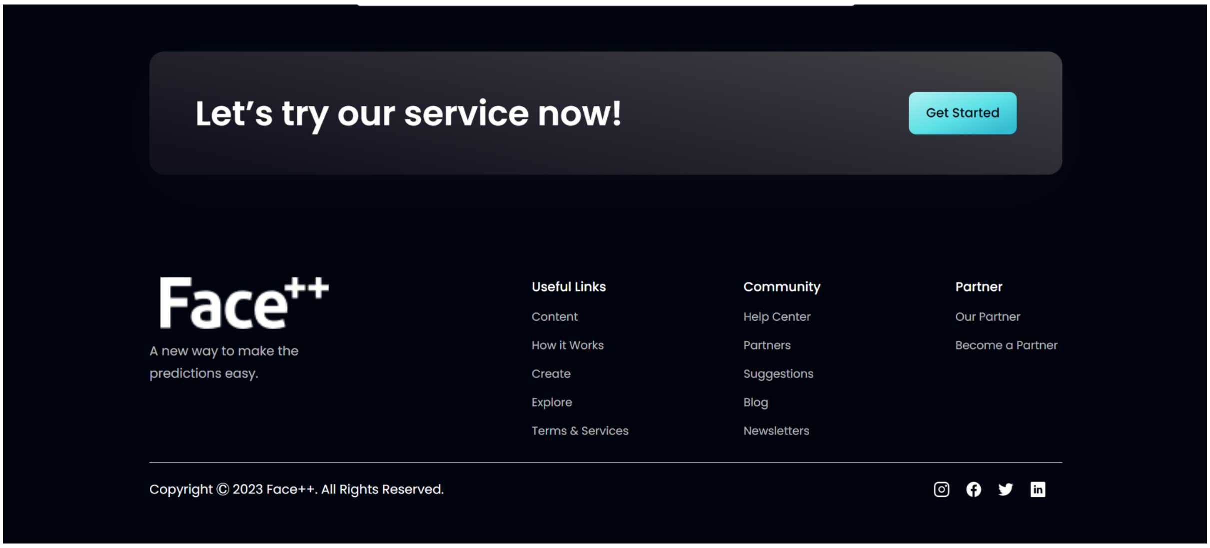
8. The web application :

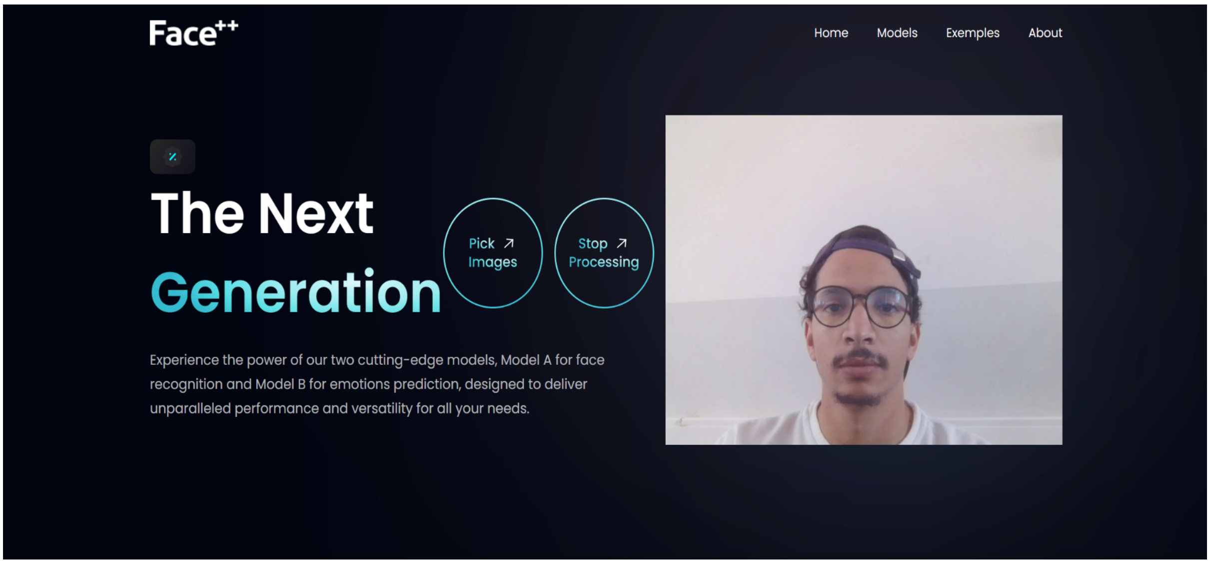
We deployed our trained models in a web application that was developed using FastAPI and ReactJS.**FastAPI** served as the backend framework, handling requests and providing the necessary endpoints for model integration.ReactJS, on the other hand, was responsible for building the frontend, providing a user-friendly interface for interacting with the application.
To enable real-time communication and instant updates between the backend and frontend, we utilized websockets. **Websockets** established a persistent connection between the two components, allowing for bidirectional communication. This facilitated seamless and immediate transmission of data, enhancing the user experience and providing real-time updates based on the model's outputs.






Who are
the royal we ?
Group of data science & artificial intelligence students at the higher school of computer science ESI-SBA
Ahmed DJELLOULI, 4th grade data & Ai student at ESI-SBA
Ahmed DJELLOULI
Founder & Leader
Sabrina BELKHODJA, 4th grade data & Ai student at ESI-SBA
Sabrina BELKHODJA
Founder & Leader
Houssem SAIDOUNE, 4th grade data & Ai student at ESI-SBA
Houssem SAIDOUNE
Founder & Leader
Assia Madani, 4th grade data & Ai student at ESI-SBA
Assia Madani
Founder & Leader
ECOLE SUPÉRIEURE EN INFORMATIQUE
8 Mai 1945 - Sidi-Bel-Abbès
ESI


Let's try our service now!
Get Started
Face++
A new way to make the predictions easy.
Useful Links
Content
How it Works
Create
Explore
Terms & Services
Community
Help Center
Partners
Suggestions
Blog
Newsletters
Partner
Our Partner
Become a Partner
Copyright © 2023 Face++. All Rights Reserved.



9. Conclusion :

In conclusion, the developed project on real-time face recognition and emotion analysis has been successfully implemented and evaluated.
This opens up numerous possibilities for enhancing user experiences, improving security systems, and enabling personalized interactions, and providing valuable insights into individuals' emotional states based on their facial expressions.