Emotion recognition

Digital forensics, Final Project

10.06.2023 | DARIA NIKOLAEVA, MALIKOV ANDREY

Introduction

The project focuses on developing a neural network for emotion recognition.

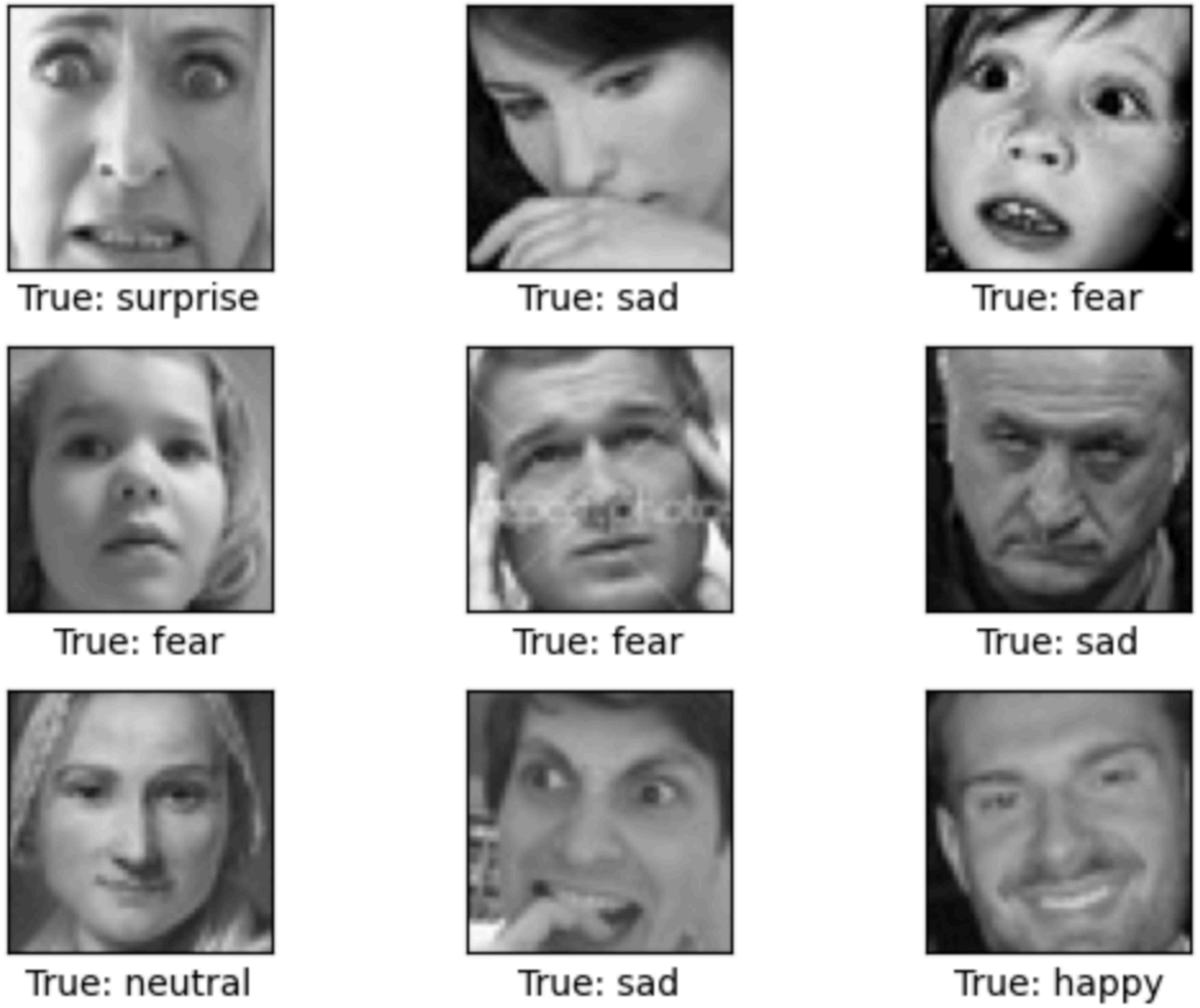
We developed a system that can accurately recognize and classify emotions from facial expressions.

Keras

dlib

Data Collection and Preprocessing

The data consists of 48x48 pixel grayscale images of faces. The training set consists of 28,709 examples and the public test set consists of 3,589 examples.

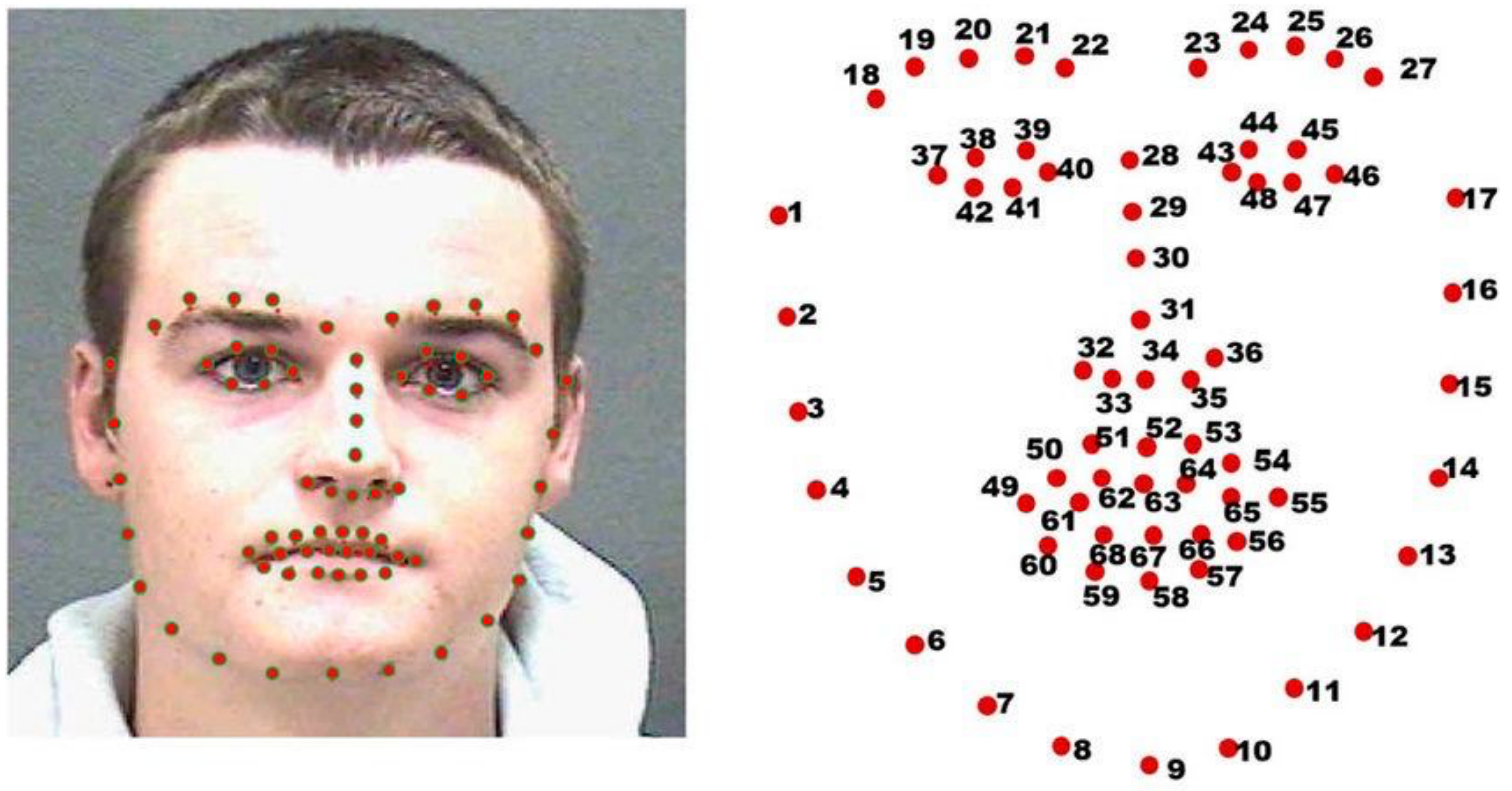


https://www.kaggle.com/datasets/msambare/fer2013

Data Collection and Preprocessing

With the help of **dlib** a face detector is created, which is able to detect the presence of faces in the image.

Then a face point predictor is created, which uses a pre-trained model. The predictor is applied to the image in the detected face area and determines the position of special face points such as eyes, nose, mouth, etc.



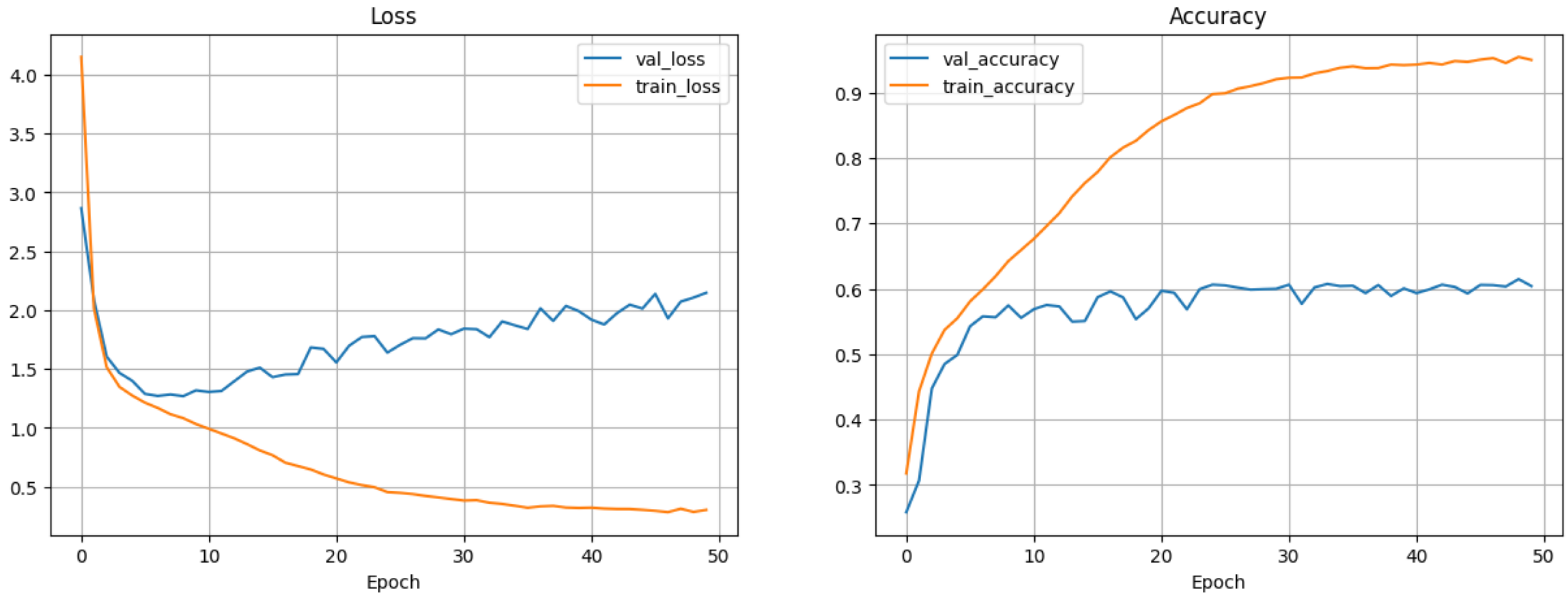
(a) (b)

Convolutional Neural Network (CNN) Architecture

```
input_images = keras.Input(shape=images_input_shape, name='images')

conv1 = layers.Conv2D(32, kernel_size=(3, 3), activation='relu')(input_images)
batch1 = layers.BatchNormalization()(conv1)

conv2 = layers.Conv2D(64, kernel_size=(3, 3), activation='relu')(batch1)
batch2 = layers.BatchNormalization()(conv2)
pool2 = layers.MaxPooling2D(pool_size=(2, 2))(batch2)
drop2 = layers.Dropout(0.25)(pool2)

conv3 = layers.Conv2D(128, kernel_size=(3, 3), activation='relu')(drop2)
batch3 = layers.BatchNormalization()(conv3)

conv4 = layers.Conv2D(128, kernel_size=(3, 3), activation='relu')(batch3)
batch4 = layers.BatchNormalization()(conv4)
pool4 = layers.MaxPooling2D(pool_size=(2, 2))(batch4)
drop4 = layers.Dropout(0.25)(pool4)

conv5 = layers.Conv2D(256, kernel_size=(3, 3), activation='relu')(drop4)
batch5 = layers.BatchNormalization()(conv5)

conv6 = layers.Conv2D(256, kernel_size=(3, 3), activation='relu')(batch5)
batch6 = layers.BatchNormalization()(conv6)
pool6 = layers.MaxPooling2D(pool_size=(2, 2))(batch6)
drop6 = layers.Dropout(0.25)(pool6)

output_images = layers.Flatten()(drop6)

input_landmarks = keras.Input(shape=landmarks_input_shape, name='landmarks')
output_landmards = layers.Flatten()(input_landmarks)

concat = keras.layers.concatenate([output_images, output_landmards])

dence7 = layers.Dense(256, activation='relu',
kernel_regularizer=keras.regularizers.l2(0.01), name='dence1')(concat)
batch7 = layers.BatchNormalization()(dence7)
drop7 = layers.Dropout(rate=0.25, name='drop4')(dence7)
output = layers.Dense(num_classes, activation='softmax', name='output')(drop7)

nn = keras.Model(inputs=[input_images,input_landmarks], outputs=output,
name='Emotion_rec_model')
nn.compile(optimizer='adam', loss='sparse_categorical_crossentropy', metrics=['accuracy'])
nn.summary()
```

Training and Evaluation

Training process over the epochs



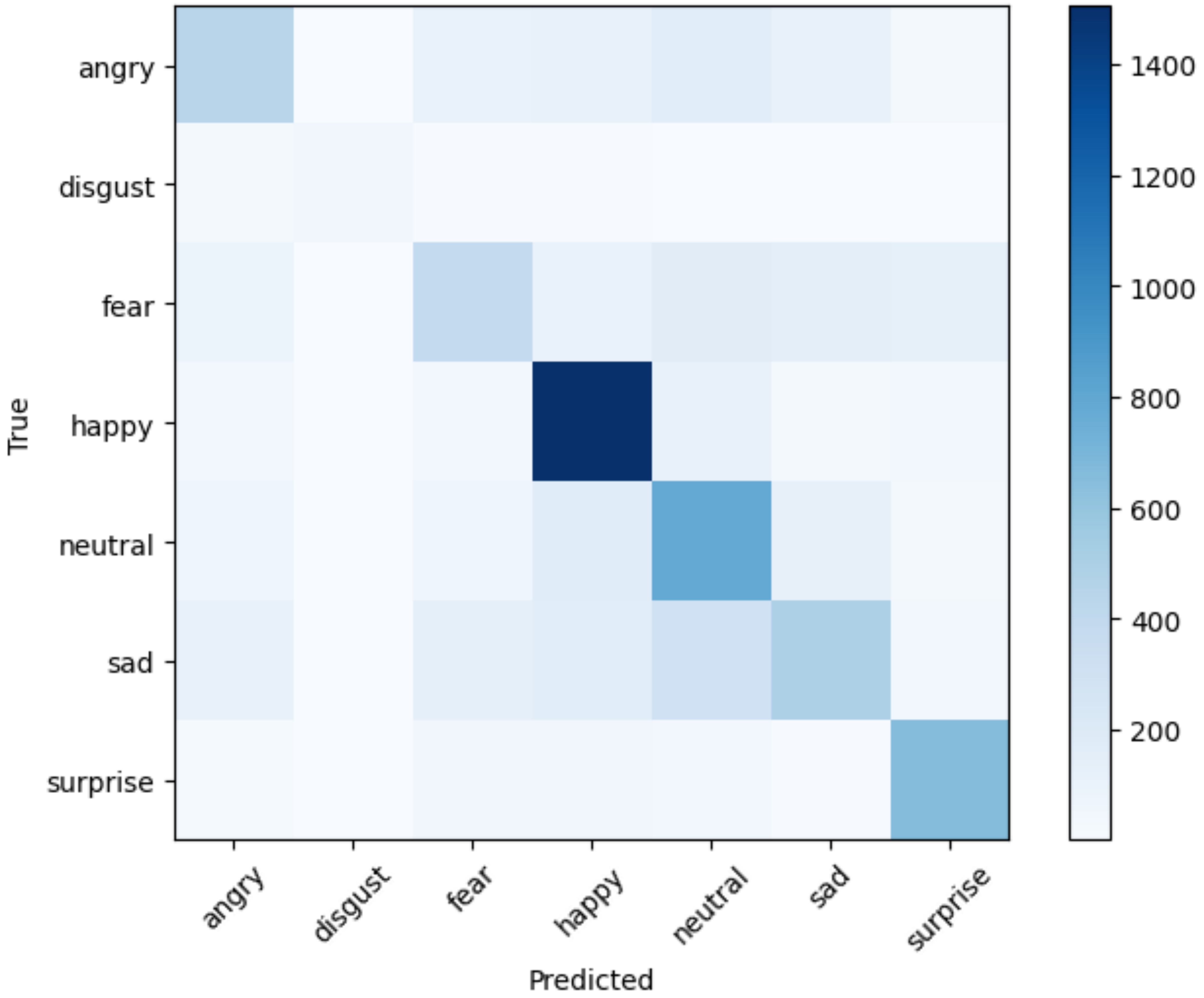
Training loss: 0.3024, training accuracy: 0.9496
Validation loss: 2.1469, validation accuracy: 0.6041
Test loss: 2.180, test accuracy: 0.6002

Training and Evaluation

Training loss: 0.3024, training accuracy: 0.9496
Validation loss: 2.1469, validation accuracy: 0.6041
Test loss: 2.180, test accuracy: 0.6002

As we can see from the confusion matrix, happiness is classified the best by the neural network and has minimal confusion. However, the model often confuses emotions such as sadness and neutrality.

Since we have a limited number of samples for "Disgust," the picture is not as clear.

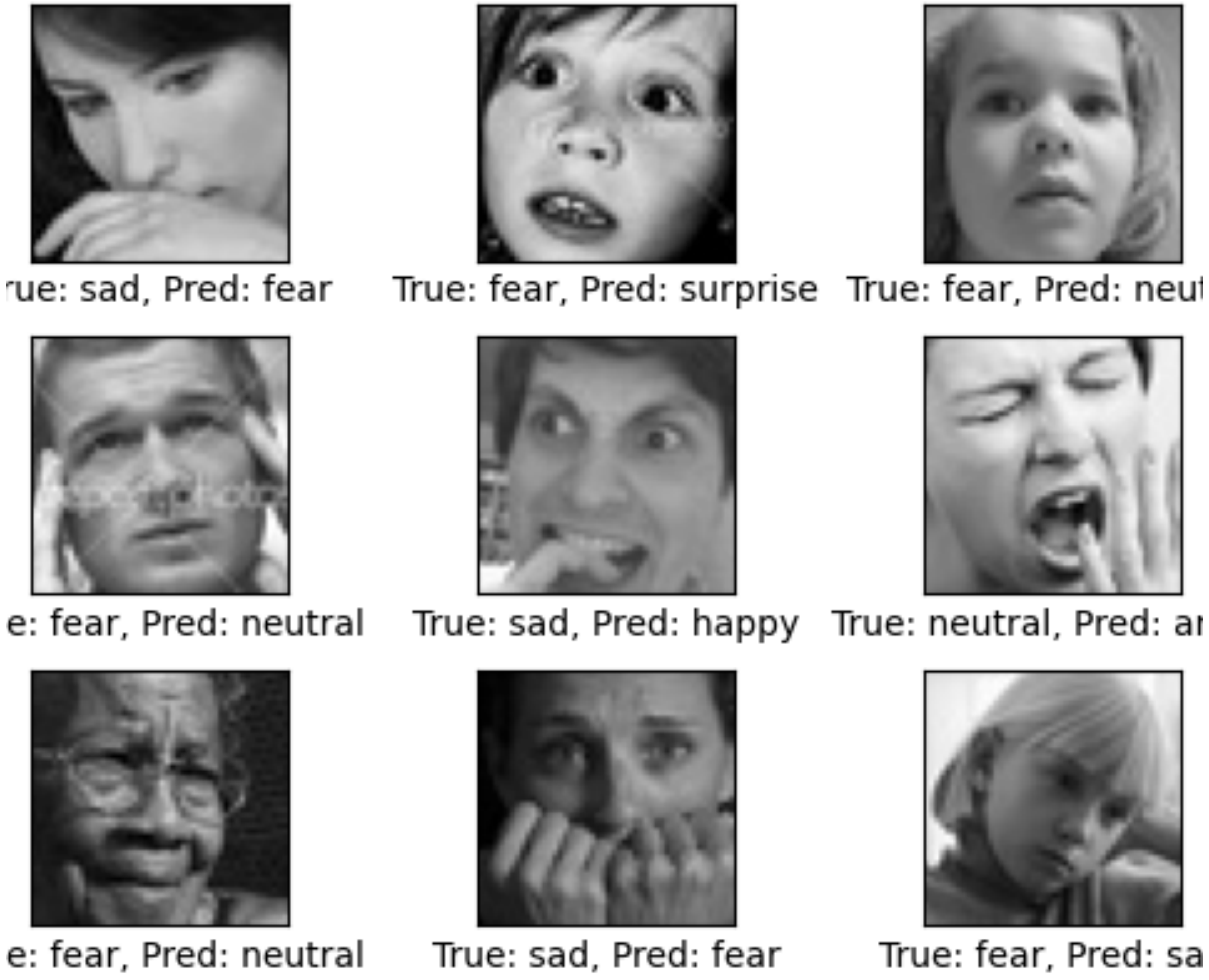


confusion matrix

Discussion and Limitations

The performance of a neural network is far from perfect because emotion recognition is a complex task. In the image, you can observe the typical classification errors. Some emotions are similar to each other, and even humans can't always distinguish them accurately.

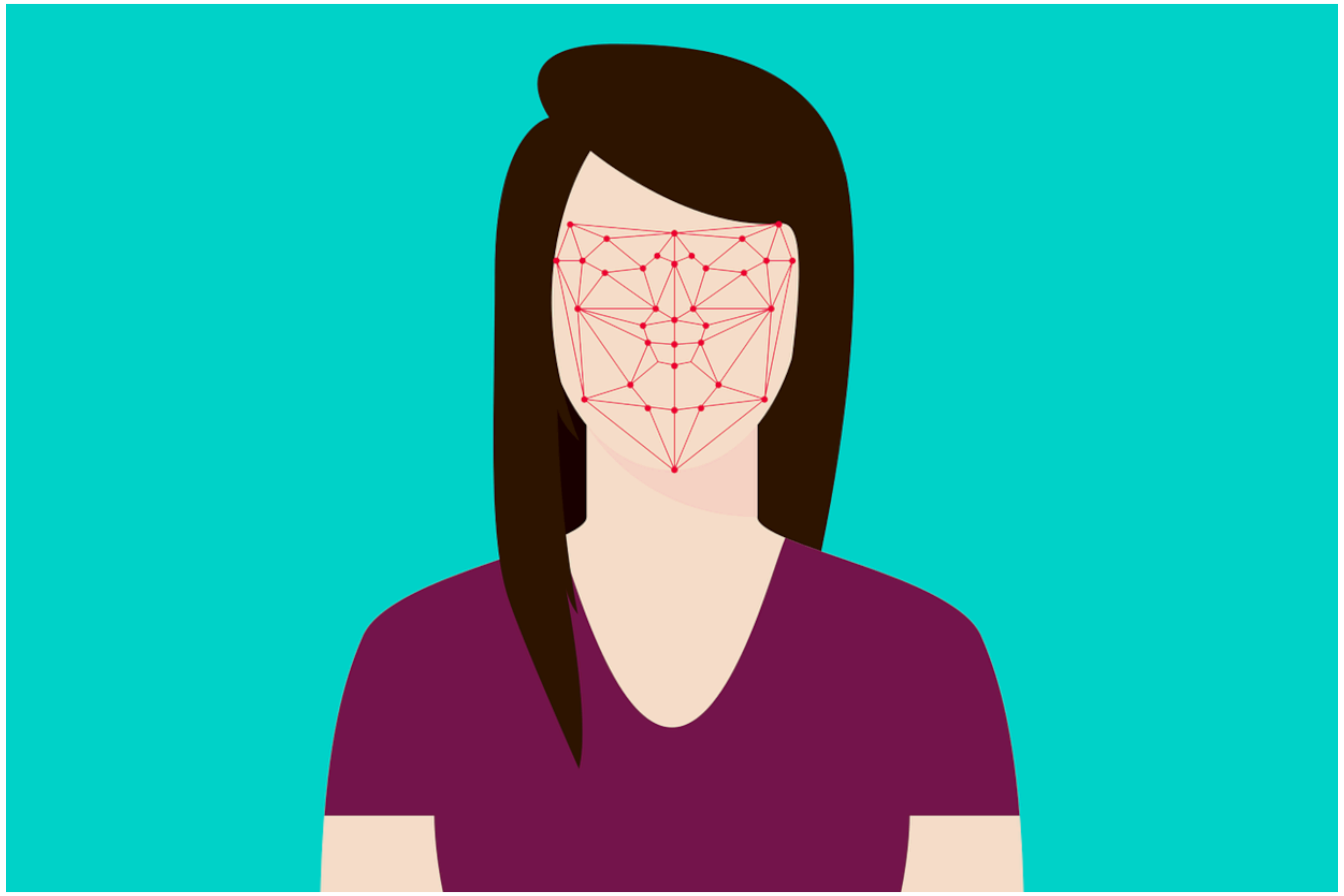
The results are also influenced by hands covering the face, as it was not possible to extract all the required landmarks.



Conclusion

In future research to improve the neural network, we need to change the dataset, make data augmentation or use a pre-trained network (e.g. alex net).

We exported the project into .h5 format and the model is now ready for further training.



Resources

List of the references and sources we consulted during the project:

- official dlib documentation http://dlib.net/python/index.html

- official keras documentation https://keras.io/

- lectures Machine Learning Pietro Zanuttigh

Thank you for your attention