Neural Network in Facial Emotion Recognition

Andrei Muravev
September, 2021

Introduction:

As one of the prominent directions in the computer vision and AI, the *facial emotion recognition* or FER plays an important role in people's daily work and life. Many companies including NEC Corporation from Japan or Google invest in FER's technologies which proves its growing acceptance. FER is based on learning and analyzing different positions of nose, eyebrows or lips from an image or video to determine the psychological emotions of the object.

Facial emotions can be divided into seven universal categories: happy, fearful, angry, surprised, disgusted, sad and neutral.

In my project, I will be teaching computers to recognize these 7 emotions from grayscale images using neural network models, perform different image operations, experiment with several deep learning models: Resnet50, VGG16 and VGG19. And, finally learn the Neural Style Transfer - technique that combines two images together to get a new one.

Part 1. Image Operations.

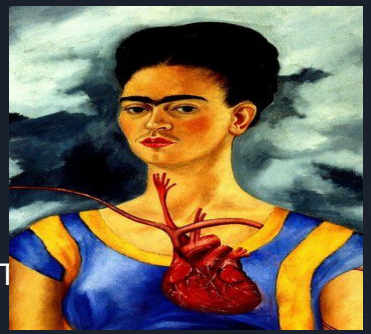
01 Importing images with Glob() and Path()

02 OpenCV Python Image Library Matplotlib

03 Cascade Classifier

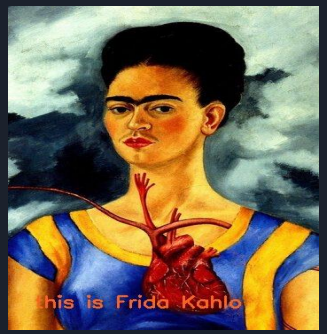
04 Preprocessing

Blurred Raphael



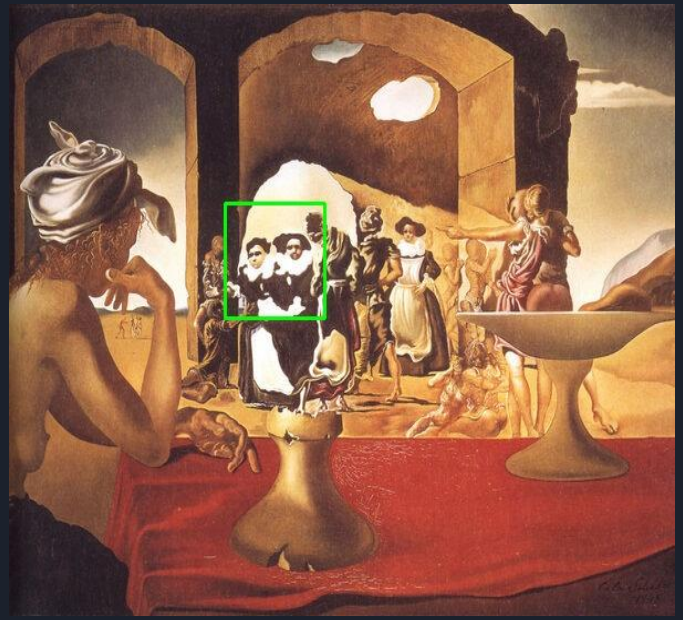
text on

Frida Kahlo



Gray Rembrandt

CASCADE CLASSIFIER
in
Recognizing Faces from Salvador Dali

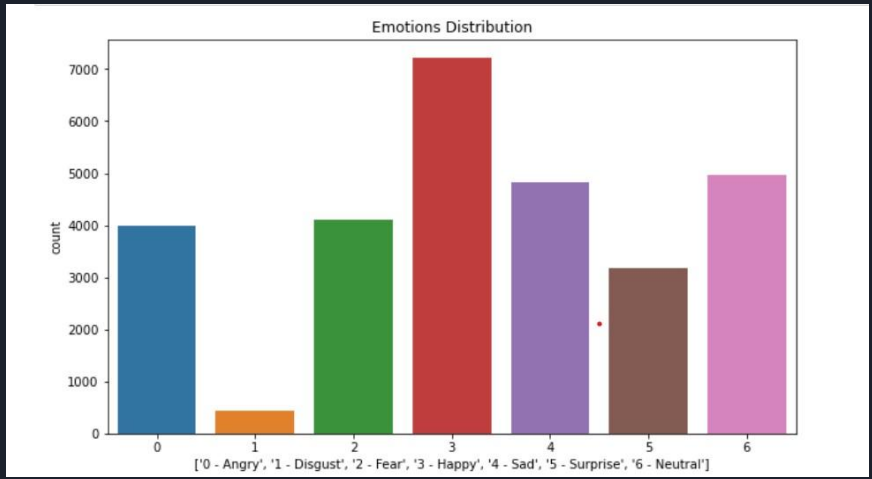


Part 2. Modeling.

Convolutional Network

Resnet50

VGG16



Modeling Process:

- 3 Convolutional and 3 Dense layers
- Batch Normalization for Convolutional layers
- Dropout for Dense layers
- MaxPooling
- Optimizer Adam with lower learning rates
- Call Back list:
  EarlyStopping() and ReduceLROnPlateau()
- Image Augmentation

```
Total params: 1,275,783
Trainable params: 1,275,335
Non-trainable params: 448
```

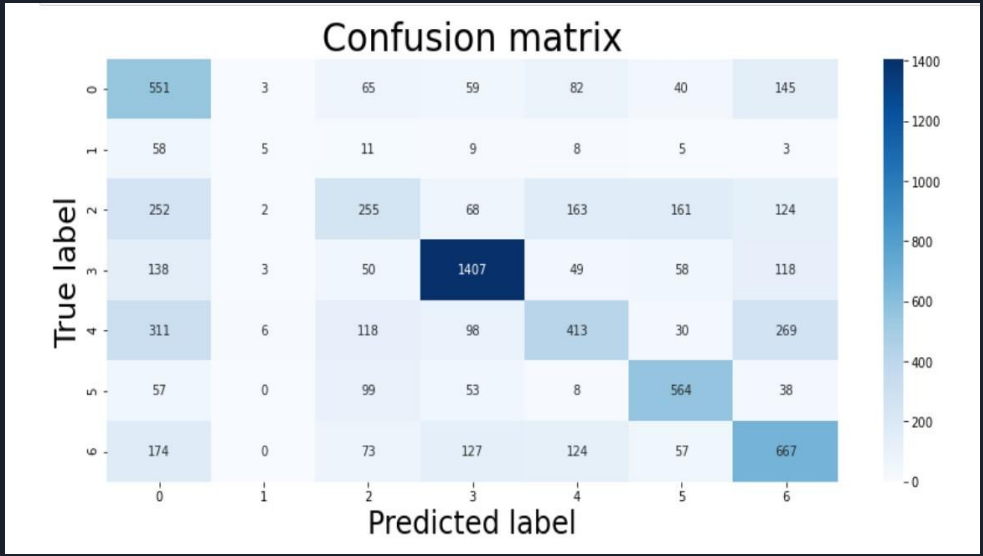
```
Total params: 27,786,119
Trainable params: 4,198,407
Non-trainable params: 23,587,712
```

```
# Instantiate the model

cnn = Sequential()

cnn.add(Conv2D(32, kernel_size = (3,3), activation ='relu', input_shape = (48,48,1)))
cnn.add(BatchNormalization())
cnn.add(MaxPool2D(pool_size=(2,2)))


cnn.add(Conv2D(64, kernel_size = (3,3), activation ='relu'))
cnn.add(BatchNormalization())
cnn.add(MaxPool2D(pool_size=(2,2)))


cnn.add(Conv2D(128, kernel_size = (3,3), activation ='relu'))
cnn.add(BatchNormalization())
cnn.add(MaxPool2D(pool_size=(2,2)))


cnn.add(Flatten())

cnn.add(Dense(512, activation = 'relu'))
cnn.add(Dropout(0.3))
cnn.add(Dense(256, activation = 'relu'))
cnn.add(Dropout(0.3))
cnn.add(Dense(7, activation = 'softmax'))
```

Part 3. Evaluation.

Classification Report that shows Recall, Precision, F-score and

Confusion Matrix



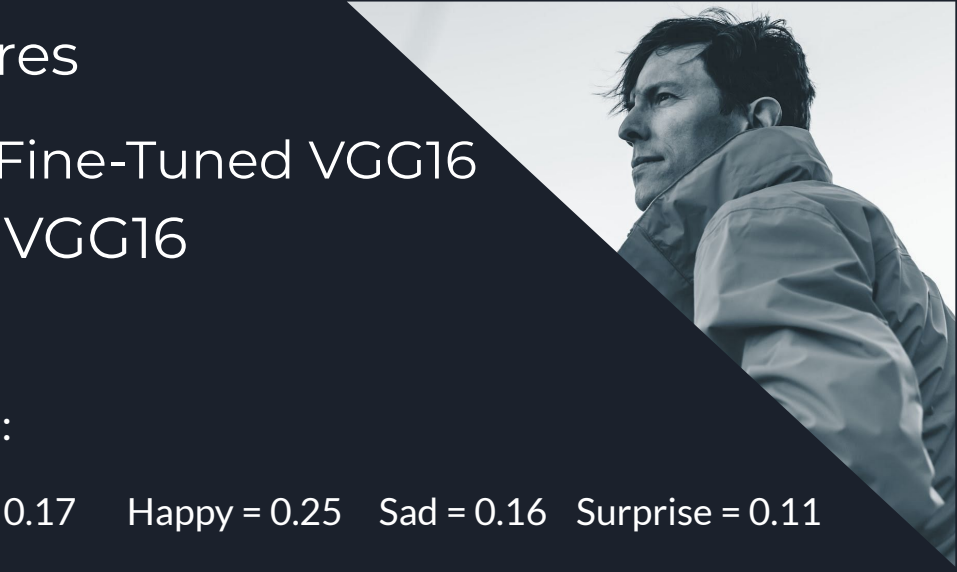
Emotion Prediction Visualization


```
['0 - Angry', '1 - Disgust',  '2 - Fear',  '3 - Happy',  '4 - Sad', '5 - Surprise', '6 - Neutral']
```

Accuracy scores

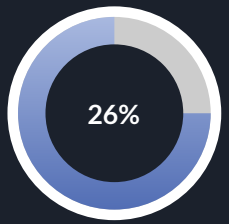
Custom Conv Net vs Fine-Tuned VGG16

Resnet50 vs VGG16

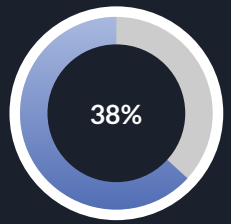


Baseline score:

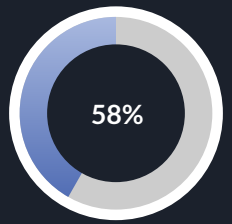
Angry = 0.13 Disgust = 0.01 Fear = 0.14 Neutral = 0.17 Happy = 0.25 Sad = 0.16 Surprise = 0.11



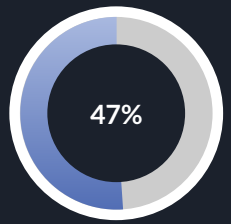
Accuracy Resnet50



Accuracy VGG16



Accuracy Custom Conv Net



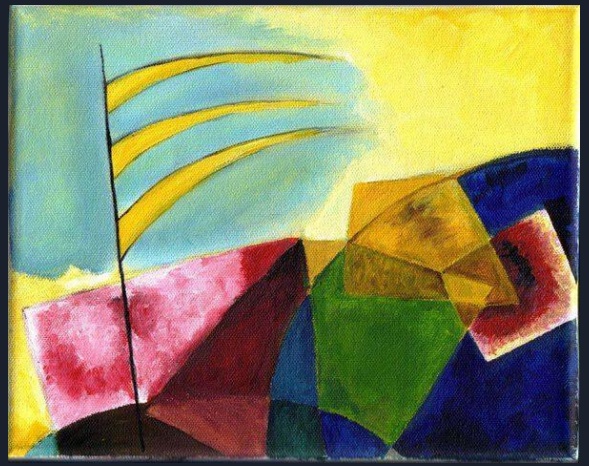
Fine-tuned VGG16

Part 4. Neural Style Transfer using VGG19

Target image

+

Style Image (Kandinskiy)



Part 5. Conclusion

Facial emotion recognition needs to be researched and high-class models to be developed every time. Many useful applications have been already implemented based on FER in people's life such as temperature detectors, health monitors, security technologies in airports and many more! I truly believe that car manufactures like Tesla, Toyota or especially electric truck companies could install systems to their vehicles to monitor driver's faces and alarm them if they are fatigue or could fall asleep to prevent car crashes. It is our future. It would be used more and more often, and I hope only for good reasons!