Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.Doi Number

Face Emotion Recognition with Image Processing and Neural Network (June 2022)

Ahmet Telçeken[1]

[1]Computer Engineering Department, University of Osmangazi, Eskişehir

Corresponding author: Ahmet Telçeken Author (e-mail: ahmettelceken16@gmail.com).

ABSTRACT People use facial expressions to show emotional states. However, facial expression recognition remained a challenging and interesting problem in computer vision. This project proposes a new facial emotional recognition model using a convolutional neural network. Therefore a convolutional neural network based solution combined with image processing is used in classification of universal emotions: Anger, Sadness, Happiness, Disgust, Fear and Surprise. Frontal face images are given as input to the system. To complete the training of the CNN network model, I use the CK+ (Extended Cohn-Kanade dataset) CNN consists of three layers of convolution together with fully connected layers. The features extracted by the HOG(Histogram of Oriented Gradients), Convolutional Neural network (CNN) from facial expressions images were fused to develop the classifcation model through training by our proposed CNN model. And then I use more advanced CNN network model with FER2013 dataset.

INDEX TERMS Emotion Recognition, Deep learning, Neural Network Convolutional Neural Network (CNN), HOG , Object Detection, CK+ Dataset, FER2013 Dataset

I. INTRODUCTION

An emotion is a mental and physiological state that is subjective and private; includes many behaviors, actions, thoughts and feelings.

Research on emotions can be traced back to the book 'The Expression of the Emotions in Man and Animals' by Charles Darwin[1]. He believed that emotions to be species-specific rather than culture-specific, but in 1969 after realizing the universality of emotions despite cultural differences, Ekman and Friesen classified expression of emotion as universal: happiness, sadness, anger, disgust, surprise and fear. [2-3- 4-5]

The ability to recognize emotions can be valuable in face recognition applications. Suspect detection systems and intelligence improvement systems are some other benefits. [6].

Haar wavelet transform[7], Gabor wavelet transform[8], Local Binary Pattern (LBP), and Active Presence Models (AAM)[9] are feature extraction methods based on static images. Whereas dynamic-based approaches assume the temporal association in the sequence of input facial expression within clinging frames. Hidden Markov Model, Support Vector Machine (SVM), AdaBoost, and Artificial Neural Networks are commonly used schemes for facial expression recognition. An important advancement in the field of deep learning and implementation of CNN[10-11-12] has quite promising. However, a big issue with the using deep learning is that a large amounts of data are need to learn successful model.

While some improvement were made in the identification of facial expression with the CNN algorithm, some breaks are still exist, including too long training times and low recognizing rates in the complex environment.

In databases, two difficulties were observed in deep learning achievements in FER methods: (1) a low number of images, and (2) images from heavily structured conditions. These concern led to the the creation of FER techniques focused on the set of Web images.[13-14-15]

II. Background Review

Recent work on the study can be broadly divided into three categories: Face detection, Facial feature extraction and Emotion classification. The number of studies carried out in each of these categories is quite large and remarkable.

A. Face Detection

Given an image, detecting the presence of a human face due to possible variations of the face is a complex task. The different sizes, angles and poses a human face might have within the image can cause this variation. The emotions which are understandable from human face and different imaging conditions such as illumination also affect facial appearances. In addition, the presence of glasses, hair, beard and makeup have a considerable effect in the facial appearance.[17]

There are many different methods for face detection, such as color space method, Haar Cascade classifiers method, using dlib libraries.[16]

B. Face Feature Extraction

The next step after face detection is removal of facial features. The permanent features of face such as eyes, nose, mouth and facial lines are detected. Feature descriptors are used to extract face features.

The features descriptor is a representation of an image created by receiving useful information and eliminating unnecessary information, thus it simplifies the image.

Typically, feature descriptor converts an image of size height, width and 3 (channels) into a feature vector or array of length n.

Histogram of Oriented Gradient (HOG), focuses on the structure or shape of an object. When it comes to edge properties, only determines whether the pixel is an edge. And this is done by subtracting gradient and direction of the edges.[16]

C. Emotion Classification

Extracted feature points are process to obtain inputs for neural network. Neural network has being trained so that the emotions happiness, sadness, anger, disgust, surprise and fear are recognized. In this project, I used the Convolutional Neural Networks (CNN) method.[18]

In deep learning, a convolutional neural network (CNN) is a class of deep neural networks most widely applied to analyze visual images. It has applications in image and video recognition, image classification, medical image analysis, financial time series, and natural language processing.

Convolution Layer extracts the features of an input image from the training dataset where the core is a small part of an input image for feature classification. The design begins with starting CNN model by taking an input image (static or dynamic) created by adding a convolution layer, flattening layer, merge layers, and dense layers. Convolution layers are added for better accuracy in large data sets.[16]

III. Emotion Recognition with Convolutional Neural Network

A simple CNN template with several building blocks that we can easily understand and associate with the proposed CNN model. Three types of layers make up a basic CNN, input, hidden, and output. The data enters the CNN via the input layer and then travels through many hidden levels before reaching the output layer. The network's prediction is reflected via the output layer. In terms of loss or error, the network's output is compared to the actual labels.

Hidden layers in the network act as a basic building element for data transformation. The four sub-functions: layer function, Pooling, Normalization and Activation can be dissected from each layer. Convolutional neural network architecture consists of the following layers. [16]

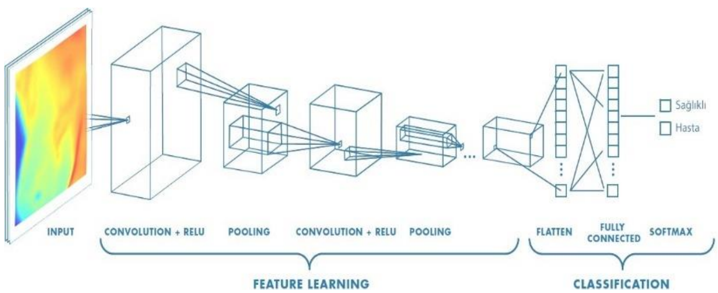

FIGURE 1.1. Understanding of Convolutional Neural Network (CNN)

- Convolutional Layer – Used to detect features
- Non-Linearity Layer – Introducing non-linearity to the system
- Pooling (Downsampling) Layer – Reduces the number of weights and controls overfitting
- Convolutional Layer – Used to detect features
- Fully-Connected Layer – Standard Neural Network used for classification[20]

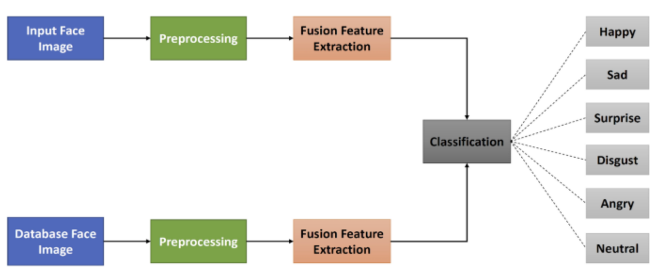

FIGURE 1.2. Facial Emotion Recognition (FER)

A. CK+ Dataset with HOG

CK + image set consisting of 981 facial images was used to recognize emotions from facial expression. First, the data set was created using the data set obtained from the CK + picture set. And this dataset has been trained through the model which we created by convolutional neural networks (CNN). The success rate of the model was calculated by entering test data into the trained model. Since the data in the complexity matrix is 20% test data, it will be 981 * 0.2 = 197, so 197 data labels were obtained.

With the python programs I use, HOG, as depicted in Fig. 2.1 of each picture in the original CK+ picture set were found.

```python
def Create_Hog_features(data):
    Feature_data = np.zeros((len(data),48,48))

    for i in range(len(data)):
        img = data[i]
        resized_img = resize(img, (128, 64))
        fd, hog_image = hog(
            resized_img,
            orientations=9,
            pixels_per_cell=(8, 8),
            cells_per_block=(2, 2),
            visualize=True,
            multichannel=True
        )
        Feature_data[i] = resize(hog_image, (48, 48))
    return Feature_data
```

FIGURE 2.1. Creating Hog Feature with Python

The data set created from the pictures obtained was applied to our CNN model is shown in figure 2.2.

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 48, 48, 6)         156

 max_pooling2d (MaxPooling2D  (None, 24, 24, 6)        0
 )

 conv2d_1 (Conv2D)           (None, 24, 24, 16)        2416

 max_pooling2d_1 (MaxPooling  (None, 12, 12, 16)       0
 2D)

 conv2d_2 (Conv2D)           (None, 10, 10, 64)        9280

 max_pooling2d_2 (MaxPooling  (None, 5, 5, 64)         0
 2D)

 conv2d_3 (Conv2D)           (None, 3, 3, 128)         73856

 max_pooling2d_3 (MaxPooling  (None, 1, 1, 128)        0
 2D)

 flatten (Flatten)           (None, 128)               0

 dense (Dense)               (None, 256)               33024

 dropout (Dropout)           (None, 256)               0

 dense_1 (Dense)             (None, 7)                 1799

=================================================================
Total params: 120,531
Trainable params: 120,531
Non-trainable params: 0
_________________________________________________________________
```

FIGURE 2.3. Cnn Model for CK+ dataset

The success rates of the facial expression recognition models created by using HOG and Original picture by CNN model and the results are given in Fig. 2.3

CNN model with different Features	Training accuracy score	Test accuracy score	Training Time(second)
Hog images features	%100	%98.98	122.935
Original images features	%100	%98.48	80.242

FIGURE 2.3. Accuracy Score comparison chart

Then, to test the accuracy of the model, I tested it with the pictures which is uploaded from the outside. I used the Cascade Classifier to find the frontal faces of the uploaded images. When the face detected is show in figure 2.4

FIGURE 2.4. Detected Face from image

I cropped the frontal face images for the model to use and I converted to Hog image is show in figure 2.5

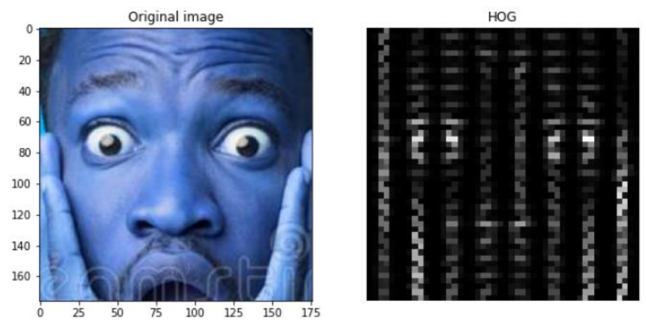

FIGURE 2.5. Cropped original image and Hog

I prepared the shape of the image for the prediction of the Cnn model is given is Figure 2.6.

```
Original Image : (176, 176, 3)
after gray:  (176, 176)
(1, 48, 48)
-----
Hog_image : (128, 64)
after gray:  (176, 176)
(1, 48, 48)
```

FIGURE 2.6. Shape of Images

And finally when I sent the prepared pictures to model to predict, I got different results from both.

Cnn trained with Hog image gave "Happy" result, Cnn trained with Normal cropped image gave "Suprise" result is shown in figure 2.7.

```
Prediction_Hog = HOG_model.predict(final_image_hog_image)
Prediction_Normal = normal_model.predict(final_image_face_roi)
```

```
print(Prediction_Hog[0])
print(Prediction_Normal[0])
```

```
[0. 0. 0. 1. 0. 0. 0.]
[0. 0. 0. 0. 0. 0. 1.]
```

```
print("Hog image = ",emotion(np.argmax(Prediction_Hog)))
```

```
Hog image =  Happy
```

```
print("Normal image = ",emotion(np.argmax(Prediction_Normal)))
```

```
Normal image =  Suprise
```

FIGURE 2.7. Predicted Emotion from Hog and Cropped Original images

B. FER2013 Dataset with different CNN

To complete the training of the CNN network model, I use the FER2013 databases and I create more advanced CNN.

Fer2013 contains approximately 30,000 facial RGB images of different expressions with size restricted to 48×48, and the main labels of it can be divided into 7 types: 0=Angry, 1=Disgust, 2=Fear, 3=Happy, 4=Sad, 5=Surprise, 6=Neutral. The Disgust expression has the minimal number of images – 600, while other labels have nearly 5,000 samples each.[21]

Unlike the Cnn we used in the CK+ dataset, I used a more advanced Cnn in the Fer2013 dataset is shown in figure 3.1.

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 48, 48, 64)        640

 conv2d_1 (Conv2D)           (None, 48, 48, 64)        36928

 dropout (Dropout)           (None, 48, 48, 64)        0

 conv2d_2 (Conv2D)           (None, 48, 48, 128)       73856

 conv2d_3 (Conv2D)           (None, 48, 48, 128)       147584

 max_pooling2d (MaxPooling2D  (None, 24, 24, 128)      0
 )

 dropout_1 (Dropout)         (None, 24, 24, 128)       0

 conv2d_4 (Conv2D)           (None, 24, 24, 256)       295168

 conv2d_5 (Conv2D)           (None, 24, 24, 256)       590080

 max_pooling2d_1 (MaxPooling  (None, 12, 12, 256)      0
 2D)

 dropout_2 (Dropout)         (None, 12, 12, 256)       0

 conv2d_6 (Conv2D)           (None, 12, 12, 512)       1180160

 conv2d_7 (Conv2D)           (None, 12, 12, 512)       2359808

 max_pooling2d_2 (MaxPooling  (None, 6, 6, 512)        0
 2D)

 dropout_3 (Dropout)         (None, 6, 6, 512)         0

 flatten (Flatten)           (None, 18432)             0

 dense (Dense)               (None, 1024)              18875392

 dropout_4 (Dropout)         (None, 1024)              0

 dense_1 (Dense)             (None, 7)                 7175

=================================================================
Total params: 23,566,791
Trainable params: 23,566,791
Non-trainable params: 0
_________________________________________________________________
```

FIGURE 3.1. Cnn Model for Fer2013 Dataset

The success rates of the facial expression recognition models created by using FER2013 dataset and Original picture by CNN model and the results are given in Fig. 3.2

Cnn Model	Training Accuracy Score	Testing Accuracy Score	Training Time(Second)
Original Cropped Images	0.8175	0.6914	3569.25

FIGURE 3.2. Accuracy Score for Cnn Model

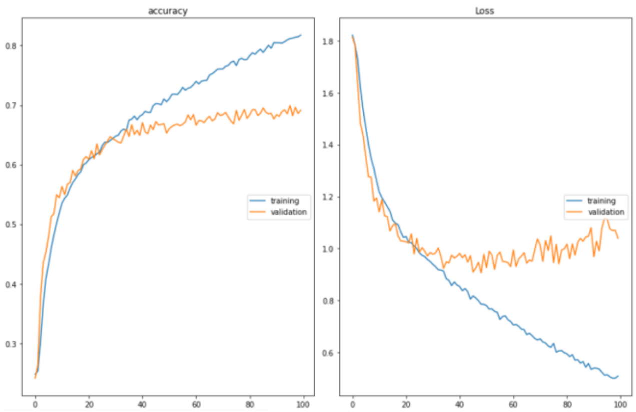

FIGURE 3.3. **Plotloss with PlotLossesKeras**

When we created a CNN model original pictures, it was found by using the confusion matrix how accurately 7 facial emotion expressions were predicted from training pictures and test pictures and the results are given in Fig. 3.4 and Fig.3.5 comparatively.

```
Confusion Matrix
[[ 550   62  555 1049  684  637  458]
 [  55    8   65  113   72   80   43]
 [ 537   67  562 1042  695  719  475]
 [ 929  111  966 1939 1323 1172  775]
 [ 669   78  658 1247  905  859  549]
 [ 670   55  669 1194  898  823  521]
 [ 406   45  424  839  559  555  343]]
Classification Report
              precision    recall  f1-score   support

       angry       0.14      0.14      0.14      3995
     disgust       0.02      0.02      0.02       436
        fear       0.14      0.14      0.14      4097
       happy       0.26      0.27      0.26      7215
     neutral       0.18      0.18      0.18      4965
         sad       0.17      0.17      0.17      4830
    surprise       0.11      0.11      0.11      3171

    accuracy                           0.18     28709
   macro avg       0.15      0.15      0.15     28709
weighted avg       0.18      0.18      0.18     28709
```

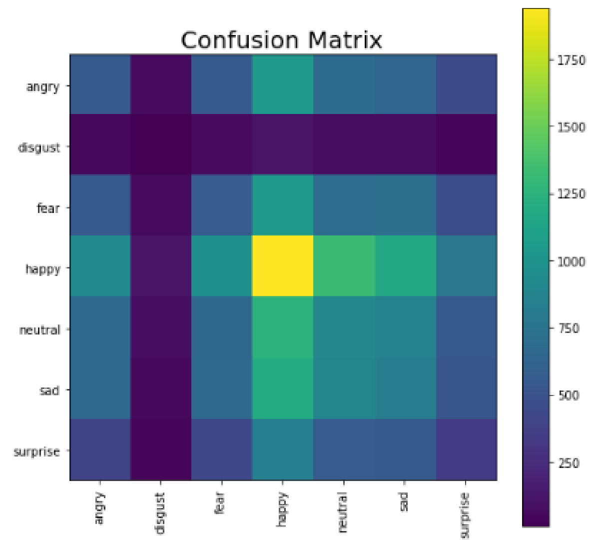

FIGURE 3.4. **Confusion Matrix and Classification on training set**

```
Confusion Matrix
[[114  14 114 224 207 184 101]
 [ 12   1  12  31  21  20  14]
 [150  11 160 233 189 154 127]
 [232  25 236 448 328 285 220]
 [176  11 145 325 243 199 134]
 [151  13 175 327 238 207 136]
 [ 96  10 104 229 158 125 109]]
Classification Report
              precision    recall  f1-score   support

       angry       0.12      0.12      0.12       958
     disgust       0.01      0.01      0.01       111
        fear       0.17      0.16      0.16      1024
       happy       0.25      0.25      0.25      1774
     neutral       0.18      0.20      0.19      1233
         sad       0.18      0.17      0.17      1247
    surprise       0.13      0.13      0.13       831

    accuracy                           0.18      7178
   macro avg       0.15      0.15      0.15      7178
weighted avg       0.18      0.18      0.18      7178
```

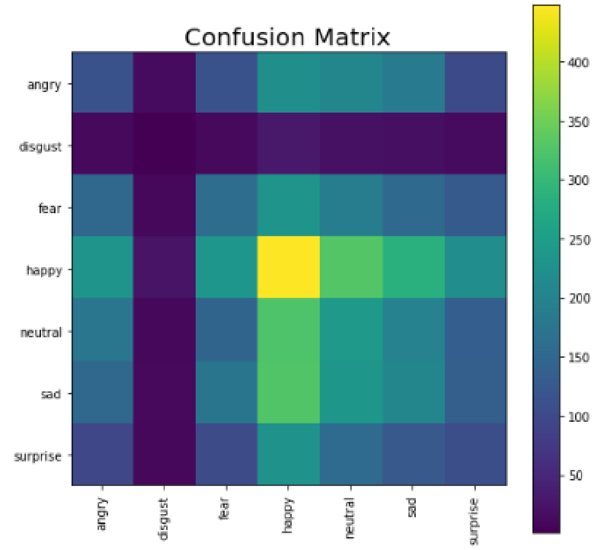

FIGURE 3.5. **Confusion Matrix and Classification on test set**

Then, I uploaded pictures from the outside and Cascade classifier detected frontal faces of the cropped images and I prepared the shape of the cropped images for the prediction of the Cnn model like I did on the Ck+dataset.

FIGURE 3.4. Original Image

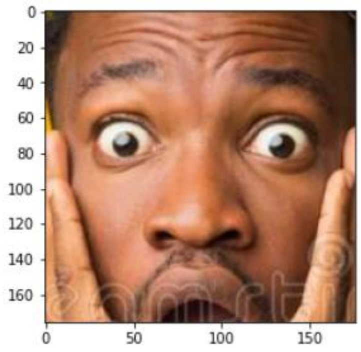

FIGURE 3.5. Cropped Image with HaarCascade

And finally I sent the prepared pictures to model, Cnn trained with Normal cropped images gave "Happy" result is shown in figure 3.6.

```
Prediction_Fer2013 = saved_model.predict(final_image)

Prediction_Fer2013[0]

array([0., 0., 1., 0., 0., 0., 0.], dtype=float32)

emotion(np.argmax(Prediction_Fer2013))

'Fear'
```

FIGURE 3.6. Predicted Emotion with advanced CNN

IV. Conclusion

HOG and CNN features based on the CK+ dataset, HOG technique was train accuracy (100%), just original image I mean without Hog technique found same train accuracy (100%).

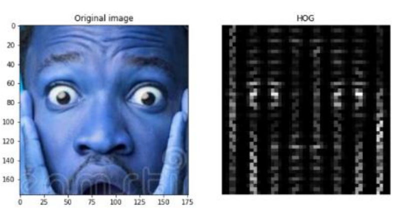

FIGURE 4.1. Test Image (Must be fear or suprise)

Despite this high accuracy, when I predicted images from the outside, I found different results in the two techniques is shown in figure 4.2.

```
print("Hog image = ",emotion(np.argmax(Prediction_Hog)))

Hog image =  Sad

print("Normal image = ",emotion(np.argmax(Prediction_Normal)))

Normal image =  Happy
```

FIGURE 4.2. Different Results HOG and Original image

Then I used a more advanced CNN model and replaced the dataset with FER2013, which contains about 35 thousand photos.

The model took a very long time to fit, but it correctly predicted a images that had never seen before is shown in figure 4.3.

```
emotion(np.argmax(Prediction_Fer2013))

'Fear'
```

FIGURE 4.3. Predicted Emotion with advanced CNN

V. References

1) Gu, H., Su, G., and Du, C., Feature points extraction from faces. 2003.

2) C. Busso, Z. Deng, S. Yildirim, M. Bulut, C. M. Lee, A. Kazemzadeh, S. Lee, U. Neumann,and S. Narayanan. Analysis of emotion recognition using facial expressions, speech and multimodal information. In ICMI '04: Proceedings of the 6th international conference on Multimodal interfaces, pages 205–211, New York, NY, USA, 2004. ACM.

3) M. N. Dailey, G. W. Cottrell, C. Padgett, and R. Adolphs. Empath: A neural network that categorizes facial expressions. J. Cognitive Neuroscience, 14(8):1158–1173, 2002

4) P. Ekman. Facial expression and emotion. American Psychologist, 48:384–392, 1993.

5) M. Grimm, D. G. Dastidar, and K. Kroschel. Recognizing emotions in spontaneous facial expressions. 2008.

6) Sanderson C., and Paliwal, K. K., Fast feature extraction method for robust face verification. Electronics Letters, 8:1648 – 1650, 2002.

7)).Kumar, N. & Bhargava, D. A scheme of features fusion for facial expression analysis: A facial action recognition. J. Stat. Manag. Syst. 20(4), 693–701 (2017).

8) Yang, B., Xiang, X., Xu, D., Wang, X. & Yang, X. 3d palm print recognition using shape index representation and fragile bits. Multimed. Tools Appl. 76(14), 15357–15375 (2017).

9)Tzimiropoulos, G. & Pantic, M. Fast algorithms for fitting active appearance models to unconstrained images. Int. J. Comput. Vis. 122(1), 17–33 (2017).

10)Goodfellow, I. J., Erhan, D., Carrier, P. L., Courville, A., Mirza, M., Hamner, B., Cukierski, W., Tang, Y., Thaler, D., Lee, D.-H. et al. Challenges in representation learning: A report on three machine learning contests. In International Conference on Neural Information Processing 117–124 (Springer, 2013).

11)Yu, Z. & Zhang, C. Image based static facial expression recognition with multiple deep network learning. In Proceedings of the 2015 ACM on International Conference on Multimodal Interaction 435–442 (2015).

12)Niu, H. et al. Deep feature learnt by conventional deep neural network. Comput. Electr. Eng. 84, 106656 (2020).

13)Pantic, M., Valstar, M., Rademaker, R., & Maat, L. Web-based database for facial expression analysis. In 2005 IEEE International Conference on Multimedia and Expo 5 (IEEE, 2005).

14)Wang, X., Feng, X., & Peng, J. A novel facial expression database construction method based on web images. In Proceedings of the Third International Conference on Internet Multimedia Computing and Service 124–127 (2011).

15) Four-layer ConvNet to facial emotion recognition with minimal epochs and the significance of data diversity
Online : https://www.nature.com/articles/s41598-022-11173-0

16) Emotion Detection from Facial Expression Using Different Feature Descriptor Methods with Convolutional Neural Networks Online: *https://dergipark.org.tr/tr/download/article-file/1500483*

17)Facial Emotion Recegnition with a Neural Network Approach
Online:
http://www.cs.utah.edu/~widanaga/papers/Widanagamaachchi.2009.thesis.pdf

18) Emotion Recognition with Image Processing and Neural Networks
Online :
http://www.cs.utah.edu/~widanaga/papers/Widanagamaachchi.2009.Emotion.pdf

19) Automated Facial Expression Recognition Using Deep Learning Techniques: An Overview
Online : https://dergipark.org.tr/en/download/article-file/1125402

20) Introduction to Convolutional Neural Networks
Online: https://rubikscode.net/2018/02/26/introduction-to-convolutional-neural-networks/

21) FER2013 (Facial Expression Recognition 2013 Dataset)
Online : https://paperswithcode.com/dataset/fer2013

22)Object Detection Techniques (Kaggle)
Online: https://www.kaggle.com/code/infernop/object-detection-techniques

23)FER2013 - Emotional Recognition with 93% accuracy (Kaggle)
Online: https://www.kaggle.com/code/sivantm/fer2013-emotional-recognition-with-93-accuracy

24)Face Mask Detection | Xception + Haar Cascade (Kaggle)
Online : https://www.kaggle.com/code/chaitanya99/face-mask-detection-xception-haar-cascade

25)Convolutional Neural Network (ConvNet yada CNN) nedir, nasıl çalışır?
Online: https://medium.com/@tuncerergin/convolutional-neural-network-convnet-yada-cnn-nedir-nasil-calisir-97a0f5d34cad

26)Object Detection using OpenCV | Python | Tutorial for beginners 2020 (Youtube)
Online:https://www.youtube.com/watch?v=RFqvTmEFtOE&ab_channel=DeepLearning_by_PhDScholar

27)Realtime Face Emotion Recognition | Tensorflow | Transfer Learning | Python | Train your own Images (Youtube)
Online:https://www.youtube.com/watch?v=avv9GQ3b6Qg&ab_channel=DeepLearning_by_PhDScholar

28)Facial_Expression_Recognition_FER2013 (Github)
Online:
https://github.com/Baticsute/Facial_Expression_Recognition_FER2013

VI. Project Source Code and Dataset

Dataset:
Ck+
https://www.kaggle.com/datasets/shawon10/ckplus

FER2013
https://www.kaggle.com/datasets/msambare/fer2013

Project Source Code:
GoogleColab
https://colab.research.google.com/drive/1xPbjMoQF7SHtY8Khsimr-xzcOQH9Sb0z?usp=sharing

Github Repo
https://github.com/mrtlckn/WorkAndStudy/blob/main/Deep%20Learning/Neural%20Network%20Project/EmotionPrediction_with_ImageProcessing_and_Neural_Network_(CNN).ipynb