Facial Expressions Recognition

Burak Kuyucu*, Yunus Emre Kepenek**

***Department:** Computer Engineering, TOBB University**, email:** bkuyucu@etu.edu.tr
**** Department:** Computer Engineering, TOBB University **email:** ykepenek@etu.edu.tr

Abstract*:* An facial expression recognition algorithm using frontal facial image is presented in this paper. The algorithm is composed of four main steps: facial landmarks detection step, creating dataset by calculating attributes step, training models step, test step. In the first step, facial landmarks are detected by 68 points. In the second step, 13 attributes are calculated for each image in the database. Then, dataset is created which consists of 60 rows and 13 attributes. In the third step, 5 models are created and one model for each facial expression is trained with the created dataset. In the fourth step, facial expression of the test images are predicted by calculating the 12 attributes and calculating predictions by the trained models. In the experimental results, facial expressions of three persons are tested. Even though, the algorithm is very successful at predicting "happy", "sleepy" and "surprised", not good at predicting "sad" and "neutral. Because, their predictions are always very close to each other. More district attributes and bigger dataset can solve the problem.

1. Introduction

In recent years, computer vision is one of the focused domains of computer engineering. With the development of technology, demand on the computer vision is increasing significantly. The computer vision is applied in many different fields. Iris or fingerprint recognition in security, pattern recognition in medical imaging, automatic target recognition in defense industry are some of the usage of computer vision. Facial expressions recognition is also a very important problem in the domain of computer vision.

Facial expressions are signs that enhance interpersonal communication and provide for better transmission of the emotion to be conveyed. It is usually easy for people to understand and interpret facial expressions. However, it is not easy for computers. We aim to recognize facial expressions by our algorithm.

2. Implementation

In the project, it is possible to detect five facial expressions: happy neutral, sad, sleepy, surprised. Detecting facial expression process composed of 4 steps.

1. Detecting the locations of facial landmarks: eyebrows, eyes, nose, mouth and jaw.
2. Creating a dataset by deriving 12 attributes which are calculated using location and shape of the facial landmarks.
3. Training a model for each facial expression by using the created dataset.
4. Testing

1. Step – Facial Landmarks Detection

To detect facial landmarks, "facial landmark detector" is used. It is a pretrained model of "dlib" library. This pretrained model detects facial landmarks by 68 points at the given image. The details of 68 points are like as following:

- Jaw - 18 points
- Right eyebrow – 5 points
- Left eyebrow – 5 points
- Nose – 8 points
- Right eye – 6 points
- Left eye – 6 points
- Mouth – 20 points

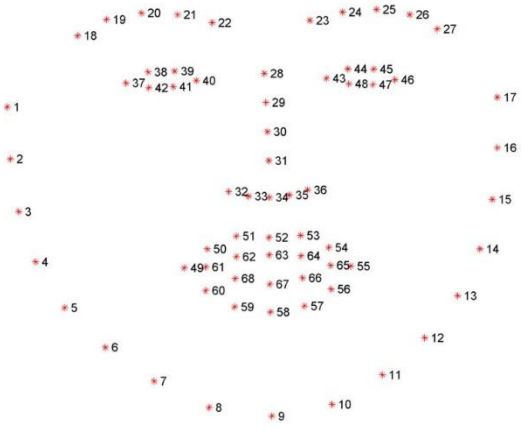


Figure 1 - facial landmark detector

Order of the points are shown in the Figure 1.

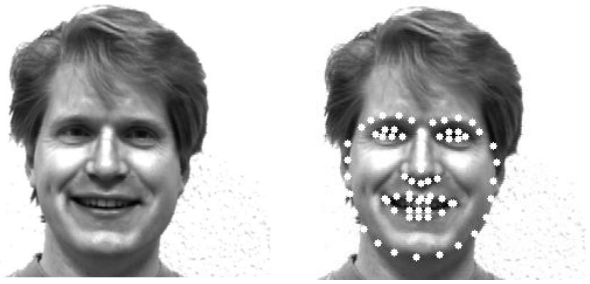


Figure 2 – example result

An example output of the pretrained model is shown in the Figure 2.

2. Step – Creating Dataset

At this step, Yale database images are used to create dataset. In the database, there are 13 persons and many facial expressions of the persons. Because of having 5 expressions in our project, 13x5=65 images are used from the database. 5 images are separated from creating dataset. Because, they are used for testing. For each image, 68 points are found by processing the first step. Then, 12 attributes are found for creating dataset. 12 attributes are found by using the following formulas:

- Mouth height / Jaw height
- Mouth width / Jaw width
- Nose width / Jaw width
- Left eyebrow height / Left eyebrow width
- Distance between the left eye and eyebrow / Left eyebrow width
- Left eye height / Left eye width
- Right eye height / Right eye width
- Left eye height / Distance between the left eye and eyebrow
- Right eye height / Distance between the right eye and eyebrow
- Mouth height / Mouth width
- Difference of nose min y coordinate and mouth max y coordinate / Mouth height
- Difference nose min y coordinate and mouth min y coordinate / Mouth height

After calculating these attributes for each image, dataset is created which consists of 60 rows and 13 attributes. 12 attributes were mentioned above. 13th attribute is the facial expression of the image. 13th attribute is assigned 1 if expression is happy, assigned 2 if expression is neutral, assigned 3 if expression is sad, assigned 4 if expression is sleepy, assigned 5 if expression is surprised.

3. Step – Model Training

The main objective of this step is machine learning. Machine learning is implemented by using Scikit learn. Support Vector Machine is used as a classifier. At this step, models are trained by using the dataset which is created at the second step. There is a different model for each facial expression. Model are named as the following:

- Happy model
- Neutral model
- Sad model
- Sleepy model
- Surprised model

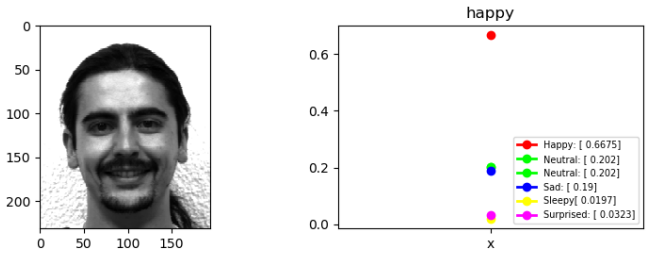
Models are trained by manipulating 13th attributes in the dataset. For instance, if sad model will be trained, 13th attributes of 48 rows are changed to 0. Then, 13th attribute of 12 rows are changed to 1 which were 3. They were 3, because 3 represents "sad" as we mentioned before. First, 12 attributes are used to train models and 13th attribute is given as a result parameter of the scikit learn.

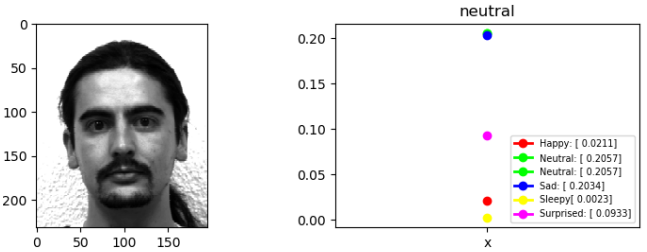
4. Step - Test

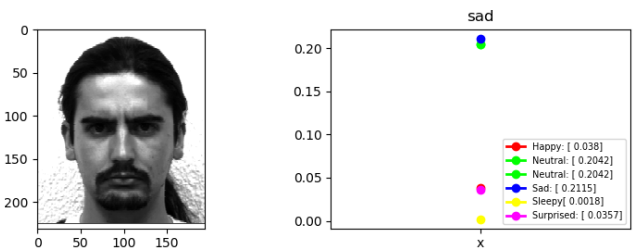
At this step, test images are tested. To test an image, 68 points are derived as mentioned in the first step. Then, 12 attributes are calculated as mentioned in the second step. Next, predictions are made by trained models which are trained in the third step. Facial expression prediction of the test image is decided according to a trained model which has highest prediction score. For instance, if happy model has the highest prediction score, our facial expression prediction is “Happy”.

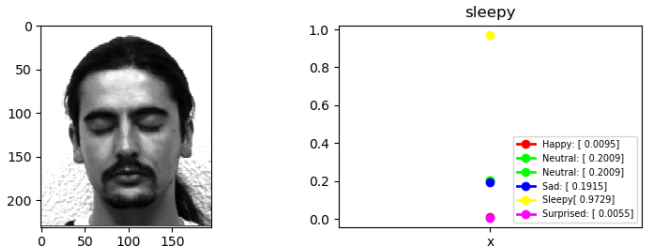
3. Experimental Results

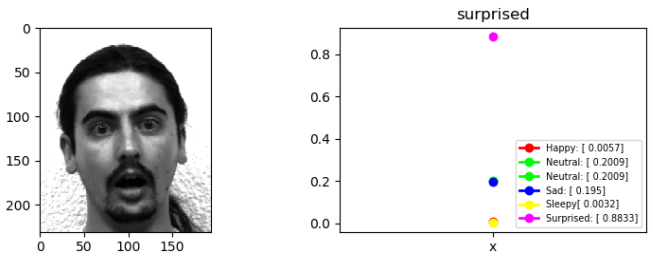
After processing 4th step, a graphic has been created to show results. At below, test results of the first person are shown:





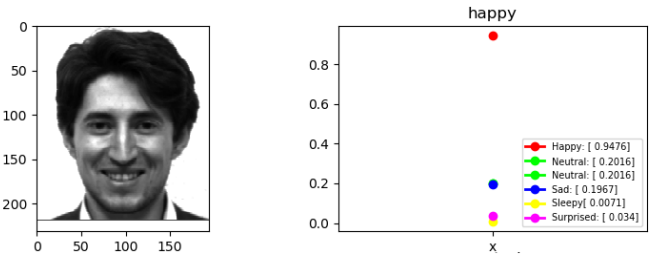


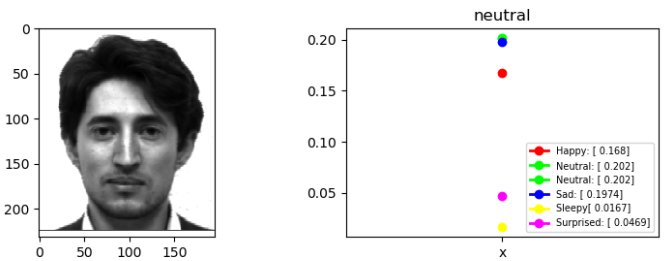


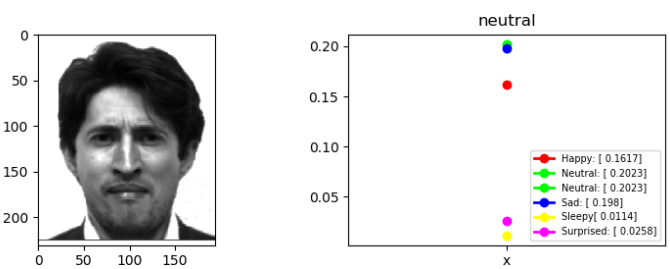


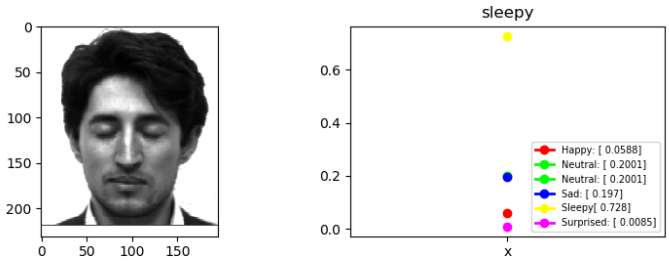
The prediction score of the models are can be seen on the right bottom of the images. As seen in the pictures, facial expressions of the images are predicted successfully. However, the success we define is not the rate of the prediction. The success we define is the big margin between the two highest predictions. As seen in the test results, recognition of "Happy", "Sleepy" and "Surprised is successful. Recognition of "sad" and "Neutral" is true but less successful than other facial expressions. Because, margin of the two highest predictions is very low. It may cause wrong predictions at other test images.

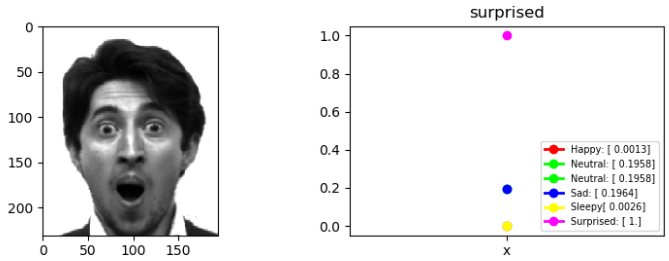
At below, test results of the second person are shown:





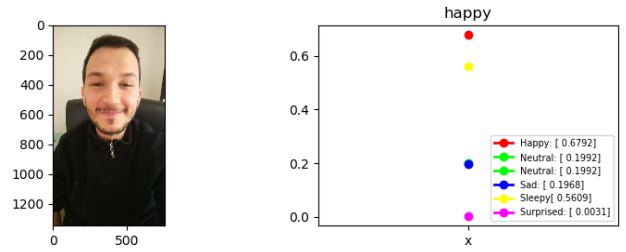


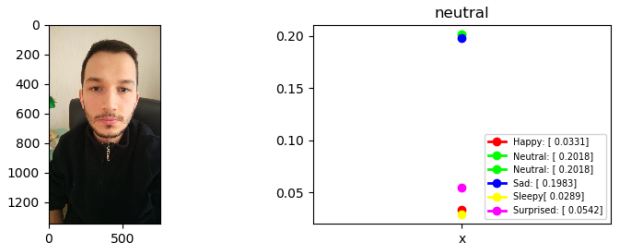


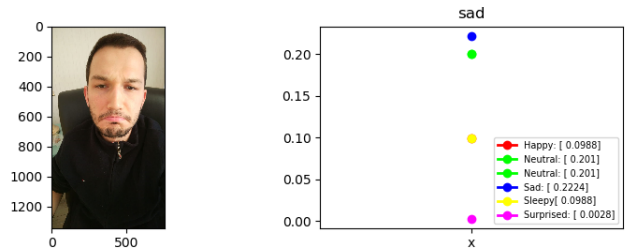


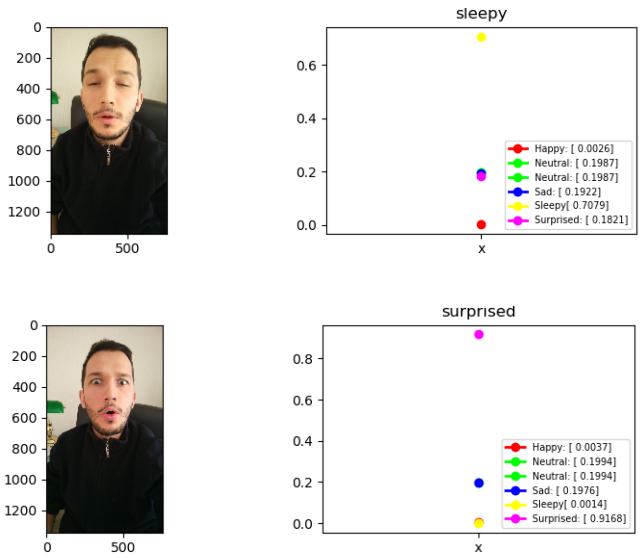
As seen in the test results of the second person, facial expressions are predicted successfully except "Sad". "Sad" picture's facial expression was predicted as "Neutral". It was mentioned that this error can happen because of not having margin between the "Sad" and "Neutral" predictions. There are two things should be done to solve this problem. First is that new attributes can be found and added to the dataset which is created at the second step. Second is that bigger dataset should be used to create the dataset.

After reaching considerable results, real person's facial expressions are tested. At below, test results of the real person are shown:









As seen in the test results of the real person, 5 facial expressions are detected successfully. However, recognition of the "happy", "sleepy" and "surprised" is more successful than recognition of the "sad" and "neutral" again.

4. Conclusion

An facial expression recognition algorithm using frontal facial image is presented in this paper. The algorithm is composed of four main steps: facial landmarks detection step, creating dataset by calculating attributes step, training models step, test step. In the first step, facial landmarks are detected by 68 points. In the second step, 13 attributes are calculated for each image in the database. Then, dataset is created which consists of 60 rows and 13 attributes. In the third step, 5 models are created and one model for each facial expression is trained with the created dataset. In the fourth step, facial expression of the test images are predicted by calculating the 12 attributes and calculating predictions by the trained models. In the experimental results, facial expressions of three persons are tested. Even though, the algorithm is very successful at predicting "happy", "sleepy" and "surprised", not good at predicting "sad" and "neutral. Because, their predictions are always very close to each other. More district attributes and bigger dataset can solve the problem.

5. References

https://www.pyimagesearch.com/2017/04/10/detect-eyes-nose-lips-jaw-dlib-opencv-python/

http://vismod.media.mit.edu/vismod/classes/mas622-00/datasets/

https://www.learnopencv.com/facial-landmark-detection/

https://www.pyimagesearch.com/2017/04/10/detect-eyes-nose-lips-jaw-dlib-opencv-python/