Facial Recognition

Oğuzhan Ulusoy  
Department of Computer Science and Engineering  
Graduate School of Science and EngineeringIşık University, Istanbul, Turkey  
oguzhan.ulusoy@isik.edu.tr

*Abstract* — *Extracting meaningful data from real time images is important. I have studied facial recognition in this study. The architecture has been used in this study is a face recognition pipeline that learns mapping from faces to a position in a multidimensional space where the distance between points directly correspond to a measure of face similarity. Including pipeline consists of three major stages: preprocessing, embedding and classification.*

Keywords — Deep learning, convolutional neural network, facial/face recognition, object prediction, neural networks.

# **Introduction**

Extracting meaningful data from real time images which have been produced in real time devices become the most important point for city security, even national security. People contribute data into big data While it is increasing, to understand it is critical. I suggest that there must be a relation between intelligent computer and city. Segmentation of images provides pre-estimation before something happened or taking the necessary action. Although the major study should be including image segmentation and alert system, I cover -*small part*- facial recognition in this study.

# **Theory**

## Architecture

Facial recognition is a combination of two major operations: face detection and face classification. FaceNet—architecture is going to implement in this study.

## Features of architecture

FaceNet is start-of-art face recognition, verification and clustering neural network. It is 22-layers deep neural network that directly trains its output to be a 128-dimensional embedding. The loss function used at the last layer is called triplet loss. It is efficient and faster computationally power. It has low memory usage and low power consumption.

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

Figure 1 Layers of FaceNet architecture

# **Stages**

*1. Pre-processing* – a method used to take a set of images and convert them all to a uniform format – in our case, a square image containing just a person’s face. A uniform dataset is useful for decreasing variance when training as we have limited computational resources.

*2. Embedding* – a process, fundamental to the way FaceNet works, which learns representations of faces in multidimensional space where distance corresponds to a measure of face similarity.

*3. Classification* – the final step which uses information given by the embedding process to separate distinct faces.

# **Implementation**

*1.* Some Python libraries – *open computer vision, keras, numpy and preprocess* are used in this study.

*2.* Application starts with main file.

*3.* Database is created using preprocess. Preprocess loads images from folder, then obtains faces.

*4.* Each image file converts into arrays using transpose and around methods in numpy library.

*5.* After database has created, a window is created, then currently taken images are captured from test video using video capture method in open computer vision library.

*6.* CascadeClassifier extracts some features. While test content is running, each frame converts to gray scale form. With this content, cascade classifier runs again for detecting multi scale. The output is face from test content. Captured images from processing by some techniques.

*7.* Captured images from test video and items in database are compared. Euclid distance is used in this step. Therefore, similarity is obtained. If threshold value is greater then similarity, a rectangle is put into face with tag.

# **Experıment Results**

I have tested with images of Sherlock and John characters from Sherlock Holmes series. The similarity rate changes between 18.\* and 14.\*. Average similarity rate is 16.\*.

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

Figure 2 Similarity rates for Sherlock and John

Then, I have created own data to test the program. I have took a few my photo, then have replaced them into train folder with a subfolder. When the similarity has been obtained, subfolder name is displayed as a tag. Moreover, I have also added subfolder name to configuration file. Test content is a video that is recorded in mp4 format. I have recorded a test video, then I have put it into test folder.

When I have started the program in second time, the similarity rate changes between 14.\* and 4.\*. Average similarity rate is 8.\*.

metin, iç mekan içeren bir resim

Açıklama otomatik olarak oluşturuldu

Figure 3 Similarity rates for Oguzhan

# **Conclusıon**

The reason for the difference between similarity rates is train data. Train data for Sherlock and John includes enhanced photos and sizes are appropriate. Train data for Oguzhan does not enhanced photos. Pixels and color are not good as well as Sherlock and John data.

##### **References**

1. Introduction to FaceNet: A Unified Embedding for Face Recognition and Clustering - Analytics Vidya
2. FaceNet FaceNet: A Unified Embedding for Face Recognition and Clustering
3. Face recognizer application using a deep learning model (Python and Keras)