Using Face Recognition for Authentication

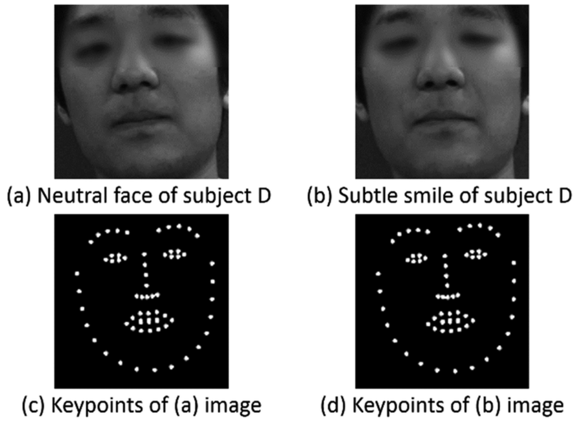
UDACITY

Capstone Proposal

Bodziony Rafał

1. Domain Background

Nowadays, internet security is one of the basic requirements of the modern world, security should consist of two or three types of authorization to be safe. Safeness can be described as Something you know, something you have or something you are.[1] The biometric data is unique and impossible to forget e.g. how we look, how we move, or what our eyes look like can be used as an authorization component and increase security. One of the basic biometric features is the face. By appropriately transforming this information, we are able to add an important component to authorization, e.g. checking employees entering the company or as an additional confirmation of payment in the bank to prove the transaction. This problem is called Face Recognition and is an important scientific issue. Chiara Turati et al. showed that even a few days children are able to distinguish a familiar face. In the case of computer vision and recognition of facial features, the face matching system presented by Takeo Kanade was a big step. He described the problem of facial recognition as a match between specific features on the face and calculating the distance between them.


(a) Neutral face of subject D

(b) Subtle smile of subject D

(c) Keypoints of (a) image

(d) Keypoints of (b) image

1. Key Points of face

Other widely used method was:

- 3-dimensional recognition
- Thermal cameras
- Eigenfaces[2]

The problem of accurate facial recognition and identification can be solved by using neural networks, the task is to extract a feature vector of a face and compare it with the vector which already exists in the database.

The motivation behind the facial recognition app is to take the next step in protecting privacy and increasing online and real-world security. If the face detection algorithm were effective enough to be used commercially, the possibilities could be enormous. From signing

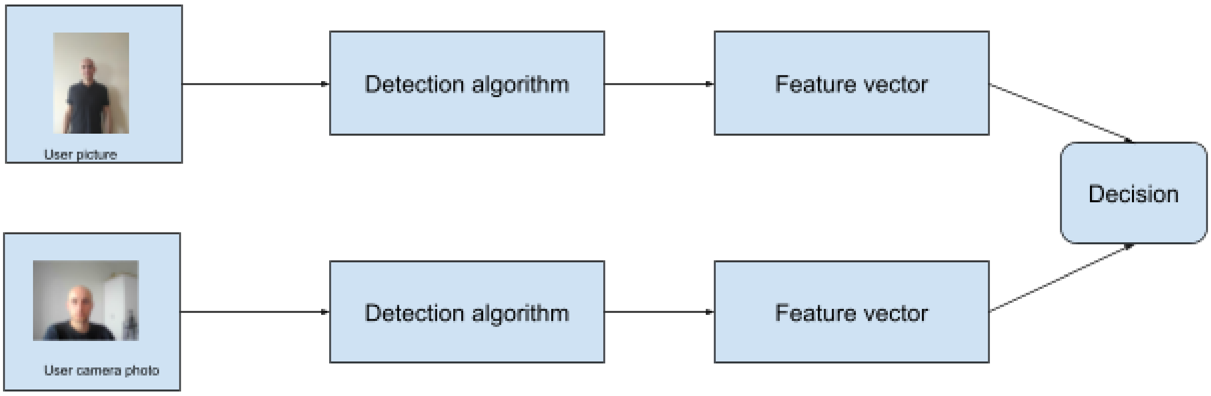
[1] Brandom, Russell (July 10, 2017). "Two-factor authentication is a mess". *The Verge*. Retrieved July 10, 2017.

[2] Face recognition with OpenCV. https://docs.opencv.org/master/da/d60/tutorial_face_main.html

messages with a key based on biometric data to simplified multi factor authentications to the automation of security in the real world, e.g. opening doors or locks.

2. Problem Statement

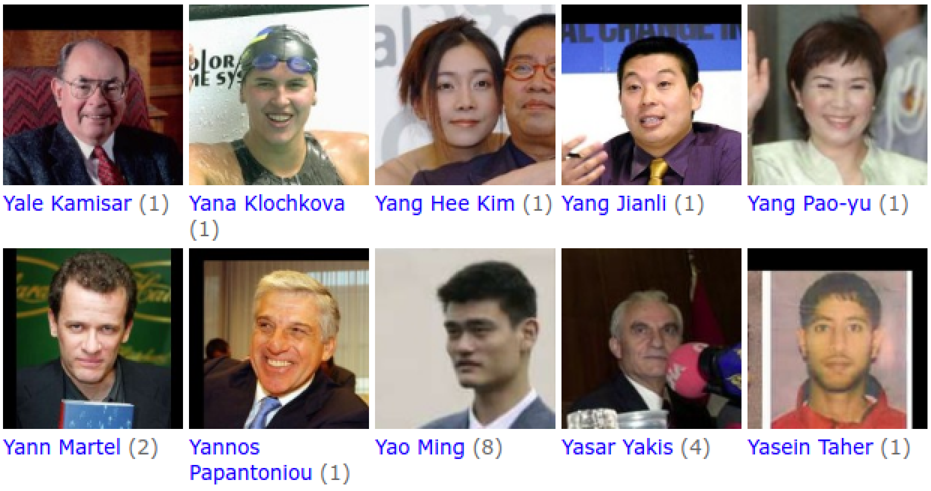
Make an application for user authentication based on facial recognition. In the first step, the user should add a photo of his face, then the algorithm via the webcam should distinguish whether he sees a real face or a photo, and finally determine whether he sees the same user.



2. Basic model scheme

3. Datasets and Inputs

For training and testing purpose and for input/output data as well I will use LFW Face Database[3]. Labeled Faces in the Wild[4] is a public benchmark for face verification, also known as pair matching.



3. Examples of images from LFW Dataset

[3] Huang, Gary & Mattar, Marwan & Berg, Tamara & Learned-Miller, Eric. (2008). Labeled Faces in the Wild: A Database forStudying Face Recognition in Unconstrained Environments. Tech. rep..
[4] http://vis-www.cs.umass.edu/lfw/

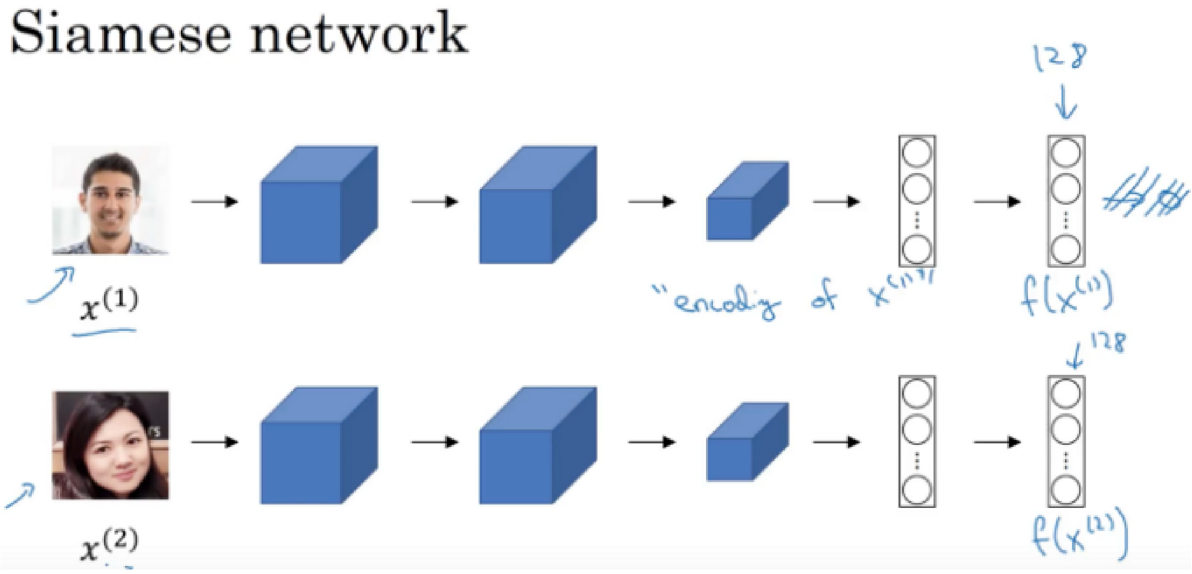
The data set contains more than 13,000 images of faces collected from the web. Each face has been labeled with the name of the person pictured. 1680 of the people pictured have two or more distinct photos in the data set. Dataset was splited to train, test, by researchers. To precise, dataset contains:

- 13233 images
- 5749 people
- 1680 people with two or more images

The dataset will be split to train and test sets based on information on the website. Then, I will use face images to distinguish pairs of the same person. I don't assume to recognize the identity information appearing on each image. The project assumed to use basic methods such as Eigenfaces[5] to prove the baseline model results and Convolutional neural network-based models. I want to choose one model used in the paperswithcode.com benchmark.[6] Images will serve as input data to the model and throughout the learning process images of the same person will be compared to minimize the distance between feature vectors. Models prepared in this way will be used to the image uploaded by the user and real-time webcam information to decide whether a face is the same or not.

4. Solution Statement

The solution assumes to use a kind of siamese network to train models and verify examples. The Siamese network is based on the assumption that the network should extract similar features for two or more images of the person. The model is trained on two images at the same time in two identical neural network architectures. Layers extract features space for images and create a vector that is used in comparison with a given distance metric function e.g. cosine. Below, the siamese network is described.

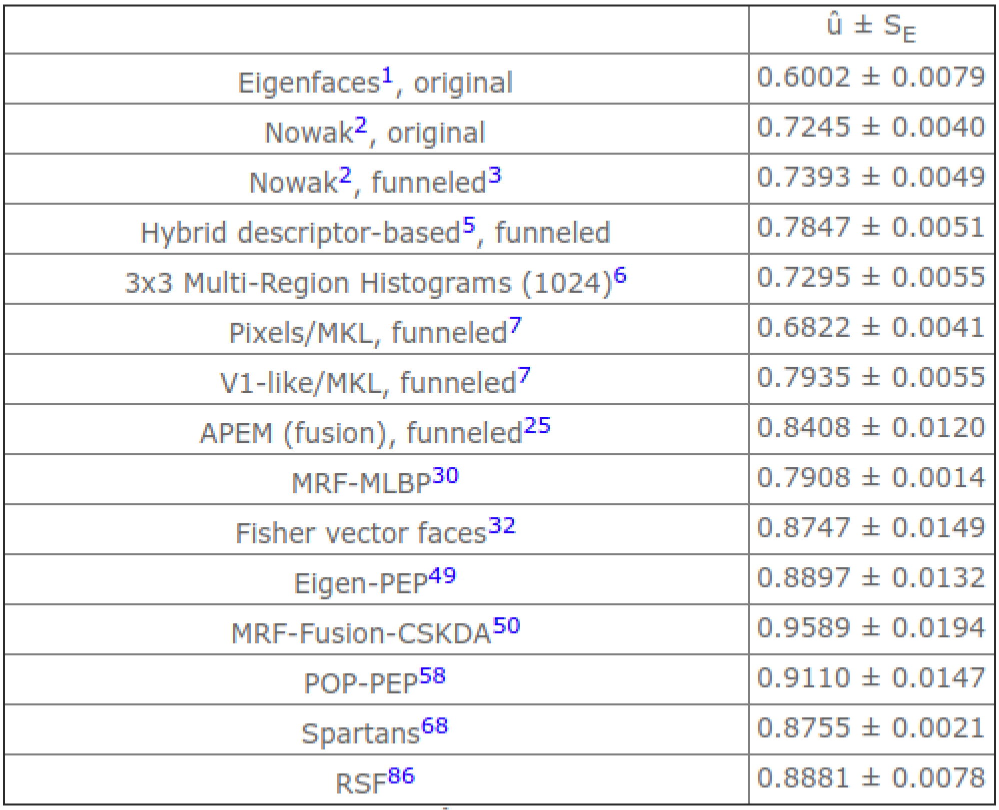


[5] Matthew A. Turk and Alex P. Pentland. Face Recognition Using Eigenfaces. Computer Vision and Pattern Recognition (CVPR), 1991.
[6] https://paperswithcode.com/sota/face-verification-on-labeled-faces-in-the

4. Siamese network explanation[7]

5. Benchmark Model

For this project, I will use Eigenface with SVM for a benchmark. Models can be found on the scikit learn examples website.[8]On http://vis-www.cs.umass.edu/lfw/results.html we can find benchmark models for Image-Restricted results without output-side data. The Table below shows benchmarked models, I will use the first Eigenface to benchmark my algorithm.



	$\hat{u} \pm S_E$
Eigenfaces[1], original	0.6002 ± 0.0079
Nowak[2], original	0.7245 ± 0.0040
Nowak[2], funneled[3]	0.7393 ± 0.0049
Hybrid descriptor-based[5], funneled	0.7847 ± 0.0051
3x3 Multi-Region Histograms (1024)[6]	0.7295 ± 0.0055
Pixels/MKL, funneled[7]	0.6822 ± 0.0041
V1-like/MKL, funneled[7]	0.7935 ± 0.0055
APEM (fusion), funneled[25]	0.8408 ± 0.0120
MRF-MLBP[30]	0.7908 ± 0.0014
Fisher vector faces[32]	0.8747 ± 0.0149
Eigen-PEP[49]	0.8897 ± 0.0132
MRF-Fusion-CSKDA[50]	0.9589 ± 0.0194
POP-PEP[58]	0.9110 ± 0.0147
Spartans[68]	0.8755 ± 0.0021
RSF[86]	0.8881 ± 0.0078

6. Mean classification accuracy û and standard error of the mean S_E[9]

7. Evaluation Metrics

For this project, I will use accuracy as an evaluation metric, suggested by the authors of the LFW dataset.

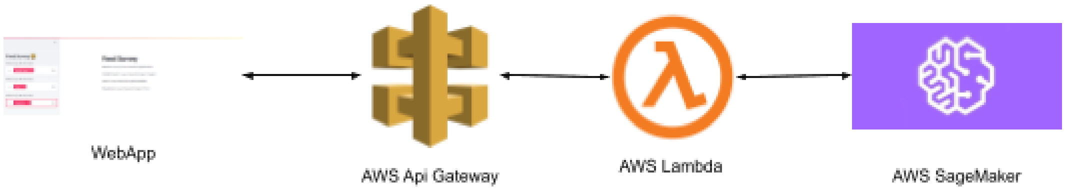
[7] https://www.youtube.com/watch?v=6jfw8MuKwpI

[8] https://scikit-learn.org/stable/auto_examples/applications/plot_face_recognition.html

[9] http://vis-www.cs.umass.edu/lfw/results.html#ImageRestrictedNo

8. Project Design

The project involves the development of an application to verify a person using computer vision algorithms. The project involves the development of an application to verify a person using computer vision algorithms. The first step will be to collect data and perform a quantitative and qualitative analysis of images, at this stage, it is also assumed to add data from other sets in order to normalize the set in terms of ethnic and age diversity. Next, models based on simple computer vision algorithms and deep neural networks will be tested. It is also assumed to create a web application on which the user will be able to place a photo and re-identify it later. The basic architecture schema of an application is presented below



7. Simple architecture scheme for Face Recognition service