

Machine Learning Engineer Nanodegree

Image Recognition Capstone Project

Andrey Semenov



I. Image recognition capstone project

Project Overview

The idea of the project has been taken from the description of one of project examples: [Computer Vision Capstone Project](#). The goal of the project was to create a learning

agent in the form of bot for Telegram messenger which is able to recognize faces of famous people.

The set of basic techniques for image processing is described in the book [Programming Computer Vision with Python](#). The basic methods and python libraries (PIL, NumPy, SciPy, Matplotlib) are described in details including the number of examples. Computer vision could be described as an field AI in which the computer is able to recognize various objects using different AI approaches from images or videos.

Face recognition is one of most challenging problems in computer vision and image processing. Good literature review of different approaches used for this problem could be found in the paper [A Survey of Face Recognition Techniques by Rabia Jafri and Hamid R. Arabnia](#). Authors divide the techniques for the problems on the following: featured-based, statistical, AI based.

The face recognition problem has a number of practical applications:

- Security surveillance
- General identity verification
- Entertaining applications (Apple Photos)

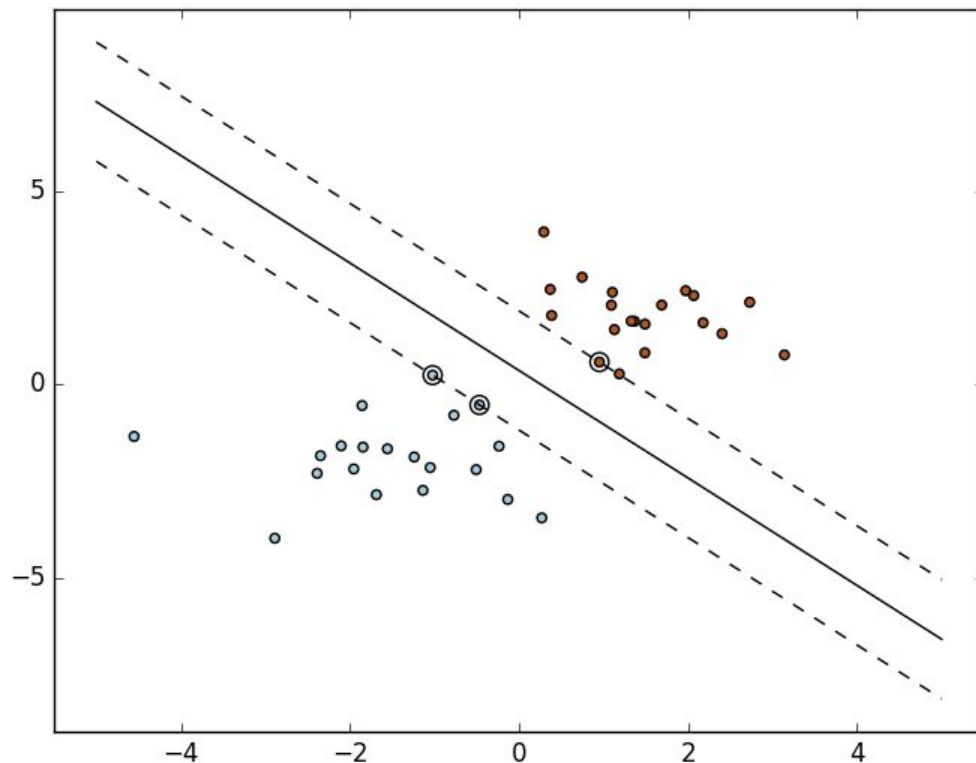
Problem Statement

The face recognition general problem can be formulated as follows: Given an input face image and a database of face images of known individuals, how can we verify or determine the identity of the person in the input image? More specific, Face Recognition or Face Identification is: given the picture of the face of an unknown person, identify the name of the person by referring to a gallery of previously seen pictures of identified persons.

The problem of face recognition is classification problem. Solution of the problems is supervised by nature in the face identification problems, because we are comparing the unknown face with already known faces.

The goal for this particular project was formulation as: create the bot for Telegram messenger allowing to identify face of famous person. Python programming language has been chosen for the implementation of user application. Telegram messenger platform has been chosen for user interface development due to the open API. [telepot](#) framework was used for the bot development.

Machine learning component of application (database of faces, functions) was taken from [sklearn face_recognition](#). In the solution of the problems the principal component analysis will be used to decrease the dimension of the problem. And Support Vector Machine ([SVM](#)) algorithm will be used for the solution of classification problem. A support vector machine constructs a hyper-plane or set of hyper-planes in a high or infinite dimensional space, which can be used for classification, regression or other tasks. Intuitively, a good separation is achieved by the hyper-plane that has the largest distance to the nearest training data points of any class (so-called functional margin), since in general the larger the margin the lower the generalization error of the classifier.



Metrics

Different models build on training dataset were evaluated by the following criteria [classification report](#):

- [Recall score](#)

Recall score= $tp/(tp+fn)$. Where tp is the number of true positive outcomes on testing data set and fn is false negative outcomes. The recall is an ability of classifier to find all positive outcomes.

- [Precision score](#)

Precision score= $tp/(tp+fp)$. Where tp is the number of true positive outcomes on testing data set and fp is false positive outcomes. The closer the precision to 1, the better the model.

- [F1 score](#)

F1 score is a harmonic average of precision and recall score ($F1=2*(precision*recall/(precision+recall))$)

Metrics above have been chosen instead of accuracy due to the better description of predictive ability of classifier due to the widely known [accuracy paradox](#) in which the model with highest accuracy is not the best one.

II. Analysis

Data Exploration

Publically available [faces](#) dataset was used. Dataset of [labeled faces in wild](#) 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. In different parts of the project, I've varied the minimum number of photos per person for the analysis, and used the portion of dataset in the range from 1288 samples to 4325. Each picture is centered on a single face. Each pixel of each channel (color in RGB) is encoded by a float in range 0.0 - 1.0. The original images are 250 x 250 pixels, but the default slice and resize arguments reduce them to 62 x 74. The dataset is described in details in the paper [let](#).

Exploratory Visualization

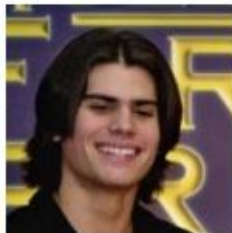
The dataset is a named series of famous faces people:

Database by name, all

[A][Alf][Ang][B][Bin][C][Che][Col][D][Daw][Don][E][Eri][F][G][Goe][H] [I][J][Jav][Jes][Joh]
[Jos][K][Kim][L][Lil][M][Mark][Mel][Mik][N][O][P] [Per][Q][R][Ric][Rog][S][Sha][Ste][T]
[Tim][U][V][W][X][Y][Z]



AJ Cook (1)



AJ Lamas (1)



Aaron Eckhart (1)



Aaron Guiel (1)



Aaron Patterson
(1)



Aaron Peirsol (4)



Aaron Pena (1)



Aaron Sorkin (2)



Aaron Tippin (1)



Abba Eban (1)



Abbas Kiarostami
(1)



Abdel Aziz Al-
Hakim (1)



Abdel Madi
Shabneh (1)



Abdel Nasser
Assidi (2)



Abdoulaye Wade
(4)



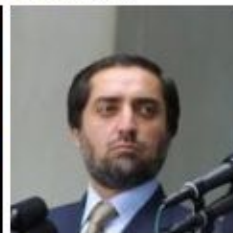
Abdul Majeed
Shobokshi (1)



Abdul Rahman (1)



Abdulaziz Kamilov
(1)



Abdullah (4)



Abdullah Ahmad
Badawi (1)

Many persons have more than 1 image (in brackets).

Algorithms and Techniques

There are three main techniques to be used for the solving this image recognition task:

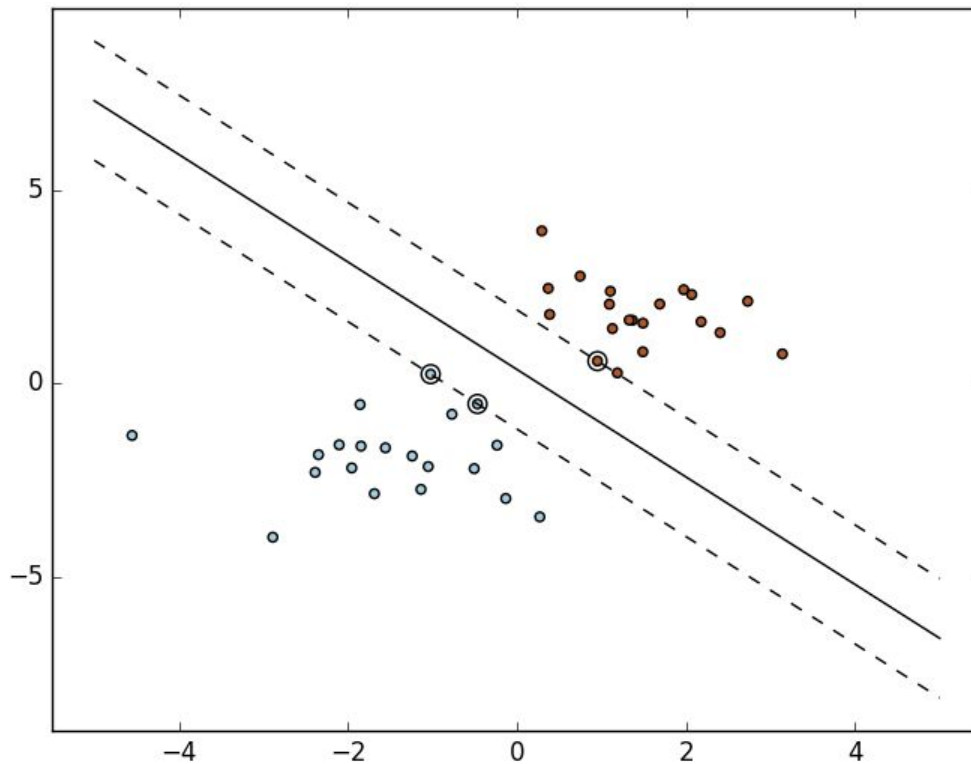
1) Principal Components Analysis [PCA](#).

Principal component analysis (PCA) is a statistical procedure that uses an orthogonal transformation to convert a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables called principal components. Detailed and clear explanation of linear algebra basics behind PCA are presented in the paper [Principal component analysis by Aapo Hyv arinen](#). Principal components are the linear combination of vectors that contain as much of the variance in the input data as possible. In order to found the principal components, we need first to create covariance matrix of input data. Then found the eighteen values of covariance matrix. Eighteen vectors of covariance matrix will be our principal components. So, the in the basis of eighteen vectors the covariance matrix is orthogonal. So mathematically speaking, the task of finding principal component is the task of calculating eighteen vectors of the covariance matrix. By transforming the observation matrix into the basis of principal components, we are minimizing the correlation among the data.

It allows to decrease the dimension of the problem for face recognition by leaving only most important features (corresponding by first n eighteen values of covariance matrix) and removing correlated data. But how many principal components to keep? Different number of features (n=5,10,50,100,150,200,250) was compared in order to achieve optimal number of principal components.

2) Machine learning problems in such case is a supervised learning problems. The model is supposed to be trained trained on the part of the dataset (training dataset), tested on testing data set and used on images received from the user. For supervised learning part the Support Vector Classifier was used [SVM](#). In the supervised learning project, the SVM classifier got one of the highest scores.

A support vector machine constructs a hyper-plane or set of hyper-planes in a high or infinite dimensional space, which can be used for classification, regression or other tasks. Intuitively, a good separation is achieved by the hyper-plane that has the largest distance to the nearest training data points of any class (so-called functional margin), since in general the larger the margin the lower the generalization error of the classifier.



The advantages of support vector machines are:

- Effective in high dimensional spaces.
- Still effective in cases where number of dimensions is greater than the number of samples.
- Uses a subset of training points in the decision function (called support vectors), so it is also memory efficient.
- Versatile: different Kernel functions can be specified for the decision function. Common kernels are provided, but it is also possible to specify custom kernels.

The disadvantages of support vector machines include:

- If the number of features is much greater than the number of samples, the method is likely to give poor performances.
- SVMs do not directly provide probability estimates, these are calculated using an expensive five-fold cross-validation.

-
- 3) In order to make the learning agent able to learn new images in addition to existing dataset, if the image is not recognized, the image will be stored in the dataset. And in the next request, the classifier will take into account the stored image received from user.

Benchmarks

The number of variable benchmarks is available for this dataset (image-restricted results with no outside data) [benchmark](#):

Technique	Mean classification accuracy and standard error of the mean
Eigenfaces, original	0.6002 ± 0.0079
Nowak, original	0.7245 ± 0.0040
Nowak, funneled	0.7393 ± 0.0049
Hybrid descriptor-based, funneled	0.7847 ± 0.0051
3x3 Multi-Region Histograms (1024)	0.7295 ± 0.0055
Pixels/MKL, funneled	0.6822 ± 0.0041
APEM (fusion), funneled	0.8408 ± 0.0120
APEM (fusion), funneled	0.8408 ± 0.0120
MRF-MLBP	0.7908 ± 0.0014
Fisher vector faces	0.8747 ± 0.0149
Eigen-PEP	0.8897 ± 0.0132
MRF-Fusion-CSKDA	0.9589 ± 0.0194
POP-PEP	0.9110 ± 0.0147
Spartans	0.8755 ± 0.0021

After analysis of available information about different techniques used for face recognition problem above, the closest method to the PCA approach used in this project,

would be method used eigenfaces described in the [paper](#). It is meant, that result with mean classifier accuracy more than 0,6 could be treated as statistically significant.

III. Methodology

Data Preprocessing

A few data preprocessing steps was used for the dataset:

- The time of the bot response is crucial. So in order to manage the part of dataset on which the model is trained, the `min_faces_per_person` variable is using in order to take the faces with the `>threshold` number of pictures. All images from dataset are resizing with `resize=(0,1]` ratio parameter.
- The next step is splitting the data between training and testing data set. For the existing data set the ratio of 3:1 was used, for the bot image recognition all images are using as testing data set and only 1 received from user as testing data set.
- After receiving the image from user, the image is resizing to the data set size.
- Also image is converting to 1D numpy array for further transformation by PCA.

Implementation

1. PCA implementation.

Standard sklearn implementation of [PCA](#) was used for the project. In the beginning the number of principal components was chosen (150 in this example). After that the training and testing datasets were converted into the space of principal components.

2. SVM classification implementation.

Support Vector Machine classifier was used with GridSearchCV object. SVM classifier from sklearn package was used with variable parameters 'C' and 'gamma' in order to achieve the best fit model. Parameters were varied in the following range: 'C': [1e3, 5e3, 1e4, 5e4, 1e5], 'gamma': [0.0001, 0.0005, 0.001, 0.005, 0.01, 0.1].

3. Recognizer bot implementation in telegram.

[Telepot](#) framework was used and class Bot was created. Bot Recognizer was created by BotFather in Telegram messenger. Received Token was used for authorization. Bot is handling messages by method: on_chat_message. In case of receiving an image, the image is storing and used for recognition.

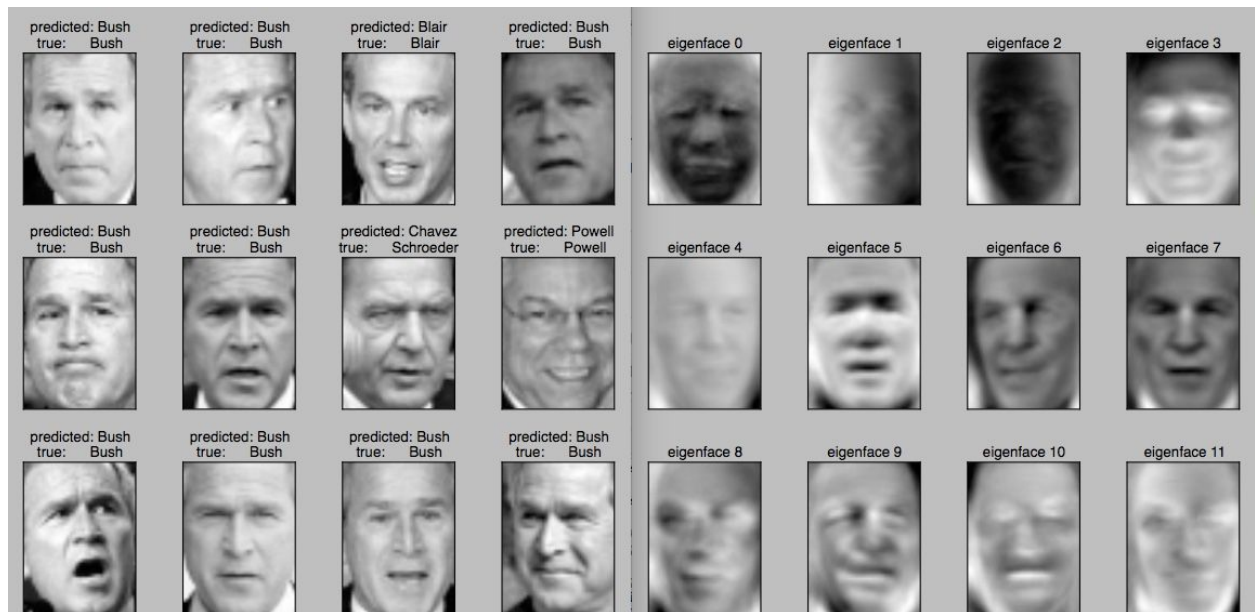
Refinement

In order to found the best solution using PCA, the following number of components was used: (5,10,50,100,150,200,250). And results log for 12 persons with the highest number of pictures are below. The performance of the model depends upon number of PC could be compared by the average precision/recall/F1 score. Total dataset size: n_samples: 1288, n_features: 1850, n_classes: 7. Results:

n_components	precision	recall	f1-score	C	gamma
5	0.41	0.34	0.36	50000	0.1
10	0.50	0.48	0.49	1000	0.1
50	0.82	0.82	0.82	1000	0.01
100	0.86	0.86	0.86	1000	0.005
150	0.86	0.85	0.85	1000	0.005
200	0.85	0.85	0.85	1000	0.001

As it could be seen from the beginning, in case of low number of PC (5,10) the dimension is not enough to recognize faces and scores are poor. But starting from 50, the average F1/Precision/Recall scores are increasing, which means that model is improving the predictive ability. The local maximum is found for n=100 components (all average scores are equal 0,86) and later increase of number of components to 150 and 200 only a little bit decrease average scores. It is mean, that 100 components are allowing to extract all information from images needed for this task. The models with highest scores (100 and 150 n_components) were derived with the same other parameters C=1000 and gamma=0.005. It justify usage of this C and gamma parameters for the future GridSearchCV.

Example of visual analysis below: prediction vs true name. And display of first 11 eighteen vectors:



IV. Results

Model Evaluation and Validation

The final model was constructed for famous people with >40 images in the database and 100 principal components and have the following parameters. The number of 100 principal components and parameters $C=1000$ and $\gamma=0.005$ were selected due to the maximization of F1 score in the analysis above.

The final model has the following characteristics (train on 1867 samples, 1850 features each):

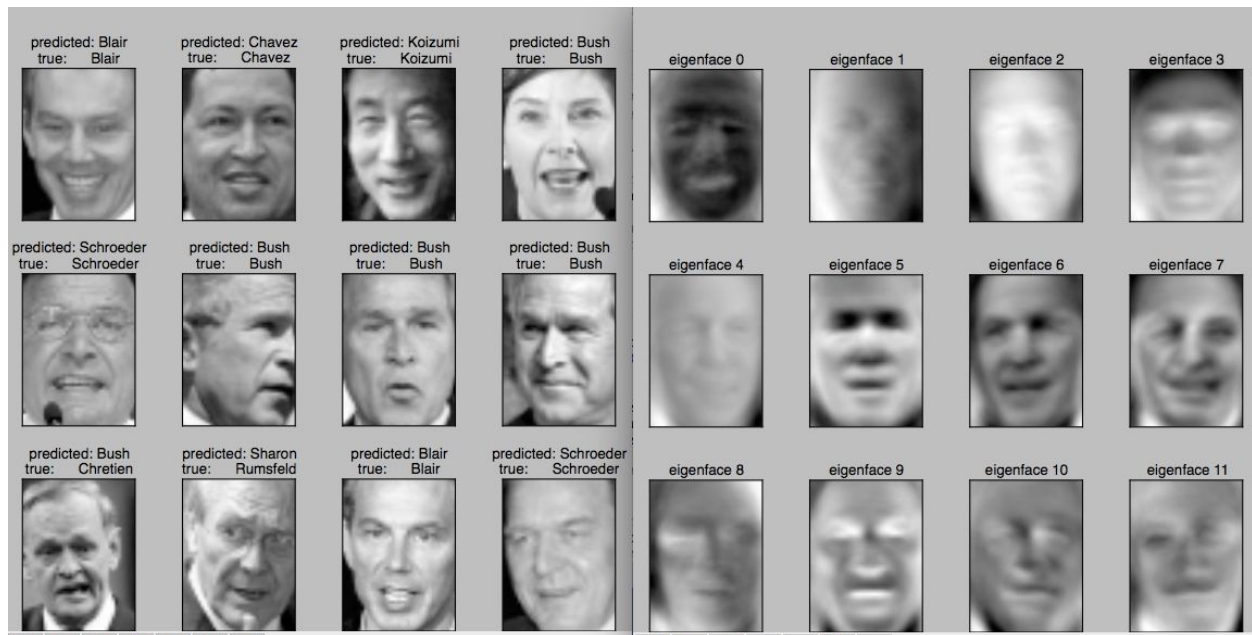
Precision	Recall	f1-score	Support	Mean classification accuracy
0.79	0.78	0.77	467	0.784

The model fitting time is less than 60 s, which make the model available for usage in the user application. Model is able to recognize 19 famous people and tell to which of them

the provided photo is similar. As well the model mean accuracy is equal to 0,784 which is more than 0,602 in established benchmark. All of mentioned above make model robust.

Justification

Based on the avg/total scores for precision, recall, $F1 > 0.77$ the model is able to recognize the face for 19 famous people. As it is shown below, the faces predicted are correct for the majority of predictions.



V. Conclusion

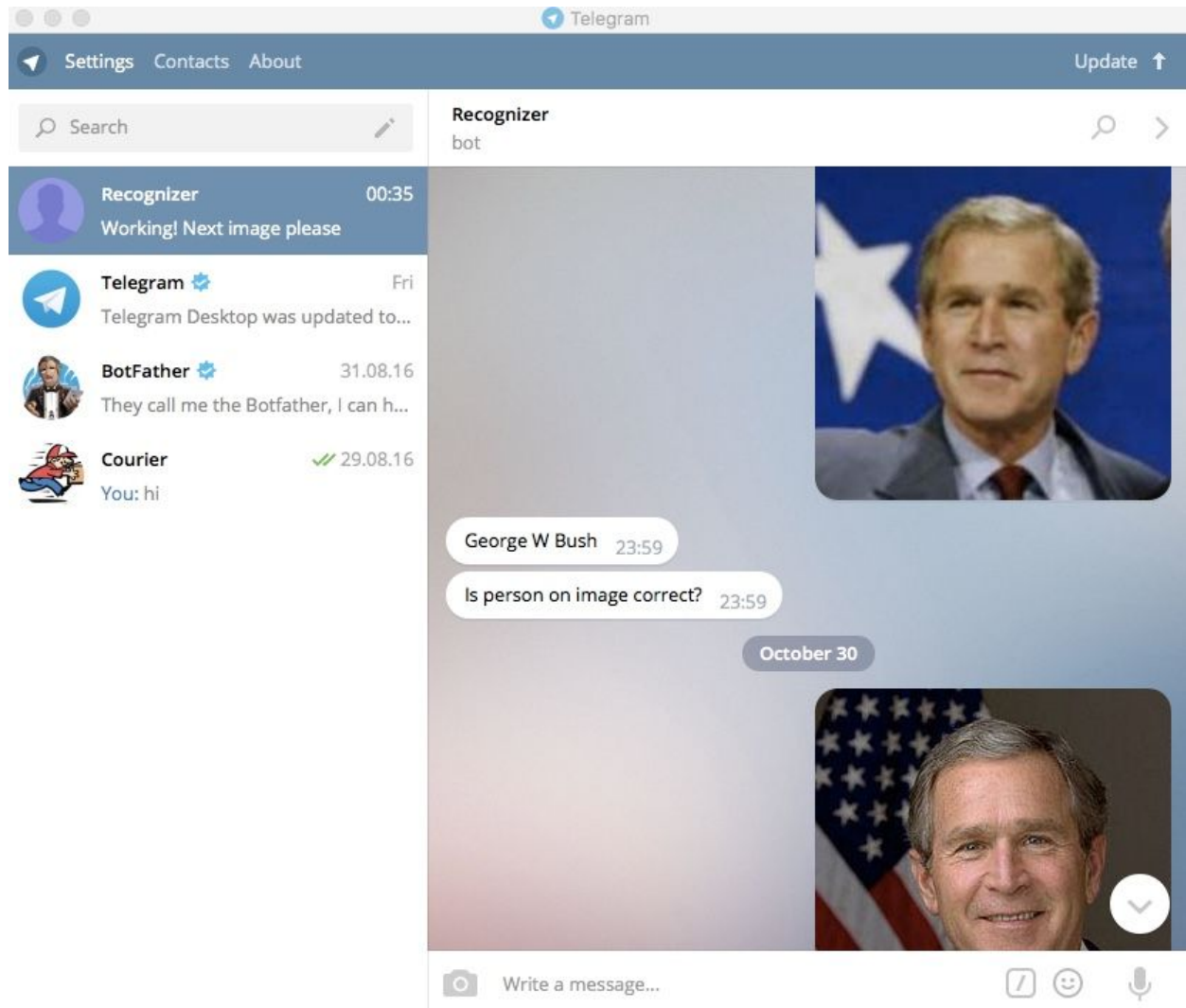
Face recognition problems are quite difficult due to the limitations in calculation power, necessity to process images and limited volume of existing databases. But use of modern filtering (PCA) and supervised learning algorithms (SVM) could found the solution of the problem in case of sufficient image database and time available. The following tasks and steps were executed on the way to project accomplishment:

1. Dataset of pictures of famous people was uploaded to the machine and analyzed. In this part I've managed to load and store database of pictures which will be used in the future for model training and validation.

-
2. The PCA was used to decrease the dimension of the problem and extract the maximum of non-correlated information from the dataset. On this step, I transform the database of images to the array form and simplify it. After this data processing, I ended up with the problem statement which could be solved using machine learning techniques. Important to note, that PCA is a universal approach and could be applied for different problems in AI.
 3. I've used GridSearchCV and SVM classifier to train it, and derive optimal parameters (n_components, C, gamma) for my future prediction model. On this step, based on the different model scores (precision, recall, f1), I were able to pick optimal parameters. Those parameters will be used for future predictions.
 4. The user interface in the form of Telegram bot was created for the derived model. It allows user to interact with the model and retrieve predictions based on the supplied images. This was particularly interesting, because it allows make derived model useful for different people in real world and build cross-platform application.

Free-Form Visualization

The user interface of application is looks like this:



The image could be sent to bot and bot will advise to which famous person the face is similar.

Reflection

There were two particularly interesting and difficult at the same time parts of the project:

1. Image recognition techniques. The whole set of instruments for image recognition: filtering, image representation in the form of arrays, PCA transform, etc... were quite new for me. Great help for me was coverage of PCA on this example in the education materials.

-
2. Bot user interface. Thanks to open API it was possible to build user interface for the application available on different platforms: mobile, desktop, etc... Interesting part was in building the bot and handling the messages.

Main lessons learned from this project for me are the techniques for image processing and image data analysis. It was first time when I've used image recognition techniques and works with images as with arrays of data. It allows me to solve the real problems. As well PCA shows good applicability to extraction of unique information from images and reduction of problems dimension. SVM classifier validate its applicability to the complex classification problems.

Improvement

Derived model is robust enough in case of the big number (>50 images) per person. In case of decreasing the threshold of images per person to 10 for example, the model has the following performance:

Precision	Recall	f1-score	Support	Fitting time
0.51	0.51	0.47	1081	620.780s

So there are 2 clear issues:

- Decrease of scores in case of using more images in the dataset and less images per person for model training. But still the derived model have scores >0.5.
- The training of the model is taking >10 minutes which makes it hard to reeducate the model after adding the images to the dataset.

The solution of those problems could significantly improve the project. In order to do it there are 2 possible solutions:

1. Train the model (for example complex neural network) not in real time. For example spent few days and huge database of image and then use it for all face recognition without training on newly received images from the user. Or train it

periodically, let's say every couple of days. 'Hot start' option for algorithm retaining information from previous training could be used as well.

2. Increase of accuracy could be achieved by increasing number of images per person. 10 images is not enough, but 40 (as it is shown before) allows to make model using SVM classifier with good F1 score.
 3. Other option could be use of different classification method: neural networks, decision trees, etc... They could improve the results.
-