

Face recognition and verification

Subhasis Chand, Abhinav Dhere, Aniket Pokale, T. Venkata Sai
Raghuram

November 29, 2017

1 Introduction

2 Datasets

3 Method

4 Results

- Results : Facial recognition
- Results : Twin recognition

5 Conclusion

Introduction

- Facial recognition is finding increasingly many applications in security, consumer products and as a biometric signature.

Introduction

- Facial recognition is finding increasingly many applications in security, consumer products and as a biometric signature.
- In this project, we employ FaceNet neural network [4] to obtain embeddings from face images and then apply a classifier to identify the faces.

Introduction

- Facial recognition is finding increasingly many applications in security, consumer products and as a biometric signature.
- In this project, we employ FaceNet neural network [4] to obtain embeddings from face images and then apply a classifier to identify the faces.
- We start by performing extensive testing on multiple facial recognition datasets. Following this, we take up a relatively unexplored problem of identifying identical twins and test our system on three celebrity twin pairs.

Face recognition datasets

We have used the following datasets in this project:

- ① Faces94 [5] : Contains 2,220 images of 153 male and female subjects sitting at a fix distance from camera and speaking with facial expression variation.

Face recognition datasets

We have used the following datasets in this project:

- ① Faces94 [5] : Contains 2,220 images of 153 male and female subjects sitting at a fix distance from camera and speaking with facial expression variation.
- ② Faces95 [6] : Contains 1,440 images of 72 subjects moving towards camera, thus containing variations in head positions.

Face recognition datasets

We have used the following datasets in this project:

- ① Faces94 [5] : Contains 2,220 images of 153 male and female subjects sitting at a fix distance from camera and speaking with facial expression variation.
- ② Faces95 [6] : Contains 1,440 images of 72 subjects moving towards camera, thus containing variations in head positions.
- ③ Grimace [7] : This dataset contains 360 images of 18 subjects. This is the most difficult data as it contains head variations and extreme grimace expressions.

Celebrity pairs for twin recognition

We have used web scraping to obtain images of following celebrity pairs and then manually cleaned the results to remove erroneous images downloaded –

- 1 Canadian actors Aaron and Shawn Ashmore.

Celebrity pairs for twin recognition

We have used web scraping to obtain images of following celebrity pairs and then manually cleaned the results to remove erroneous images downloaded –

- ① Canadian actors Aaron and Shawn Ashmore.
- ② British actors James and Oliver Phelps (known for Harry Potter franchise).

Celebrity pairs for twin recognition

We have used web scraping to obtain images of following celebrity pairs and then manually cleaned the results to remove erroneous images downloaded –

- ① Canadian actors Aaron and Shawn Ashmore.
- ② British actors James and Oliver Phelps (known for Harry Potter franchise).
- ③ Veteran actor Will Ferrell and renowned drummer Chad Smith.

Face detection and preprocessing

- We use built in function for Haar Cascade face detection [8] in OpenCV to detect the frontal faces in each image.

Face detection and preprocessing

- We use built in function for Haar Cascade face detection [8] in OpenCV to detect the frontal faces in each image.
- Following this we crop the detected face and resize the image to 96x96 size for input to the neural net. For this we have written code in OpenCV Python.

Face detection and preprocessing

- We use built in function for Haar Cascade face detection [8] in OpenCV to detect the frontal faces in each image.
- Following this we crop the detected face and resize the image to 96x96 size for input to the neural net. For this we have written code in OpenCV Python.
- We also attempted histogram of gradients method[3], followed by affine transform, to detect, straighten and use non-frontal faces. It introduced a bluish color defect in the images due to inconsistency in the way Scikit-learn and OpenCV read images.

Obtaining face embeddings using FaceNet

- Schroff et al. at Google Inc. proposed a deep convolutional neural network to obtain face embedding, called as FaceNet.

Obtaining face embeddings using FaceNet

- Schroff et al. at Google Inc. proposed a deep convolutional neural network to obtain face embedding, called as FaceNet.
- The network has been trained on 100-200 million face thumbnails of 8 million different identities.

Obtaining face embeddings using FaceNet

- Schroff et al. at Google Inc. proposed a deep convolutional neural network to obtain face embedding, called as FaceNet.
- The network has been trained on 100-200 million face thumbnails of 8 million different identities.
- Due to computational limitations, we have downloaded and used trained model of one of the architectures of FaceNet called NN4. This network has been made open source by OpenFace group at Carnegie Melon University [2, 1].

Obtaining face embeddings using FaceNet

- Schroff et al. at Google Inc. proposed a deep convolutional neural network to obtain face embedding, called as FaceNet.
- The network has been trained on 100-200 million face thumbnails of 8 million different identities.
- Due to computational limitations, we have downloaded and used trained model of one of the architectures of FaceNet called NN4. This network has been made open source by OpenFace group at Carnegie Mellon University [2, 1].
- We have used two models of NN4 architecture (different in number of training images) which take face thumbnails of size 96x96 as input and generate face embeddings which are vectors of 128 length for each image.

Adoption of OpenFace trained net

- The neural net provided by OpenFace is in t7 format usable in Torch using the Lua language. For convenience of Python, we attempted to load the model in PyTorch. However, the SpatialConvolutionMM layer used in the model cannot be serialized for use in PyTorch.

Adoption of OpenFace trained net

- The neural net provided by OpenFace is in t7 format usable in Torch using the Lua language. For convenience of Python, we attempted to load the model in PyTorch. However, the SpatialConvolutionMM layer used in the model cannot be serialized for use in PyTorch.
- Pete Kim of Intel used 'torch-hdf5' module to obtain the model data as t7 and manually recoded the layers to provide an approximate PyTorch model on Github. This adaption had some bugs which we debugged and managed to run the PyTorch model.

Adoption of OpenFace trained net

- The neural net provided by OpenFace is in t7 format usable in Torch using the Lua language. For convenience of Python, we attempted to load the model in PyTorch. However, the SpatialConvolutionMM layer used in the model cannot be serialized for use in PyTorch.
- Pete Kim of Intel used 'torch-hdf5' module to obtain the model data as t7 and manually recoded the layers to provide an approximate PyTorch model on Github. This adaption had some bugs which we debugged and managed to run the PyTorch model.
- However, since this solution was not scalable to other architectures and networks of FaceNet, we proceeded to code in Lua using Torch for obtaining face embeddings.

Classification based on face embeddings

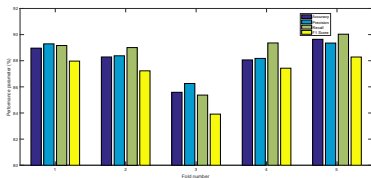
- After obtaining face embeddings from neural network, we perform classification using these values as 128 separate features. We have experimented with two different classifiers - k nearest neighbors and SVM.

Classification based on face embeddings

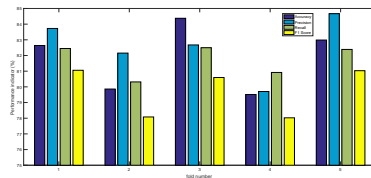
- After obtaining face embeddings from neural network, we perform classification using these values as 128 separate features. We have experimented with two different classifiers - k nearest neighbors and SVM.
- To avoid overfitting, we have used 5 fold cross validation in both classifiers.

Performance comparison

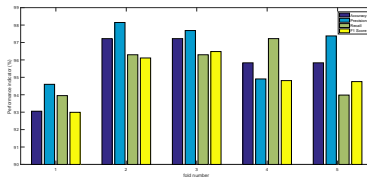
5 fold performance indicators



(a) Faces94



(b) Faces95



(c) Grimace

Figure: 5 fold performance indicators.

Performance comparison

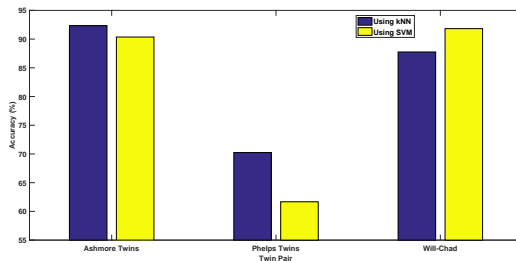
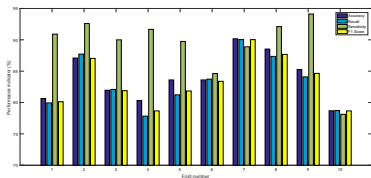
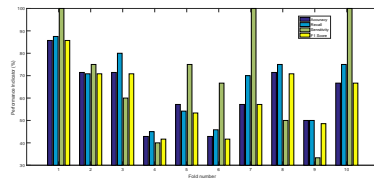


Figure: Accuracy using small NN4 model

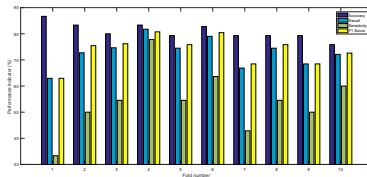
10 fold performance indicators



(a) Ashmore Twins



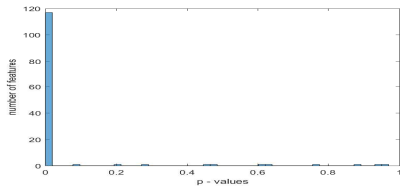
(b) Phelps Twins



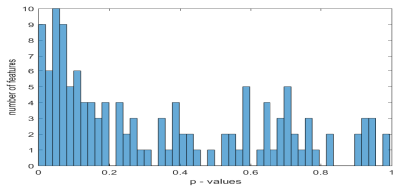
(c) Will Ferrell, Chad Smith

Figure: 10 fold performance indicators.

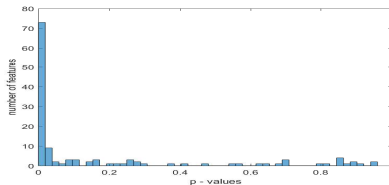
p-Values



(a) Ashmore Twins



(b) Phelps Twins



(c) Will Ferrell, Chad Smith

Figure: p-Values from Kruskal Wallis test.

Conclusion

- We implement a facial recognition system and test it on three different datasets for face recognition and images of three celebrity twins for identical twin identification.

Conclusion

- We implement a facial recognition system and test it on three different datasets for face recognition and images of three celebrity twins for identical twin identification.
- We perform a comparison between results of kNN and SVM classifiers and use two different models of FaceNet for the classification.

Conclusion

- We implement a facial recognition system and test it on three different datasets for face recognition and images of three celebrity twins for identical twin identification.
- We perform a comparison between results of kNN and SVM classifiers and use two different models of FaceNet for the classification.
- Finally we present a detailed analysis of the results using both classifiers and the two neural network models considered.

References I



Brandon Amos.

Openface.

<https://github.com/cmusatyalab/openface>.



Brandon Amos, Bartosz Ludwiczuk, and Mahadev Satyanarayanan.

Openface: A general-purpose face recognition library with mobile applications.

Technical report, CMU-CS-16-118, CMU School of Computer Science, 2016.



Navneet Dalal and Bill Triggs.

Histograms of oriented gradients for human detection.

In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 886–893. IEEE, 2005.



Florian Schroff, Dmitry Kalenichenko, and James Philbin.

Facenet: A unified embedding for face recognition and clustering.

In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 815–823, 2015.



Libor Spacek.

Faces94 dataset.

<http://cswwww.essex.ac.uk/mv/allfaces/faces94.html>.



Libor Spacek.

Faces94 dataset.

<http://cswwww.essex.ac.uk/mv/allfaces/faces95.html>.



Libor Spacek.

Faces94 dataset.

<http://cswwww.essex.ac.uk/mv/allfaces/grimace.html>.

References II



Paul Viola and Michael Jones.

Rapid object detection using a boosted cascade of simple features.

In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, volume 1, pages I-I. IEEE, 2001.