SCHULICH
School of Engineering
UNIVERSITY OF CALGARY
DEPARTMENT OF ELECTRICAL
AND COMPUTER ENGINEERING

ENEL 645 – Data Mining & Machine Learning

Final Project Report: Team 8

"Face Detection Using Siamese Network"

<u>**Submitted to:**</u>

Dr. Roberto Medeiros de Souza

Department of Electrical & Computer Engineering

University of Calgary

<u>**Submitted by:**</u>

Jasneet Kaur Chahal – UCID#30119845

Ahmad Elkholy – UCID#30120003

Ramneek Kaur – UCID#30138283

Lakshmi Teja Makineni – UCID#30126350

Muhammad Ammaar Raihan – UCID#30139255

Parth Satishkumar Shah – UCID#30136276

Team Members' Contribution:

All Members have contributed to the project in the following manner:

Jasneet, Ahmed, Ammaar, and Parth has contributed in building a model, training it with different epochs and batch size.

Ramneek and Lakshmi has helped with data collection and data pre-processing. They also helped in preparing the report.

Overall, it was a collectively team-effort to make this project. All the team members contributed equally towards it.

Score Table:

Name	Score
Jasneet Kaur Chahal	3
Ahmad Elkholy	3
Ramneek Kaur	3
Lakshmi Teja Makineni	3
Muhammad Ammaar Raihan	3
Parth Satishkumar Shah	3

GitHub Repository:

Below is the link for GitHub repository for all the source files and the data set:

https://github.com/jasneetchahal/ENEL-645-group-8.git

Below is the description of the relevant the folders on GitHub:

- 'Test_Images' folder contains the test dataset.
- 'Augmented_dataSet' folder contains the training dataset i.e., 12 images belonging to each team member.
- 'Dataset' folder contains the unaugment training data i.e., 4 images belonging to each team member.
- 'Original_dataset' contains the unaugmented training data set we initially used i.e 8 images per person.
- 'Original_Augmented_dataSet' folder contains the training data set of total 192 images in which we have used 3 transformations. It has 32 images per team member.
- 'Output' folder contains NumPy arrays and Lables – Unable to upload due to the limit on GitHub.
- 'src_files/Test_data_augmentation.ipynb' and 'src_files/Augmented_data.ipynb' includes the code for the generation of augmented test and the training data, respectively.
- 'src_files/SiameseModel.ipynb' includes the build Siamese model.
- 'src_files/DataPreProcessFile.ipynb' includes the code for the generation of the NumPy arrays required for training the model.
- 'src_files/Main.ipynb' includes the code for training the model and predicting the images.

ENEL 645 Data Mining & Machine Learning

Face Detection using Siamese Network

Jasneet Kaur[1], Ahmed Elkholy[2], Ramneek Kaur[3], Lakshmi Teja Makineni[4], Muhammad Ammaar Raihan[5], Parth Satishkumar Shah[6]
MEng Students, Department of Electrical and Computer Engineering, University of Calgary
Calgary, AB, CANADA

***Abstract* - Nowadays, computer-based face recognition is a mature and reliable mechanism that is significantly used in many access control scenarios along with other biometric methods. There has been huge growth in the smart mobile industry for using face verification. Perceiving faces is a difficult job however Siamese Networks (fake brain organizations) give a productive answer for the comparable issue articulations. Siamese networks are incredibly powerful networks, responsible for significant increases in face recognition, signature verification, and prescription pill identification applications**

Keywords: Siamese network, face recognition.

1. INTRODUCTION

A facial identification system is a technology that can match a human face from a digital image or a video frame against a database of faces. It works by locating and measuring facial features from a given image, and is commonly used to verify users through ID verification services. Facial identification is a difficult pattern identification problem in computing, despite the fact that humans can recognize faces without much effort. Based on a two-dimensional photograph, facial identification systems attempt to detect a three-dimensional human face that changes appearance with lighting and facial emotion. Facial identification systems go through four steps to complete this computational problem. The face is first segmented from the image background using face detection. The segmented face picture is aligned in the second step to account for face posture, image size, and photographic qualities like lighting and grayscale. The goal of the alignment method is to allow for accurate facial feature localization in the third step, facial feature extraction. To depict the face, features such as the eyes, nose, and mouth are located and measured in the image. The face's established feature vector is then matched against a database of faces in the fourth step [1].

Face identification is a practical requirement in the real world, primarily for human identification and surveillance. Face images provide a number of advantages over conventional biometrics (such as fingerprints and iris) [2]. The capture is non-intrusive and can be carried out from a distance. Facial identification systems are employed throughout the world today by governments and private companies.

2. MOTIVATION & SIGNIFICANCE

The scientific community, as well as the general public, has recently become increasingly interested in face identification. The general public's interest originates primarily from recent terrorist attacks throughout the world, which have boosted demand for useful security measures [3]. The above-mentioned security applications are far from the only ones that use facial identification.

The development of facial recognition technology has allowed security and police agencies to process photographs and videos from numerous sources, such as body-worn, smartphone, and in-vehicle camera systems, to identify people of interest swiftly and effectively in real-time, speeding up investigations.

3. METHODOLOGY

3.1. Creating Dataset

For this project, we created our own dataset. Everyone clicked 4 images of themselves; so, in total there were 48 original images. We performed data augmentation to artificially expand our existing dataset. Data augmentation is the process of modifying, or augmenting a dataset with additional data. We performed the following 2 transformations: (1) Rotating the image anti-clockwise (2) Blurring the Images. After this, we had total of 72 images.

For the test data set we have used 2 images per person and used transformations for data augmentation. The transformations used are clockwise 45-degree rotation, and anti-clockwise 45-degree rotation. Therefore, there are total of 6 images per persons and 36 images in the test data set. So, in total 72 images are added as the training dataset and 36 images are added as a test dataset.

Fig. 1. Examples of Augmented Images

3.2. Data Pre-Processing

In the wake of characterizing the datasets, the pictures are stacked and resized for the characterized component of (128, 128) pixels utilizing Python Imaging Library. This outcome looks like each picture as (128, 128, 3). To make a real preparation dataset, the characterized preparing pictures are then changed over to NumPy exhibits with the end goal that each picture is matched with each and every other picture and a label is appointed to each match. Since preparing dataset comprises of 72 pictures, the previously mentioned blend brings about preparing dataset of 5184 sets of pictures.

For images belonging to the same class label is assigned as 1 and it is 0 for every image of different class. The images of the data set stored in above characterized NumPy exhibits are x1.npy and x2.npy records. Each of the 'x1.npy' and 'x2.npy' contains a picture from each pair, to such an extent that nth picture from each file, makes the nth pair with corresponding label stored at the nth position in 'y.npy' record. Consequently, accordingly, we have 2 arrangements of pictures, X1 and X2 of shape (5184, 128, 128, 3) and 5184 labels.

3.3. Siamese Network

Neural networks are practically perfect at almost every activity in the present Deep learning era, but they rely on additional data to perform properly. However, we can't always rely on more data for specific problems like face recognition and signature verification; to handle these challenges, we have a new form of neural network architecture called Siamese Networks. To obtain better predictions, it only employs a small number of images. Siamese networks have been more popular in recent years due to their capacity to learn from relatively little data.

Figure 2 explains a basic Siamese network architecture.

- On the *left* we present two example images (from our dataset) to the Siamese model. Our goal is to determine if these images belong to the *same class(person)* or not.
- The *middle* shows the Siamese network itself. These two subnetworks have the *same* architecture and *same* parameters, and they *mirror* each other — if the weights in one subnetwork are updated, then the weights in the other subnetwork(s) are updated as well.
- Each subnetwork's output is a fully-connected (FC) layer. The Euclidean distance between these outputs is often computed and fed through a sigmoid activation to assess how similar the two input images are. Closer values of the sigmoid activation function to "1" indicate "more similar," while values closer to "0" indicate "less similar."

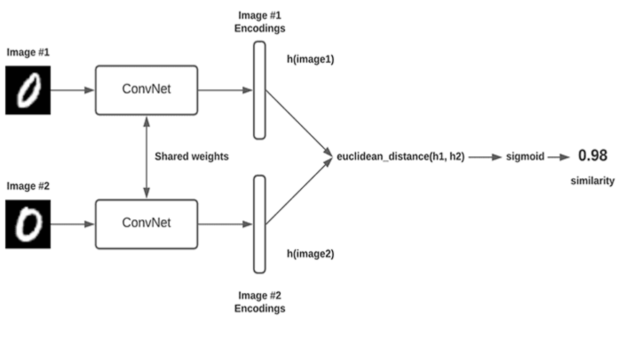

Fig. 2. A basic Siamese network architecture implementation accepts two input images (left), has identical CNN subnetworks for each input with each subnetwork ending in a fully-connected layer (middle), computes the Euclidean distance between the fully-connected layer outputs, and then passes the distance through a sigmoid activation function to determine similarity (right)

3.4. Training the Model

We have trained the model as defined in the image, and then tested the same on the data set. The test data set consists of 32 images from different classes, which are then passed to the model that has been trained to calculate the similarity score. The similarity scores are calculated corresponding to each image in the training dataset. The image is classified into the class for which the highest similarity score is observed.

We trained the model with different batch size and epochs. The results in the report are achieved with a batch size of 7 and 100 epochs. The performance of the model is also noted by varying the split ratio of training and validation data, and the model performed best when the ratio of training and validation data is 4000:1184.

4. RESULTS

Each pair of images in the training dataset has a label associated with it. If both images in a pair are from the same class, the label is 1, otherwise it is zero. The images are saved in x1.npy and x2.npy files as NumPy arrays. Each of the files 'x1.npy' and 'x2.npy' contains an image from each pair, so that the n^{th} image from each file forms the nth pair with the appropriate label placed at the nth position in the 'y.npy' file.

These three files are successfully getting created on the execution of the program. Based on these files, the model is trained and tested. We tested our model for 36 test images. Corresponding results shows the best confidence scores of each image along with the class into which the image is classified.

4.1. Predicted vs Actual images along with confidence scores

The confidence score is the score calculated by the model between the predicted image and the actual image. The figure below describes the labels of the class which belongs to the particular image against the image that has been predicted by the model.

```
IMAGE 5 is 1 with confidence of 0.2508563697338104
IMAGE 5 is 3 with confidence of 0.9203009009361267
IMAGE 5 is 5 with confidence of 0.8605433106422424
IMAGE 5 is 2 with confidence of 0.34786778688430786
IMAGE 5 is 2 with confidence of 0.776971161365509
IMAGE 5 is 2 with confidence of 0.8814347386360168
IMAGE 4 is 3 with confidence of 0.6120381355285645
IMAGE 4 is 3 with confidence of 0.9455202221870422
IMAGE 4 is 4 with confidence of 0.542770266532898
IMAGE 4 is 4 with confidence of 0.937578022480011
IMAGE 4 is 3 with confidence of 0.7609069347381592
IMAGE 4 is 4 with confidence of 0.8488909602165222
IMAGE 3 is 0 with confidence of 0.2718435227870941
IMAGE 3 is 0 with confidence of 0.34591159224510193
IMAGE 3 is 3 with confidence of 0.868130624294281
IMAGE 3 is 3 with confidence of 0.9010767340660095
IMAGE 3 is 5 with confidence of 0.45263078808784485
IMAGE 3 is 3 with confidence of 0.3800944983959198
IMAGE 2 is 2 with confidence of 0.49752742052078247
IMAGE 2 is 5 with confidence of 0.5349675416946411
IMAGE 2 is 2 with confidence of 0.6718438267707825
IMAGE 2 is 2 with confidence of 0.8331968188285828
IMAGE 2 is 2 with confidence of 0.5600950717926025
IMAGE 2 is 2 with confidence of 0.8245675563812256
```

```
IMAGE 1 is 1 with confidence of 0.7819284796714783
IMAGE 1 is 1 with confidence of 0.7782650589942932
IMAGE 1 is 1 with confidence of 0.8425878286361694
IMAGE 1 is 1 with confidence of 0.9382176995277405
IMAGE 1 is 2 with confidence of 0.7749021053314209
IMAGE 1 is 5 with confidence of 0.806168794631958
IMAGE 0 is 2 with confidence of 0.7616654634475708
IMAGE 0 is 2 with confidence of 0.8978510499000549
IMAGE 0 is 2 with confidence of 0.7474689483642578
IMAGE 0 is 2 with confidence of 0.3844742774963379
IMAGE 0 is 2 with confidence of 0.7975437045097351
IMAGE 0 is 0 with confidence of 0.7412831783294678
```

Fig. 3. Confidence Score

4.2. *Confusion Matrix*

A confusion matrix is a summary of classification problem prediction outcomes. The number of right and unsuccessful predictions is summarized and broken down by class using count values. The confusion matrix's key is this. The confusion matrix depicts the various ways in which your classification model becomes perplexed when making predictions.

The confusion matrix shows the true and false labels corresponding to different classes. Class Labels 0,1,2,3,4,5 corresponds to the images of 6 people in the group i.e., Ahmad, Ammaar, Jasneet, Lakshmi, Parth and Ramneek.

Class_Lable	**0**	**1**	**2**	**3**	**4**	**5**
0	1	0	5	0	0	0
1	0	4	1	0	0	1
2	0	0	5	0	0	1
3	2	0	0	3	0	1
4	0	0	0	3	3	0
5	0	1	3	1	0	1

Table 1. Class Lables

```
Confusion Matrix

[[1 0 5 0 0 0]
 [0 4 1 0 0 1]
 [0 0 5 0 0 1]
 [2 0 0 3 0 1]
 [0 0 0 3 3 0]
 [0 1 3 1 0 1]]

Accuracy: 0.47

Micro Precision: 0.47
Micro Recall: 0.47
Micro F1-score: 0.47
```

Fig. 4. Confusion Matrix

In the matrix, cell with the same row and same column represents the correctly classified images of the person corresponding to that label.

4.3. *Classification Report*

Below image represents the classification. First five rows give the precision, recall, f1-score for the individual classes. 'Support' represents the number of test images belonging to a particular class. Last three rows in the image gives the stats corresponding to the micro, macro, and average precision, recall, and f1-scores. So, according to the micro scores, overall accuracy of the model achieved is 47%.

```
              precision    recall  f1-score   support

       Ahmad       0.33      0.17      0.22         6
      Ammaar       0.80      0.67      0.73         6
     Jasneet       0.36      0.83      0.50         6
     Lakshmi       0.43      0.50      0.46         6
       Parth       1.00      0.50      0.67         6
     Ramneek       0.25      0.17      0.20         6

    accuracy                           0.47        36
   macro avg       0.53      0.47      0.46        36
weighted avg       0.53      0.47      0.46        36
```

Fig. 5. Classification Report

4.4. *Actual vs Predicted Images*

Below images show the custom images and the predicted images. Custom images represent the test dataset and predicted images are the images which got the highest score among all the images of the trained dataset.

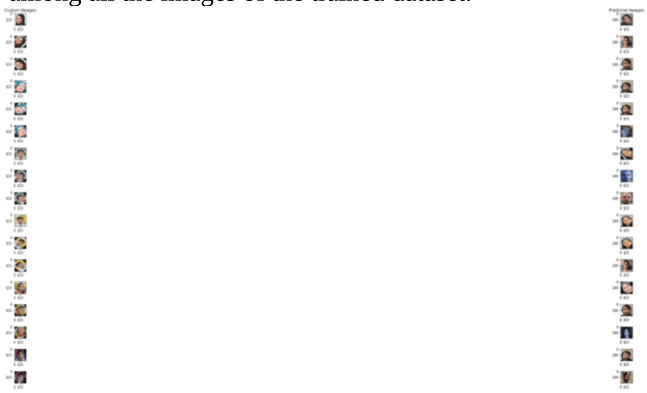

Fig. 6. Actual vs Predicted Images

5. Challenges Faced

Google Collab has a limit on the number of GPU runtime resources we can use, thus training a model with different number of possibilities for longer duration was difficult.

We initially started with the data set of 8 images per person, however we could not train our model with such a huge number of images due to the limitation of GPU runtime resources.

6. Conclusion

Overall, our results are not as good as we had hoped. Despite trying multiple different configurations for the model and the dataset, our best result only amounted to a 47% accuracy. Had we more time, I believe all of us would like to approach this project one more time and try to achieve a better result. However, in consideration of all of our workload this semester with the rest of the courses, we believe we did our best given the challenges and the time given to us.

ACKNOWLEDGMENT

ENEL 645 Data Mining & Machine Learning

We would like to express our gratitude to Professor Dr. Roberto Medeiros de Souza, our course instructor for Data Mining & Machine Learning, for his teachings and guidance all through the semester. We would also like to appreciate the efforts of our course TAs Michael Lasby and Youssef Beauferris.

REFERENCES

[1] Niranjani, N., Tharmila, B., Sukirtha, C., Kamalraj, K., & Thanujan, S. (2017). *The real time face detection and recognition system*.

[2] Khalil, T. S., & Jilani, S. A. K. (2016, December). *Face Recognition using Siamese Architecture and Convolutional Neural Networks*. IJSETR. Retrieved April 3, 2022.

[3] Saloman, G., Britto, A., Vareto, R. H., & Manotti, D. (2021, May 14). *Open-set Face Recognition for Small Galleries Using Siamese Networks*. arXiv. Retrieved April 3, 2022.

[4] https://medium.com/@enoshshr/learning-similarity-with-siamese-neural-networks

[5] https://github.com/adityajn105/Face-Recognition-Siamese-Network

[6] https://github.com/shubham0204/Face_Recognition_with_TF

[7] https://medium.com/predict/face-recognition-from-scratch-using-siamese-networks-andtensorflowdf03e32f8cd0

[8] https://en.wikipedia.org/wiki/Siamese_neural_network

[9] https://towardsdatascience.com/train-validation-and-test-sets-72cb40cba9e7

[10] https://towardsdatascience.com/siamese-networks-line-by-line-explanation-for-beginners55b8be1d2fc6

[11] https://towardsdatascience.com/confusion-matrix-for-your-multi-class-machine-learning-modelff9aa3bf7826

[12] https://towardsdatascience.com/image-augmentation-using-python-numpy-opencv-and-skimageef027e9898da