Real-Time Facial Recognition System

Name: Anuj Agrawal

Resources used:

1. https://www.coursera.org/learn/neural-networks-deep-learning?specialization=deep-learning
2. https://www.coursera.org/learn/convolutional-neural-networks/home/week/4
3. https://www.coursera.org/learn/advanced-learning-algorithms?specialization=machine-learning-introduction
4. Research paper-Siamese Neural Networks for One-shot Image Recognition(Authors-Gregory Koch, Richard Zemel, Ruslan Salakhutdinov)

Topics that I learnt:

- Intro to Deep Learning
- Neural Network basics
- Shallow neural network
- Deep neural network
- Neural Network training
- Decision trees
- Foundation of CNN
- Siamese Network

About the model:

I have added all the necessary comments in the code file. So, you will get the entire idea about my code from there. But some images in the comments may not be visible on that uploaded file because it opens on drive. They are visible on the local Jupyter notebook.
So I am attaching those 2 images in the comments on that code file here:

Image 1:

Fig. Top 2 rows have a set of 10 faces of the same person. Bottom 2 rows have faces of 10 different people shown. Pictures taken from Olivetti Faces from sklearn.datasets API. Image by Girija Behera

Image 2:

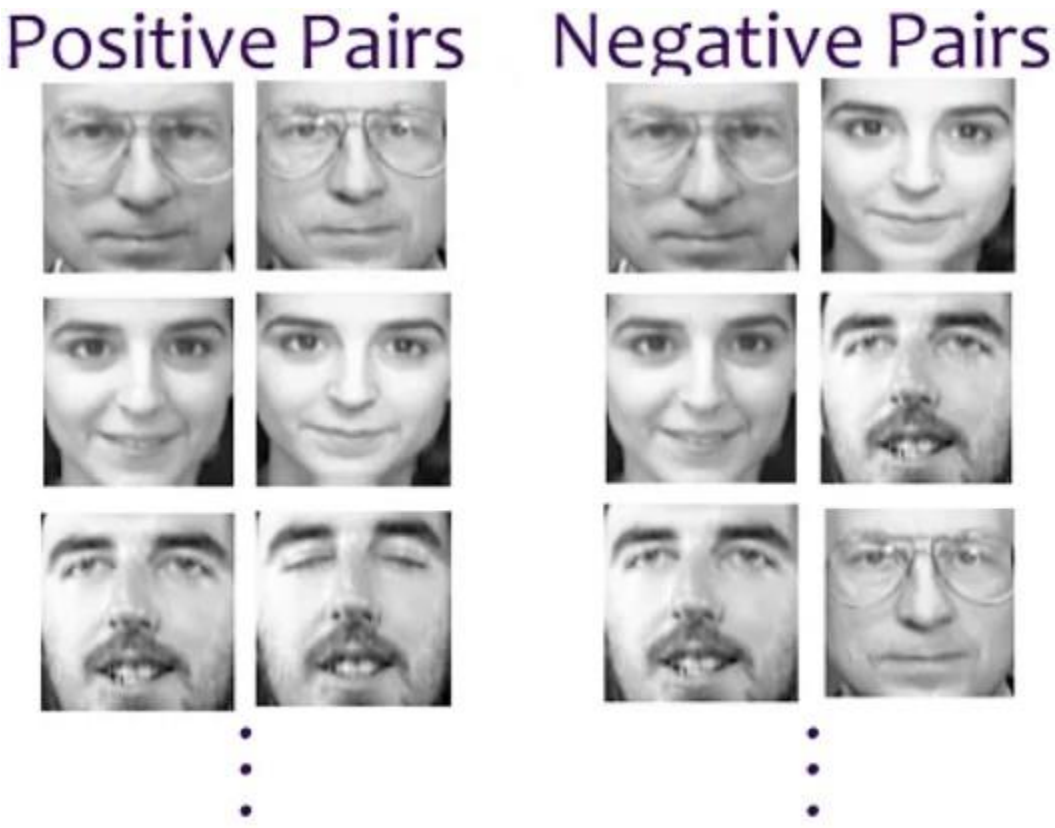

Fig. Sample Image Pairs generated for Siamese Network Training

Siamese Neural Network for Face Recognition: Project Report

Overview

This project involves developing a Siamese Neural Network for a face recognition system using the Olivetti face dataset. The dataset comprises 400 grayscale images of 40 individuals, with 10 images per person. The network is designed to compare pairs of images and determine whether they depict the same person.

Network Architecture

Feature Extractor Model

- **Input**: Images of shape (64, 64, 1).
- **Layers**: The model consists of three sets of Convolutional (Conv2D) layers, each followed by MaxPooling and Dropout for feature extraction.
  - Conv2D layers with 96, 256, and 384 filters of sizes 11x11, 5x5, and 3x3, respectively.
  - MaxPooling2D layers with a pool size of (2, 2).
  - Dropout layers with a rate of 0.3 to reduce overfitting.
- **Output**: Dense layer with 128 nodes, acting as the embedding layer for the input image.

Siamese Network Model

-**Inputs**: Pairs of images, each processed by the feature extractor model.
-**Distance Calculation**: Euclidean distance between the feature vectors of the two input images, calculated using a Lambda layer.
- **Output**: Sigmoid activation function applied to the distance, yielding a probability score indicating the likelihood of the images being of the same person.

Training and Validation

- **Dataset Generation**: Pairs of images are generated, labeled as '1' for images of the same person (positive pair) and '0' for images of different people (negative pair).
- **Model Compilation**: The model is compiled with binary cross-entropy loss and the Adam optimizer.
- **Training**: The model is trained for 100 epochs with a batch size of 64.

Results Analysis

- **Performance**: The validation loss decreases consistently up to 80 epochs, then stabilizes. The validation accuracy shows moderate improvement.
- **Thresholding**: Based on observations, a similarity threshold of 0.65 is set to classify whether pairs are of the same person.
- **Limitations**: The model shows some false positives, indicating room for improvement.

Model Improvement Strategies

1. **Convolutional Layers**: Experiment with filter sizes and counts.
2. **Additional Layers**: Consider adding more convolutional layers.
3. **Batch Normalization**: Incorporate BatchNormalization layers.
4. **Dropout Rate Tuning**: Experiment with different dropout rates.

5. **Advanced Layers**: Explore more advanced layers like Residual Connections.
6. **Dense Layer Tuning**: Adjust the number of neurons in Dense layers.
7. **Learning Rate Scheduling**: Implement adaptive learning rate optimizers.
8. **Data Augmentation**: Increase training data size with variations.
9. **Regularization**: Apply L1 or L2 regularization in layers.
10. **Hyperparameter Optimization**: Use techniques like grid search for tuning.
11. **Activation Functions**: Evaluate other activations like LeakyReLU or ELU.
12. **Model Complexity**: Increase complexity carefully to capture patterns.
13. **Transfer Learning**: Use pretrained models for transfer learning.

Benefits of Siamese Networks

- **Flexibility**: New data can be added without retraining the entire model.
- **Efficiency**: Suitable for datasets with a limited number of images per class.

Conclusion

The Siamese Neural Network shows promise in face recognition tasks with limited data. However, the current model needs further tuning and experimentation to improve accuracy and reduce false positives. Advanced techniques and careful hyperparameter optimization may enhance its performance significantly.