Real-Time Face Verification Using Siamese Neural Networks

ECE-GY 7123 Deep Learning
Shanmuk Reddy, Nathaniel Sehati

New York University Tandon School of Engineering
6 MetroTech Center
Brooklyn, New York 11201 USA

Abstract

This paper details a comprehensive approach to building and implementing a Siamese Network for facial verification, illustrating the integration of deep learning techniques to enhance verification accuracy. The discussion includes the network design, data preparation, model training, and the evaluation framework, showcasing the practical application of advanced computational methods in improving security systems.

Code Availability: https://github.com/sk11331/DeepLearning-FinalProject/tree/main.

Introduction

Facial verification systems are crucial in various security applications, requiring high accuracy and robustness against diverse conditions. This project develops a Siamese Network designed for facial verification, utilizing a structured approach to model training and evaluation that emphasizes the model's ability to learn detailed facial features for accurate verification.

Literature Survey

This section reviews relevant literature on deep learning-based face recognition, focusing particularly on the use of Siamese networks. Recent advancements have demonstrated significant promise in employing Siamese networks for face verification without the need for extensively labeled datasets. According to Solomon, Woubie, and Emiru (2024), Siamese networks can be effectively trained using unsupervised methods to generate training pairs, which significantly alleviates the challenge of acquiring large labeled datasets. Their approach utilizes a double-branch network with a VGG encoder to process both negative and positive training pairs, optimizing the training process through binary cross-entropy loss without relying on labeled data.

Our project builds upon these findings by implementing a Siamese network that also aims to reduce dependence on labeled data. However, unlike the approach taken by Solomon, Woubie, and Emiru (2024), which generates training pairs entirely in an unsupervised manner, our method incorporates a hybrid approach that uses a minimal amount of labeled data to enhance the precision of the training process. This modification is intended to strike a balance between the unsupervised flexibility and the accuracy afforded by supervised methods.

Comparison with Existing Methods

The study by Solomon, Woubie, and Emiru (2024) highlights that their unsupervised Siamese network achieves comparable performance to supervised baselines on the Labeled Faces in the Wild (LFW) dataset. Our experimental results aim to explore whether the inclusion of a limited set of labeled data can further bridge the performance gap typically observed in purely unsupervised systems.

System Overview

The implemented system encompasses several components: data collection, preprocessing, model architecture design, training, and testing. Each component is crucial in ensuring the effectiveness of the facial verification system.

Data Collection and Management

Data for this project is captured in real-time using a webcam, programmed through OpenCV. This setup is critical as it allows for the collection of a dataset that is representative of practical scenarios, encompassing a variety of lighting conditions, facial expressions, and angles. Specifically, the Python script initiates the camera, captures frames on keypress events, and saves images into designated folders for anchors and positives. This ensures that the data not only feeds the model with real-world variability but also categorically segregates the data to facilitate differential learning specific to the Siamese architecture.

Additionally, pre-loaded negative images are sourced from the 'Labeled Faces in the Wild' (LFW) dataset of Massachusetts Amherst (2024). This dataset provides a broad range of faces, which helps in refining the model's ability to differentiate between non-identical pairs, enhancing the robustness of the facial verification system.

For the initial testing phase of this project, we limited the dataset to roughly 200 images for anchors and positives due to constraints in time, resources, and the experimental scope of our study. This limited data set is intended to provide preliminary insights into the model's effectiveness, with plans to expand the data breadth in future work to further evaluate and enhance the model's performance.

Data Preprocessing

Preprocessing the input data ensures consistency across all training and validation samples. The images are resized to 100x100 pixels and pixel values normalized to a range of [0, 1], enhancing model stability and training speed. This implementation using TensorFlow's API is crucial for the model's training efficiency and helps in achieving faster convergence during the learning process.

Model Architecture

The Siamese Network is structured as a twin network, where two separate input streams share weights, an approach that ensures both sides of the network extract similar kinds of features from each input pair. This enables the model to accurately compare and contrast the extracted features from the paired input images.

Detailed Network Architecture The core of the Siamese Network consists of multiple convolutional blocks, each designed to perform feature extraction at various levels of abstraction. Each block comprises a convolutional layer followed by batch normalization and max-pooling. This specific arrangement allows the network to adapt to the intricacies and variations in facial features.

Following feature extraction, the network employs dense layers that aim to integrate these features into a more abstract representation. Dropout layers interspersed between these dense layers serve to mitigate overfitting by randomly dropping units during training, thus ensuring that the model generalizes well to new, unseen data.

Siamese Network Diagram Figure 1 illustrates the structure of the Siamese Network used for facial recognition. The network takes a pair of images as input and processes them through identical convolutional networks. These networks encode the images into feature vectors which are then compared using a distance function. The outcome of this comparison is a similarity score, which quantifies the likelihood of the images belonging to the same person (Hamdani et al. 2023).

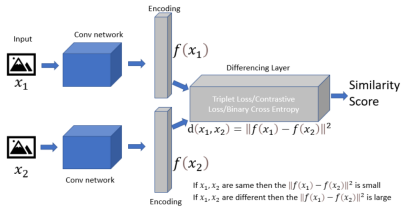

Figure 1: Illustration of the Siamese Network architecture

Embedded Network Architecture The embedded model within the Siamese Network is intricately designed to extract and process facial features from the input images. This model is crucial for generating embeddings that facilitate the comparison between pairs of images.

Architecture Details The embedded model is constructed with sequential layers that incrementally process the input image to extract increasingly abstract features. Here is a breakdown of the architecture and its components:

- **Input Layer:** The model starts with an input layer that accepts images of size $100 \times 100 \times 3$, representing the height, width, and color channels respectively.
- **Convolutional Layers:** The architecture includes four convolutional layers. The first layer has 128 filters of size 7×7, followed by layers with 256 and two layers with 292 filters of size 3×3.
- **Batch Normalization:** Each convolutional layer is followed by a batch normalization layer, which helps in stabilizing the learning process by normalizing the activations of the previous layer.
- **Max Pooling Layers:** Subsequent to each batch normalization, max pooling is applied with a pool size of 2×2. This reduces the spatial dimensions of the output from the previous layer, thus reducing the number of parameters and computation in the network.
- **Flattening Layer:** Post convolutional operations, the feature maps are flattened into a single vector, preparing the model to transition to fully connected layers.
- **Dense and Dropout Layers:** The flattened output is then fed into a series of dense layers with dropout layers in between to prevent overfitting. The dense layers are configured with 2048 and 4096 neurons, enriched with the ReLU activation function, providing the necessary computational power to learn complex facial representations.

This structured layer configuration ensures that the network learns detailed and robust feature representations required for accurate facial verification. Below is the summary of the model's architecture, illustrating the sequential and interconnected nature of the layers:

Model: "embedded_model_v2"

Layer (type)	Output Shape	Param #
input_image (InputLayer)	[(None, 100, 100, 3)]	0
conv2d_4 (Conv2D)	(None, 100, 100, 128)	18944
batch_normalization_4 (BatchNormalization)	(None, 100, 100, 128)	512
max_pooling2d_4 (MaxPooling2D)	(None, 50, 50, 128)	0
conv2d_5 (Conv2D)	(None, 50, 50, 256)	819456
batch_normalization_5 (BatchNormalization)	(None, 50, 50, 256)	1024
max_pooling2d_5 (MaxPooling2D)	(None, 25, 25, 256)	0
conv2d_6 (Conv2D)	(None, 25, 25, 292)	673060
batch_normalization_6 (BatchNormalization)	(None, 25, 25, 292)	1168
max_pooling2d_6 (MaxPooling2D)	(None, 13, 13, 292)	0
conv2d_7 (Conv2D)	(None, 13, 13, 292)	767668
batch_normalization_7 (BatchNormalization)	(None, 13, 13, 292)	1168
max_pooling2d_7 (MaxPooling2D)	(None, 7, 7, 292)	0
flatten_1 (Flatten)	(None, 14308)	0
dense_4 (Dense)	(None, 2048)	29304832
dropout_2 (Dropout)	(None, 2048)	0
dense_5 (Dense)	(None, 2048)	4196352
dropout_3 (Dropout)	(None, 2048)	0
dense_6 (Dense)	(None, 4096)	8392704

Total params: 44,176,888
Trainable params: 44,174,952
Non-trainable params: 1,936

Figure 2: Architecture of the Embedded Model ('embedded model v2') used in the Siamese Network.

Embedding and Distance Calculation After processing through the convolutional and dense layers, the network computes embeddings for each image. These embeddings are vectors in a high-dimensional space, where similar images cluster closer together. The L1 distance (absolute difference) between these embeddings is then calculated. This distance metric is critical as it quantifies the similarity between the paired images, forming the basis for the final verification decision.

Incorporating Validation Image

The following sample validation image (Figure 3) illustrates the type of input processed by our model. This image represents the data's real-world complexity and variability, highlighting the challenges in facial verification tasks that our Siamese Network addresses.

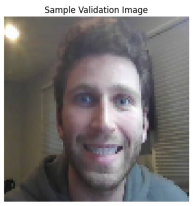

Figure 3: Sample Validation Image used in the Siamese Network for facial verification.

Training, Evaluation and Results

The training process of the Siamese Network involves adjusting model parameters to minimize the binary cross-entropy loss. This loss function is pivotal in teaching the model to accurately distinguish between pairs of images that depict either the same person or different persons.

Training Procedure

Training the Siamese Network is executed over multiple epochs, where each epoch processes mini-batches of data. This approach is beneficial for handling large datasets and also helps in stabilizing the gradient descent optimization by providing a more frequent update of the weights, potentially leading to a more robust convergence.

During each epoch, the network's parameters are updated by calculating gradients of the binary cross-entropy loss function. This function measures how well the network is performing at distinguishing between matched and unmatched pairs of images. If the labels and the predictions are close, the loss is small; if they are far apart, the loss is large.

The model is trained iteratively using mini-batches, with performance regularly assessed on a validation set to monitor progress and adjust hyperparameters accordingly. A typical training operation involves:

- Unpacking the batch data into inputs and labels.
- Performing a forward pass to generate predictions.
- Computing the loss between predictions and actual labels.
- Calculating gradients and updating the model weights using the optimizer.

Performance Metrics

To evaluate the model's effectiveness, several metrics are employed during training and testing phases. These metrics provide a comprehensive view of the model's performance and its ability to generalize to new data.

- **Loss:** The model's error rate or cost computed during the training process. A lower loss value indicates better performance, as it signifies that the model's predictions are closer to the actual values.
- **Accuracy:** Measures the proportion of true results (both true positives and true negatives) among the total number of cases examined. High accuracy means that the model can correctly identify both matching and non-matching pairs.
- **Precision:** Indicates the proportion of positive identifications that were actually correct. This metric is crucial for situations where the cost of a false positive is high.

Each of these metrics is crucial for understanding different aspects of the model's performance. The comprehensive use of these metrics ensures a balanced approach to evaluating the Siamese Network, reflecting its practical effectiveness in facial verification tasks.

Model Evaluation and Tuning

During training, the model's performance is regularly assessed on a validation set to monitor progress and adjust hyperparameters accordingly. This continuous evaluation helps in detecting overfitting early and ensures that the model generalizes well to new, unseen data.

The model's evaluation extends beyond simple loss metrics, encompassing precision, accuracy, and a detailed analysis of prediction outcomes. After training, the model is rigorously tested with unseen data to confirm its ability to accurately differentiate between similar and dissimilar facial pairs. This phase involves predicting similarities or differences between facial pairs as demonstrated in the test and validation images.

Testing and Validation Images

Figure 4 shows a sample test input and validation image used during the model's evaluation phase. These images illustrate typical inputs the model processes and provide a visual representation of the model's predictions versus actual data.

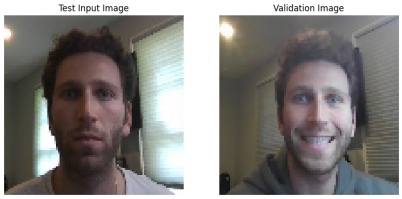

Figure 4: Example of a test input image and a validation image with model predictions.

Quantitative Performance Metrics The following sample predictions highlight the model's precision in distinguishing between pairs of faces, specifically the pair in figure 5.

```
Sample prediction from loaded model: [[9.9971467e-01]
 [1.1072530e-09]
 [8.8626254e-01]
 [8.8107300e-01]
 [3.0504829e-10]
 [5.3446425e-10]
 [9.9970871e-01]
 [4.9928939e-10]
 [8.1070393e-01]
 [9.9946934e-01]
 [3.6127465e-10]
 [9.9969113e-01]
 [8.1992638e-01]
 [9.9942279e-01]
 [1.6030067e-09]
 [3.4475500e-10]]
```

Figure 5: Visualization of sample predictions showing high confidence in both similarity and dissimilarity between facial pairs.

The predictions close to 1 indicate a high confidence in the similarity between the compared faces, suggesting they are likely the same person. Conversely, predictions near zero signify a high confidence in their dissimilarity. These results are visually supported by the test and validation images, providing a direct correlation between the model's numerical outputs and the visual inputs.

Results: Real-Time Authentication

The real-time authentication functionality of the model is showcased through a user interface that captures live video input. The system processes this input to verify the identity of an individual by comparing the captured image against a dataset of known faces. Figure 6 illustrates the application interface during an authentication attempt.

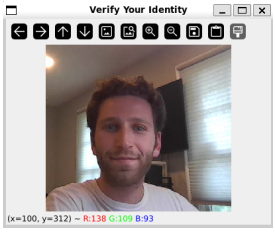

Figure 6: Real-time authentication interface

The authentication process dynamically outputs results, including detailed L1 distance statistics such as average, minimum, and maximum values for each authentication attempt. These results are demonstrated in Figure 7, where each attempt is labeled as either "Recognized!" or "Recognition Failed". Each attempt corresponds to different scenarios: the first with the original user smiling, the second with a printed picture of a person, the third with a different user smiling, and the fourth with the original user exhibiting a different facial expression.

```
Recognized!
Average L1 Distance: 0.988, Min: 0.979, Max: 0.994
Recognition Failed
Average L1 Distance: 0.002, Min: 0.000, Max: 0.004
Recognition Failed
Average L1 Distance: 0.545, Min: 0.377, Max: 0.698
Recognized!
Average L1 Distance: 0.987, Min: 0.979, Max: 0.992
```

Figure 7: Sample output from real-time authentication attempts, showing L1 distance statistics.

Model Limitations and Areas for Improvement

While the model performs well under controlled conditions, the system occasionally reports false positives or negatives, influenced by variations in lighting, facial expressions, and other environmental factors. This is partly due to the limited diversity in the training dataset and the number of epochs used during the training phase. Increasing the dataset size and diversity, improving model accuracy/complexity, and extending the training duration could improve the model's robustness.

Another significant factor is the model's complexity. The current architecture, while sufficient for basic verification tasks, may benefit from integrating more advanced neural network techniques, such as attention mechanisms or GANs, to enhance its discriminative capabilities.

Further, implementation of adaptive thresholding mechanisms for decision-making could reduce the rate of false positives and negatives, making the system more adaptable to varying user conditions, thereby increasing its practical applicability.

Conclusion

This paper presents the implementation of a Siamese Network model for facial verification, demonstrating promising results in real-time identity verification while highlighting the integration of deep learning techniques. Despite high accuracy in controlled environments, the model's performance under varied conditions reveals its sensitivity to environmental factors such as lighting and facial expressions.

Future enhancements should focus on refining the network architecture by integrating advanced techniques to improve feature extraction, expanding the training dataset to cover a broader range of scenarios, and optimizing the training process for better generalization. These steps are aimed at enhancing the model's robustness, making it more suitable for diverse real-world applications in facial verification.

References

Hamdani, N.; Bousahba, N.; Bousbai, A.; and Braikia, A. 2023. Face Detection and Recognition using Siamese Neural Network.

of Massachusetts Amherst, U. 2024. Labeled Faces in the Wild. https://vis-www.cs.umass.edu/lfw/. Accessed: 2024-05-13.

Solomon, E.; Woubie, A.; and Emiru, E. S. 2024. Deep Learning Based Face Recognition Method using Siamese Network. *arXiv preprint arXiv:2312.14001*. Version 2.