Inference Report: Intelligent Face Recognition System using Siamese Networks

Md Zohaib

May 05, 2025

1 Introduction

This report presents the inference results of the **Intelligent Face Recognition System using Siamese Networks**, developed to achieve real-time identity verification. The objective is to create a system that detects faces, compares them against known identities using similarity scores, and displays results through a Flask-based web application. The system leverages a Siamese Neural Network trained on the Labeled Faces in the Wild (LFW) dataset, achieving a training loss of 0.07 and a test accuracy of 78.30%.

2 Methodology

The system comprises three main components:

- **Face Detection**: Utilizes OpenCV's Haar Cascade classifier for real-time face detection from webcam feeds and uploaded images.
- **Siamese Neural Network**: A custom convolutional neural network (CNN) with three convolutional layers (64, 128, 256 filters) followed by fully connected layers, producing 256-dimensional embeddings. The network is trained using contrastive loss (margin=2.0) to minimize the distance between embeddings of the same person and maximize it for different people.
- **Flask Web Application**: A user-friendly interface for uploading image pairs, registering known identities, and streaming webcam feeds, displaying bounding boxes, similarity scores, and identity labels.

The LFW dataset, containing over 13,000 images of 5,749 individuals, is used for training and testing. The dataset is preprocessed with face alignment (using Dlib's 68-landmark predictor) and normalized to 100x100 pixels.

3 Training

The Siamese Network was trained for 10 epochs on the LFW dataset's `pairs.csv` using the `LFWDataset` class (defined in `utils.py`). The training configuration is as follows:

- **Batch Size**: 32
- **Optimizer**: Adam (learning rate = 0.001)
- **Loss Function**: Contrastive loss (margin = 2.0)
- **Device**: MPS (Apple Silicon) or CPU

The training loss decreased steadily, reaching **0.07** by the 10th epoch, indicating effective learning of discriminative embeddings. The model was saved as `siamese_model.pth` in the `models/` directory.

4 Testing

The trained model was evaluated on a test set derived from `matchpairsDevTest.csv` and `mismatchpairsDevTest.csv`, processed by `create_test_pairs.py` into `test_pairs.csv`. The test dataset contains pairs of images labeled as same (1) or different (0). The evaluation used a Euclidean distance threshold of 0.5.

Key metrics:

- **Accuracy**: 78.30%
- **Precision, Recall, F1-Score**: Computed and visualized (see `sample_predictions.png`)
- **ROC Curve**: Generated with an Area Under Curve (AUC) plotted in `roc_curve.png`.

The test accuracy of 78.30% demonstrates the model's ability to distinguish between same and different individuals effectively, though there is room for improvement in handling challenging cases (e.g., lighting variations).

5 Real-Time Inference

The system supports real-time inference through two modes:

1. **Image Upload**: Users upload two images via `upload.html`, and the system detects faces, computes embeddings, and displays similarity scores and predictions (same/different) in `result.html`. Known identities are matched against `embeddings.pkl`, showing names or ”Unknown.”
2. **Webcam Streaming**: The `webcam.html` interface streams live video, detecting faces and labeling them with names and similarity scores in real-time, meeting the project's real-time verification goal.

The Flask app, implemented in `app.py`, integrates OpenCV for face detection and PyTorch for embedding computation, providing a seamless user experience.

6 Conclusion

The Intelligent Face Recognition System successfully achieves real-time identity verification using a Siamese Neural Network, with a training loss of 0.07 and a test accuracy of 78.30%. The Flask-based web app provides an intuitive interface for image-based and webcam-based inference, meeting the project objectives.

Future improvements include:

- Enhancing face detection with MTCNN for better accuracy.
- Supporting video uploads for dynamic verification.
- Optimizing the model with transfer learning (e.g., ResNet backbone).
- Adding a statistics dashboard to track matching performance.

This project demonstrates the potential of Siamese Networks for face recognition and lays a strong foundation for further advancements in biometric systems.