Robust Face Recognition Under Challenging Visual Conditions Using Deep Learning

Team:Code_Slayers

Date: June 29, 2025

Abstract

Face recognition models often perform poorly in non-ideal conditions such as blur, fog, rain, overexposure, and low lighting. To tackle this, we designed a deep learning pipeline capable of robust face recognition even under such distorted scenarios. The dataset includes original and synthetically distorted images labeled by identity, enabling the system to learn invariant features. Using PyTorch, Scikit-learn, PIL, NumPy, and tqdm, we trained and evaluated our model on a structured dataset containing facial images and various distortion variants. Our model achieves strong recognition performance, demonstrating resilience to severe image degradation.

1. Introduction

Face recognition has rapidly evolved with deep learning, yet its performance significantly degrades in the presence of visual distortions. In real-world applications, surveillance systems, biometric access controls, and forensic analysis must operate under diverse environmental conditions. This research focuses on building a face recognition model that retains identity recognition capability across distorted visual inputs. The task was part of a broader challenge to evaluate model robustness under non-ideal settings.

2. Dataset Description

The dataset was divided into two primary folders: train and val. Each folder contained subfolders named in the format 001frontal, 002frontal, and so on. These subfolders include:
- A clean reference image of a subject (frontal face).
- A "distortion" subfolder containing distorted images of the same identity, simulating various conditions like blur, fog, rain, etc.

This structure allowed supervised learning of identity classification while explicitly testing model robustness to image degradation.

3. Methodology

We implemented our face recognition system using a combination of PyTorch, Scikit-learn, PIL, NumPy, and tqdm. The pipeline was designed to handle distorted inputs and generalize identity recognition under such conditions.

3.1 Tools and Libraries

- PyTorch: For deep learning model development and GPU-accelerated training.
- Scikit-learn: For evaluation metrics and classification analysis.
- PIL: For loading and transforming images.
- NumPy: For numerical and array-based computations.
- tqdm: For tracking training progress interactively.

3.2 Data Loading and Preprocessing

The clean and distorted images were loaded using PyTorch's ImageFolder and custom Dataset classes. All images were resized to a fixed resolution, normalized, and converted to PyTorch tensors. Each distorted image was associated with its clean identity label, enabling the model to learn invariant features.

Preprocessing steps:
- Resizing to 224x224 pixels.
- Normalization using dataset-wide mean and standard deviation.
- Augmentation through horizontal flipping and random rotation.

3.3 Model Architecture

We used a ResNet-like CNN architecture designed to extract robust facial features from both clean and distorted images. The architecture included:
- Convolutional blocks with batch normalization and ReLU activations.
- MaxPooling layers to downsample spatial dimensions.
- Fully connected layers to project into an embedding space.
- Final classification layer mapping features to identity labels.

Alternatively, the model could be trained using triplet loss or contrastive learning to further enforce identity similarity across distortions.

3.4 Training and Optimization

We trained the model using the CrossEntropyLoss function with the Adam optimizer. Training was accelerated using CUDA-enabled GPUs. Each epoch included:
- Forward pass: passing inputs through the model.
- Loss calculation: comparing predicted identity with ground truth.
- Backward pass: computing gradients.
- Optimization: updating weights.

All progress was tracked using tqdm to visualize batch processing.

4. Results and Evaluation

The model was evaluated using top-1 accuracy, confusion matrices, and feature space visualizations (e.g., t-SNE or PCA). The system performed well across most distortion types, particularly under mild fog and blur, but showed reduced performance under extreme overexposure or occlusion. The confusion matrix highlighted effective identity clustering, even for distorted samples.

5. Discussion

This study shows that training a deep neural network with distorted and clean image pairs enables robust face recognition. Augmenting the model with distorted data significantly improved its generalization. However, limitations were observed in the model’s performance when the number of distorted images per identity was imbalanced. Future iterations may include architectural adjustments and contrastive learning strategies to further enhance distortion-invariant representation learning.

6.Conclusion

We successfully developed a deep learning model capable of recognizing faces across a variety of visual distortions.

References

1. He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. CVPR.

2. PyTorch Documentation: https://pytorch.org

3. Scikit-learn API: https://scikit-learn.org

4. Schroff, F., Kalenichenko, D., & Philbin, J. (2015). FaceNet: A unified embedding for face recognition and clustering.