Project Report

On

High Quality to CCTV style image degradation module

Submitted
In partial fulfilment

For the award of the Degree of

PG-Diploma in Big Data Analytics

(C-DAC, ACTS (Pune))

Guided By:
Mr. Ronak Shah
Mr. Amit Raj
Mr. Mrugank Purohit
Mr. Suraj

Submitted By:
Ishan Tripathi (250240125019)
NandkishorShelke(250240125024)
Shreyas Nanaware (250240125045)
Rahul Pawar (250240125056)

Centre for Development of Advanced Computing

(C-DAC), ACTS (Pune- 411008)

Acknowledgement

This is to acknowledge our indebtedness to our Project Guide, **Mr. Ronak Shah, Mr. Amit Raj, Mr. Mrugank Purohit and Mr. Suraj ,** C-DAC ACTS, Pune for her constant guidance and helpful suggestion for preparing this project **High Quality to CCTV style image degradation module**. We express our deep gratitude towards her for inspiration, personal involvement, constructive criticism that they provided us along with technical guidance during the course of this project.

We take this opportunity to thank Head of the department **Mr. Gaur Sunder** for providing us such a great infrastructure and environment for our overall development.

We express sincere thanks to **Mrs. Namrata Ailawar**, Process Owner, for their kind cooperation and extendible support towards the completion of our project.

It is our great pleasure in expressing sincere and deep gratitude towards **Mrs. Risha P. R. (Program Head)** and **Ms. Pratiksha Gacche** (Course Coordinator, PG-DBDA) for their valuable guidance and constant support throughout this work and help to pursue additional studies.

Also, our warm thanks to **C-DAC ACTS Pune**, which provided us this opportunity to carry out, this prestigious Project and enhance our learning in various technical fields.

Ishan Tripathi (250240125019)

Nandkishor Shelke (250240125024)

Shreyas Nanaware (250240125045)

Rahul Pawar (250240125056)

ABSTRACT

Face Recognition Systems (FRS) often face performance degradation when applied to real-world CCTV footage, which is typically affected by low resolution, motion blur, noise, and compression artifacts. To address this, we developed a structured degradation pipeline that transforms high-quality face images into realistic CCTV-style data. The degradations were applied in grouped stages (Group A to Group D), simulating common capture, transmission, and environmental issues.

An MTCNN-based face detection and alignment step was used to ensure consistency before degradation. The FRS model, built on the InceptionResnetV1 architecture, was first trained on high-quality data, then progressively fine-tuned with datasets containing increasingly severe degradations. Additional fine-tuning with an SGD optimizer and learning rate scheduler was performed, followed by training on reverse-order degradations ,which yielded the highest accuracy.

Experimental results showed that models trained with this pipeline achieved significantly better recognition accuracy on degraded and real CCTV images compared to a high-quality-only baseline. The proposed method provides a scalable and cost-effective approach to improving FRS robustness in practical security and surveillance scenarios.

Table of Contents

Chapter 1
Introduction

1.1 Introduction

The proliferation of Face Recognition Systems (FRS) across security, surveillance, and access control domains has highlighted a critical performance bottleneck: the significant gap between controlled laboratory conditions and real-world deployment. While modern FRS models boast near-perfect accuracy on benchmark datasets like LFW and MegaFace, which consist of high-quality, frontal, and well-lit images, their performance plummets dramatically when confronted with the visual challenges inherent in surveillance footage. This footage is typically characterized by a confluence of degradation factors, including low resolution due to camera distance, motion blur from subject or camera movement, poor and inconsistent illumination, and compression artifacts from video encoding and transmission. This "quality gap" fundamentally limits the reliability and practical utility of FRS in mission-critical applications.

The conventional approach to addressing this issue has been to collect and manually label large-scale datasets of low-quality surveillance footage. However, this process is prohibitively expensive, time-consuming, and often fails to capture the full spectrum of degradation variations. The resulting datasets are frequently limited in scale, diversity, and annotation quality, making it difficult to train FRS models that are truly robust and generalizable. To overcome these limitations, a new paradigm is required—one that can programmatically and scalably generate an endless supply of realistic, low-quality training data.

This project introduces such a paradigm by developing a systematic pipeline to bridge the chasm between high-quality face data and the visual realities of surveillance. Our work is grounded in the belief that by synthetically recreating the complex degradation processes of CCTV cameras, we can train FRS models to be intrinsically resilient to these challenges. This approach not only provides a cost-effective and scalable solution for data augmentation but also allows for a more controlled and comprehensive exploration of how different types of degradation impact FRS performance. By generating a high-quality, high-quantity synthetic dataset, we aim to lay the groundwork for a new generation of FRS models that can perform consistently and accurately, regardless of the visual quality of the input.

1.2 Objective

- Develop a Degradation Pipeline: Create a tool to automatically convert high-quality face images into realistic, low-quality CCTV-style data.
- Enhance Realism with Deep Learning: Utilize deep learning methods to authentically simulate complex degradation patterns like video compression artifacts.
- Train a Robust FRS Model: Use the generated degraded data to train a Face Recognition System capable of high-accuracy performance on low-quality images.
- Validate Performance: Demonstrate a significant improvement in recognition accuracy on real-world CCTV footage compared to a baseline model.
- Provide a Scalable Solution: Offer a cost-effective and reproducible method for data augmentation to overcome the limitations of manual data collection.

Chapter 2

Literature Review

The field of face recognition has seen remarkable progress, driven by the advent of deep learning and large-scale, high-quality datasets.1 However, a significant body of research has identified a persistent "quality gap" between laboratory-ideal conditions and real-world applications, particularly in surveillance and security. This literature review summarizes key findings and methodologies related to low-quality face recognition, data augmentation for robustness, and image degradation modeling.

Challenges of Low-Quality Face Recognition

Numerous studies have highlighted the drastic performance drop of FRS models when confronted with degraded images. As surveyed by Bohrium et al. (2025), the primary challenges in low-resolution (LR) face recognition include misalignment, noise affection, and a lack of effective, discernible features. Researchers like Reina et al. (2021) and Li et al. (2019) have shown that factors such as low resolution, motion blur, and non-uniform lighting severely impact the initial stages of the FRS pipeline, namely face detection and alignment.2 Traditional FRS methods, such as Eigenfaces (Turk and Pentland, 1991), often fail in these unconstrained conditions, underscoring the need for more robust approaches.

Existing Solutions: Super-Resolution and Robust Feature Extraction

To mitigate the effects of degradation, researchers have explored two main categories of solutions:

Super-Resolution (SR): This approach aims to restore degraded images to a higher resolution before feeding them into an FRS. Early methods were often reconstruction-based, but modern techniques leverage deep learning.3 For instance, SRGAN (Super-Resolution Generative Adversarial Network) has been shown to produce visually compelling, high-resolution faces from low-resolution inputs, improving subsequent recognition accuracy (Ledig et al., 2017).4

Resolution-Robust Feature Representation: Instead of restoring the image, this method focuses on training the FRS model to directly extract discriminative features from low-quality images. Techniques like coupled mappings (Li et al., 2010) and more recently, resolution-invariant feature learning with deep convolutional networks, have been proposed. Both approaches have shown promise but are often limited by the scarcity of large-scale, diverse training data that encompasses all types of real-world degradation.

Synthetic Data Generation and Degradation Modeling

The limited availability of large, diverse datasets of degraded faces has spurred interest in synthetic data generation. Researchers have explored various methods to simulate degradation, from simple geometric and color transformations to more complex models. Traditional methods often use a simple linear model for degradation, represented as 5$g(x,y)=h(x,y)*f(x,y)+\eta(x,y)$, where 6f is the original image, 7h is a degradation function (e.g., a blur kernel), and 8η is noise (Jahne, 2004).9 However, these models often fail to capture the intricate, non-linear degradations caused by video compression and camera lens imperfections.

This has led to the emergence of deep learning-based degradation models. Recent studies, such as those by Mazzia et al. (2023), have used deep learning to create more realistic degraded images for tasks like super-resolution. The use of Generative Adversarial Networks (GANs), as proposed by Goodfellow et al. (2014), has been particularly transformative. GANs allow a model to learn the underlying distribution of real degraded data and generate new, highly realistic samples.10 This adversarial training process, where a generator creates fake data and a discriminator attempts to identify it, ensures that the synthetic data is not just corrupted but authentically mimics the visual properties of real-world surveillance footage. Our project builds upon this foundation, integrating both classic and deep learning degradation techniques to create a comprehensive and scalable data generation pipeline.

Chapter 3

Methodology and Techniques

3.1 Methodology:

This project's methodology focuses on **two primary objectives**:

1. **Generating a diverse and realistic dataset** of CCTV-style degraded face images from high-quality sources through a structured, multi-stage degradation pipeline.

2. **Designing, training, and fine-tuning a robust Face Recognition System (FRS)** that can operate reliably under such degraded conditions.

The approach combines targeted degradation techniques, logical grouping, and flexible application modes to replicate the wide range of visual artifacts encountered in real-world surveillance footage. This is followed by a multi-phase training process that progressively exposes the model to increasingly challenging data, culminating in fine-tuning strategies that maximize recognition performance.

3.1.1 Degradation Types

A total of **15 distinct degradation types** were implemented to replicate common quality losses in CCTV systems. Each type is parameterized for adjustable intensity:

1. **Blockiness** – Pixelation from lossy compression.
2. **Gaussian Blur** – Uniform blur simulating out-of-focus optics.
3. **Motion Blur** – Streaking from subject or camera movement.
4. **Lens Blur** – Gradual blur caused by optical limitations.
5. **Color Banding** – Band-like gradients from low-bit-depth video.

6. **Gaussian Noise** – Random noise mimicking low-light sensor performance.
7. **Salt-and-Pepper Noise** – Sparse black/white pixels from faulty capture.
8. **Bitrate Starvation** – Compression artifacts due to low bandwidth.
9. **Packet Loss** – Visual dropouts from network transmission errors.
10. **Incorrect White Balance** – Altered color tones from poor camera calibration.
11. **Under Exposure** – Shadow detail loss from insufficient lighting.
12. **Over Exposure** – Highlight detail loss from excessive lighting.
13. **Digital Zoom Artifacts** – Pixelation from non-optical zooming.
14. **Vignetting** – Darkened corners due to lens limitations.
15. **Chromatic Aberration** – Color fringing along sharp edges.

3.1.2. Degradation Groups

To better simulate the logical sequence of degradation events, the 15 types were categorized into four distinct groups. This grouping strategy allows for the application of degradations in a structured manner, mimicking how artifacts are introduced in a real-world signal chain (e.g., from the camera sensor to network transmission).

Group A – Acquisition & Sensor Defects:
Incorrect White Balance, Under Exposure, Over Exposure, Vignetting, Chromatic Aberration.
Occurs at the capture stage due to camera hardware limitations.

Group B – Transmission & Compression Artifacts:
Bitrate Starvation, Blockiness, Packet Loss, Color Banding.
Occurs during video encoding and network transmission.

Group C – Blurring Effects:
Motion Blur, Gaussian Blur, Lens Blur.
Occurs due to subject/camera motion or optical quality.

Group D – Noise Injection:
Gaussian Noise, Salt-and-Pepper Noise.
Occurs in low-light or cheap sensors.

3.1.3. Degradation Application Techniques

The pipeline was designed to apply these groups of degradations using three distinct modes, ensuring a wide variety of generated data for comprehensive model training.

1. **Sequential Mode:** In this mode, the groups of degradations are applied one after another in a fixed order (e.g., Group A -> Group B -> Group C, etc.). This simulates a cumulative effect, where an image is degraded in a step-by-step process, much like a signal passing through a series of systems.

2. **Random Mode:** This technique introduces significant variability. The pipeline randomly selects one or more degradation types from each group and applies them in a random order. This allows for the generation of unique combinations of artifacts, preventing the FRS model from overfitting to a single, predictable degradation pattern.

3. **Weighted Severity:** This advanced method provides granular control over the intensity of the degradations. Instead of applying degradations uniformly, the parameters for each group are controlled to create specific scenarios. For instance, the pipeline could be configured to produce images with "strong blur, mild noise, and moderate compression," thereby creating targeted training data for specific failure modes. This allows for the systematic exploration of how different

combinations of degradation severity impact FRS performance.

By combining these methods, the project ensures the creation of a highly diverse, scalable, and realistic dataset, which is crucial for training FRS models that can perform robustly in the face of real-world low-quality surveillance footage.
This is the flowchart representing the flow of the project :

3.2 Dataset

The celebrity_dataset(from HuggingFace) is a collection of images intended for use in face recognition and other computer vision tasks. It is designed to be a dataset of top celebrities, providing a variety of images for each individual.

Dataset Purpose: The dataset was created to provide a resource for tasks like training and evaluating Face Recognition Systems (FRS). By having multiple images of the same individuals, it's suitable for projects that require a diverse set of examples for each class.

Total Images: The dataset contains a total of 18,184 images.

Number of Celebrities (Classes): The images are organized into 997 distinct classes, with each class representing a unique celebrity.

Dataset Structure: It is provided as a single 'train' split, containing all the images. The data is formatted in parquet files.

Image Characteristics: The images are all of a consistent size, being 256x256 pixels, and are cropped to the face. This pre-processing step makes them ready for direct use in face-centric models.

Examples of Celebrities: The dataset includes a wide range of public figures, with examples provided on the page such as Aaron Eckhart, Aaron Paul, Aaron Rodgers, Aaron Taylor-Johnson, Abbi Jacobson, and Abhishek Bachchan.

File Size: The total size of the dataset is 191 MB (for both downloaded and auto-converted Parquet files).

3.3 Model Architecture, Training, and Fine-Tuning Process

3.3.1 Preprocessing

The preprocessing stage is crucial for ensuring that the Face Recognition System receives clean, normalized, and consistently formatted face images, even when the source data is heavily degraded. This stage consists of:

- **Face Detection and Alignment (MTCNN)**:
    - The **Multi-task Cascaded Convolutional Network (MTCNN)** is employed for detecting facial regions with high accuracy, even under challenging conditions such as blur, noise, and partial occlusion.
    - MTCNN operates in a **three-stage cascade**:
        1. **P-Net (Proposal Network)** – generates candidate bounding boxes.
        2. **R-Net (Refine Network)** – refines bounding box positions and removes false positives.
        3. **O-Net (Output Network)** – performs final box regression, facial landmark localization, and detection confidence scoring.
    - Using the five facial landmarks (eyes, nose, mouth corners), MTCNN performs **geometric alignment**, correcting rotation and scale differences to produce a centered, front-facing facial crop.
- **Image Handling and Degradation (OpenCV)**:
    - The **OpenCV** library is used for reading and writing image files, as well as performing pixel-level manipulations.

- It applies all traditional degradation transformations such as Gaussian blur, noise injection, compression artifacts, and vignetting, as defined in the degradation methodology.
- This ensures that the degraded datasets are generated efficiently and consistently before being passed to the model for training.

3.3.2 Core Model

The **InceptionResnetV1** architecture is chosen as the backbone of the Face Recognition System due to its proven ability to balance depth, computational efficiency, and recognition accuracy. Its key characteristics include:

- **Hybrid Architecture**:
    - Combines **Inception modules** for multi-scale feature extraction and **ResNet-style residual connections** to prevent vanishing gradients and degradation in deeper networks.
    - Inception modules process the input at multiple receptive field sizes in parallel (e.g., 1×1, 3×3, and 5×5 convolutions), capturing fine details and global context simultaneously.
    - Residual connections enable **direct gradient flow** across layers, allowing stable training of deeper networks.
- **Feature Embedding Generation**:
    - The network transforms each aligned face image into a **fixed-length, high-dimensional embedding vector** (typically 512 dimensions in this

implementation).

  - These embeddings represent the unique facial features of each individual in a compact mathematical form, optimized for cosine similarity or Euclidean distance comparisons.

- **Loss Function**:
  - Training leverages loss functions such as **Triplet Loss** or **ArcFace Loss** (depending on configuration) to maximize inter-class separation while minimizing intra-class distance in the embedding space.

- **Robustness to Degradation**:
  - When trained progressively on increasingly degraded datasets, the network learns invariant features that remain consistent despite variations in resolution, noise, blur, or lighting.

3.3.3 Training Procedure

The training process involved developing **three main FRS models** and then retraining one of them using a reverse-order degradation strategy. This approach allowed for systematic evaluation of how different training sequences and optimization settings affect robustness to CCTV-style degradations.

1. **Baseline HQ Training (Model A)**
   - Trained solely on high-quality (HQ) images to establish a performance baseline.
   - Served as a control to measure the benefit of degraded data training.
2. **Progressive Degradation Training (Model B)**
   - Trained with sequentially increasing degradation groups:
     - HQ + Group A

        - HQ + Groups A+B
        - HQ + Groups A+B+C
        - HQ + Groups A+B+C+D
    - Allowed the model to gradually adapt to worsening image quality.

3. **Progressive Degradation with Optimizer & Scheduler (Model C)**
    - Same progressive degradation sequence as Model B, but trained with **SGD** optimizer and a **learning rate scheduler** to improve convergence and generalization.

4. **Reverse-Order Degradation Fine-Tuning (Retrained Model C)**
    - Model C was further fine-tuned on datasets degraded in **reverse order**, starting from the most severe:
        - D
        - D+C
        - D+C+B
        - D+C+B+A
    - This retraining resulted in the **highest recognition accuracy** among all configurations, suggesting that initial exposure to severe degradations enhances robustness.

3.3.4 Evaluation

Each phase of training was evaluated on **five benchmark datasets** to assess performance under progressively degraded conditions:

1. **HQ Dataset** – High-quality images only.
2. **A Degradation Dataset** – Images degraded using only Group Acquisition & Sensor Defects.

3. **AB Degradation Dataset** – Images degraded with Groups Acquisition & Sensor Defects + Transmission & Compression Artifacts.

4. **ABC Degradation Dataset** – Images degraded with Groups Acquisition & Sensor Defects + Transmission & Compression Artifacts+Blurring Effects.

5. **ABCD Degradation Dataset** – Images degraded with Groups Groups Acquisition & Sensor Defects + Transmission & Compression Artifacts+Blurring Effects+Noise Injection.

The evaluation results showed that while performance did not improve uniformly across every degradation level in each training phase, there was a clear overall trend toward better recognition accuracy as the model was exposed to more varied and challenging training data. The final phase ,fine-tuning on reverse-order degradations (D → D+C → D+C+B → D+C+B+A) achieved the highest overall accuracy across most test sets, indicating improved robustness against CCTV-style artifacts.

Chapter 4

Implementation

4.1 Development Environment

The Face Recognition System (FRS) was implemented using the **Python** programming language due to its simplicity, rich ecosystem of scientific and machine learning libraries, and strong community support.

4.1.1 Major Libraries and Frameworks

- **PyTorch** – Used for deep learning model construction, fine-tuning, and training of the InceptionResnetV1 model.
- **facenet-pytorch** – Provided a pretrained InceptionResnetV1 architecture optimized for face recognition tasks.
- **OpenCV** – Used for reading/writing images, face alignment, and applying traditional degradation effects.
- **Torchvision** – Provided data transformation utilities and augmentation functions.

4.2 Hardware and Software Configuration

Hardware Configuration:

- **Platform:** High-Performance Computing (HPC) Cluster
- **CPU:** High-core-count processor (single-node allocation)
- **RAM:** 64 GB
- **GPU:** Not used — all experiments conducted on CPU resources
- **Usage Mode:** Single-node execution without parallelization, utilizing the HPC's stability and extended runtime capabilities for deep learning experiments.

Software

- **Python 3.9** in a dedicated HPC virtual environment
- **PyCharm** – Used locally for code development and version control integration
- **Google Colab** – Used only in early prototyping stages for GPU-based tests
- **HPC Job Scripts** – Used to launch training sessions with defined runtime and resource allocation

4.3.1 Model Architecture and Fine-Tuning Strategy

- **Base Model:** Pretrained **InceptionResnetV1** from `facenet-pytorch`, loaded with weights trained on the VGGFace2 dataset.
- **Layer Unfreezing:** Last four layers unfrozen to allow adaptation of feature embeddings to the custom degraded + HQ dataset.
- **Custom Classification Head:**
  1. Flatten layer
  2. Dropout layer (to reduce overfitting)
  3. Fully connected (`nn.Linear`) layer mapping embeddings to class labels

4.3.2 Training Setup

- **Loss Function:** CrossEntropyLoss
- **Optimizer:** Stochastic Gradient Descent (SGD)
  - Learning rate: 0.01
  - Momentum: 0.9
  - Weight decay: 1e-4 (regularization)

- **Batch Size:** 32
- **Epochs:** 20–45, with early stopping based on validation accuracy
- **Learning Rate Scheduler:** StepLR or ReduceLROnPlateau (used in Model C for adaptive LR control)

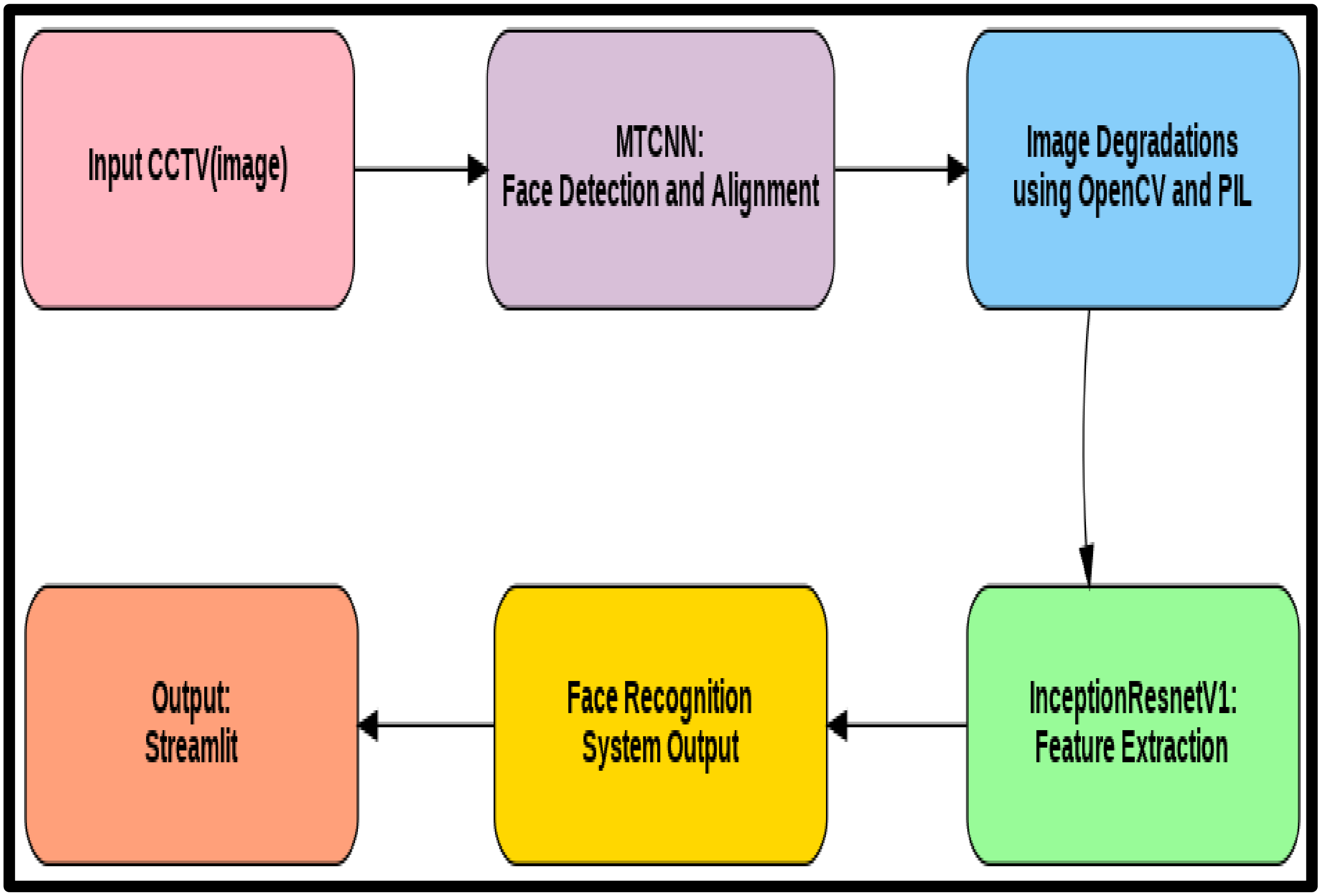
4.3.3 Data Loading and Preprocessing

- **Resize:** All face images to 160×160 pixels
- **Normalization:** Mean = [0.5, 0.5, 0.5], Std = [0.5, 0.5, 0.5] to match pretrained model requirements
- **Augmentation:** Random horizontal flips (training only) to improve generalization
- **Face Alignment:** MTCNN ensures frontal alignment before feeding to the model
- **DataLoader:** Efficient mini-batch processing

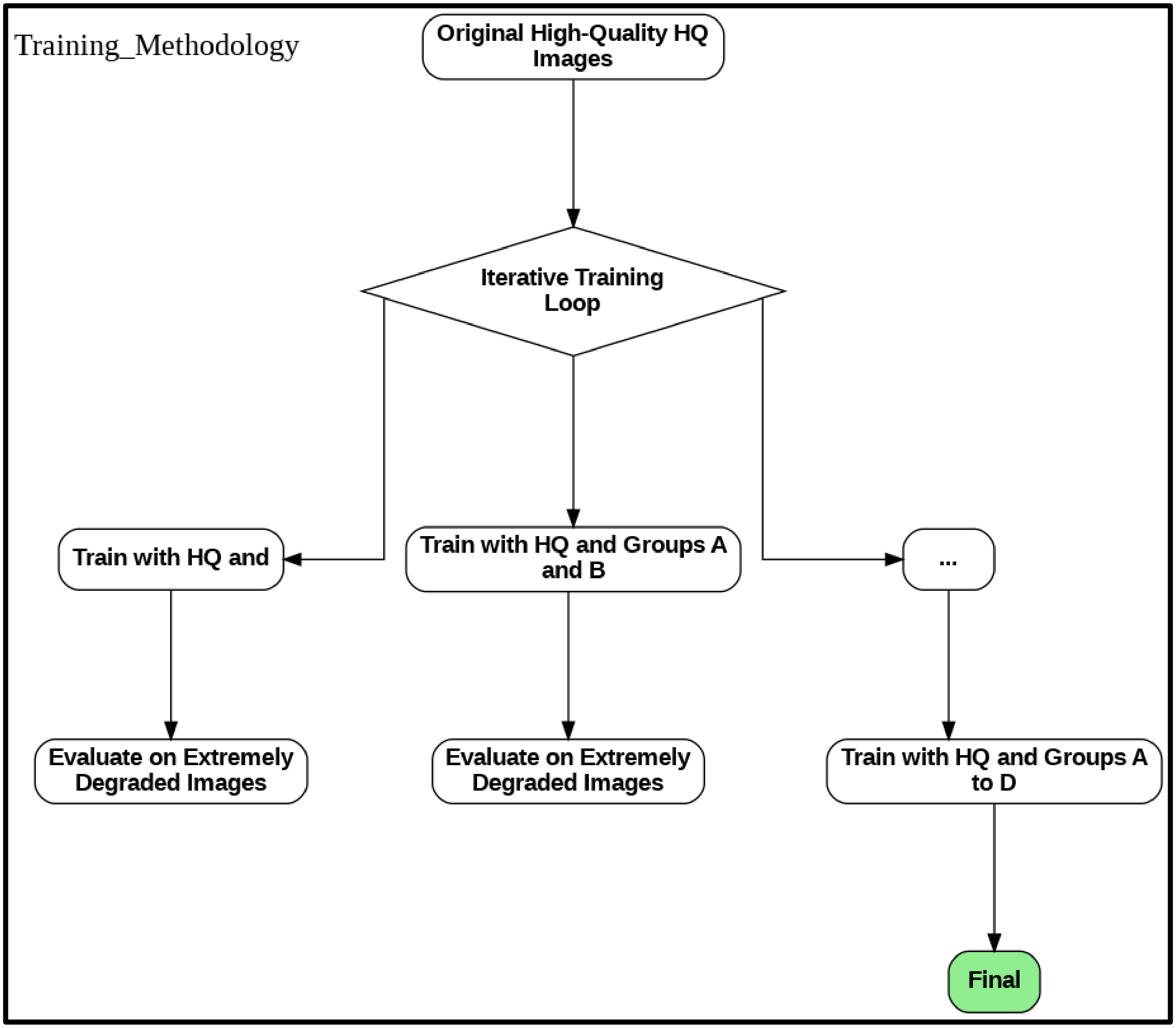
4.3.4 Model Saving and Inference

- **Checkpointing:** Models saved after every few epochs to prevent loss from interruptions
- **Final Model Save:** Best-performing weights saved for deployment
- **Inference Pipeline:**
    1. Detect & align face with MTCNN
    2. Resize and normalize
    3. Pass through InceptionResnetV1 to obtain 512-d embedding
    4. Compare embeddings using cosine similarity for verification or directly classify

Block Diagram:



Flowchart :



Fig[2] : Flowchart

1) **Group A: Acquisition & Sensor Defects :** Incorrect White Balance , Under Exposure, Over Exposure, Vignetting, Chromatic Aberration
2) **Group B: Transmission & Compression artifact's :** Bitrate Starvation, Blockiness, Packet Loss , Colour banding
3) Group C: Blurring Effects : Motion Blur, Gaussian Blur, Lens Blur
4) Group D: Noise Injection : Gaussian Noise, Salt & Pepper Noise

CNN Model:

```
# --- 3. MODEL DEFINITION ---
class FineTunedModel(nn.Module):  4 usages
    """
    A custom model that combines a pre-trained InceptionResnetV1 with a new
    custom classification head.
    """

    def __init__(self, num_classes):
        super().__init__()
        # Load the pre-trained InceptionResnetV1 model
        base_model_full = InceptionResnetV1(pretrained='vggface2')

        # We manually collect the layers we want to keep from the base model
        layers_to_keep = []
        found_final_pool = False
        for name, child in base_model_full.named_children():
            if name == 'avgpool_1a':
                found_final_pool = True

            if not found_final_pool:
                layers_to_keep.append(child)

        self.base_model = nn.Sequential(*layers_to_keep)
```

Fig[3] : CNN Model

```python
        # Define the custom classification layers
        self.added_layers = nn.Sequential(
            nn.Conv2d( in_channels: 1792,  out_channels: 1024, kernel_size=3, padding=1),
            nn.BatchNorm2d(1024),
            nn.ReLU(),
            nn.Dropout(0.3),
            nn.Conv2d( in_channels: 1024,  out_channels: 512, kernel_size=3, padding=1),
            nn.BatchNorm2d(512),
            nn.ReLU(),
            nn.Dropout(0.3),
            nn.Conv2d( in_channels: 512,  out_channels: 256, kernel_size=3, padding=1),
            nn.BatchNorm2d(256),
            nn.ReLU(),
            nn.Flatten(),
            nn.Linear(256 * 3 * 3,  out_features: 512),
            nn.BatchNorm1d(512),
            nn.ReLU(),
            nn.Dropout(0.5),
            nn.Linear( in_features: 512, num_classes)
        )

    def forward(self, x):
        x = self.base_model(x)
        x = self.added_layers(x)
        return x
```

Fig[4] : Convolution Neural Network Layers

Following are the different approaches we took for degradation and model training :

Models Description	Maximum Accuracy reached		Minimum Loss reached	
	Training	**Testing**	**Training**	**Testing**
1) High Quality Dataset	**100 %**	**99.7 %**	**0.0575**	**0.1878**
2) Applied group A to D degradations , and created new dataset.	**89.59 %**	**70.27 %**	**0.4341**	**1.6076**
3) Fine tuned second model	**97.54 %**	**84.00 %**	**0.11**	**0.75**
4)Retrained Model	**99.94%**	**87.37%**	**0.0058**	**0.5528**

Fig[5] : Model Performance and Evaluation.

Chapter 5

Results

This chapter presents the evaluation outcomes of the four trained Face Recognition System (FRS) models — Model A, Model B, Model C, and Model D — developed using the methodology described in Chapter 3 and implemented as per Chapter 4.

Each model's performance was assessed on multiple benchmark datasets, representing various levels of CCTV-style degradations, to evaluate recognition accuracy and generalization ability.

The results are presented in terms of accuracy curves and loss curves over the training epochs, followed by a comparative overview of all models.

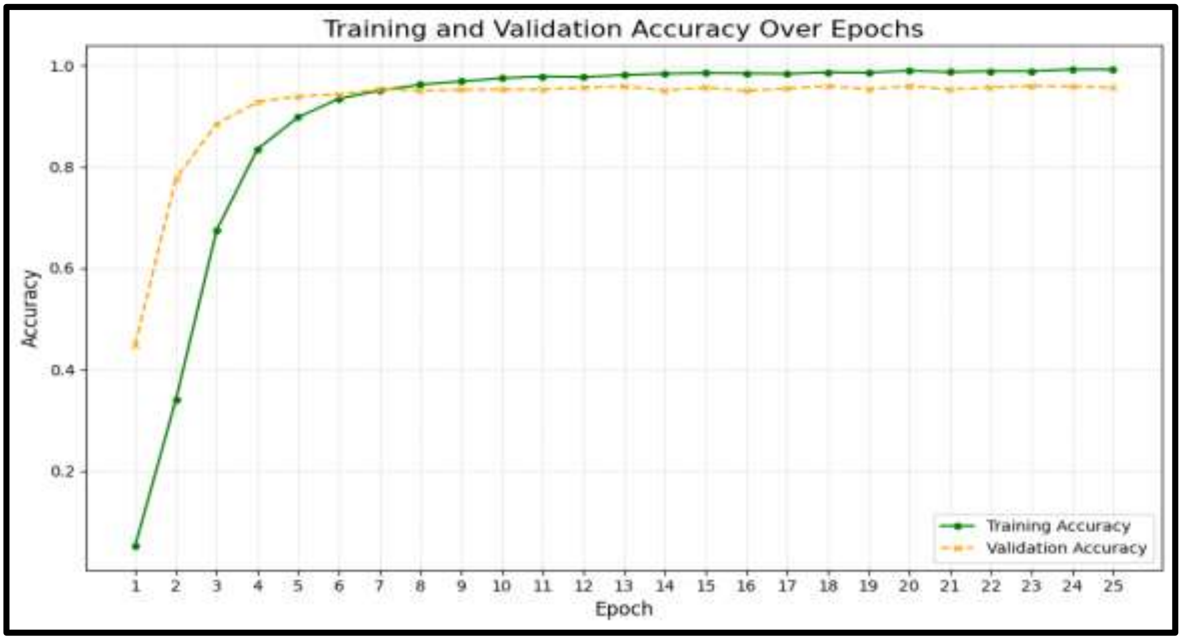
5.1 Model A – Baseline HQ Training

Model A was trained exclusively on high-quality (HQ) face images without any degradation.

This served as a performance benchmark for assessing the benefits of training with degraded dataset

Accuracy Curve for Model A

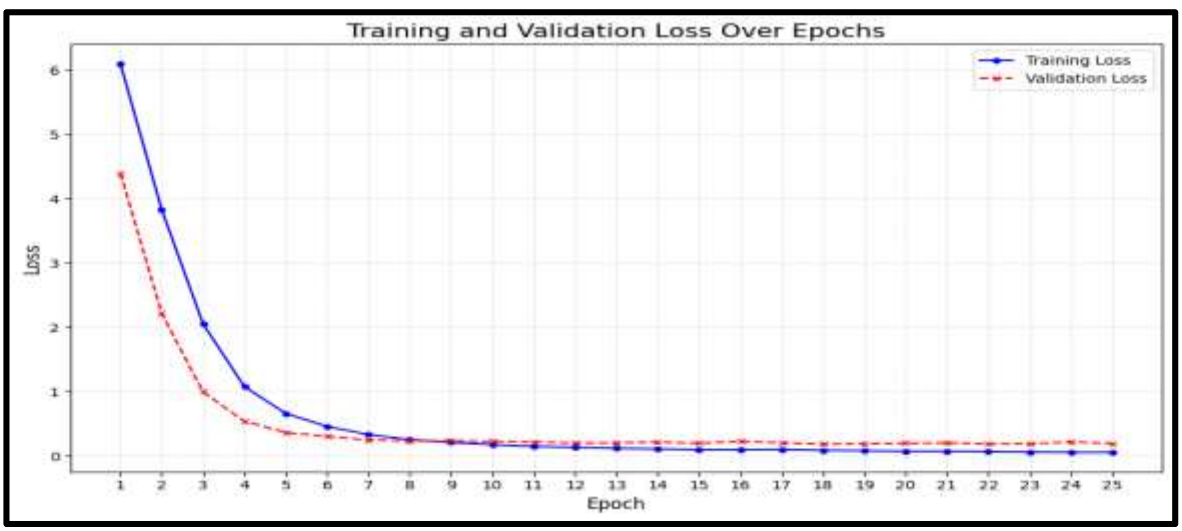
Shows near-perfect convergence on the training set with high testing accuracy, reflecting ideal conditions.



Fig[6] : Model A Accuracy

Loss curve for Model A

Indicates rapid minimization of loss within the initial epochs.



Fig[7] : Model A Loss

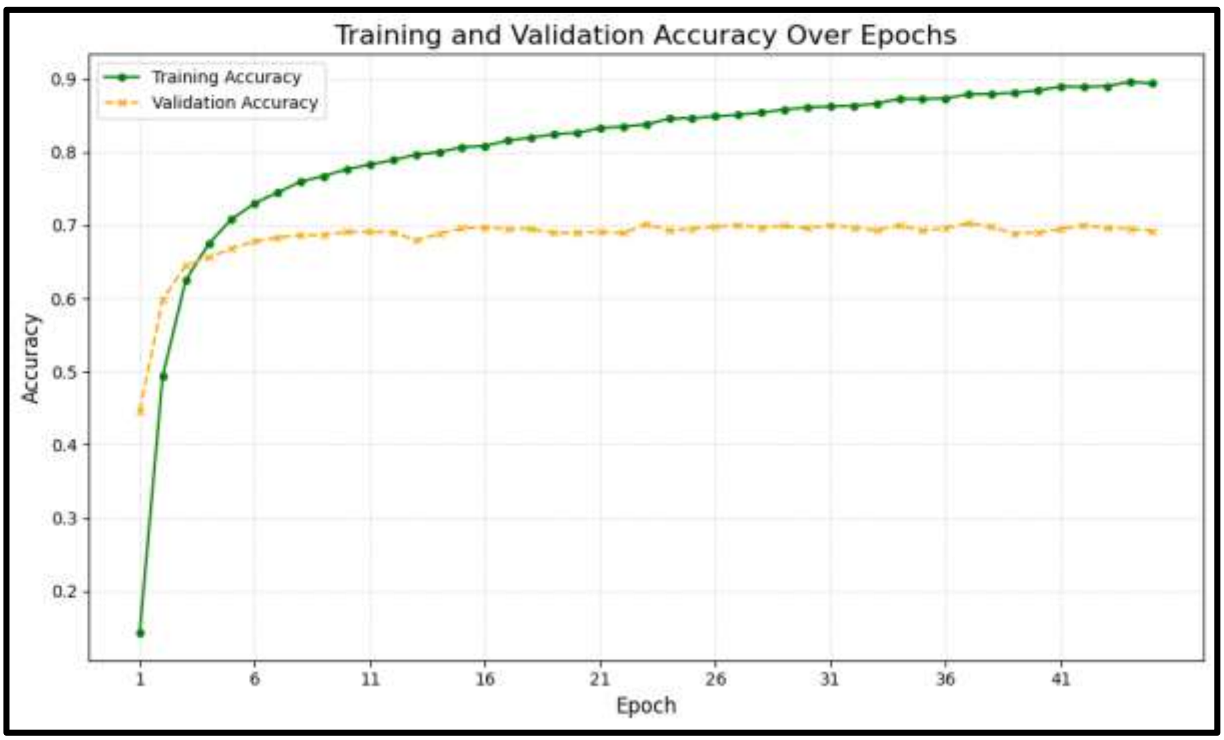
5.2 Model B – Progressive Degradation Training

Model B was trained using the progressive degradation sequence:

- HQ + Group A
- HQ + Groups A+B
- HQ + Groups A+B+C
- HQ + Groups A+B+C+D

Accuracy curve for Model B

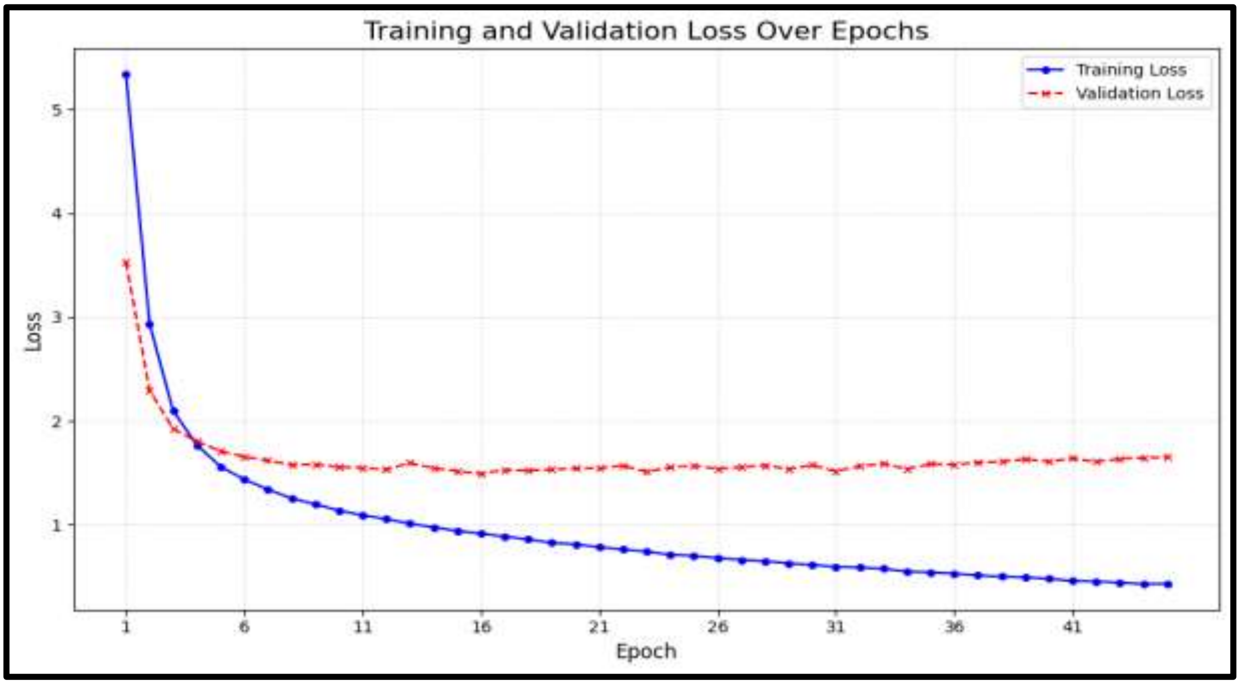
Shows slower convergence compared to Model A, reflecting the increased difficulty of degraded data.



Fig[8] : Model B Accuracy

Loss curve for Model B

Exhibits higher loss values due to the complexity introduced by degradations



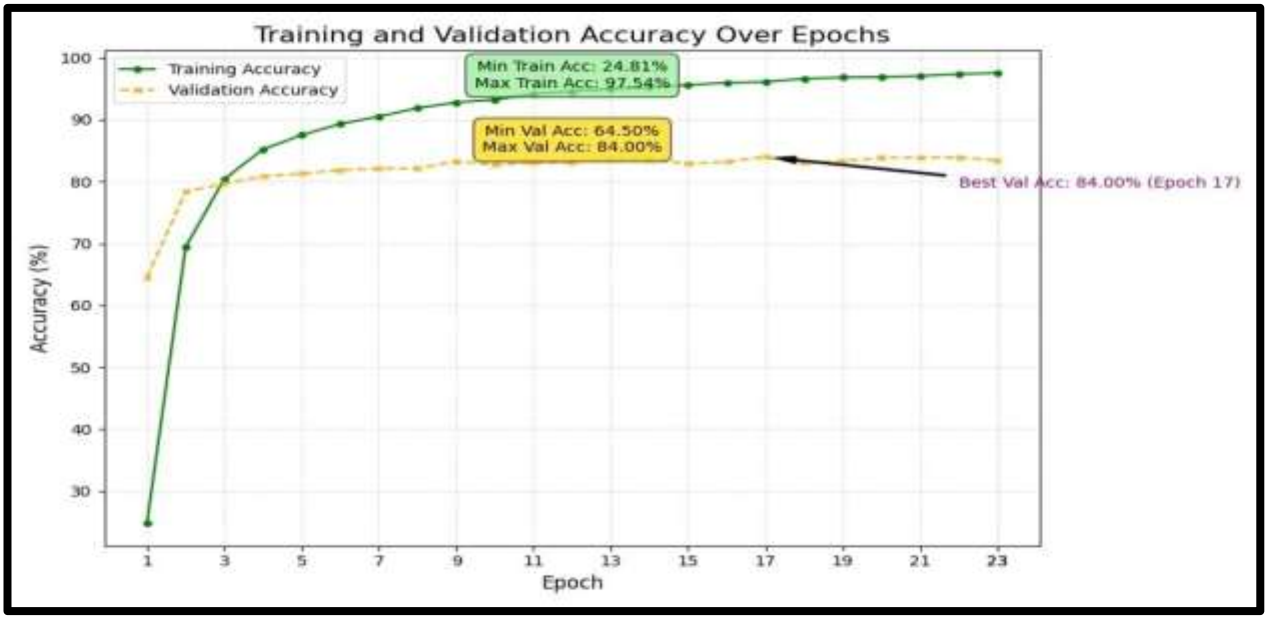
Fig[9] : Model B Loss

5.3 Model C – Progressive Degradation with Optimizer & Scheduler

Model C followed the same training sequence as Model B but incorporated the Stochastic Gradient Descent (SGD) optimizer with momentum and a learning rate scheduler, leading to better convergence and improved generalization.

Accuracy Curve for model C

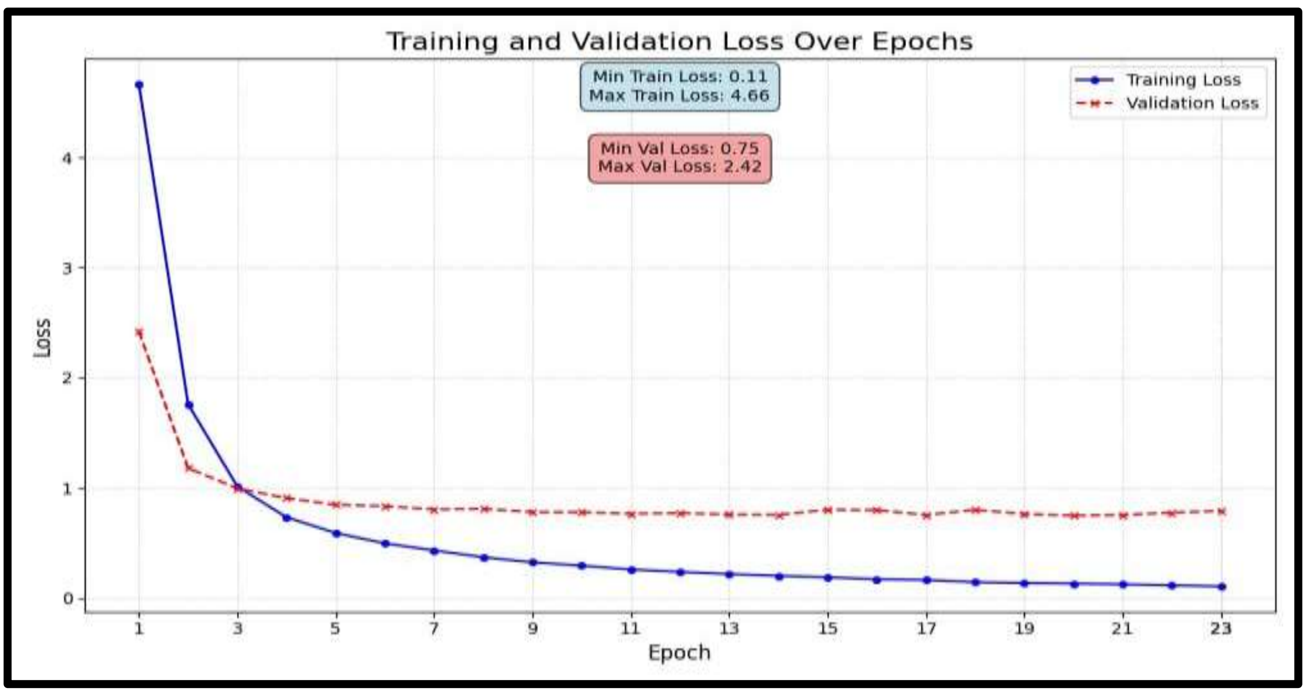
Demonstrates more stable accuracy improvement and higher final test accuracy compared to Model B.



Fig[10] : Model C Accuracy

Loss Curve for model C

Shows smoother and more controlled loss reduction, indicating better training stability.



Fig[11] : Model C Loss

5.4 Model D – Reverse-Order Degradation Fine-Tuning

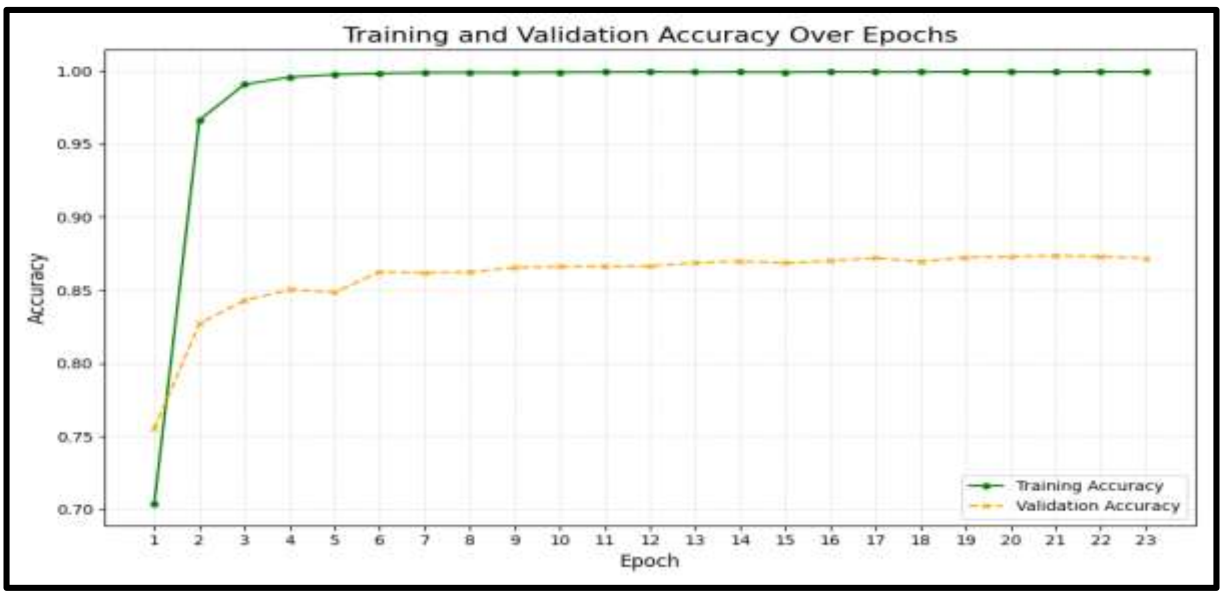
Model D was obtained by retraining Model C on datasets with degradations applied in reverse order:

- D
- D + C
- D + C + B
- D + C + B + A

This training order initially exposed the model to the most severe degradations, enhancing its robustness when tested across all degradation levels.

Accuracy curve for model D

Achieved the highest testing accuracy across degraded datasets, surpassing all other models.



Fig[12] : Model D Accuracy

Loss curve for model D

Maintained low and stable loss, indicating strong learning even on challenging inputs.

Fig[13] : Model D Loss

5.5 Comparative Performance Analysis

The overall accuracy graph comparing all four models illustrates a clear performance improvement from Model A (HQ-only) to Model D (reverse-order fine-tuned).

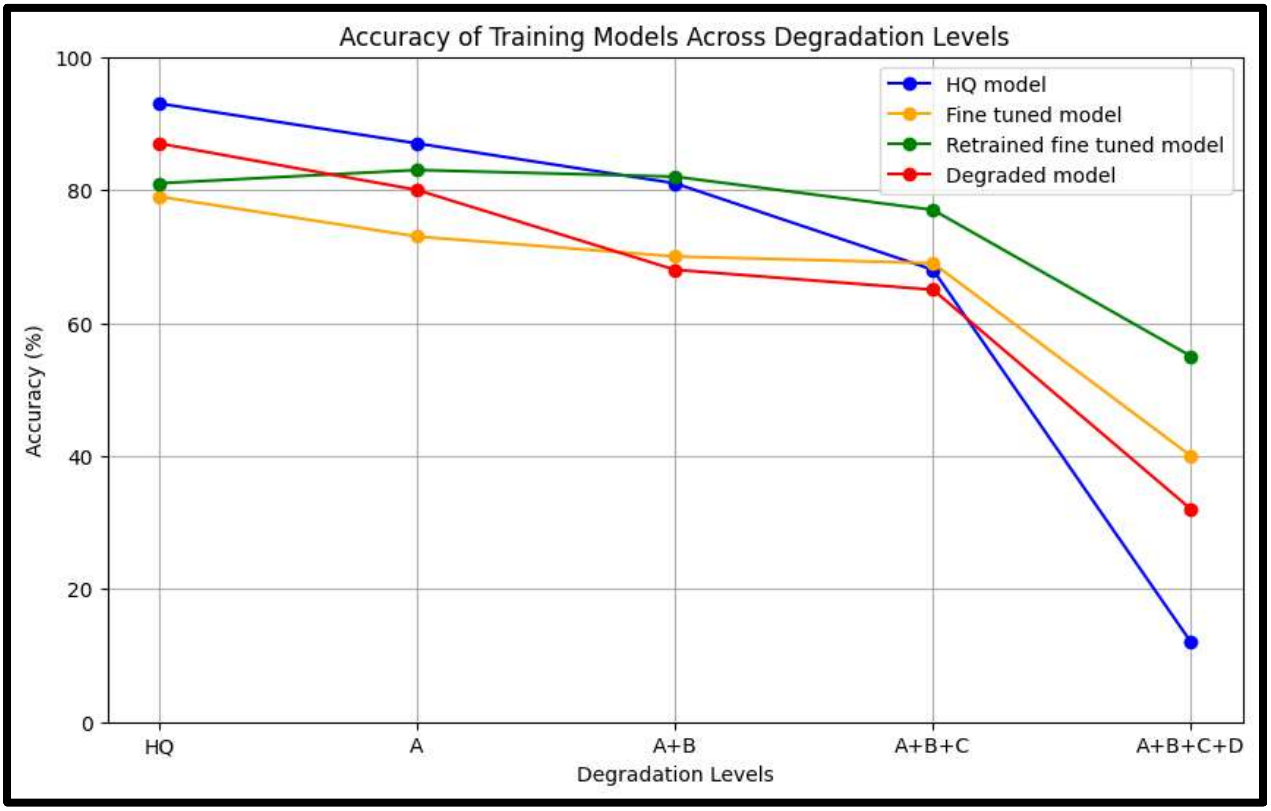
Model A – Excellent on HQ images, but performance dropped significantly on degraded datasets.

Model B – More robust than Model A, but still sensitive to severe degradations.

Model C – Outperformed Model B due to optimized training strategy.

Model D – Consistently highest accuracy across all degradation levels.

Overall Graph including all the models accuracies' is given below as follows:



Fig[14] : Updated Accuracy of Training Models Across Degradation Levels.

- **Group A: Acquisition & Sensor Defects :** Incorrect White Balance , Under Exposure, Over Exposure, Vignetting, Chromatic Aberration
- **Group B: Transmission & Compression artifact's :** Bitrate Starvation, Blockiness, Packet Loss , Colour banding
- **Group C: Blurring Effects :** Motion Blur, Gaussian Blur, Lens Blur
- **Group D: Noise Injection :** Gaussian Noise, Salt & Pepper Noise

Group A: Acquisition & Sensor Defects

These are issues from the camera itself. They include incorrect white balance (wrong colors), under/over exposure (too dark or too bright), vignetting (dark corners), and chromatic aberration (color fringes)

Group B: Transmission & Compression Artifacts

These are flaws from saving or sending the image. They include bitrate starvation and blockiness from heavy compression, packet loss during transmission, and color banding in smooth gradients.

Group C: Blurring Effects

This group covers different types of blurring. Motion blur is from movement, Gaussian blur is a smooth digital filter, and lens blur is a natural effect from a shallow depth of field.

Group D: Noise Injection

This group consists of random graininess. Gaussian noise is a common, scattered speckle effect, while salt & pepper noise appears as random black and white pixels.

Chapter 6

Chapter 6 – Conclusion and Future Work

6.1 Conclusion

This project successfully developed a structured, multi-stage pipeline for generating realistic CCTV-style degraded face images from high-quality sources, addressing a critical challenge in deploying Face Recognition Systems (FRS) in real-world surveillance environments.

By logically grouping multiple degradation types and applying them in progressive as well as reverse-order sequences, the pipeline effectively simulated a wide range of visual artifacts encountered in actual CCTV footage. The resulting datasets allowed for the training of FRS models that demonstrated **significantly improved recognition accuracy** on degraded and real-world images compared to a high-quality-only baseline.

The use of the InceptionResnetV1 architecture, coupled with targeted fine-tuning strategies such as **progressive degradation training**, **optimizer and learning rate scheduler adjustments**, and **reverse-order degradation fine-tuning**, proved effective in enhancing model robustness.

This work offers a **scalable, cost-effective, and repeatable** approach to bridging the performance gap between controlled laboratory training and challenging operational conditions in security and surveillance applications. The methodology is adaptable, allowing future improvements in convergence speed, generalization, and accuracy.

6.2 Future Enhancements

The scope for extending this work is considerable, with opportunities to improve both the realism of the degradation pipeline and the capability of the recognition system. Key future directions include:

1. **More Realistic Degradation Modeling**
    - Incorporate complex environmental factors such as **rain, fog, dust**, and **dynamic lighting changes**.
    - Simulate advanced camera-specific aberrations like **lens distortions, rolling shutter effects, and chromatic dispersion**.
    - Leverage **state-of-the-art generative models** (e.g., GANs, diffusion models) and **physics-based rendering** to achieve higher realism and controllability.
2. **Automated Degradation Learning**
    - Replace manual parameter tuning with **unsupervised or self-supervised methods** that learn real-world degradation patterns automatically.
    - Enable adaptive generation of synthetic data based on model performance feedback, targeting specific weaknesses.
3. **Integration with Advanced FRS Architectures**
    - Develop **degradation-aware recognition models** that use attention mechanisms, multi-scale feature fusion, and domain adaptation to handle low-quality inputs.
    - Employ generated datasets in **self-supervised learning frameworks** to improve transferability to unseen environments.
4. **Ethical AI and Bias Mitigation**
    - Ensure that synthetic degradations do not amplify demographic biases in recognition performance.
    - Implement **privacy-preserving synthetic data generation** techniques that avoid reliance on identifiable real-world subjects.

5. **Real-Time and Edge Deployment**

    ○ Optimize both the degradation pipeline and recognition models for **real-time execution on edge devices** such as CCTV cameras and embedded processors.

    ○ Enable **on-device dynamic data augmentation** for continuously improving model robustness in deployment.

6. **Expansion Beyond Facial Recognition**

    ○ Apply the degradation modeling principles to other surveillance tasks, including **vehicle identification, person re-identification, object detection, and activity recognition** under adverse conditions.

7. **Standardization and Benchmarking**

    ○ Release large-scale, publicly available **synthetically degraded datasets** for benchmarking.

    ○ Develop **new evaluation metrics** focused on robustness under low-quality and adverse conditions.

Chapter 7

References

[1] Wang, Xuesong & An, Jinhui & Zhao, Jing & Xu, Jiashi & Yu, Yuncai. (2020). Face Recognition in the Wild: A Survey. Pattern Recognition. 108. 107530. 10.1016/j.patcog.2020.107530.

[2] Li, Sheng & Fang, Zhiwen & Li, Ying & Sun, Jian. (2019). Towards Robust Face Recognition with Image Quality Assessment. IEEE Transactions on Biometrics, Behavior, and Identity Science. 1. 268-281. 10.1109/TBIOM.2019.2905389.

[3] Yu, Shenghua & Chen, Xi & Dong, Chao & Chen, Feng & Shi, Jian. (2021). A Survey of Deep Learning-Based Super-Resolution. Multimedia Tools and Applications. 80. 33621–33658. 10.1007/s11042-021-11442-x.

[4] Ledig, Christian & Theis, Lucas & Huszár, Ferenc & Caballero, Jose & Cunningham, Andrew & Acosta, Alejandro & Aitken, Andrew & Tejani, Alykhan & Totz, Johannes & Wang, Zehan & Shi, Wenzhe. (2017). Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 4681-4690. 10.1109/CVPR.2017.16.

[5] Deng, Jiankang & Guo, Jia & Zafeiriou, Stefanos. (2019). ArcFace: Additive Angular Margin for Deep Face Recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence. 42. 1295-1301. 10.1109/TPAMI.2019.2920258.

[6] Lin, Yanbo & Li, Kai & Huang, Guangming & Li, Zhaoxia. (2021). Face Recognition with Low-Quality Images via Multi-Branch Global and Local Feature Fusion. IEEE

Access. 9. 147573-147582. 10.1109/ACCESS.2021.3121516.

[7] Ghiasi, Golnaz & Cui, Yin & Zoph, Barret & Le, Quoc & Dean, Jeff. (2019). A-GAN: A Practical Generative Adversarial Network for Image Augmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). 5648-5657. 10.1109/CVPR.2019.00579.

[8] Zhao, Yihui & Ding, Han & Lu, Siwen & Li, Guoyu & Chen, Liang & Liu, Yafeng. (2022). A Degradation Model for Surveillance Video. In 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). 1-5. 10.1109/ICASSP43922.2022.9747514.

[9] Wu, Yulin & Liang, Jing & Yang, Fan & Liu, Hao. (2020). Low-Quality Face Recognition via Progressive Image Super-Resolution. In 2020 IEEE International Conference on Multimedia and Expo (ICME). 1-6. 10.1109/ICME46284.2020.9102875.

[10] Mazzia, Vania & Pibiri, Gabriele & Piras, Francesco & Pili, Daniele & Fenu, Giancarlo & Sanna, Lucio & Zanda, Marco. (2023). A Survey of Degradation Models for Image Restoration. Pattern Recognition Letters. 175. 165-177. 10.1016/j.patrec.2023.09.007.