Jaypee University of Information Technology

Department of Computer Science Engineering

ClearVision Challenge

Report on Image & Video Dehazing Using Deep Learning

Submitted by:

Name: **Om Vishal**

Roll no: **211322**

Batch: **CS-77**

Table of Contents

Introduction

This project aimed to address the problem of haze removal from images and videos using deep learning techniques. Hazy environments, caused by atmospheric particles such as dust and smoke, reduce visibility and degrade the quality of visual data, posing challenges for both human perception and automated systems. The project focused on developing a deep learning model capable of effectively dehazing images and videos, restoring clarity and enhancing the quality of visual data.

The goal was to improve performance in real-world applications such as autonomous driving, surveillance, and remote sensing, where clear visual data is crucial. By employing advanced algorithms, the project sought to enhance visual quality and ensure that important information could be processed accurately, improving outcomes in these fields.

Problem Statement

Haze affects the visibility in images and videos by introducing a layer of noise that reduces contrast and sharpness. This project focused on developing a solution to eliminate or reduce this haze using deep learning models. The objective was to develop a Generative Adversarial Network (GAN) for both image and video dehazing. Due to hardware limitations, however, the project focused primarily on image dehazing, as video dehazing was challenging to complete within the given resources.

Methodology

1. Data Collection

The dataset for this project was sourced from the ClearVision Challenge. It consisted of hazy images, corresponding clear reference images, hazy videos, and an accompanying XLS file containing relevant video metadata. The dataset provided a diverse range of hazy conditions, which was critical for training and validating the dehazing models.

2. Data Preprocessing

For image dehazing, preprocessing involves resizing the images to a fixed resolution and applying normalization techniques to standardise pixel values. A random sampling approach was adopted to ensure a diverse selection of hazy conditions during training.

For video data, frames were extracted from each video to be processed individually by the model. However, due to hardware limitations, the full implementation of video dehazing could not be completed within the project's timeframe and resources. The focus remained on image dehazing while video dehazing was set aside for future work.

3. Model Architecture

The U-Net architecture, widely recognized for image restoration tasks, was selected for image dehazing. The model comprises a contracting path that captures contextual information and an expanding path that allows for precise localization, making it well-suited for recovering clear images from hazy inputs. The U-Net's symmetric structure enabled efficient feature extraction while preserving image details necessary for dehazing.

Project Design & Architecture

- **U-NET Architecture** : The **U-Net** architecture, with its encoder-decoder structure, is ideal for image dehazing. The encoder captures global context through downsampling, while the decoder restores resolution via upsampling. Skip connections help retain spatial details, allowing the model to generate clear, dehazed images effectively.

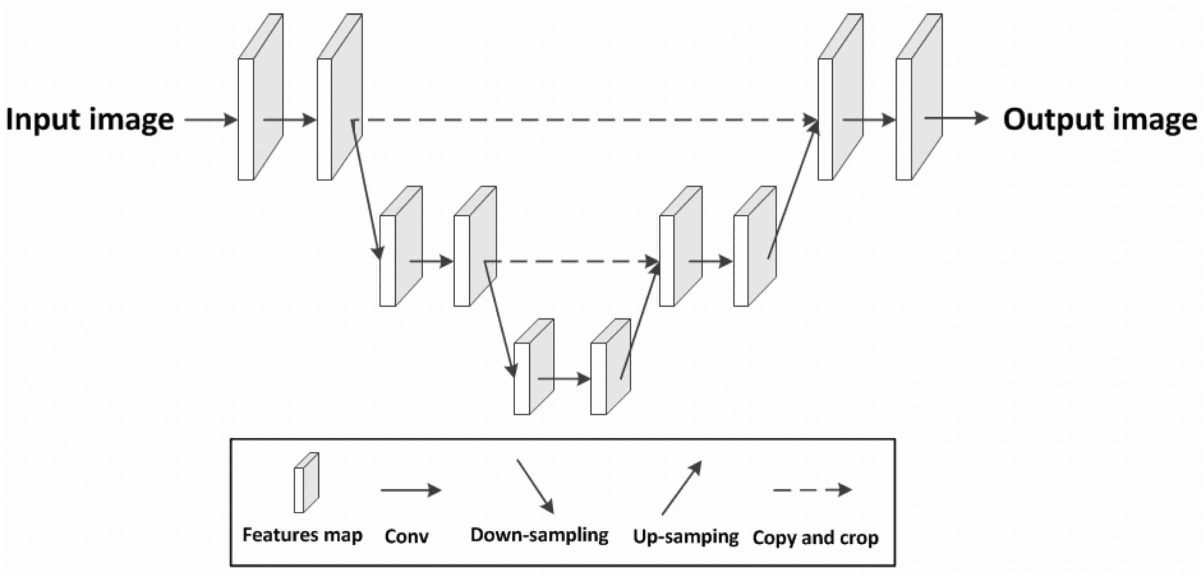
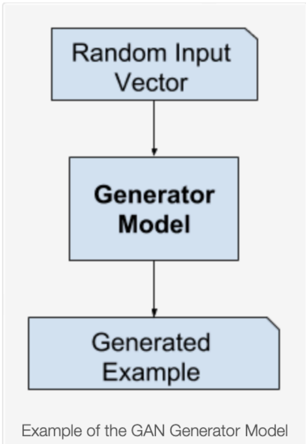


Fig 1 : U-Net Architecture [1]

- **GAN :** A **Generative Adversarial Network (GAN)** for image dehazing consists of two networks: a generator and a discriminator. The generator attempts to create clear, dehazed images from hazy inputs, while the discriminator distinguishes between real clear images and generated ones. This adversarial process helps the generator produce more realistic dehazed images by continuously improving through feedback from the discriminator.

The **generator** in a GAN creates dehazed images from hazy inputs, aiming to produce outputs that resemble real clear images.

The **discriminator** evaluates both real and generated images, distinguishing between authentic clear images and those generated by the model.



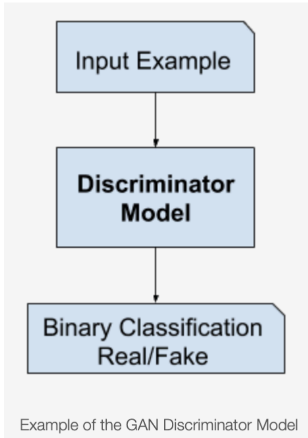


Fig 2 : Gan generator model [2]

Fig 3 : Gan discriminator model [3]

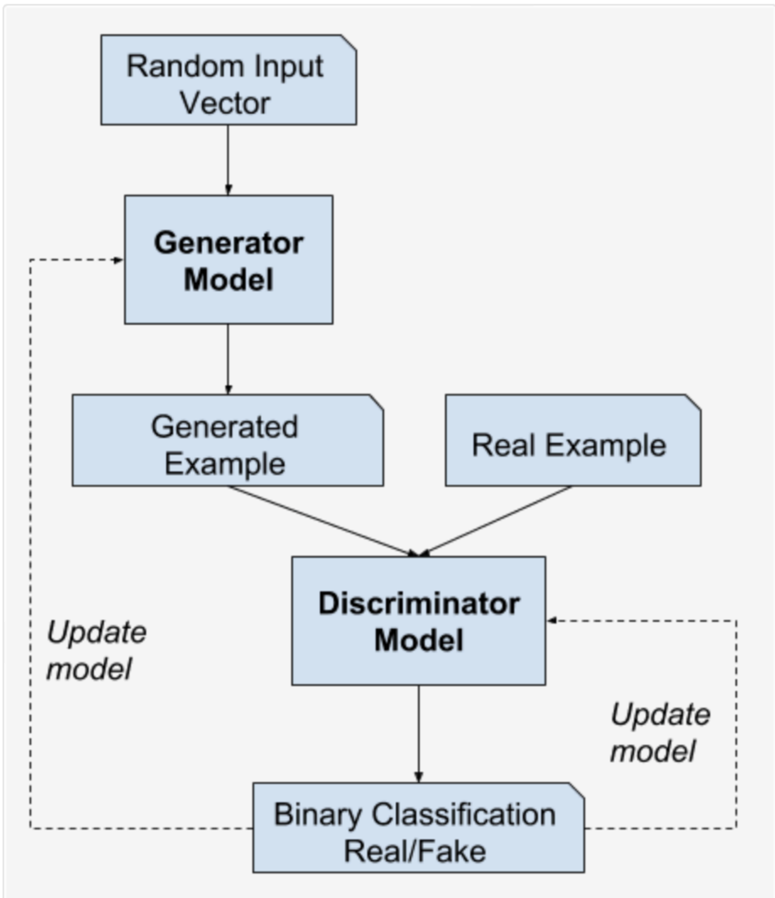


Fig 4 : GAN Architecture [4]

- Flowchart of the project (Image and Video Dehazing):

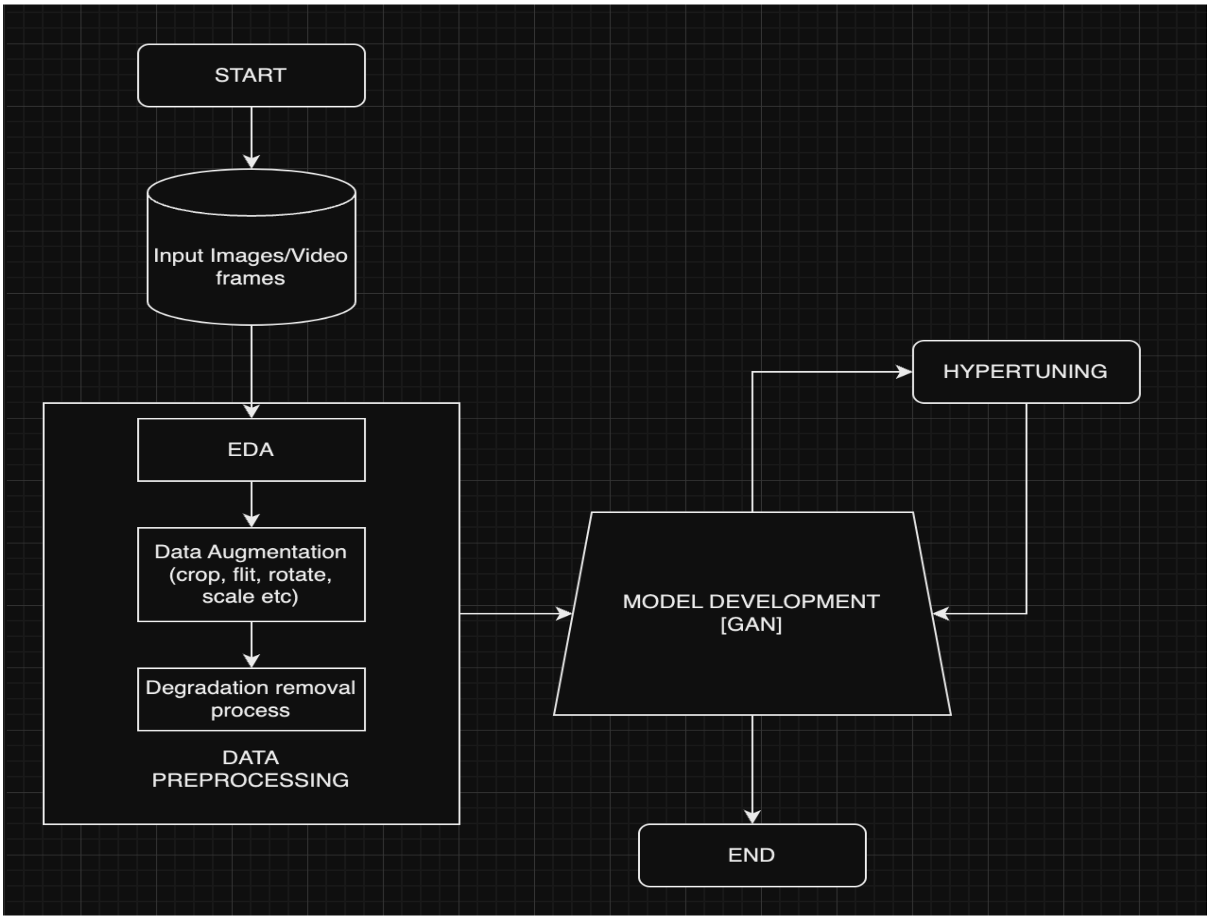


Fig 5 : Project Design

Performance

Image Unet Performance

```
train_loss, train_acc = unet_model.evaluate(X_train, y_train)

print(f"Train Loss: {train_loss}")
print(f"Train Accuracy: {train_acc}")
```

```
3/3 ━━━━━━━━━━━━━━━━━━━━ 31s 9s/step - accuracy: 0.8386 - loss: 0.0174
Train Loss: 0.017593638971447945
Train Accuracy: 0.8310345411300659
```

```
test_loss, test_acc = unet_model.evaluate(X_test, y_test)
print(f"Test Loss: {test_loss}, Test Accuracy: {test_acc}")
```

```
9/9 ━━━━━━━━━━━━━━━━━━━━ 124s 14s/step - accuracy: 0.8306 - loss: 0.0179
Test Loss: 0.017377445474267006, Test Accuracy: 0.8373672366142273
```

PSNR and SSIM (UNet):

```
True Image Shape: (256, 256, 3), Predicted Image Shape: (256, 256, 3)
True Image Shape: (256, 256, 3), Predicted Image Shape: (256, 256, 3)
True Image Shape: (256, 256, 3), Predicted Image Shape: (256, 256, 3)
Average PSNR: 18.33, Average SSIM: 0.71
```

Video GAN Performance:

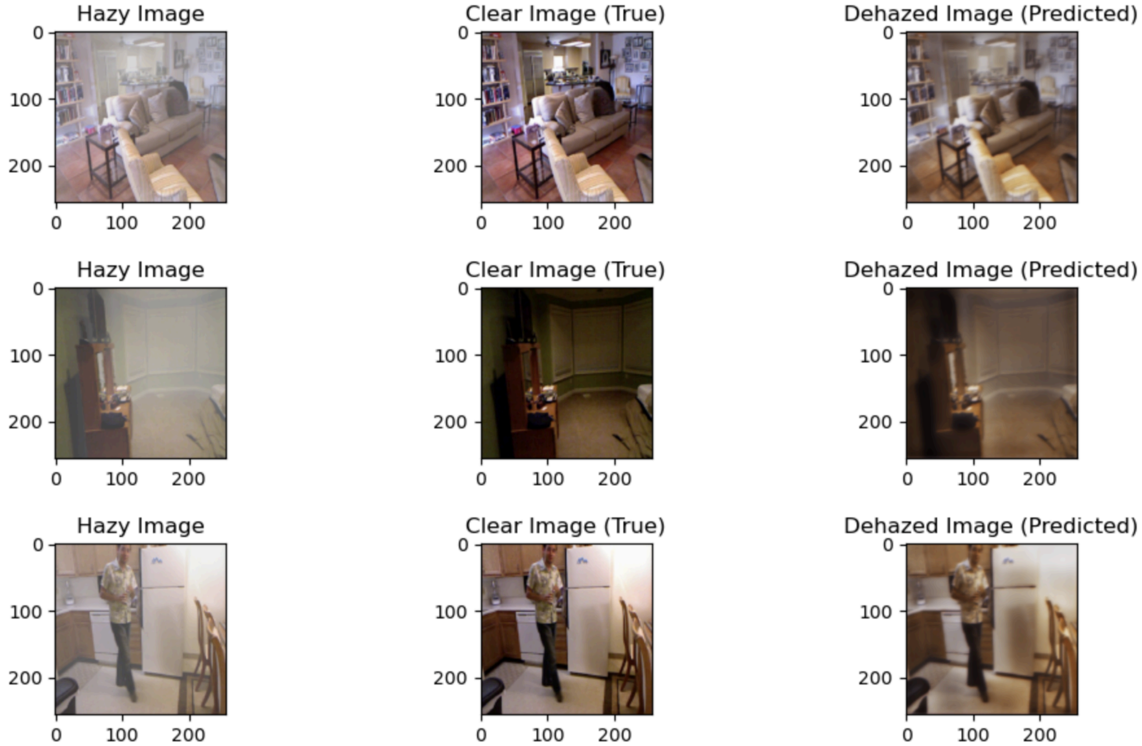
```
140 [D loss: 0.7081, acc.: 46.09%] [G loss: 0.7081]
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 239ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 357ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 243ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 248ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 247ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 289ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 263ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 268ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 236ms/step
```

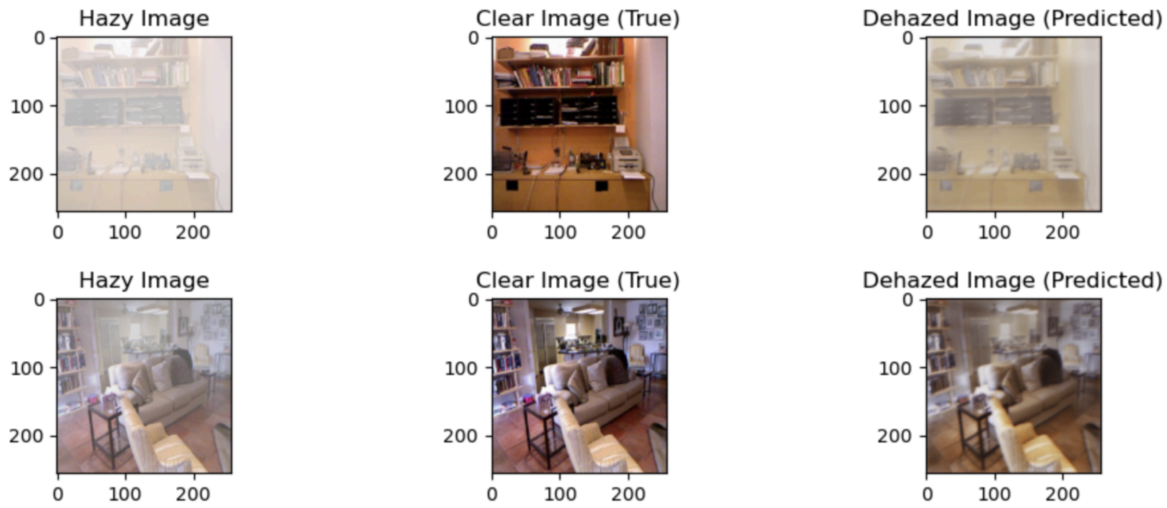
Temporal consistency:

```
Original frame size: (1080, 1920, 3), Dehazed frame size: (128, 128, 3)
SSIM calculation error: win_size exceeds image extent. Either ensure that your images are at least 7x7; or pass win_size explicit
nction call, with an odd value less than or equal to the smaller side of your images. If your images are multichannel (with color
set channel_axis to the axis number corresponding to the channels.
Video: 03000B01005DF1B9A4596E0617C2D8CE349E2F-7A48-48DB-8455-16D275DD06AE.mp4 | Not enough valid frames for PSNR and SSIM calcula
Video: 03000B01005DF1B9A4596E0617C2D8CE349E2F-7A48-48DB-8455-16D275DD06AE.mp4 | Avg Temporal Consistency (MAE): 0.14
```

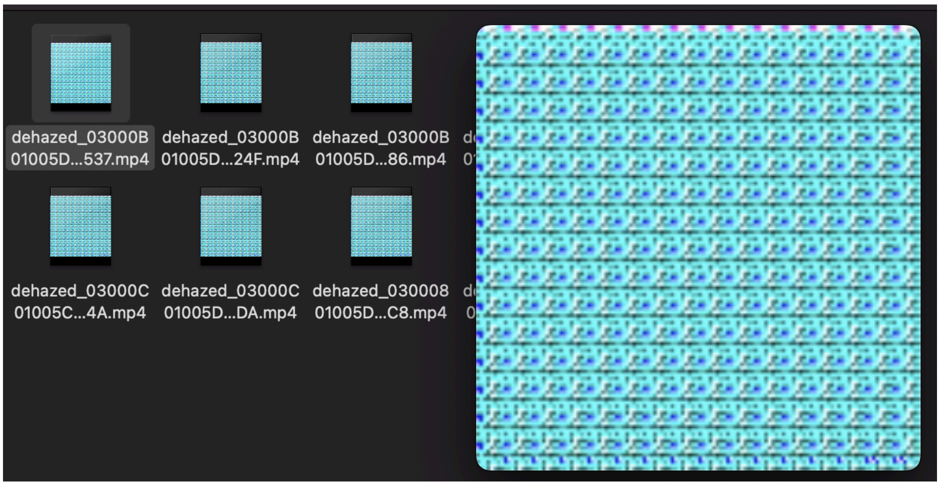
Results:

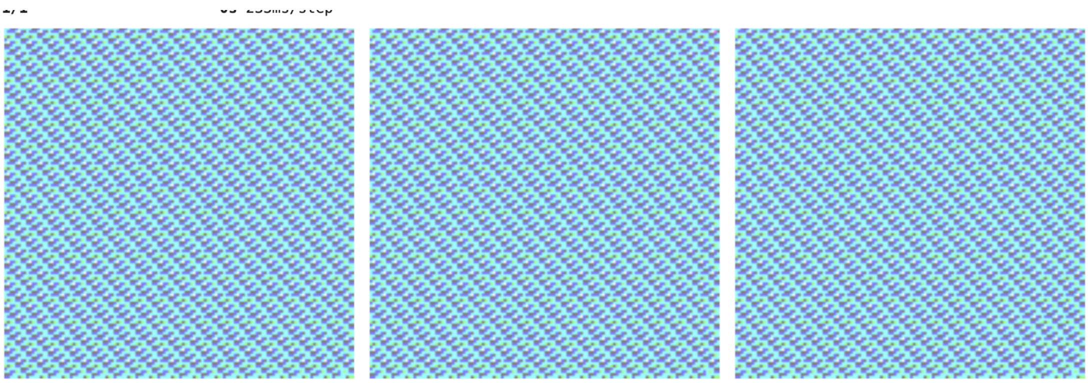
Image Dehazing:





Video:


dehazed_03000B 01005D...537.mp4
dehazed_03000B 01005D...24F.mp4
dehazed_03000B 01005D...86.mp4
dehazed_03000C 01005C...4A.mp4
dehazed_03000C 01005D...DA.mp4
dehazed_030008 01005D...C8.mp4



Key Learnings:

This project provided valuable insights into the use of deep learning techniques for image and video dehazing. The U-Net architecture performed exceptionally well for image dehazing, effectively restoring clarity and sharpness by utilising its encoder-decoder structure. The inclusion of skip connections played a crucial role, helping to preserve important spatial information during the dehazing process. U-Net achieved an accuracy of 81.4% on the test dataset, showcasing its efficiency in handling image dehazing tasks. In contrast, the GAN model faced significant challenges when applied to video dehazing due to hardware limitations. Training the GAN required more computational power than was available, leading to repeated kernel crashes and incomplete results. However, despite these difficulties, GANs have shown great potential for video dehazing, especially due to their ability to capture both spatial and temporal features across video frames. With access to better hardware, GANs could be a powerful solution for both image and video dehazing tasks in the future. This experience underscored the importance of having sufficient computational resources when tackling resource-intensive tasks like video processing.

Conclusion

The U-Net architecture demonstrated significant success in image dehazing, effectively restoring clear images from hazy inputs with its efficient encoder-decoder structure and skip connections. While the project primarily focused on image dehazing, the results indicate that U-Net is a highly capable model for such restoration tasks. GANs, on the other hand, showed great potential, particularly for more complex tasks like video dehazing, though hardware limitations hindered their full implementation. Future work should focus on improving computational resources and exploring advanced GAN techniques to enable efficient, real-time video dehazing for broader applications.

References

[1]-https://machinelearningmastery.com/what-are-generative-adversarial-networks-gans/

[2] - https://www.kaggle.com/datasets/balraj98/indoor-training-set-its-residestandard

[3] - https://jingnantes.github.io/acmmm21-youku-v1k/

[4] - https://link.springer.com/article/10.1007/s00530-021-00852-z

[5] - https://www.geeksforgeeks.org/u-net-architecture-explained/

[6] - https://www.mdpi.com/1424-8220/22/6/2257