TRIBHUWAN UNIVERSITY

INSTITUTE OF ENGINEERING

NATIONAL COLLEGE OF ENGINEERING

A

PROJECT PROPOSAL

ON

"VIDEO ENHANCEMENT USING DEEP LEARNING"

SUBMITTED BY:

Aayush Shrestha	077BCT002
Ishita Chalise	077BCT012
Pradish Tamrakar	077BCT019

SUBMITTED TO:

DEPARTMENT OF ELECTRONICS AND COMPUTER ENGINEERING

LALITPUR, NEPAL

MAY, 2024

Table of Contents

List of Figures

List of Abbreviations

CNN	Convolutional Neural Network
ECBAM	Enhanced Convolutional Block Attention Module
ERNB	Enhanced Residual Non-Local Block
GAN	Generative Adversarial Network
GPU	Graphics Processing Unit
HDD	Hard Disk Drive
HEVC	High Efficiency Video Coding
MSE	Mean Squared Error
OCT	Optical Coherence Tomography
PP	Post-Processing
PSNR	Peak Signal-to-Noise Ratio
ReSphereGAN	Relativistic Sphere Generative Adversarial Network
SRA	Spatial Resolution Adaptation
SRGAN	Super-Resolution Generative Adversarial Network
SSD	Solid State Drive
SVM	Support Vector Machine
VSRGAN	Video Super-Resolution Generative Adversarial Network
VVC	Versatile Video Coding

1. Introduction

1.1. Background

Video enhancement has evolved significantly with the advent of deep learning techniques, particularly Generative Adversarial Networks (GANs). Beginning with basic signal processing methods, the field progressed to embrace more sophisticated algorithms as computational power increased. Machine learning, through models like Support Vector Machines and Random Forests, demonstrated notable improvements over traditional approaches by learning mappings between low and high-quality video pairs. However, it was the emergence of deep learning, notably Convolutional Neural Networks (CNNs), that marked a paradigm shift in video enhancement. CNNs, with their ability to learn hierarchical features directly from raw pixel data, became instrumental in tasks such as super-resolution, as demonstrated by the Super-Resolution Convolutional Neural Network (SRCNN).

The introduction of GANs by Goodfellow et al. [1] in 2014 assisted in the new era of video enhancement. GANs, comprising of a generator and a discriminator trained adversarial, demonstrated remarkable capabilities in generating high-quality, realistic images and videos. The Super-Resolution GAN (SRGAN), introduced by Ledig et al. [2] in 2017, further advanced the field by combining adversarial and content losses to produce photo-realistic high-resolution images from low-resolution inputs. The extension of GANs to video enhancement tasks, such as the Video Super-Resolution GAN (VSRGAN), addressed the challenge of maintaining temporal consistency across frames. Techniques like VSRGAN incorporated temporal consistency terms in the loss function, reducing artifacts and ensuring smooth playback, thereby pushing the boundaries of video quality enhancement.

Despite these advancements, several challenges persist. The demand for large, high-quality training datasets, computational complexity, and the need to maintain temporal coherence in enhanced videos remain significant hurdles. Innovations in neural network architectures, such as transformer models and attention mechanisms, continue to improve model performance and efficiency. Ethical considerations, particularly regarding privacy in surveillance applications, underscore the importance of responsible deployment and compliance with data protection regulations. In conclusion, leveraging advanced deep learning techniques, particularly GANs,

holds immense promise in significantly enhancing video quality, with potential applications spanning entertainment, education, communication, and surveillance.

1.2. Problem Statement

Enhancing video quality is a fundamental challenge in various domains, ranging from entertainment to surveillance. The task involves addressing issues like low resolution, noise, compression, and poor lighting conditions to improve the overall visual fidelity of video content. In this project, we aim to develop a system for enhancing the quality of low-resolution and degraded videos using SRGAN. SRGAN, renowned for their effectiveness in natural language processing tasks, will be adapted and optimized to process video sequences efficiently while capturing both spatial and temporal dependencies. The proposed system will focus on tasks such as super-resolution, denoising, deblurring, and enhancing overall visual quality, with an emphasis on scalability and performance. Through this endeavor, we seek to advance the state-of-the-art in video enhancement technology, benefiting a wide range of applications and stakeholders, including researchers, industry professionals, and the general public.

1.3. Aim and Objectives

The aim of our project is to develop a video enhancement system that utilizes SRGAN to improve the quality of low-resolution and degraded videos.

The objectives of our project are:

- To design and develop a model capable of enhancing the visual quality of low-resolution.
- To develop a user-friendly interface for the video enhancement system.

1.4. Scope

The scope of the project encompasses the development of advanced models for enhancing the visual quality of low-resolution and degraded videos. It leverages deep learning and computer vision techniques, utilizing Convolutional Neural Network (CNN) and Super Resolution General Adversarial Network (SRGAN). The project involves designing the architecture of the video enhancement model, optimizing it for efficient training and inference, and curating datasets of

low-resolution and high-resolution video pairs for training and evaluation. Additionally, ensuring robustness to variations in input video quality and providing a user-friendly interface for intuitive interaction are key considerations in the project's scope.

2. Literature Review

2.1. Related Theory

2.1.1. CNN

A class of artificial neural network called a convolutional neural network, sometimes referred to as CNN or ConvNet, is mostly used for image processing. By utilizing convolutional layers, it can recognize patterns in images, aiding in image analysis. Input and output layers, as well as one or more hidden layers, are present in CNN, like most neural networks. In CNN, the hidden layers read the inputs from the input layer then perform convolution on the input values. Convolution in this context denotes a dot product or matrix multiplication. The Rectified Linear Unit (ReLU), a nonlinearity activation function, is used by CNN after matrix multiplication, followed by additional convolutions like pooling layers. To minimize the dimensionality of the data, pooling layers compute the outputs using functions like maximum pooling or average pooling. CNN typically has three layers: a convolutional layer, a pooling layer, and a fully connected layer.

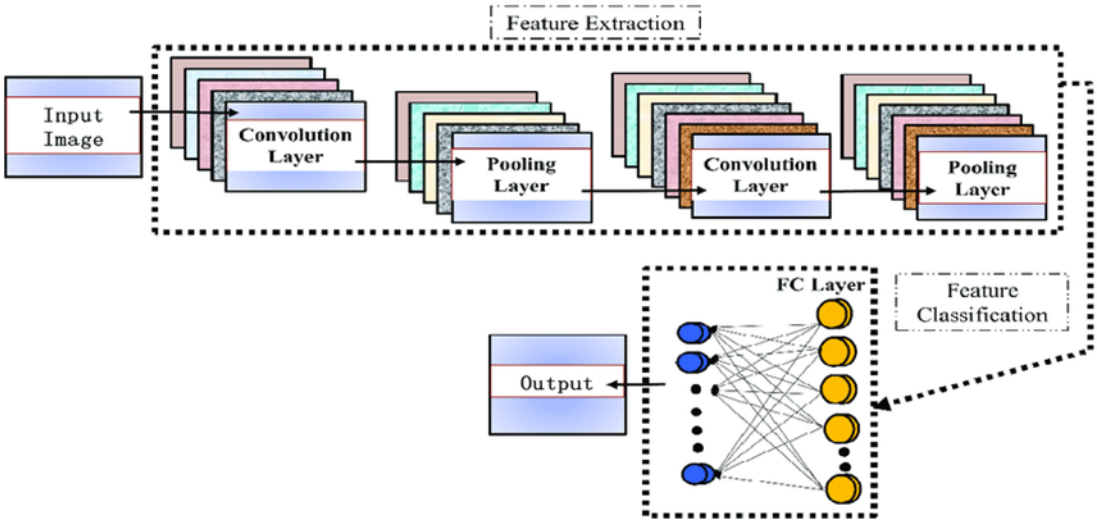

Figure 1:Architecture of CNN [6]

A convolutional layer is the main part of CNN model that performs dot product between two matrices where one matrix is the filter/kernel which is set of learnable parameters and the other matrix is the matrix representation of image. The feature detector, also known as a kernel or a filter, moves across the receptive fields of the image, to extract the feature present in the image.

This process is known as convolution. This helps for pattern recognition. 6 During the convolution operation the filter slides across the height and width of the image two-dimensional representation of the image known as activation map is generated which gives response of the kernel at corresponding spatial position of the image. The sliding size of the kernel is called a stride. Pooling is down-sampling in order to reduce the complexity for further layers. It can be compared to reducing the resolution when it comes to image processing. Pooling has no effect on the number of filters. One of the most popular kinds of pooling techniques is max-pooling. It partitions the image to sub-region rectangles, and it only returns the maximum value of the inside of that sub-region. Down-sampling does not preserve the position of the information. Therefore, it should be applied only when the presence of information is important. Moreover, pooling can be used with non-equal filters and strides to improve efficiency. The fully-connected layer is similar to the way that neurons are arranged in 16 traditional neural networks. Therefore, each node in a fully-connected layer is directly connected to every node in both the previous and in the next layer. It is a part of the classification layer which processes the final classification output. There can be no convolutional layers after a fully connected layer.

2.1.2. Super-resolution

Super-resolution is a technique in digital image processing aimed at enhancing the resolution of an image. It involves reconstructing a high-resolution image from one or more low-resolution versions of the same scene. The process can be categorized into two primary approaches: single-image super-resolution (SISR) and multi-image super-resolution (MISR). SISR focuses on enhancing the resolution of a single image, often utilizing sophisticated algorithms like deep learning models, particularly convolutional neural networks (CNNs), to predict high-frequency details and textures that are absent in the low-resolution image. MISR, on the other hand, leverages multiple low-resolution images taken from slightly different perspectives or at different times to reconstruct a higher-resolution image, often using techniques like image registration and fusion to combine the information from these multiple sources.

The field of super-resolution has seen significant advancements due to the integration of machine learning techniques. Deep learning models, such as Generative Adversarial Networks (GANs) and CNNs, have shown remarkable success in learning the complex mappings from low-resolution to high-resolution images. These models are trained on large datasets of paired low-

and high-resolution images, enabling them to understand and predict the missing high-frequency details effectively. Applications of super-resolution are vast and diverse, ranging from medical imaging, where high-resolution images are crucial for accurate diagnosis, to enhancing the quality of satellite imagery, security surveillance footage, and even improving the visual experience in consumer electronics like televisions and smartphones. The continuous development in computational power and algorithmic efficiency suggests that super-resolution will keep evolving, pushing the boundaries of what is possible in digital image enhancement.

2.1.3. GAN

Generative Adversarial Networks (GANs) are a class of machine learning frameworks designed by Ian Goodfellow and his colleagues in 2014.[1] GANs consist of two neural networks, the generator and the discriminator, which are set against each other in a game-theoretic scenario. The generator creates synthetic data samples, trying to mimic the true data distribution, while the discriminator evaluates these samples, aiming to distinguish between real and generated data. This adversarial process continues iteratively: the generator improves its ability to create realistic data to fool the discriminator, and the discriminator gets better at identifying the generated data. Through this dynamic, both networks enhance their capabilities, leading to the generation of highly realistic data samples.

The training process of GANs involves back-and-forth optimization. The generator learns to map a random input (often noise) into plausible data samples, while the discriminator learns to assign probabilities that any given input is real or fake. The goal for the generator is to maximize the probability of the discriminator making a mistake, while the discriminator aims to minimize this probability. This minimax game results in a unique equilibrium where the generator produces data that is indistinguishable from the real data according to the discriminator's evaluation. Despite challenges such as training instability and mode collapse, GANs have achieved remarkable success in various applications, including image and video generation, data augmentation, and even style transfer.

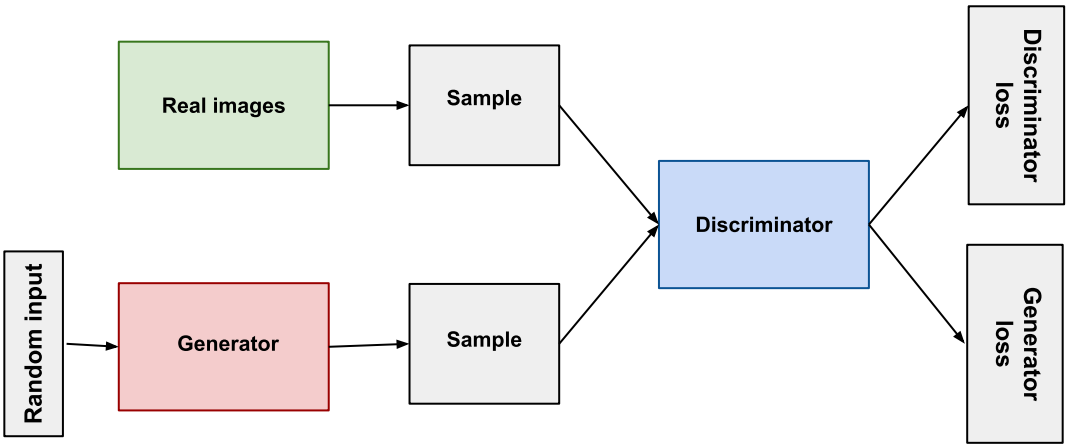

Figure 2: Architecture of GAN [7]

2.1.4. SRGAN

Super-Resolution Generative Adversarial Networks (SRGAN) are a significant advancement in image super-resolution, which aims to enhance the resolution of low-resolution images. Introduced by Ledig et al. in 2017 [2], SRGAN leverages the power of Generative Adversarial Networks (GANs) to generate high-resolution images from low-resolution inputs. The architecture consists of two primary components: a generator and a discriminator. The generator attempts to produce high-resolution images that are as realistic as possible, while the discriminator evaluates the authenticity of these generated images by distinguishing them from real high-resolution images. Through this adversarial process, the generator learns to create increasingly convincing high-resolution images.

The SRGAN framework incorporates perceptual loss functions, which include both content loss and adversarial loss, to ensure the generated images are not only high in resolution but also perceptually indistinguishable from real images. Content loss, typically measured using the mean squared error or feature maps from a pre-trained network like VGG, ensures that the generated images maintain the essential structure and details of the input images. Adversarial loss, on the other hand, encourages the generator to produce images with more realistic textures and fine details by attempting to fool the discriminator. The combination of these loss functions allows SRGAN to outperform traditional super-resolution methods, particularly in generating images

with high-frequency details and natural textures, making it a powerful tool for applications ranging from medical imaging to video enhancement.

2.2. Related Works

CVEGAN: A Perceptually-inspired GAN for Compressed Video Enhancement" by Di Ma, Fan Zhang, and David R. Bull [3] proposes a novel GAN architecture to enhance compressed video frames. The generator uses a Multi-Resolution block (Mul2Res) with multiple residual learning branches, an Enhanced Residual Non-Local Block (ERNB), and an Enhanced Convolutional Block Attention Module (ECBAM). The training strategy employs a relativistic sphere GAN (ReSphereGAN) and new perceptual loss functions. Evaluated within MPEG HEVC and VVC test models, CVEGAN achieves significant coding gains, with up to 28% improvement in post-processing (PP) and 38% in spatial resolution adaptation (SRA) for HM 16.20, and up to 8.0% and 20.3% respectively for VTM 7.0

“GANs-Based Intracoronary Optical Coherence Tomography Image Augmentation for Improved Plaques Characterization Using Deep Neural Networks” by Haroon Zafar et al. [4] The study presents a method for augmenting a dataset of intracoronary optical coherence tomography (OCT) images using conditional generative adversarial networks (cGANs). The goal is to enhance the classification of coronary plaques. The dataset consists of OCT images from 51 patients, which were augmented by factors of 5×, 10×, 50×, and 100× using cGANs. The augmented images were used to train an AlexNet model, and it was found that augmenting the dataset by a factor of 50× improved classification accuracy by 15.8%. The study demonstrates that synthetic images generated by cGANs can effectively complement real images in training deep learning models, resulting in better classification performance.

"Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network" by Christian Ledig et al. [2] This paper introduces SRGAN, a Generative Adversarial Network (GAN) for single image super-resolution. The SRGAN aims to produce photo-realistic high-resolution images from low-resolution inputs by employing a perceptual loss function that combines an adversarial loss with a content loss. The adversarial loss helps the generator network produce images that are indistinguishable from real high-resolution images, while the content loss, based on VGG network feature maps, focuses on perceptual similarity rather than pixel-wise accuracy. The SRGAN significantly improves the visual quality of super-resolved images,

particularly for high upscaling factors like 4×, surpassing traditional methods that optimize for mean squared error (MSE) and peak signal-to-noise ratio (PSNR)

"A Comparative Analysis of SRGAN Models" by Fatemeh Rezapoor Nikroo et al. [5] evaluates the performance of several state-of-the-art Super-Resolution Generative Adversarial Network (SRGAN) models on real-world images. The Enhanced Deep Super-Resolution (EDSR) model is highlighted for its effectiveness, employing deep convolutional neural networks to achieve high-quality image reconstructions. Similarly, the Enhanced Super-Resolution Generative Adversarial Network (ESRGAN) and its upgraded version, Real-ESRGAN, are noted for their ability to generate realistic textures and high-resolution images through advanced architectural modifications like Residual-in-Residual Dense Blocks (RRDB). The study also emphasizes the importance of GANs in enhancing image quality, particularly for applications involving Optical Character Recognition (OCR), where improved visual fidelity directly correlates with better text recognition accuracy.

3. Methodology

3.1. Feasibility Study

3.1.1. Technical Feasibility

The project is a comprehensive desktop-based application aimed at enhancing video quality using advanced deep learning models. The primary technologies and tools involved in the system include PyTorch, TensorFlow, high-performance GPUs (such as NVIDIA Tesla or RTX series), OpenCV for video processing, and scikit-image for image quality assessment. Each of these technologies is readily available and the skills required to extract meaningful outputs are manageable. The time limitations for product development align well with the ease of implementation using these technologies. From these considerations, it is evident that the product is technically feasible.

3.1.2. Operational Feasibility

The main purpose of the system is to develop a desktop application that facilitates users in enhancing the visual quality of low-resolution and degraded videos. The system will provide easy access to its features, requiring minimal knowledge from the users to operate it to its full

capability. With a user-friendly interface designed using modern web and desktop application frameworks like React and Electron, the end users will find the system intuitive and easy to use. Therefore, it is clear that the system is operationally feasible.

3.1.3. Economic Feasibility

From an economic standpoint, the system's development cost is based primarily on the expenses for acquiring high-performance GPUs, software licenses, and cloud computing resources. However, the software used for development, such as PyTorch and TensorFlow, is open-source and freely available, reducing the overall cost. Thus, the system is economically feasible.

3.1.4 Schedule Feasibility

The goals and principles guiding the development of the system are well-understood and can be accomplished within the given timeframe. The project is expected to span 12 months, with phases for research, development, testing, and deployment. By adhering to a strict schedule and keeping the project objectives in mind, it is anticipated that the system will be completed within the designated timeline. Therefore, the system is schedule feasible.

3.2. Requirement Analysis

Functional Requirements

Video Input and Output: The system must accept various video formats and output enhanced videos in the same format as the input or offer a choice of formats.

User Interface: Users should be able to upload videos, preview them, and view processing progress.

Processing and Enhancement: Users should be able to select different pre-trained models, process multiple videos in batches, and optionally perform real-time enhancement.

Error Handling: Clear error messages for unsupported formats, corrupted files, or processing failures.

Non-Functional Requirements

Performance: Efficient processing with GPU acceleration, handling large files and batches without performance degradation.

Usability: Intuitive interface and comprehensive documentation for users of varying technical expertise.

Reliability: Robust performance with minimal downtime, and mechanisms for backup and recovery.

Security: Secure storage and processing of user data.

Compatibility: Compatibility with major operating systems and clear management of software dependencies.

Environmental Impact: Optimization for energy efficiency, especially during intensive processing tasks.

3.3. Proposed System Design

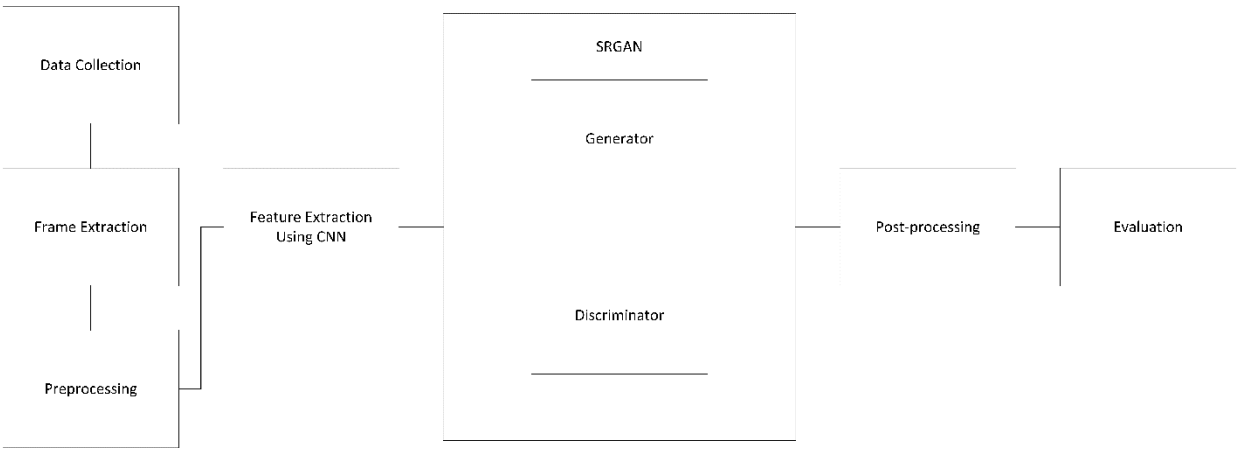

Figure 3: System Block Diagram

1. Data Collection:

Collect a diverse dataset of low-resolution video clips relevant to your application domain. Ensure the dataset covers various scenes, lighting conditions, and motion types.

2. Frame Extraction:

Extract individual frames from the collected low-resolution video clips using appropriate tools or libraries. Pay attention to maintaining the temporal order of frames.

3. Preprocessing:

Normalize the pixel values of the frames to a suitable range [0, 1]

Apply preprocessing techniques such as resizing, cropping, and data augmentation to enhance the quality and diversity of the training data.

4. SRGAN Architecture Implementation:

Implement the SRGAN architecture, consisting of a generator and a discriminator, using TensorFlow.

4.1. Generator Training:

Train the generator network using the low-resolution frames as input and the corresponding high-resolution frames as ground truth labels.

Utilize loss functions such as content loss (e.g., L1 or L2 loss) and adversarial loss (e.g., binary cross-entropy loss) to guide the training process.

4.2. Discriminator Training:

Train the discriminator network to distinguish between real high-resolution frames and generated ones.

Use adversarial training to improve the discriminator's ability to differentiate between real and generated frames.

5. Post-processing

Reconstruct the enhanced high-resolution frames into complete video sequences.

Combine the individual enhanced frames in the correct temporal order to recreate the high-resolution videos.

6. Evaluation:

Evaluate the performance of the trained SRGAN model using quantitative metrics such as PSNR, SSIM, and LPIPS. Fine-tune the model parameters and hyperparameters based on the evaluation results to improve performance.

3.4. Workflow of Application

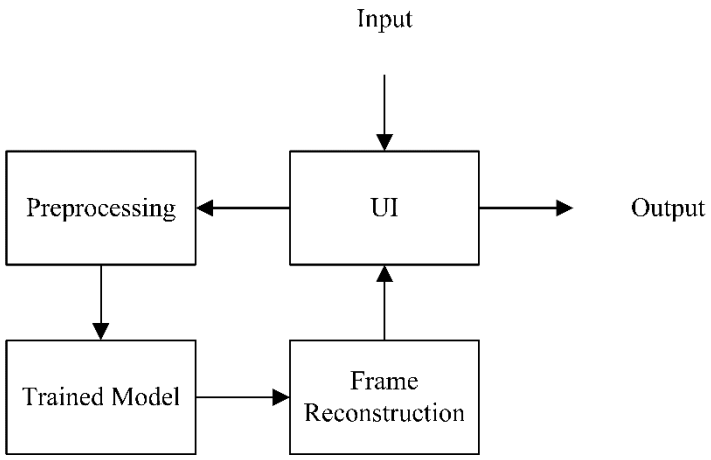

Figure 4: Workflow of the Application

User Interface (UI)

- Input: Users upload low-resolution videos through the application interface.
- Output: The enhanced video is presented back to the user for preview and download.

Preprocessing

- The video frames are normalized and resized.

Trained Model

- The preprocessed frames are enhanced using the model.

Frame Reconstruction

- The enhanced frames are combined to reconstruct the high-resolution video.

3.5. System Requirements

3.5.1. Hardware Requirements

A computer system with the following specifications

• CPU: Intel Core i5 (10th generation or newer) or AMD Ryzen 5 (4000 series or newer)

• Installed memory (RAM): 16 GB or above

• Storage Device: SSD or HDD

• GPU: CUDA enabled GPU (NVIDIA GTX 3050)

3.5.2. Software Requirements

- Pythons
- Flask
- Keras
- TensorFlow
- PyTorch
- Vscode
- Jupyter Notebook

3.6. Algorithm

SRGAN is a generative adversarial network for single image super-resolution. It uses a perceptual loss function which consists of an adversarial loss and a content loss. The adversarial loss pushes the solution to the natural image manifold using a discriminator network that is trained to differentiate between the super-resolved images and original photo-realistic images. In addition, the authors use a content loss motivated by perceptual similarity instead of similarity in pixel space. The actual networks - depicted in the Figure to the right - consist mainly of residual blocks for feature extraction.

Formally we write the perceptual loss function as a weighted sum of a content loss l_X^{SR} and an adversarial loss component l_{GEN}^{SR}:

$$l^{SR} = l_X^{SR} + 10^{-3} l_{Gen}^{SR}$$

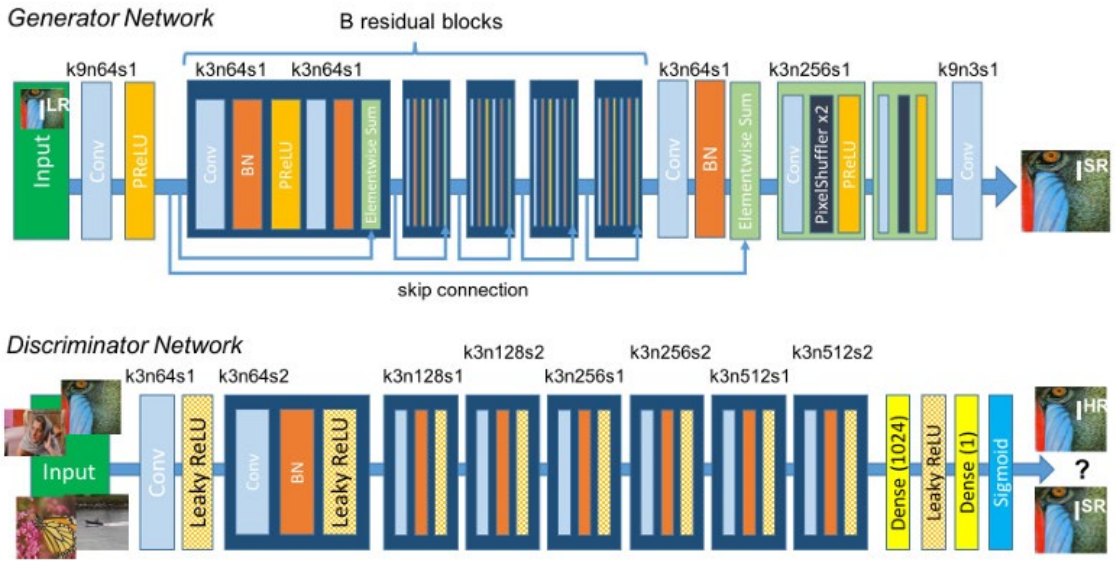

Figure 5: Architecture of Generator and Discriminator Network

4. Time Schedule

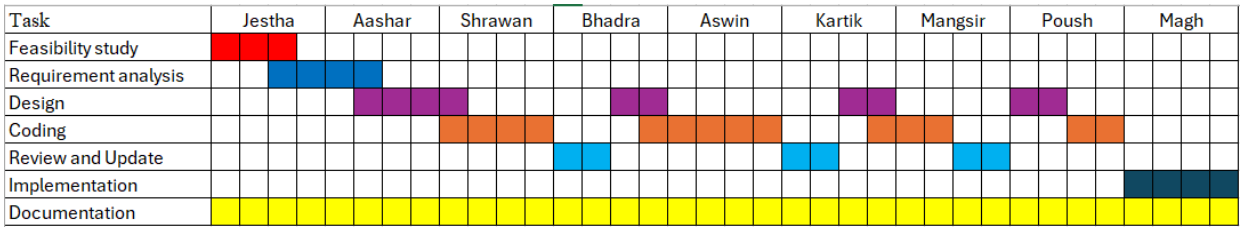

Figure 6: Gantt Chart

5. Expected Output

The expected output of the video enhancement system is a high-resolution, high-quality video generated from the user-uploaded low-resolution video. The system will significantly improve the resolution, sharpening fine details that were previously blurry or pixelated. The enhanced video will be visually appealing, offering clearer and more vivid video, thereby increasing its utility for various applications such as entertainment, surveillance, and more. Users will be able to preview the enhanced video through the application interface and download it in standard formats like MP4, with optimized file sizes for easy storage and sharing. Overall, the system aims to provide a user-friendly experience that delivers significantly improved video quality.

References

[1] Goodfellow, Ian J., et al. “Generative Adversarial Networks.” ArXiv.org, 2014, arxiv.org/abs/1406.2661.

[2] Ledig, Christian, et al. “Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network.” ArXiv.org, 2016, arxiv.org/abs/1609.04802.

[3] Ma, Di, et al. “CVEGAN: A Perceptually-Inspired GAN for Compressed Video Enhancement.” ArXiv.org, 2020, arxiv.org/abs/2011.09190.

[4] Zafar, Haroon, et al. “GANs-Based Intracoronary Optical Coherence Tomography Image Augmentation for Improved Plaques Characterization Using Deep Neural Networks.” Optics, vol. 4, no. 2, Multidisciplinary Digital Publishing Institute, Mar. 2023, pp. 288–99.

[5] Nikroo, Fatemeh Rezapoor, et al. “A Comparative Analysis of SRGAN Models.” ArXiv.org, 2023, arxiv.org/abs/2307.09456.

[6] https://www.researchgate.net/figure/Structure-of-CNN-There-are-some-special-structural-features-in-the-CNN-architecture_fig1_344807764

[7] “Overview of GAN Structure.” Google for Developers, 2022, developers.google.com/machine-learning/gan/gan_structure.