TRIBHUVAN UNIVERSITY

INSTITUTE OF ENGINEERING

PULCHOWK CAMPUS

A

FINAL REPORT

ON

VIDEO UPSAMPLING OF CCTV FOOTAGES

SUBMITTED BY:

NIKHIL ARYAL (PUL075BCT053)

SANDESH POKHREL (PUL075BCT076)

SANJAY BHANDARI (PUL075BCT079)

SANTOSH PANGENI (PUL075BCT082)

SUBMITTED TO:

DEPARTMENT OF ELECTRONICS & COMPUTER ENGINEERING

MAY 2, 2023

Page of Approval

TRIBHUVAN UNIVERSIY
INSTITUTE OF ENGINEERING
PULCHOWK CAMPUS
DEPARTMENT OF ELECTRONICS AND COMPUTER ENGINEERING

The undersigned certifies that they have read and recommended to the Institute of Engineering for acceptance of a project report entitled **"Video Upsampling of CCTV Footages"** submitted by **Nikhil Aryal**, **Sandesh Pokhrel**, **Sanjay Bhandari**, **Santosh Pangeni** in partial fulfillment of the requirements for the Bachelor's degree in Electronics & Computer Engineering.

.....
Supervisor
Dr. Sanjeeb Prasad Panday
Associate Professor
Department of Electronics and Computer Engineering,
Pulchowk Campus, IOE, TU.

.....
Internal examiner

.....
External examiner

Date of approval:

Copyright

Acknowledgments

We would like to express our sincerest gratitude to **Associate Prof. Dr. Sanjeeb Prasad Panday**, our supervisor, for his invaluable insights and feedback throughout the project. His unwavering support and participation played a significant role in the completion of our project and helped us take it one step further.

We would also like to acknowledge the **Department of Electronics and Computer Engineering** for their collaborative efforts and coordination in the successful completion of this project.

Additionally, we express our heartfelt appreciation to our teachers and instructors for their guidance and support, which helped us to overcome various challenges we faced throughout the project.

Sincerely,
Nikhil Aryal
Sandesh Pokhrel
Sanjay Bhandari
Santosh Pangeni

Abstract

The idea of super resolution and image upsampling have taken the field of computer vision by storm. New methods to upsample a grainy and low resolution videos are now the new chase. Our research is focused on upsampling a CCTV video through the use of deep learning techniques. Video superresolution often show sub-par results because they tend to have more components to process than their image counterparts, namely temporal dimension apart from the usual spatial dimension. In this research, we have studied these components and developed a pipeline that effectively processes the spatio-temporal information through optical flow, backed up by novel deep learning based VSR practices such as feature alignment, aggregation and upsampling. We examined and improved the pipeline based on the BasicVSR architecture and developed a model of our own by introducing residual in residual dense blocks. The new model RD-BasicVSR, was successful in surpassing the results of BasicVSR in both PSNR and SSIM metrics at same experimental settings.

Keywords: *VSR, Basic VSR CNN, Spatial upsampling, Spatio-temporal upsampling, Residual Blocks, Optical Flow*

Contents

List of Figures

List of Tables

List of Abbreviations

CCTV Closed-Circuit Television

CNN Convolutional Neural Network

dB decibel

DRRN Deep Recursive Residual Network

GAN Generative Adversarial Networks

HR High Resolution

JPEG Joint Photographic Experts Group

LR Low Resolution

MEVA Multiview Extended Video with Activities

MP Megapixel

MSE Mean Squared Error

PNG Portable Network Graphics

PSNR Peak Signal to Noise Ratio

RDN Residual Dense Network

REDS Realistic and Dynamic Scenes

RRDB Residual in Residual Dense BLock

RRN Recursive Residual Network

SISR Single Image Super Resolution

SSIM Structural Similarity Index

VFX Visual Effects

VSR Video Super Resolution

1. Introduction

1.1 Background

Computer vision is a field of computer science which tries to make the machine understand human perception of an image. A machine sees an image as a simple grid of numbers crammed together in a sequence. This grid of number, pixels, is what determines the resolution of an image. The greater the number of pixels, the better the quality of the image as perceived by a human eye and the better the detail.

With improvement in technology we are able to render large number of pixels. A pixel size of 2k has become customary whereas 4k is sought after. The only thing limiting us to using smaller images is the size and speed with with we can render these images. To get better detail researchers started working on compression algorithms like JPEG and PNG, rendering methods were improved and lastly artificial intelligence entered the fray with deep learning techniques utilizing CNN [1], [2],[3], [4] , GAN [5], [6], [7],[8], [9], [10] and diffusion models [11], [12].

Aforementioned methods work well on still images however there is much room for improvement in the domain of videos as you not only have to consider spatial presence of features in images but also the temporal constraint needs to be factored in. The size and computation requirements of videos also exceed that of images. For high end devices, the quality of video is indiscernible for human eyes. For images and videos taken from CCTV footage the quality is not satisfactory compared to what technology has achieved in other general use cases.

1.2 Problem Statement

The quality of CCTV videos is not satisfactory and comes at a considerable cost of equipments. Even with the advancement of deep learning in the field of computer vision, upscaling a video into higher definition has not become a primary part of application and remains a active research problem. The clear difference in the quality we intend to dive upon the problem of spatio-temporal upsampling of videos focusing on how the literature and research can be useful in CCTV footages. With the use of novel Deep Learning methods, we propose to design a system that improves the quality of videos that are captured by ordinary CCTV.

1.3 Objectives

- Improve the quality of CCTV videos through spatio-temporal upscaling
- Improve the perceived quality of videos

1.4 Scope

The principal perk of this project is that it bridges the gap that prohibits dozens of effective image super resolution models to work just as effective in the field of videos upsampling. The final end product will provide a system that will increase the resolution of live incoming CCTV videos. However, the project offers much more than that. The system can be implemented in computationally medium and high range mobile devices and PCs to upsample the recordings and live videos. It can also be extended to be used as a part of gaming and VFX designing to improve the quality of animation and rendering.

Apart from direct applications in products, the research conducted in this project will definitely provide much needed aid to the study of related factors and elements that contribute to the improvement of Video based generative models by targeting the existing shortcomings in the field and proposing the changes.

1.5 Organization of Report

The project report is organized into six chapters. After this introductory chapter, chapter two describes the underlying literature in the field of image and video super resolution using deep learning methods like convolutional networks, generative adversarial networks and diffusion model.

Chapter three describes the relevant theory necessary to understand the project. Image and video super resolution, convolutional neural network and their construction, bilinear and bicubic methods of upsampling and downsampling.

The fourth chapter describes the crux of the project and how we upsample a low resolution video to higher resolution. It is achieved through a series of computation on the image frames of the video namely, propagation, alignment and aggregation steps. This chapter also includes the datasets, training settings and the metrics we used for evaluation.

In chapter five we have presented the results of our research and the essential discussions that go with those obtained results. The upsampled results of the video, perceived difference

in quality between BasicVSR and our model RD-BasicVSR is included in this chapter.

In the last chapter we summarize our research project and discuss the recurring limitations. Future work describes how we can improve upon the limitations moving forward in the same area of research.

2. Literature Review

The field of upsampling videos for better quality is not a novel one. The quality of video is determined by the resolution of frame images and the frame rate of an image. A standard CCTV has 30 frames per second while its resolution depends on camera quality, with varying ranges(generally 2MP-8MP) according to the manufacturer. In most use cases however CCTV captures are of low resolution(2MP or lower) resulting in grainy videos.

The improvement of such videos can thus be done through image upscaling and super resolution techniques, a popular domain in computer vision. The most basic attempts to super resolution include interpolations like nearest neighbour, bilinear and bicubic interpolations which increase the number of pixels by specifying pixel values based on nearest neighbour or interpolating a group of pixels around a known pixel value. Advancement and improvement of shading techniques are used in video games to upsample the quality of video without affecting the player's experience.[13]

Dong et. al [14] proposed Single Image Super Resolution (SISR) which aims at retrieving a high resolution (HR) image from a low resolution (LR) input. It originally used deep convolutional neural networks for the task. Ever since, a lot of architectures and strategies have been involved for developing better models. Later on, Generative models such as GANs and Diffusion models were involved that resulted in massive boost in performance of these models. SRGAN [15] and ESRGAN [8] are two such widely popular models that took the field by the storm.

The transition of such methods from images to videos has been relatively slower. The aforementioned models were all used in this process but the results expected due to their massive success in image based systems could not be replicated. The problem in learning and upsampling spatio-temporal images turned out to be one of the main issues. Chu et. al [16] proposed a self-supervised model for GAN based video generation addressing this issue. Although the output was pretty stable, the model suffered from sub-optimal details in videos.

Wang et. al [17] proposed EDVR in 2019 that incorporated components to capture large motion and preserve important features using spatial and temporal attention. Similarly, they proposed BasicVSR [18] in 2021 to exploit additional dimensions needed for video

super resolution. These papers are effective for video upsampling tasks but they fall short to address the live feed upsampling, which is the goal in most live surveillance systems such as CCTV cameras.

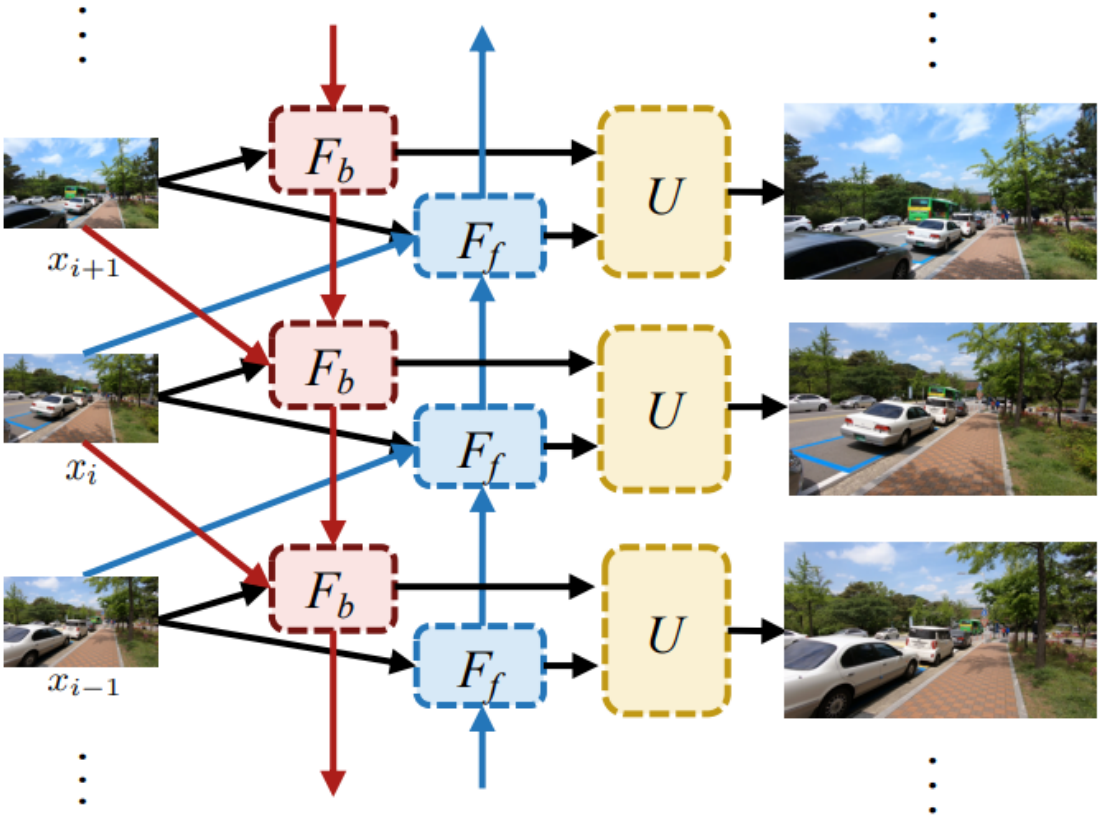

Figure 2.1: Architecture and components of BasicVSR

Liu et. al [19] studied several novel video upsampling techniques by categorizing them into seven different categories based on the methods they employed to leverage information contained in the video frames. The key takeaways from the survey provide detailed insights on working of a number of super-resolution architectures and their weak points that will eventually be used as topics of research for our project. Video super-resolution are the video counterparts of image super-resolution techniques, and are thus derived from different image based methods such as the ones using Convolutional Neural Networks, Residual Networks, Recurrent Neural Networks and Generative models based methods [19].

The two basic frameworks for existing VSR techniques [18] [20] [21] are sliding-window and recurrent. Previous techniques [22] in the sliding window architecture execute spatial warping for alignment and forecast the optical flow between low resolution (LR) frames. Subsequent methods use an implicit alignment strategy that is more advanced. Deformable

convolutions (DCNs) [23] [24], for instance, are used by TDAN [25] to align several frames at the feature level. DCNs are additionally utilized by EDVR [17] in a multi-scale manner for more precise alignment. DUF [26] uses dynamic upsampling filters to implicitly handle motions. Some strategies adopt a recurrent framework. A hidden state adaption module and a recurrent detail structural block are suggested by RSDN [27] to increase robustness to appearance change and error accumulation. The aforementioned studies have led to many new and sophisticated components to address the propagation and alignment problems in VSR. RRN [28] adopts a residual mapping between layers with identity skip connections to ensure a fluent information flow and preserve the texture information over long periods.

Learning temporal coherence along with spatial flow is the most important step in video super resolution. Optical flow based methods have been used for quite some time in VSR projects. The majority of the traditional optical flow algorithms, which date back to Horn and Schunck (1981) [29], have aimed to reduce hand-crafted energy terms for picture alignment and flow smoothness [30] [31]. Other cutting-edge techniques, such as DC Flow [32] and EpicFlow [33], further take advantage of image boundary and segment cues to enhance flow interpolation amid sparse matches. Recently, end-to-end deep learning techniques enabling quicker inference were proposed [34] [35].

The drawback of traditional approaches is that they frequently erroneously assume the image brightness change and the spatial organization of the flow. Many techniques concentrate on increasing robustness by altering the assumptions. The main benefit of studying flow computation is that we don't have to manually adjust these presumptions [36]. Instead, the learnt network incorporates the fluctuation in image brightness and spatial smoothness. The concept of employing a spatial pyramid also has a lengthy history, going back to [37], with its earliest application in the formulation of the classical flow appearing in [38].

Stacking residual-in-residual dense blocks (RRDB) has shown improved performance in SR problems and has been adopted by many SR methods such as [39] and RealESRGAN [40] [41]. RRDB blocks have been used in vanilla form and in their 2D forms in different image super resolution works. It was initially proposed by Wang et. al. [8] as a mechanism to improve generator's performance in GAN based networks. It has been demonstrated that removing BN layers improves performance and lowers computing cost for a variety of PSNR-oriented activities, including SR [42] and deblurring. Moreover, deleting BN layers aids in enhancing generalization potential as well as lowering computational complexity and memory utilization. This block has been used in our project as an alternative to vanilla residual blocks proposed in the BasicVSR paper.

3. Related Theory

3.1 Image Super-Resolution

Image Super-Resolution refers to the task of enhancing the resolution of an image from LR (Low Resolution) to HR (High Resolution). It can be thought of as an Image to Image translation task. Deep learning techniques have been fairly successful in solving the problem of image and video super-resolution. It is an important class of image processing techniques in computer vision and image processing and enjoys a wide range of real-world applications, such as medical imaging, satellite imaging, surveillance and security, astronomical imaging, amongst others.

Figure 3.1: LR image on the left & corresponding HR image generated using SRResNet[5]

3.2 Video Super-Resolution

Video Super-Resolution is a technique that enhances the resolution of low-resolution videos to a higher resolution version. This technique is useful when working with old or low-quality video footage. In deep learning, video superresolution is achieved using convolutional neural networks. These networks are effective at processing images and videos by learning to recognize patterns through multiple layers of convolutional filters.

To perform video superresolution using deep learning, a CNN or Transformer based model is trained on a dataset of low-resolution and high-resolution video pairs. The network learns to map low-resolution video frames to their corresponding high-resolution frames during training. After training, the model can be used to superresolve new low-resolution videos by passing each frame through the network.

3.3 CNN (Convolutional Neural Networks)

Convolutional Neural Networks CNN have become a cornerstone in the field of computer vision and are widely used in various applications, including image and video superresolution. CNNs are a type of artificial neural network that are inspired by the structure and function of the visual cortex in animals. In a CNN, an input image or video is passed through a series of convolutional layers that apply filters to extract relevant features. The output of each convolutional layer is then passed through a non-linear activation function, such as ReLU, to introduce non-linearity in the model. This allows the CNN to learn complex relationships between input and output data.

CNNs are widely used in super resolution applications, where the goal is to generate a high-resolution image from a low-resolution input. By training on pairs of low-resolution and high-resolution images, a CNN can learn to extract features from the low-resolution input and use them to generate a high-resolution output. This makes CNNs an effective tool for a wide range of image processing tasks, including super resolution.

Several CNN-based approaches have been proposed for video superresolution, including those that use deep architectures with skip connections, attention mechanisms, and adversarial loss functions. Some notable works include DUF-Net [26], TDAN [25], and BasicVSR [18].

3.4 Bicubic Downsampling

Bicubic downsampling is a technique used to reduce the size of an image by resampling it using a bicubic interpolation function. The process involves averaging the values of adjacent pixels in the original image to generate new pixels in the downsampled image. The bicubic interpolation function is used to generate these new pixels by fitting a 2D cubic polynomial to a set of neighbouring pixels in the original image.

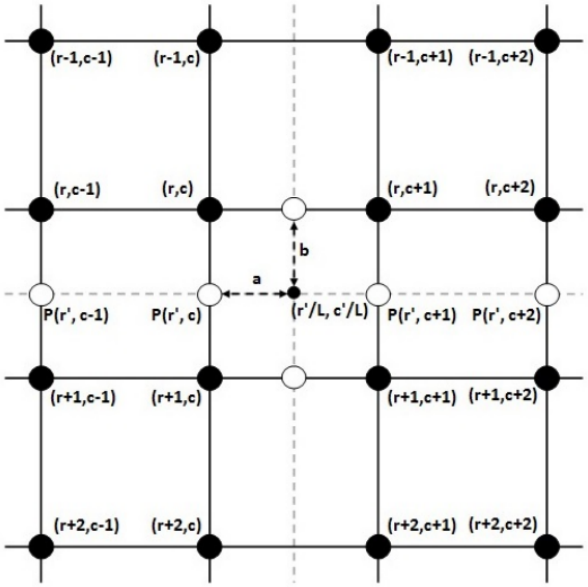

Figure 3.2: Bicubic Interpolation

The first stage of downsampling involves interpolating the values of each row in the grid using a cubic spline interpolation, such as the Catmull-Rom spline. This results is a set of intermediate values, which are then used in the second stage of interpolation to estimate the value of the point of interest.

3.5 Bilinear Upsampling

Bilinear upsampling is a technique used to increase the resolution of an image by resampling it using a bilinear interpolation function. The process involves generating new pixels in the upsampled image by averaging the values of adjacent pixels in the original image. It works by fitting a 2D plane to four neighboring pixels in the original image and using this plane to generate new pixel values in the upsampled image. The new pixel values are calculated as weighted averages of the neighboring pixel values, with the weights determined by the distance between the new pixel and each of the four neighboring pixels.

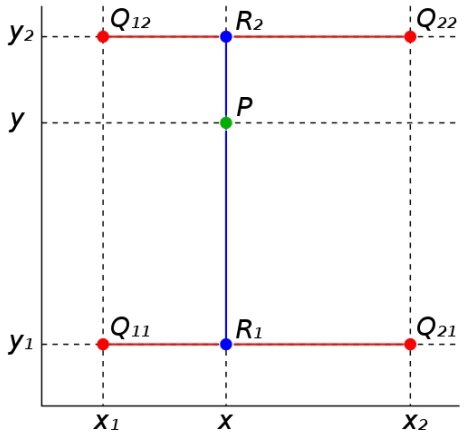

Figure 3.3: Bilinear Interpolation

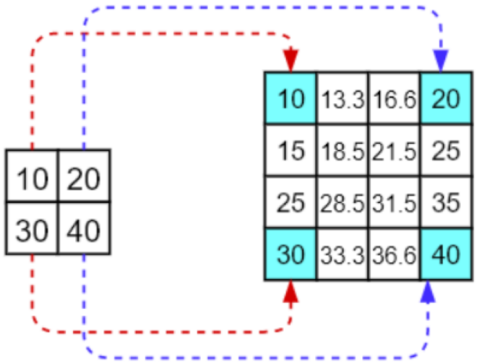

Figure 3.4: Bilinear Upsampling

Interpolating for P from four neighbouring points, first linear interpolation is done in the x-direction for the two rows. This yields,

$$f(x, y_1) = \frac{x_2 - x}{x_2 - x_1} f(Q_{11}) + \frac{x - x_1}{x_2 - x_1} f(Q_{21}) \tag{3.5.1}$$

$$f(x, y_2) = \frac{x_2 - x}{x_2 - x_1} f(Q_{12}) + \frac{x - x_1}{x_2 - x_1} f(Q_{22}) \tag{3.5.2}$$

To obtain the desired estimate, interpolation in the y-direction is performed as follows,

$$f(x, y) = \frac{y_2 - y}{y_2 - y_1} f(x, y_1) + \frac{y - y_1}{y_2 - y_1} f(x, y_2) \tag{3.5.3}$$

4. Methodology

The process of upsampling a video involves a series of image manipulations and the use of modules to abstract these steps. The overall procedure can be summarized into four main phases, namely propagation, alignment, aggregation and upsampling. Propagation specifies how the information in a video sequence is leveraged and deals with the transfer of features from one frame to another. Our architecture involves the use of a bidirectional propagation network, which allows the features to be propagated forward and backward in time independently. The alignment phase, which was inspired by the BasicVSR [18] technique, involves the computation of optical flow for spatial alignment of features using a pretrained Spynet [36] model, rather than optical flow for image alignment. During the aggregation step in BasicVSR, the features extracted from multiple frames are concatenated along the channel dimension to generate a single set of features for each frame. Finally, the upsampling step comprises the four-fold bilinear upsampling of the re-calculated features from the previous stage, as well as the separate computation of pixel shuffle twice to upsample the image four times. The results of the bilinear and pixel shuffle procedures are combined to determine the final output of the overall process.
BasicVSR utilizes a number of features such as bidirectional propagation, feature alignment, aggregation and upsampling [18]. These features have been found to be very useful in capturing the spatio-temporal domains associated with video dataset. These elements led to the usage of the BasicVSR model in our research.

4.1 Propagation

Propagation is one of the most influential components in VSR. It specifies how the information in a video sequence is leveraged. The proposed bidirectional propagation method of BasicVSR incorporates propagation of features in forward and backward direction in time independently. This eventually solves two main problems seen in traditional local and unidirectional propagation: loss of information of distant frames using sliding window approach in local propagation and imbalanced reception of information in unidirectional propagation. These problems have been verified by significant drop in PSNR values, some as much as 0.5 dB compared to the bidirectional approach [18].

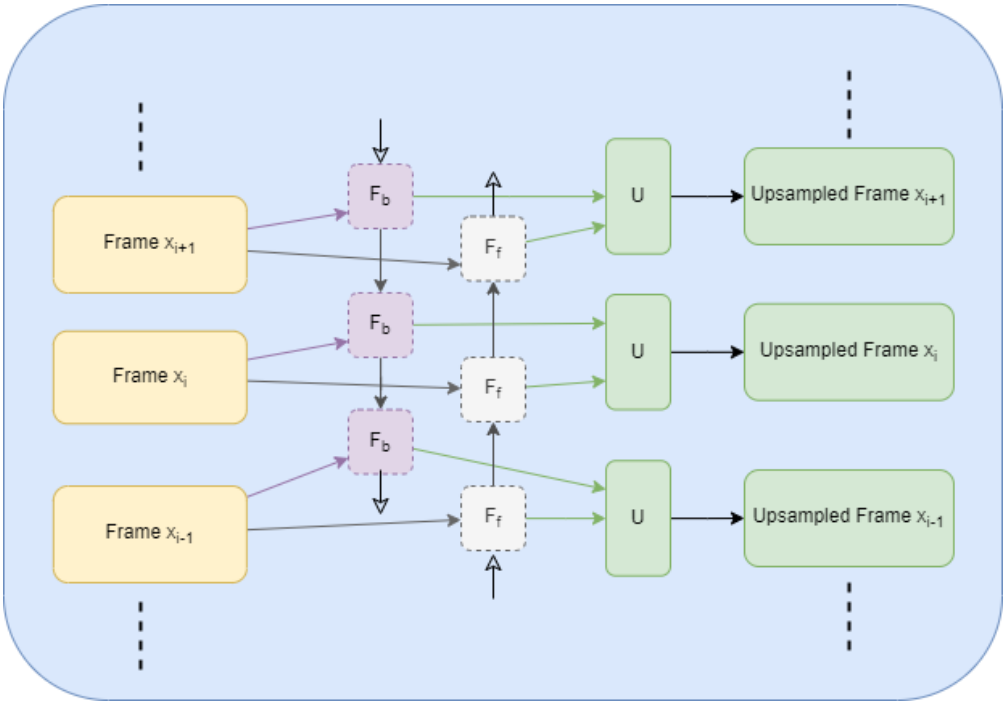

Figure 4.1: Shared architecture of BasicVSR and RD-BasicVSR

$$h_i^b = F_b(x_i, x_{i+1}, h_(i+1)^b) \tag{4.1.1}$$

$$h_i^f = F_f(x_i, x_{i-1}, h_(i-1)^f), \tag{4.1.2}$$

In BasicVSR [18], the bidirectional propagation network consists of flow estimation module, spatial warping and residual blocks. Our project proposes a novel architecture with a combination of RRDB and Residual blocks in series. Similarly, ESRGAN [8] propose some improvements over the generator section from the general super resolution section of SRGAN [15], namely removal of Batch Normalization layer and replacement of original basic block with Residual in Residual Dense Block to further improve the quality generated images. Our architecture combines these two elements from the aforementioned implementations into a single unit.

4.1.1 Residual Blocks

Residual blocks have become a fundamental building block in many deep neural network architectures due to their ability to improve gradient flow and facilitate the training of deep networks. In the context of image and video superresolution, residual blocks have

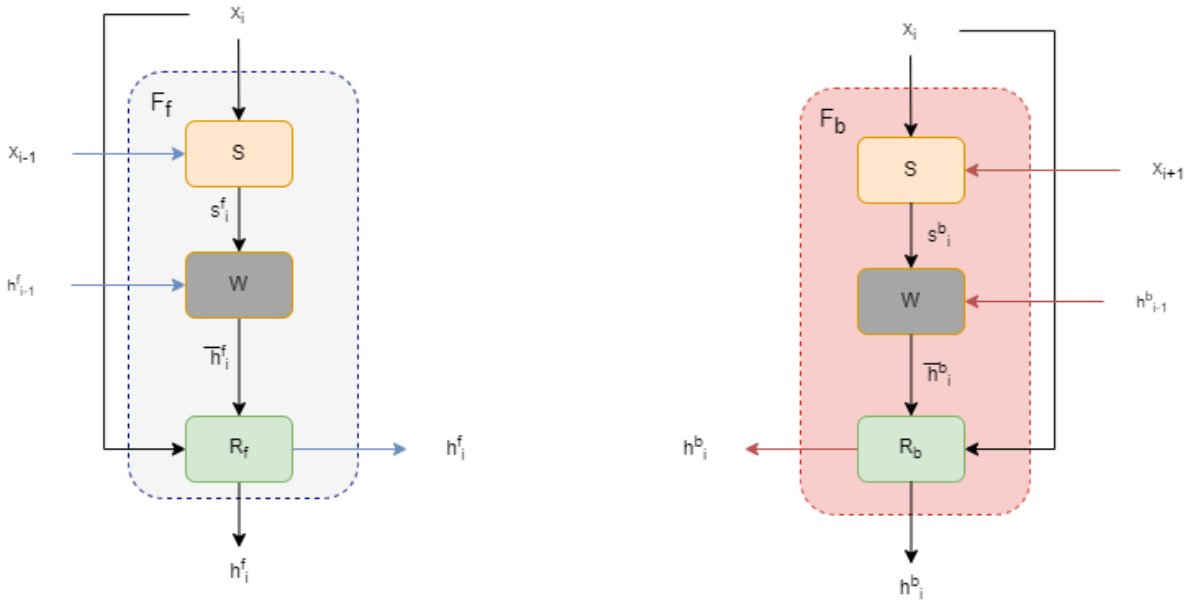

Figure 4.2: Forward and backward propagation branches in BasicVSR

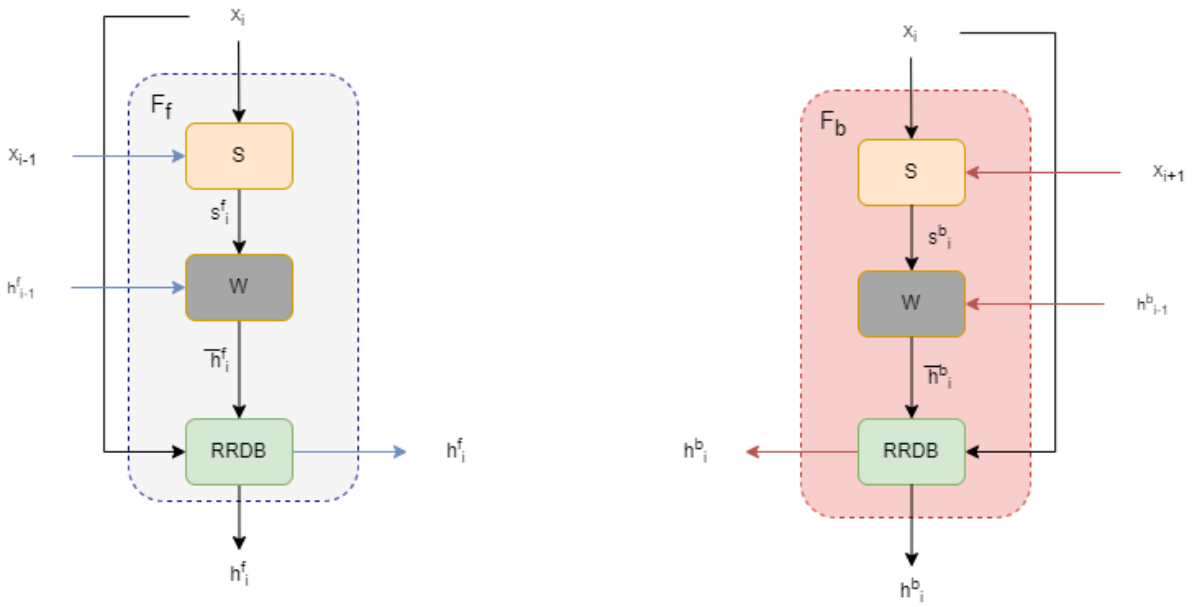

Figure 4.3: Forward and backward propagation branches in RD-BasicVSR

been used to effectively extract and propagate features through the network, enabling the generation of high-quality output frames. The propagation block consists of optical flow estimation denoted by 'S' in the block diagram, spatial warping 'W' and a residual block R_f, in BasicVSR, whereas the the RD-BasicVSR model consists of RRDB block in place of the residual block in BasicVSR.

A residual block is a building block that contains one or more convolutional layers and a

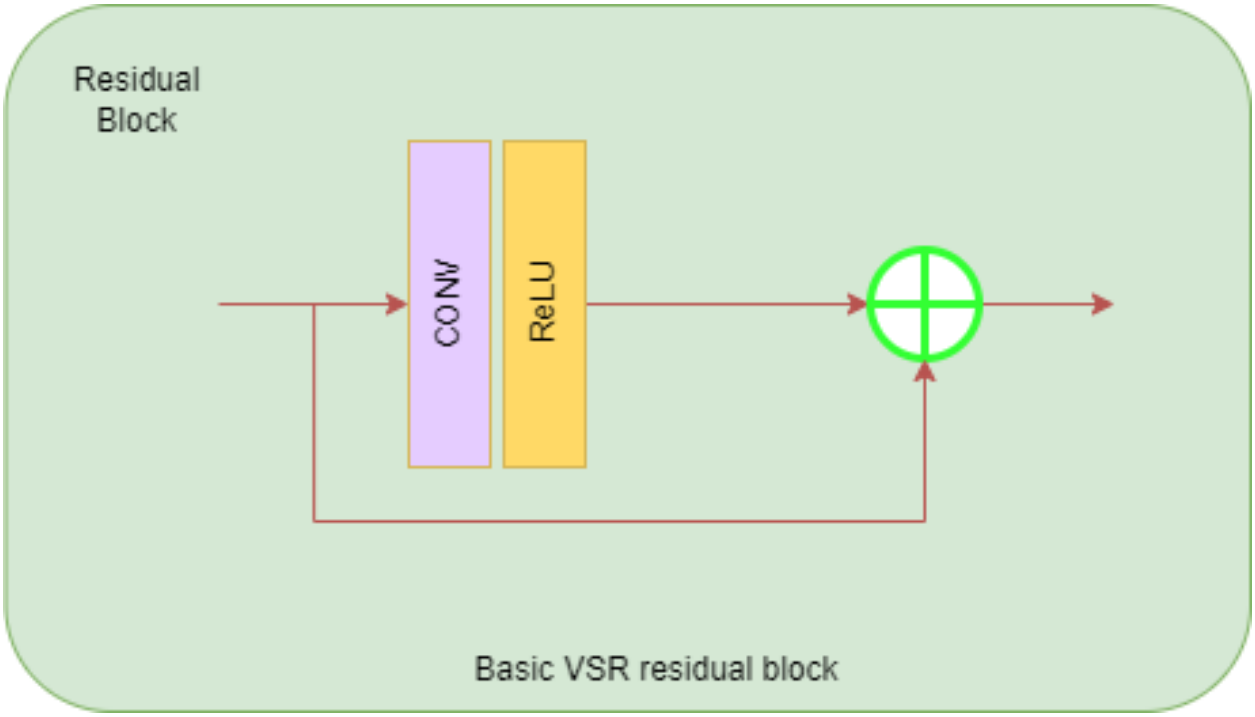

Figure 4.4: Residual block used in BasicVSR

residual connection. The residual connection allows the gradient to flow through the network more efficiently, improving the training process and avoiding the vanishing gradient problem. In video superresolution, residual blocks can be used to extract features from input frames and generate high-quality output frames by propagating features across multiple layers. [42].

Residual blocks have been used in various video superresolution methods, including but not limited to, Recursive Residual Network RRN, Residual Dense Network RDN, and Deep Recursive Residual Network DRRN. By utilizing residual blocks, video superresolution methods have been able to achieve state-of-the-art performance and generate high-quality output frames with improved visual quality and structure.

4.1.2 RRDB(Residual in residual dense blocks)

Residual in Residual Dense Block (RRDB) is an enhancement of Residual Dense Block. The basic idea behind the RRDB is to add an additional residual connection to the original RDN architecture. This residual connection allows the network to better capture long-range dependencies and improve the quality of the output.

The RRDB architecture is composed of several layers, each of which is made up of dense blocks. A dense block is a group of convolutional layers, which are connected in a densely

connected manner. Each dense block is followed by a residual connection, which ensures that the output of the block is added to the original input [42].

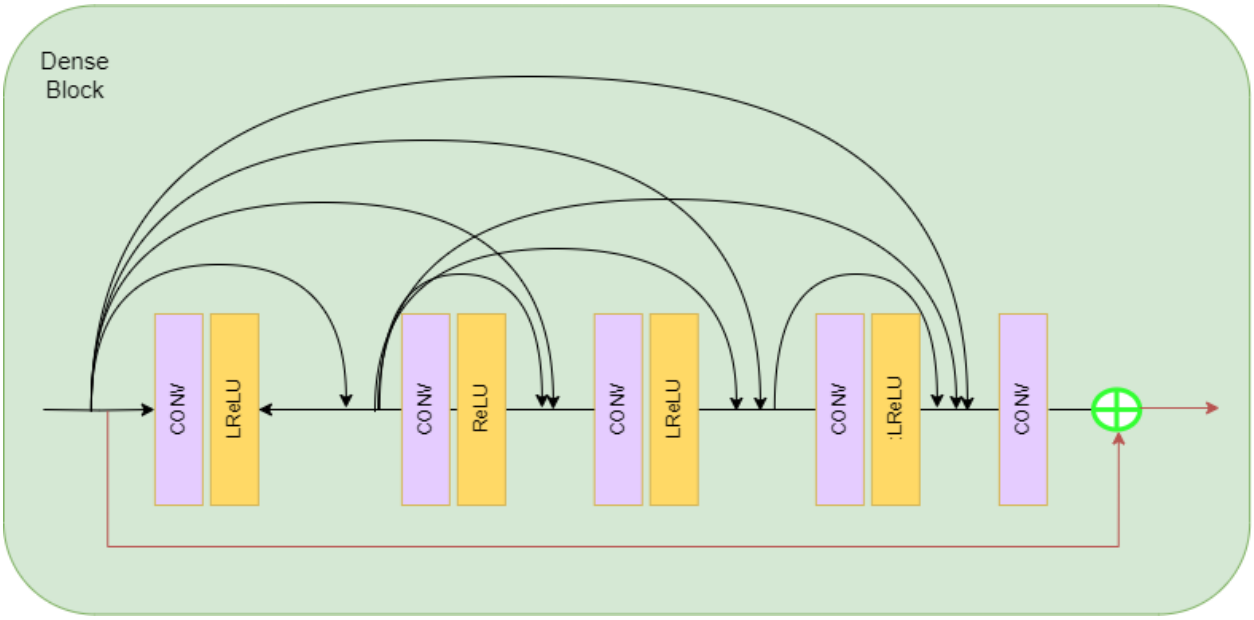

Figure 4.5: Illustration of a Dense block from ESRGAN

A single dense block has been shown in Figure 4.5. It consists of four blocks with each block consisting of a convolutional block and an LRelu block stacked together. The initial input is also concatenated to the computed output at the end.

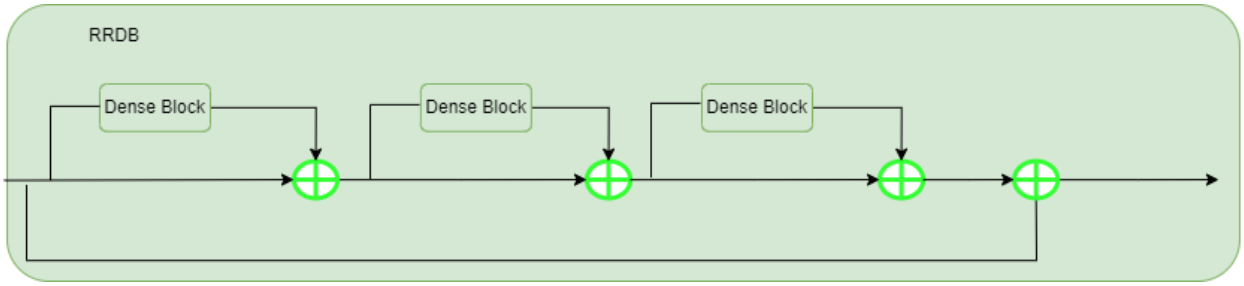

Figure 4.6: Single RRDB

An RRDB block, as explained in the ESRGan paper [41], consists of 3 dense blocks described above stacked together, with individual input concatenated to their outputs forming a residual structure. It is shown in figure 4.5. Finally, the RRDB module described in our architecture for propagation consists of a residual block and two RRDB blocks described above and as shown in figure 6. In essence, there are a total of 64 convolutional blocks in BasicVSR model while our proposed architecture consists of only 30 convolutional blocks.

In the RRDB architecture, the residual connection is added to the output of the dense block,

Figure 4.7: RRDB block used in our model

which is then passed through another set of convolutional layers. This process is repeated several times, resulting in a highly complex architecture that can capture subtle details and improve the quality of the output. It is particularly well-suited for video superresolution tasks because it can help to generate high-quality outputs by effectively capturing the temporal dependencies between frames.

4.2 Alignment

Alignment is another important factor addressed in BasicVSR and used in our research. Spatial alignment plays an important role in VSR as it is responsible for aligning highly related but misaligned images/features for subsequent aggregation. Alignment can be carried out in three different ways for VSR works: without alignment, image alignment and feature alignment. The paper presents suboptimal performance of using without alignment approach, supported by difference in PSNR values. Similarly, the image alignment approach, that performs alignment by computing optical flow and warping images prior to restoration, also performs quite poorly as compared to feature level alignment [18].

BasicVSR uses an optical flow approach for spatial alignment, which performs warping on the features instead of images level before eventually being fed to the residual blocks. Formally, it can be represented as:

$$s_i^{b,f} = S(x_i, x_{i\pm1}), \tag{4.2.1}$$

$$\overline{h}_i^{b,f} = W(h_{i\pm1}^{b,f}, s_i^{b,f}), \tag{4.2.2}$$

$$h_i^{b,f} = R_{b,f}(x_i, \overline{h}_i^{b,f}), \tag{4.2.3}$$

where, S and W denote the flow estimation and spatial warping modules respectively whereas $R_{b,f}$ denotes a stack of residual blocks.
Optical flow is the general method to capture this component in VSR. To calculate optical

flow, we use a neural network based model, SPyNet to calculate the motion vector of the pixels. Understanding the motion or flow of frames requires more than simple convolutional networks that process spatial information. Optical flow is one such method for processing inter-frame information. BasicVSR utilized deep learning techniques to estimate the optical flow [18]. The use of the spatial pyramid network [36] (SPyNet) solves the temporal learning problem in two ways: it uses a spatial pyramid and convolutional filters to estimate the temporal structure and uses traditional approaches to determine the long-range correlations in frames [36].

4.2.1 Optical Flow

Optical flow is a technique used in computer vision to estimate the motion of objects between frames of a video sequence. It refers to the apparent motion of pixels in an image or video sequence due to the motion of the objects in the scene.

The concept of optical flow is based on the assumption that pixels in an image or video sequence move in a smooth and continuous way, and that the brightness of a pixel remains constant over time. By analyzing the changes in brightness values of pixels between frames, optical flow algorithms can estimate the motion vectors of objects in the scene.

In video superresolution, optical flow is used to estimate the motion vectors between adjacent frames, which can then be used to align the low-resolution frames and generate a high-resolution output frame. It allows for the reconstruction of high-resolution images or videos from low-resolution ones by utilizing the information from neighboring frames. By estimating the motion between frames, it is possible to warp the low-resolution frames to a high-resolution grid that can then be used to generate a high-resolution image or video.

4.2.2 SpyNet

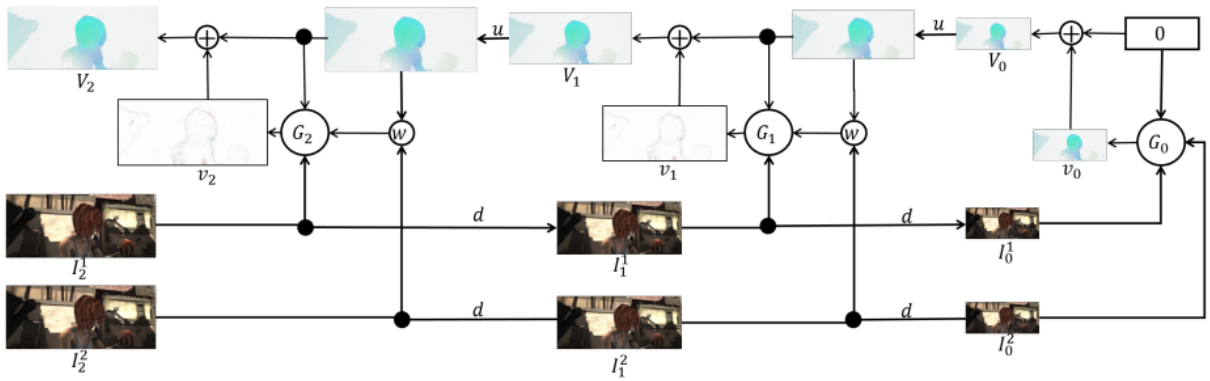

Figure 4.8: SpyNet module [36]

We use SpyNet block to estimate the optical flow in our proposed model [36]. SpyNet introduces machine learning to address the weaknesses of earlier flow estimation models which improves performance over the classical slower models and also reduces memory requirements to make the application more suited to embedded and mobile devices. Here, we use deep neural networks at each level of the spatial pyramid and train them to estimate a flow update at each level. This approach means that each network has less work to do than a fully generic flow method that has to estimate arbitrarily large motions. At each pyramid level we assume that the motion is small (on the order of a pixel) [36].

In figure 10, 'G' refers to the training neural network, which computes a residual flow that propagates to next lower levels of pyramids in turn, to finally obtain flow 'V' at the highest resolution.

4.3 Aggregation and Upsampling

Aggregation and Upsampling is another main component of the architecture. During the aggregation step in BasicVSR, the features extracted from multiple frames of the input video are concatenated along the channel dimension to generate a single set of features for each frame of the output high-resolution video. This concatenation operation allows the network to use information from multiple frames and combine the aligned features to generate better-quality high-resolution frames. Upsampling block utilizes the information from previous blocks to enhance the quality of individual frames to produce high resolution output frames. The intermediate information produced from the Optical Flow module is passed through Convolutional blocks and a Pixel Shuffle module to produce HR output. Similarly, the LR input frames are also scaled four times using Bilinear Upsampling to generate HR frames. The final frames are then obtained by combining the results of the two upsampling processes.

4.3.1 Pixel Shuffle

Pixel shuffle is a technique used in deep neural networks for image and video superresolution, which can increase the spatial resolution of an image by rearranging the pixels in a low-resolution feature map. Pixel shuffle is a type of upsampling method that involves three steps: subpixel convolution, reshaping, and shuffling.

In the first step, a subpixel convolutional layer is applied to the low-resolution feature map to increase the number of channels. The subpixel convolutional layer is designed to learn the upsampling filter that transforms a low-resolution feature map into a high-resolution feature map. The output of the subpixel convolutional layer is a high-resolution feature map with a

larger number of channels.

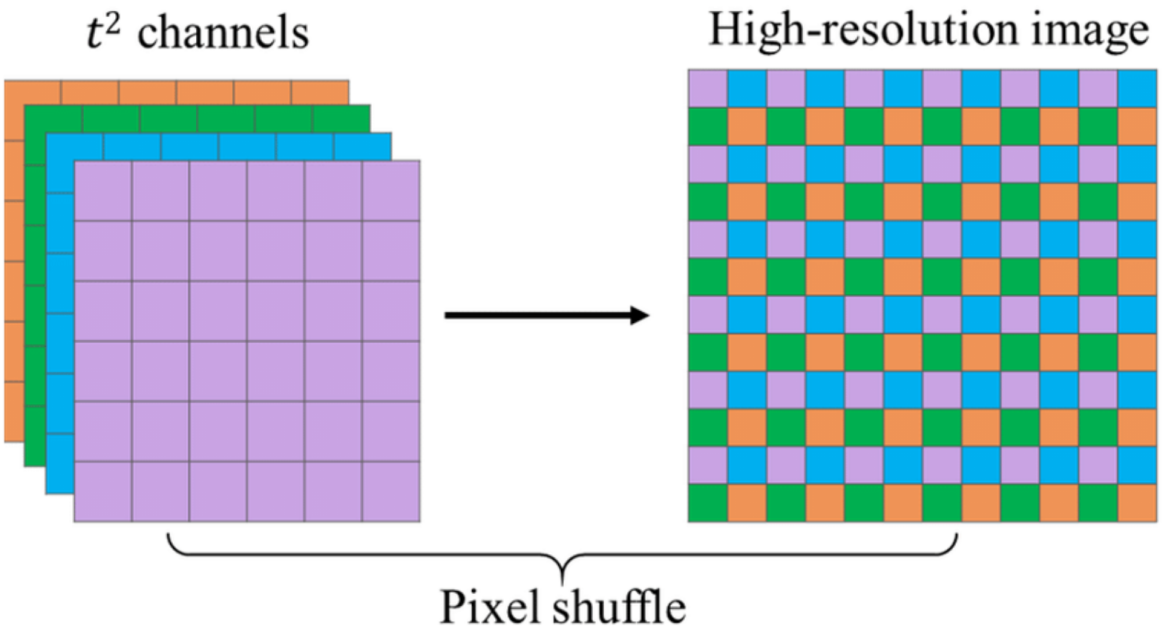

Figure 4.9: PixelShuffle final step[43]

In the second step, the high-resolution feature map is reshaped by rearranging the pixels into a block structure. The block size corresponds to the desired scale factor, which determines the level of upsampling. For example, a block size of 2x2 would correspond to a scale factor of 2x.

In the final step, the reshaped high-resolution feature map is shuffled by swapping the positions of the pixels within each block. This shuffling operation effectively spreads out the high-frequency information that was learned by the subpixel convolutional layer, resulting in a higher-resolution image.

4.4 Dataset and Settings

Two datasets have been used in the experiments in the model. REDS [44] and 160 CCTV footages extracted from MEVA dataset [45] are used for training, validation and testing purposes. Out of all the 270 folders available in the REDS dataset, folders 000, 011, 015 and 020 have been separated as testing datasets. And handpicked four videos from MEVA dataset have been seperated for testing purpose for finetuned model. The entirety of the remaining dataset has been used for training purposes. A subset of the training dataset, folders 000,001,006,017 has further been partitioned as a validation set. The videos we used for training consist of 24 frames per second and are five seconds in length for the REDS

dataset and 30 frames per second for the MEVA dataset. 30 frames per second is a standard recording frame rate for CCTV cameras.

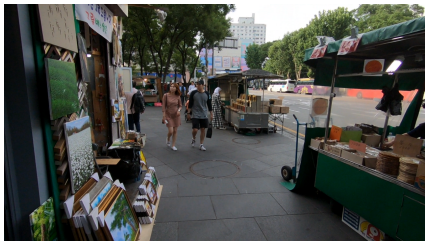

Figure 4.10: Image from REDS dataset

Figure 4.11: Image from REDS dataset

Figure 4.12: Image from MEVA dataset

Figure 4.13: Image from MEVA dataset

In our approach, we utilize pre-trained SPyNet model for flow estimation. The initial learning rates for flow estimator is set to $2.5 * 10^{-5}$, while the learning rate for all other modules is set to $2 * 10^{-4}$. We run a total of 350K iterations and freeze the weights of the flow estimator for the first 50,000 iterations. We use a batch size of 4 and the input LR frames are in a patch size of 64×64. When training, we use a sequence of 15 frames as inputs, and loss is computed for the 15 output images. The Charbonnier loss is used as loss function.

4.5 Metrics

We used following metrics to validate our designed architecture. The metrics are inspired from the BasicVSR paper [18] and are general methods to evaluate components involving images and videos.

4.5.1 PSNR (Peak Signal-to-Noise Ratio)

PSNR is a commonly used metric for measuring the quality of reconstructed or compressed images and videos. It measures the ratio of the peak signal power to the power of the noise in the image or video.

PSNR is calculated by comparing the original, high-quality image or video to the compressed

or reconstructed version of the image or video. The difference between the two is measured in terms of the MSE, which is the average of the squared differences between the pixel values of the two images or videos.

$$PSNR = 10. \log_{10}(\frac{MAX_I^2}{MSE}) \tag{4.5.1}$$

which can be simplified to

$$PSNR = 20. \log_{10}(MAX_I) - 10. \log_{10}(MSE) \tag{4.5.2}$$

where,
MAX_I=Maximum Possible Value of each pixel
MSE=Mean Squared Error

The PSNR value is expressed in dB which is a logarithmic scale that ranges from 0 to infinity. A higher PSNR value indicates a higher quality reconstructed or compressed image or video, and vice versa. A PSNR value of 30 dB or higher is generally considered to be of high quality, while a value below 20 dB is considered to be of low quality.

While PSNR is a widely used metric for image and video quality assessment, it has some limitations. For example, it doesn't always reflect how humans perceive the quality of an image or video. Additionally, it assumes that all errors are equally important, which may not be the case in certain applications. Therefore, other metrics such as Structural Similarity Index (SSIM) and Perceptual Quality Assessment (PQA) are often used in conjunction with PSNR to provide a more comprehensive assessment of image and video quality.

4.5.2 SSIM (Structural Similarity Index)

SSIM is a widely used image quality metric that measures the similarity between two images, particularly in terms of structural information. It is often used in conjunction with other metrics like PSNR to provide a more comprehensive assessment of image quality.

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)} \tag{4.5.3}$$

where,
$\mu_x \, is \, the \, average \, of \, x$
$\mu_y \, is \, the \, average \, of \, y$
$\sigma_x^2 \, is \, the \, variance \, of \, x$
$\sigma_y^2 \, is \, the \, variance \, of \, y$

σ_{xy} *is the covariance of* x *and* y

$c_1 = (k_1 L)^2, c_2 = (k_2 L)^2$ are two variables to stabilize the division with weak denominator
L the dynamic range of the pixel-values$(2^{bits\,per\,pixel} - 1)$
$k_1 = 0.01$ and k_2=0.03 by default
The SSIM metric is based on the idea that the perceived quality of an image depends on its structural information, luminance, and contrast. SSIM compares two images by calculating three components: luminance (brightness), contrast (image depth), and structure (spatial arrangement of pixels). The luminance component measures the average brightness of the images, while the contrast component measures the difference in the range of pixel values between the images. The structure component measures the similarity of the patterns in the images. The SSIM score is obtained by calculating the product of these three components, with higher scores indicating greater similarity between the images. The SSIM score ranges between 0 and 1, with a value of 1 indicating perfect similarity.

SSIM has several advantages over other image quality metrics like PSNR. It correlates better with human perception of image quality, is more robust to compression artifacts, and can better distinguish between images that are perceptually similar but have different pixel values.

4.5.3 Charbonnier Loss

Charbonnier Loss is a loss function used in image processing tasks, particularly in image restoration and superresolution, to measure the difference between the output image and the target image.

$$\mathcal{L} = \frac{1}{N} \sum_{i=0}^{N} \rho(y_i - z_i) \tag{4.5.4}$$

where $\rho(x) = \sqrt{x^2 + \epsilon^2}, \epsilon = 1 * 10^{-8}$,
z_i denotes the ground-truth HR frame, and
N denotes to the number of pixels.
The Charbonnier loss function is defined as the square root of the sum of the squared pixel differences between the target image and the output image, raised to a power of p, where p is typically set to 0.5 or 1. The function is more robust than the traditional L2 loss function because it is less sensitive to outliers, which can occur in real-world image processing tasks.

Compared to other loss functions like MSE, Charbonnier loss can lead to better image quality by preserving edges and texture details. This is because the Charbonnier loss function places less emphasis on large pixel differences and more emphasis on small pixel differences.

5. Results and Discussion

To train our model, we followed a two-phase approach: pretraining and finetuning. Initially, we trained the model on the REDS dataset for 300k iterations using the settings described earlier. We then finetuned the model on the CCTV dataset for 50k iterations keeping the initial learning rate $2 * 10^{-5}$.
We recorded the log of our training and validation sets in Wandb.ai.

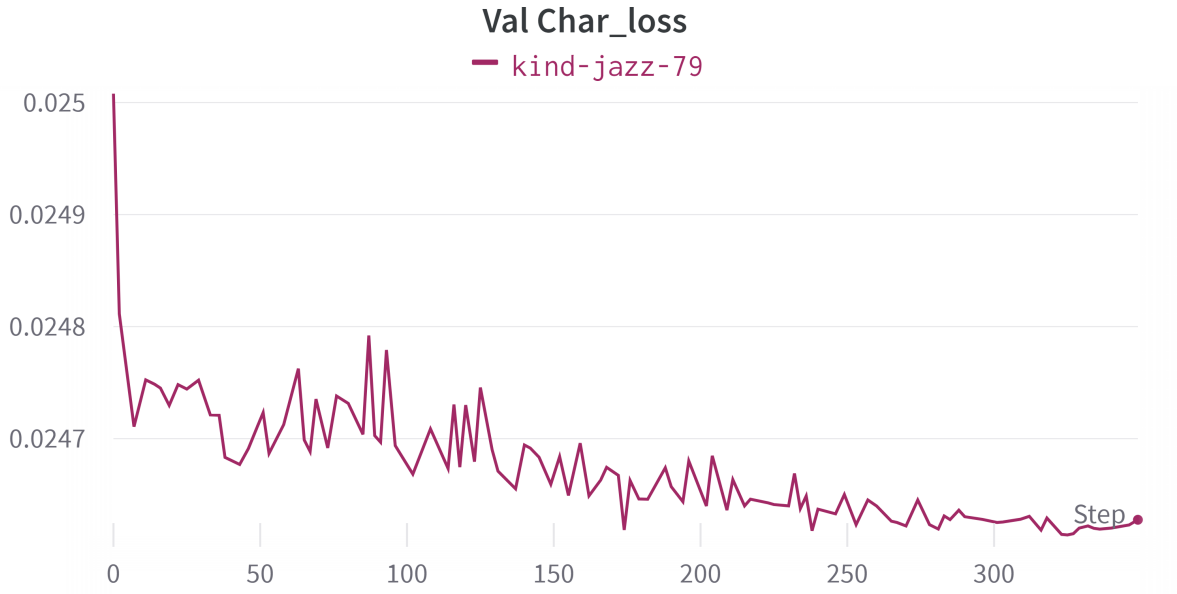

Figure 5.1: Validation Charbonnier Loss

Figure 5.1 illustrates training progress over the last 350 iterations after loading a saved checkpoint. The horizontal axis represents the iteration number, while the vertical axis shows the Charbonnier loss. The graph shows the validation curve, indicating an overall downward trend with occasional fluctuations which suggests that the model is making steady progress towards convergence.

Figure 5.2 illustrates the SSIM index of the model over the last 350 iterations after loading a saved checkpoint. The horizontal axis represents the iteration number, while the vertical axis shows the SSIM index. The graph shows an overall upward trend with occasional fluctuations which suggests that model's performance on the validation dataset is gradually improving

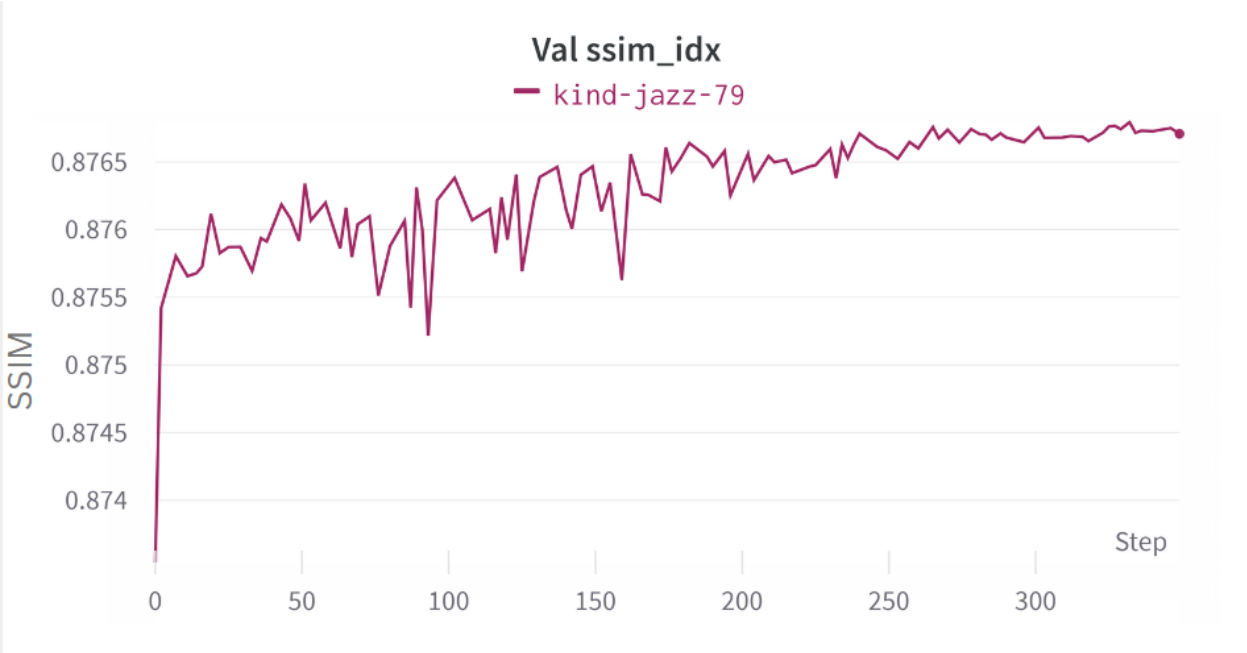

Figure 5.2: Validation SSIM Index Curve

over time.

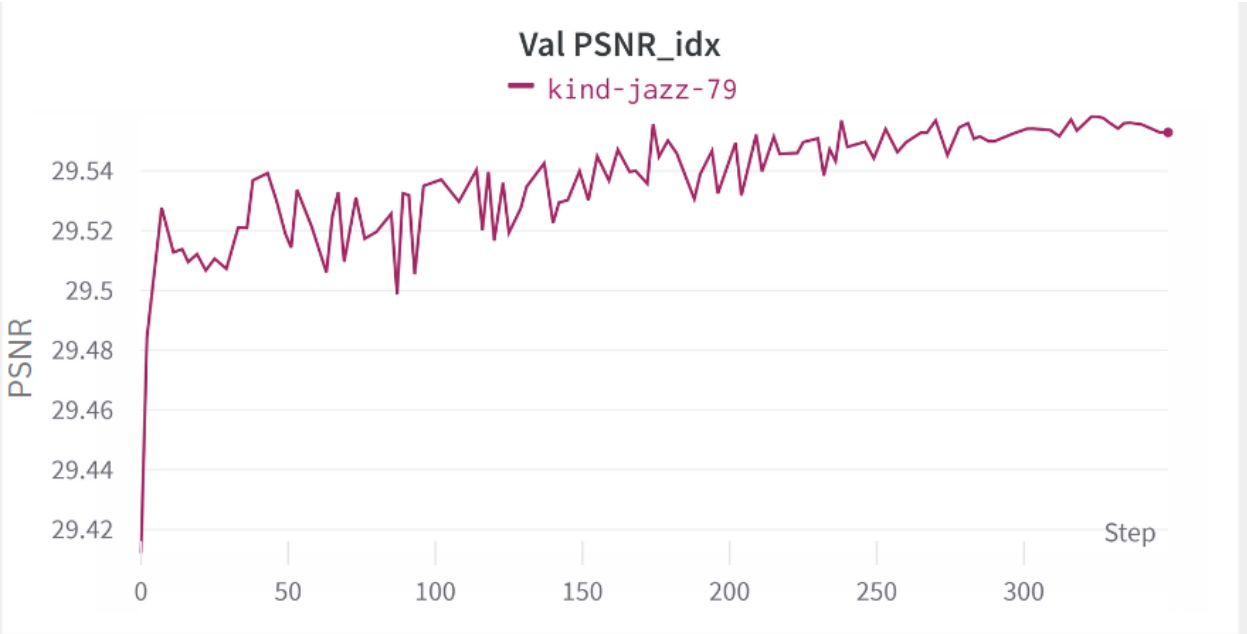

Figure 5.3: Validation PSNR Index Curve

Figure 5.3 illustrates the PSNR index of the model over 350 iterations from a saved checkpoint. The horizontal & vertical axes represent number of iterations & PSNR value respectively. The graph shows an overall upward trend with occasional fluctuations which suggests that model's performance on the validation dataset is gradually improving over time.

5.1 Quantitative Comparison with the State of the Art

Table 5.1: Quantitative Comparison in terms of PSNR and SSIM on REDS4 dataset

Model	**PSNR**	**SSIM**
Bicubic	26.14	0.7292
TOFlow [30]	27.98	0.7990
DUF [26]	28.63	0.8251
RBPN [46]	30.09	0.8590
PFNL [47]	29.63	0.8502
EDVR-M [17]	30.53	0.8699
BasicVSR [18]	**29.287**	**0.8665**
RD-BasicVSR(**ours**)	**29.52**	**0.8758**

Observing the table 5.1, it is evidently clear that our proposed model outperforms many of the state-of-the-art models, for the REDS4 dataset. The PSNR and SSIM values of our model at 29.52 and 0.8758 show results closer to the state-of-the-art BasicVSR model and thus is an effective tool in VSR literature. The output on validation frames of REDS dataset an out-of-the-wild inference on CCTV footage. The output generated is produced by our baseline model and not by the fine tuned model.

Table 5.2: Quantitative result obtained on CCTV test dataset

Model	**PSNR**	**SSIM**
RD-BasicVSR (ours)	33.88	0.9505

Even though the training input images are of 64x64, for inference and deployment we can input images/video of any size. There will obviously be similar increase in requirements of the inference machine. When run on NVIDIA RTX 3060 the inference time was 20.8 seconds for 150 frames, which accounts to about 140ms per frame for frames of size 64x64 upsampled by four times. However this value is influenced by the size of the low resolution frames and takes longer for upsampling 320x180 to 1280x720.

Figure 5.4, 5.5, 5.6, 5.7 show the comparison of a low-resolution image (left) with the same image after being upscaled by our model (right). The image on the right demonstrates the effectiveness of our model in increasing the resolution and restoring details that were lost in

the original low-resolution image. The upscaled image shows sharper edges and more visible details, resulting in a visually more appealing and higher-quality image.

Figure 5.4: The low resolution image from MEVA dataset(left) upscaled by our model(right)

Figure 5.5: The low resolution image from REDS dataset(left) upscaled by our model(right)

Figure 5.6: The low resolution test image(left) upscaled by our model(right)

Figure 5.7: The low resolution test image(left) upscaled by our model(right)

5.2 Qualitative Comparison with BasicVSR

Qualitative comparison of BasicVSR with RD-BasicVSR(ours) yields a more clearer picture of the improvement in the perceived quality of video frames. In the real world scenario, perception plays a stronger role in determining the quality of image processing algorithms.

Figure 5.8: The back of a vehicle

Figure 5.8 shows the quality comparison of the back of an automobile. It is evident that the upsampled images are significantly better at distinguishing separate regions. The improvement is even more evident when a localized portion of the frame is taken into account. For example in Figure 5.9, the letter at the back of the van is clearly distinguishable in both

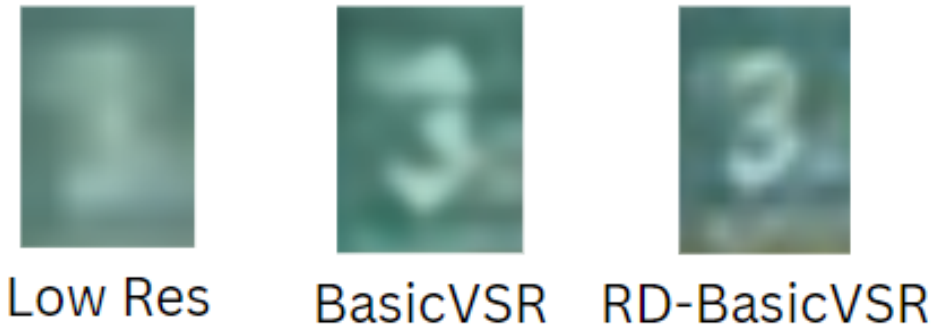

Figure 5.9: Focus on the number

upsampled images however the texture in RD-BasicVSR is smoother and free from artifacts than in BasicVSR.

Figure 5.10: Monitoring Traffic

Figure 5.10 is from a CCTV footage of a heavy traffic area, at first glance both BasicVSR and RD-BasicVSR seem to have similar results rightfully as the quantitative metrics for both are pretty close to one another. On closer inspection(Figure 5.11) it is clear that there is

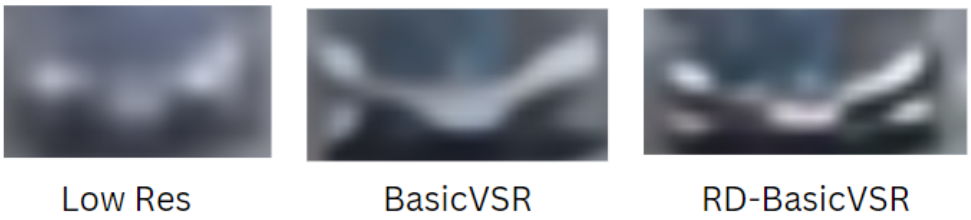

Figure 5.11: Headlight comparison

better contrast on the upsampled frame from RD-BasicVSR.

RD-BasicVSR also shows improvement over BasicVSR in terms of curves and edges. Figure 5.12 is a body cam image from a street in Thamel. Both models have significantly improved the quality of output frames, however RD-BasicVSR's output seems much less grainer than

Figure 5.12: Body Cam

BasicVSR's. On closer inspection of the tapestry, it can be seen on Figure 5.13 that the

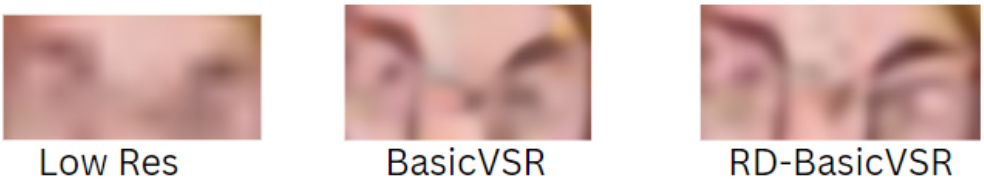

Figure 5.13: Contrasting curves

contrast at curves are much better pronounced in our model. The model is capable of retaining vivid details even on videos that it has not directly been trained on.

Figure 5.14: Tiger Painting from Reds Dataset

Similarly on videos with high contrast within itself RD-BasicVSR clearly provides a better perceived quality. The contours on the tiger painting(Figure 5.14) seen in one of the videos of REDS dataset illustrates it clearly.

6. Epilogue

6.1 Conclusion

Video quality upsampling and super resolution falls at the relatively less explored spectrum of computer vision. The research work focuses on the impact of less explored spatio-temporal domain of video superresolution and the techniques used to process such information. Using optical flow for recording the temporal information among frames, along with modern techniques such as pixel shuffle, aggregation, feature alignment. The analysis and aggregation of these tools using novel deep learning based architectures, led to the development of our model, RD-BasicVSR, which performs remarkably well on the CCTV data. The developed model not only shows close to state-of-the-art results but also promises an effective tool to improve quality of low-scale CCTV footages.

6.2 Limitations

Some limitations and challenges that were evident during our architecture and model development and application have been enumerated below:

1. Inference time: Despite the effort to develop a 'light-weight' model, our model still has high inference time (run time). This is especially important considering the fact that our model is supposed to upsample CCTV footages, which require near perfect runtime with minimal delay. However, for tasks other than CCTV upsampling, the inference time does not pose a great threat to the model.

2. Training duration: Sufficient training time is essential for success of any deep learning based project. Due to lack of abundant training resources, our model could not be trained to its full potential, especially considering the training time of the used baseline model, BasicVSR. A greater training period could lead to better model performance.

3. Lack of CCTV focused resources: VSR models discussed in this research are designed for general upsampling uses. This brings forward problems such as lack of well-curated dataset, proper baseline models designed for quick run-time purposes etc. Lack of literature in CCTV focused research is clear due to these reasons.

4. Difficulty in detecting error: The evaluation metrics for computer vision based image and video projects involve noise and similarity based metrics. These metrics are associated with clarity of generated outputs. For sensitive applications and those requiring information extraction from image frames and videos while using our model in production, it is difficult to observe the performance of model.

6.3 Future Works

Video super-resolution is an active research area, and there are several directions for future research and optimization for our project. Possible future works may include the following areas.

1. Real-time super-resolution: We could investigate methods for achieving real-time super-resolution, which could be useful for applications such as video surveillance or live streaming. This would require research for less complex network architecture to develop a light-weight model with lesser inference time.

2. Fine-tuning the existing model: The existing model could be fine-tuned by training it on a larger dataset or incorporating more training data to improve its performance.

3. Exploring different loss functions: Different types of loss functions such as perceptual loss, adversarial loss, or content loss may be investigated to enhance the quality of the output images.

4. Tailor video super-resolution models: The project could be used as a baseline to build tailored video super-resolution models that can generate high-resolution video sequences from low-resolution inputs from different fields.

5. Evaluating the performance of the model on different datasets: The performance of the model on different datasets, especially those that have different characteristics and features, could be evaluated to test its generalization ability.

References

[1] A. Shocher, N. Cohen, and M. Irani, *"zero-shot" super-resolution using deep internal learning*, 2017. DOI: 10.48550/ARXIV.1712.06087. [Online]. Available: https://arxiv.org/abs/1712.06087.

[2] S. Kim, D. Jun, B.-G. Kim, H. Lee, and E. Rhee, "Single image super-resolution method using cnn-based lightweight neural networks," *Applied Sciences*, vol. 11, p. 1092, Jan. 2021. DOI: 10.3390/app11031092.

[3] J. Yamanaka, S. Kuwashima, and T. Kurita, "Fast and accurate image super resolution by deep cnn with skip connection and network in network," 2017. DOI: 10.48550/ARXIV.1707.05425. [Online]. Available: https://arxiv.org/abs/1707.05425.

[4] M.-I. Georgescu, R. T. Ionescu, A.-I. Miron, *et al.*, *Multimodal multi-head convolutional attention with various kernel sizes for medical image super-resolution*, 2022. DOI: 10.48550/ARXIV.2204.04218. [Online]. Available: https://arxiv.org/abs/2204.04218.

[5] C. Ledig, L. Theis, F. Huszar, *et al.*, *Photo-realistic single image super-resolution using a generative adversarial network*, 2016. DOI: 10.48550/ARXIV.1609.04802. [Online]. Available: https://arxiv.org/abs/1609.04802.

[6] X. Hu, X. Liu, Z. Wang, X. Li, W. Peng, and G. Cheng, "Rtsrgan: Real-time super-resolution generative adversarial networks," in *2019 Seventh International Conference on Advanced Cloud and Big Data (CBD)*, 2019, pp. 321–326. DOI: 10.1109/CBD.2019.00064.

[7] H. Wang, P. Chen, B. Zhuang, and C. Shen, *Fully quantized image super-resolution networks*, 2020. DOI: 10.48550/ARXIV.2011.14265. [Online]. Available: https://arxiv.org/abs/2011.14265.

[8] X. Wang, K. Yu, S. Wu, *et al.*, *Esrgan: Enhanced super-resolution generative adversarial networks*, 2018. DOI: 10.48550/ARXIV.1809.00219. [Online]. Available: https://arxiv.org/abs/1809.00219.

[9] V. Bhatia and Y. Kumar, *Attaining real-time super-resolution for microscopic images using gan*, 2020. DOI: 10.48550/ARXIV.2010.04634. [Online]. Available: https://arxiv.org/abs/2010.04634.

[10] D. Hazra and Y. Byun, "Upsampling real-time, low-resolution cctv videos using generative adversarial networks," *Electronics*, vol. 9, p. 1312, Aug. 2020. DOI: 10.3390/electronics9081312.

[11] C. Saharia, J. Ho, W. Chan, T. Salimans, D. J. Fleet, and M. Norouzi, *Image super-resolution via iterative refinement*, 2021. DOI: 10.48550/ARXIV.2104.07636. [Online]. Available: https://arxiv.org/abs/2104.07636.

[12] J. Ho, T. Salimans, A. Gritsenko, W. Chan, M. Norouzi, and D. J. Fleet, *Video diffusion models*, 2022. DOI: 10.48550/ARXIV.2204.03458. [Online]. Available: https://arxiv.org/abs/2204.03458.

[13] AMD, *Fidelityfx super resolution 2.0.1 (fsr 2.0)*.

[14] C. Dong, C. C. Loy, K. He, and X. Tang, *Image super-resolution using deep convolutional networks*, 2014. DOI: 10.48550/ARXIV.1501.00092. [Online]. Available: https://arxiv.org/abs/1501.00092.

[15] C. Ledig, L. Theis, F. Huszar, *et al.*, *Photo-realistic single image super-resolution using a generative adversarial network*, 2016. DOI: 10.48550/ARXIV.1609.04802. [Online]. Available: https://arxiv.org/abs/1609.04802v5.

[16] M. Chu, Y. Xie, J. Mayer, L. Leal-Taxie, and N. Thuerey, *Learning temporal coherence via self-supervision for gan-based video generation*, 2018. DOI: 10.48550/ARXIV.1811.09393. [Online]. Available: https://arxiv.org/abs/1811.09393.

[17] X. Wang, K. C. K. Chan, K. Yu, C. Dong, and C. C. Loy, *Edvr: Video restoration with enhanced deformable convolutional networks*, 2019. DOI: 10.48550/ARXIV.1905.02716. [Online]. Available: https://arxiv.org/abs/1905.02716.

[18] X. Wang, K. C. K. Chan, K. Yu, C. Dong, and C. C. Loy, *Basicvsr: The search for essential components in video super-resolution and beyond*, 2020. DOI: 10.48550/ARXIV.2012.02181. [Online]. Available: https://arxiv.org/abs/2012.02181v2.

[19] H. Liu, Z. Ruan, P. Zhao, *et al.*, *Video super-resolution based on deep learning: A comprehensive survey*, 2022. DOI: 10.48550/ARXIV.2007.12928v3. [Online]. Available: https://arxiv.org/abs/2007.12928v3.

[20] Y. Huang, W. Wang, and L. Wang, *Bidirectional recurrent convolutional networks for multi-frame super-resolution*, C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, Eds., 2015. [Online]. Available: https://proceedings.neurips.cc/paper/2015/file/c45147dee729311ef5b5c3003946c48f-Paper.pdf.

[21] T. Isobe, F. Zhu, X. Jia, and S. Wang, *Revisiting temporal modeling for video super-resolution*, 2020. DOI: 10.48550/ARXIV.2008.05765. [Online]. Available: https://arxiv.org/abs/2008.05765.

[22] J. Caballero, C. Ledig, A. Aitken, *et al.*, *Real-time video super-resolution with spatio-temporal networks and motion compensation*, 2016. DOI: 10.48550/ARXIV.1611.05250. [Online]. Available: https://arxiv.org/abs/1611.05250.

[23] J. Dai, H. Qi, Y. Xiong, *et al.*, *Deformable convolutional networks*, 2017. DOI: 10.48550/ARXIV.1703.06211. [Online]. Available: https://arxiv.org/abs/1703.06211.

[24] X. Zhu, H. Hu, S. Lin, and J. Dai, *Deformable convnets v2: More deformable, better results*, 2018. DOI: 10.48550/ARXIV.1811.11168. [Online]. Available: https://arxiv.org/abs/1811.11168.

[25] Y. Tian, Y. Zhang, Y. Fu, and C. Xu, *Tdan: Temporally deformable alignment network for video super-resolution*, 2018. DOI: 10.48550/ARXIV.1812.02898. [Online]. Available: https://arxiv.org/abs/1812.02898.

[26] *Deep video super-resolution network using dynamic upsampling filters without explicit motion compensation*, Dec. 2018. DOI: 10.1109/CVPR.2018.00340.

[27] T. Isobe, X. Jia, S. Gu, S. Li, S. Wang, and Q. Tian, *Video super-resolution with recurrent structure-detail network*, 2020. DOI: 10.48550/ARXIV.2008.00455. [Online]. Available: https://arxiv.org/abs/2008.00455.

[28] T. Isobe, F. Zhu, X. Jia, and S. Wang, "Revisiting temporal modeling for video super-resolution," *ArXiv*, vol. abs/2008.05765, 2020.

[29] B. K. Horn and B. G. Schunck, "Determining optical flow," *Artificial Intelligence*, vol. 17, no. 1, pp. 185–203, 1981, ISSN: 0004-3702. DOI: https://doi.org/10.1016/0004-3702(81)90024-2. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0004370281900242.

[30] T. Xue, B. Chen, J. Wu, D. Wei, and W. T. Freeman, "Video enhancement with task-oriented flow," *International Journal of Computer Vision*, vol. 127, no. 8, pp. 1106–1125, Feb. 2019. DOI: 10.1007/s11263-018-01144-2. [Online]. Available: https://doi.org/10.1007%2Fs11263-018-01144-2.

[31] E. Memin and P. Perez, *Dense estimation and object-based segmentation of the optical flow with robust techniques*, 1998. DOI: 10.1109/83.668027.

[32] J. Xu, R. Ranftl, and V. Koltun, *Accurate optical flow via direct cost volume processing*, 2017. DOI: 10.48550/ARXIV.1704.07325. [Online]. Available: https://arxiv.org/abs/1704.07325.

[33] J. Revaud, P. Weinzaepfel, Z. Harchaoui, and C. Schmid, *Epicflow: Edge-preserving interpolation of correspondences for optical flow*, 2015. DOI: 10.48550/ARXIV.1501.02565. [Online]. Available: https://arxiv.org/abs/1501.02565.

[34] P. Fischer, A. Dosovitskiy, E. Ilg, *et al.*, *Flownet: Learning optical flow with convolutional networks*, 2015. DOI: 10.48550/ARXIV.1504.06852. [Online]. Available: https://arxiv.org/abs/1504.06852.

[35] J. J. Yu, A. W. Harley, and K. G. Derpanis, *Back to basics: Unsupervised learning of optical flow via brightness constancy and motion smoothness*, 2016. DOI: 10.48550/ARXIV.1608.05842. [Online]. Available: https://arxiv.org/abs/1608.05842.

[36] A. Ranjan and M. J. Black, *Optical flow estimation using a spatial pyramid network*, 2016. DOI: 10.48550/ARXIV.1611.00850. [Online]. Available: https://arxiv.org/abs/1611.00850.

[37] P. Burt and E. Adelson, "The laplacian pyramid as a compact image code," *IEEE Transactions on Communications*, vol. 31, no. 4, pp. 532–540, 1983. DOI: 10.1109/TCOM.1983.1095851.

[38] E. Riseman and A. Hanson, "Hierarchical motion detection," *International Journal of Computer Vision*, vol. 2, pp. 199–207, 1989. DOI: 10.1007/BF00158164.

[39] K. Zhang, J. Liang, L. Van Gool, and R. Timofte, *Designing a practical degradation model for deep blind image super-resolution*, 2021. DOI: 10.48550/ARXIV.2103.14006. [Online]. Available: https://arxiv.org/abs/2103.14006.

[40] X. Wang, L. Xie, C. Dong, and Y. Shan, *Real-esrgan: Training real-world blind super-resolution with pure synthetic data*, 2021. DOI: 10.48550/ARXIV.2107.10833. [Online]. Available: https://arxiv.org/abs/2107.10833.

[41] Z. Wei, Y. Huang, Y. Chen, C. Zheng, and J. Gao, *A-esrgan: Training real-world blind super-resolution with attention u-net discriminators*, 2021. DOI: 10.48550/ARXIV.2112.10046. [Online]. Available: https://arxiv.org/abs/2112.10046.

[42] B. Lim, S. Son, H. Kim, S. Nah, and K. M. Lee, *Enhanced deep residual networks for single image super-resolution*, 2017. DOI: 10.48550/ARXIV.1707.02921. [Online]. Available: https://arxiv.org/abs/1707.02921.

[43] W. Shi, J. Caballero, F. Huszár, *et al.*, "Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network," *CoRR*, vol. abs/1609.05158, 2016. arXiv: 1609.05158. [Online]. Available: http://arxiv.org/abs/1609.05158.

[44] S. Nah, S. Baik, S. Hong, *et al.*, "Ntire 2019 challenge on video deblurring and super-resolution: Dataset and study," in *CVPR Workshops*, Jun. 2019.

[45] K. Corona, K. Osterdahl, R. Collins, and A. Hoogs, "Meva: A large-scale multiview, multimodal video dataset for activity detection," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, Jan. 2021, pp. 1060–1068.

[46] M. Haris, G. Shakhnarovich, and N. Ukita, *Recurrent back-projection network for video super-resolution*, 2019. DOI: 10.48550/ARXIV.1903.10128. [Online]. Available: https://arxiv.org/abs/1903.10128.

[47] P. Yi, Z. Wang, K. Jiang, J. Jiang, and J. Ma, "Progressive fusion video super-resolution network via exploiting non-local spatio-temporal correlations," 2019, pp. 3106–3115. DOI: 10.1109/ICCV.2019.00320.