Video Super-Resolution

Abhranil Chakrabarti, Jan Reig Torra

[a]*Massachusetts Institute of Technology*

Abstract

This project proposes a new approach to enhance video super-resolution by integrating image super-resolution techniques with a temporal dimension. The first proposed method introduces a modified version of an SRCNN, which combines multiple neighboring motion-compensated frames using a new architecture. The second approach uses a signal-process technique to combine the frames, super-resolved by a pre-trained ESRGAN. Moreover, the study proposes and experiments with different loss functions, including Feature Space Loss and Motion Aware Loss, and compares them to the traditional MSE. To train the networks, pre-trained SRCNN models are used to initialize the weights, and the proposed approach is evaluated on different scenes from the CDVL dataset. The evaluation is performed using objective metrics such as PSNR and SSIM. The proposed approach offers a promising solution for real-world video super-resolution applications.

Keywords: Video Super-Resolution, Computer Vision, SRCNN, Motion Compensation

1. Introduction

Video super-resolution is a field of research that focuses on enhancing the resolution and quality of low-resolution videos. With video being a ubiquitous multimedia format in our daily lives, the need for super-resolution of low-resolution videos has grown significantly. While image super-resolution methods typically handle individual images independently, video super-resolution algorithms specifically work with multiple sequential frames to leverage inter-frame relationships for enhancing the target frame.

Video super-resolution finds diverse applications across various industries. In media and entertainment, video super-resolution improves the quality of streaming videos and facilitates the remastering of older content. Post-production workflows benefit from video super-resolution by seamlessly integrating low-resolution clips with higher-resolution footage and enabling visual effects. In medical imaging, it enhances the resolution of medical videos for more accurate diagnosis while remote sensing and aerial imagery applications benefit from video super-resolution by extracting finer details for environmental monitoring and disaster management.

2. Motivation and Related Work

Utilizing temporal information in video processing can enhance the quality of upscaling. While single image super-resolution methods can generate high-resolution frames independently, they are less effective and can introduce temporal instability. This is because modeling and estimating motion between frames can provide additional information due to subpatch motion, which can be captured through multiple frames for enhanced quality.

Recently, CNN-based image super-resolution algorithms have achieved promising results due to their ability to efficiently learn from large training databases and perform purely feed-forward processing. The current approaches can be classified into two broad categories: methods with alignment and methods without alignment (4).

Methods that incorporate alignment techniques aim to explicitly align neighboring frames with the target frame by leveraging motion information. This alignment is achieved through motion estimation and motion compensation (MEMC) or deformable convolution, which are widely used techniques for aligning frames in video superresolution. These methods utilize extracted motion information to ensure proper alignment before the subsequent reconstruction process. The alternative methods which do not align neighboring frames mainly rely on spatial or spatio-temporal information for feature extraction which can be achieved through 2D or 3D convolution methods, recurrent CNNs or other non-network based methods.

Some of the key challenges in the current methods is the replication of complex textures present in the videos. High training times and computational requirements for training large networks is another big challenge. Besides these, the need for better metrics (than mean-squared error) for capturing motion between frames is also something which needs to addressed.

3. Methodology and Approach

In this project, we introduce two approaches - a multi-frame CNN and an image fusion technique based on four key ideas:

- **Multiple Neighbouring Frames**: Using multiple neighbouring frames instead of single frame super-resolution. For this, we use the bicubic-upscaled motion-compensated neighbouring frames

- **Modified Objective Function**: Instead of relying on pixel-wise mean squared error, we explore two alternate loss functions (loss in the feature space (2) and motion-aware loss (5)) which we believe are more appropriate in this context
- **Warm Starts**: For the neural network models, to address the large data and computational requirements, we share some architecture with single-image super-resolution models and use those weights for warm starts during training (1)
- **Information Recombination**: Finally, to combine information from multiple frames, we experiment with two techniques - image processing (multi-sensor image fusion) and a neural network architecture

Hypothesis: The hypotheses we validate is that our methodology (which is a combination of the above four ideas), performs better than our baseline - independent frame by frame super-resolution on the task of video super-resolution, using PSNR and SSIM as the evaluation metrics on the CDVL dataset.

3.1. Dataset

For the purpose of our experiments, we use videos from the CDVL dataset. We chose an upscaling factor of 4 throughout our experiments.

For training, patches of 72x72 sampled from the CDVL dataset were used.

- Low Resolution: Patches sampled from LR frames - 25600x(72/4)x(72/4)
- High Resolution: Patches sampled from HR frames - 25600x72x72
- LR Bicubic-Upscaled Motion-Compensated: Patches sampled from Matlab bicubic interpolation upscaled and optical flow motion compensated frames - 25600x5x72x72

For Testing, whole videos are used instead of sample patches. The HR videos have a resolution of 1920*1080 (1920x1080xnum_frames) while the LR videos have a resolution of (1920/4)x(1080/4) for our upscaling factor of 4.

Note: The videos have one channel and are not RGB.

The motion-compensation is performed after estimating optical flow via Celiu optical flow estimation.

3.2. Loss Functions

Using a pixel-wise mean-squared error loss is inadequate for two reasons:

- Inability to adequately capture motion information.
- Pixel-Level loss is often unable to capture the texture in the videos and can be perceptually unsatisfactory

As shown in figure 1, say we have two estimates $g_1(t)$ and $g_2(t)$ for a given function $f(t)$. The mean-squared error computed will be the same for both the estimates since the deviation is the same for both. However, we know that $g_1(t)$ is likely a much better estimate than $g_2(t)$ since it captures the essence of the signal much better.

3.2.1. Motion Smoothed Loss

For a measure to be effective, it should align with a human observer's perception. The commonly used Sum of Squared Differences (SSD) for color information in image completion is insufficient for achieving satisfactory outcomes in videos, even when considering various color spaces. This inadequacy is primarily due to the human eye's heightened sensitivity to motion. Maintaining motion continuity is more important than finding the exact spatial pattern match within an image of the video.

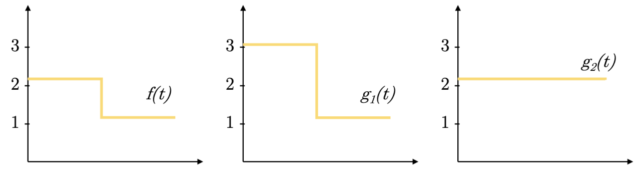


Figure 1: Perception Loss

We use a motion-aware loss as used in (5) for video inpainting. Consider sequence Y, which comprises the grayscale (intensity) data extracted from the color sequence. At each pixel, we calculate the temporal and spatial gradients Y_x, Y_y and Y_t

We then compute $u = \frac{Y_t}{Y_x}$ as the instantaneous motion in the x-direction and $v = \frac{Y_t}{Y_y}$ as the instantaneous motion in the y-direction. To compute the temporal and spatial gradients, we make use of the Sobel filter.

The loss is calculated by computing the mean-squared error between the motion u and v **(normalized)** at the pixel level for the output and the target.

3.2.2. Feature Space Loss

An alternative to the mean-squared error at the pixel level, we compute this at a feature level by extracting features from both the output and the target using a pre-trained VGG network. The feature space loss is computed by extracting features from the output and the target using a pre-trained VGG-16 architecture with features extracted from layer 31 (Appendix C).

3.3. Network Architecture

Multi-frame SRCNN

Including neighboring frames in the recovery process has been found to be beneficial for video super-resolution (SR), both for model-based approaches and learning-based approaches. The proposed CNN for video SR takes multiple LR frames as input (including the objective frame and its neighbors) and generates a single HR output frame. In this novel architecture, neighboring frames are included in the process to improve the quality of the output. Figure 2 illustrates how this is done by incorporating the two frames before and after the current frame t in the process. Initially, a single input frame with dimensions 1 × W × H is used, where W and H are the width and height of the input image, respectively. The first convolution is applied to all five frames. The two previous frames and the current frame (three frames in total) are concatenated along the first dimension, and the same is done for the two next frames and the current frame. The two resulting combinations of frames are then passed through the second layer, and the outputs are concatenated before passing through a ReLU activation function.

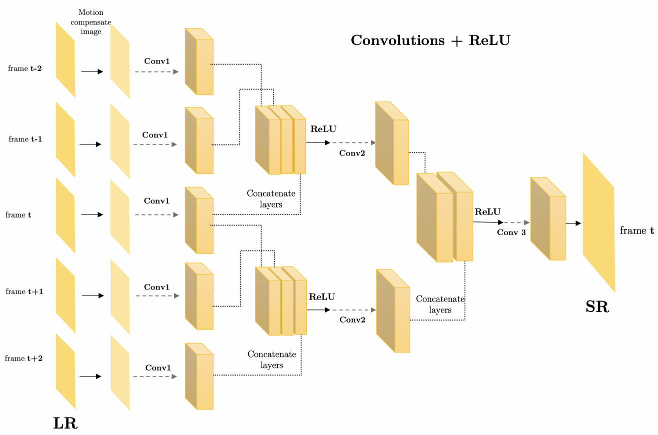


Figure 2: SRCNN Architecture

In order to speed up the video SR architecture and to bypass the need for a large video database, the reference CNN architecture is pre-trained on images. The weights of our model are then initialized using the SRCNN model (Appendix B) trained on images to provide warm starts during training. For this reason, the network parameters (kernel width, height, and their number) are shared with the SRCNN network. The difference arises in the concatenation layers and the filter depth in layer 2 and layer 3 which are five and two times larger, respectively.

3.4. Approach

3.4.1. Modified SRCNN with MSE Loss

In this approach (as shown in Figure 2), we use our modified multi-frame SRCNN architecture with a simple pixel-level MSE Loss. Weights for training are initialized by providing warm starts from single image superresolution-based SRCNN. Training is performed for 20 epochs using a learning rate of 0.001

3.4.2. Modified SRCNN with MSE+VGG Loss

In this approach (as shown in Figure 2), we use the same modified multi-frame SRCNN architecture but combine the pixel-level MSE loss with the feature space VGG loss. The loss is a weighted sum of the pixel-level and the feature space loss (2). For the feature space loss, we use a pre-trained VGG-16 architecture with features extracted from layer 31 (after the convolutional layers but before the linear layers). Features are extracted from the output and the target and the feature mean-squared error is computed. Since mean-squared error is not scale-invariant, we take a weighted sum of the losses with the same weights as in the SRGAN paper. Weights for training are initialized by providing warm starts from single image superresolution-based SRCNN.

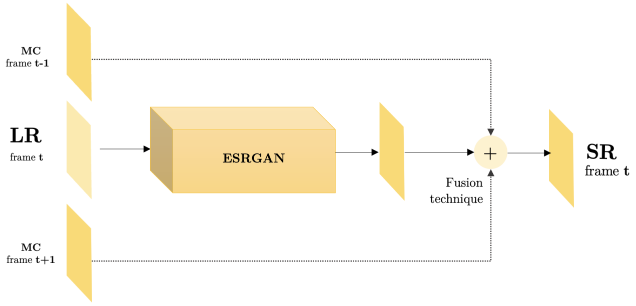


Figure 3: Image Fusion Architecture

3.4.3. Modified SRCNN with Motion-Aware Loss

As discussed previously, we utilize the augmented motion-aware loss intended to capture the motion between frames. The spatial derivatives are calculated on the current frame for both the output and the target. For the temporal derivatives across frames, we use current frame and the bicubic upscaled (previous and next) neighbouring frames i.e. the centered difference via the **sobel filter** for the current output with the bicubic upscaled neighbouring frames and similarly the centered difference for the current target with the bicubic upscaled neighbouring frames. The loss is a sum of the pixel-space loss and the gradient loss with **normalized gradients to keep the scale consistent**. Weights for training are initialized by providing warm starts from single image superresolution-based SRCNN.

3.4.4. Multi-Sensor Image Fusion

In this approach (as shown in Figure 3), we explore combining complementary information from multiple sources to create new images that are more suitable for visual perception. The two sources of information in our case are the current LR frame super-resolved by a pre-trained network (ESRGAN in our case) and the neighbouring upsampled motion-compensated frames. The images are recombined in a transformed domain with the help of a **wavelet transform** (3). The weights for recombination are 0.9 for the central frame and 0.1 for the neighbouring frames. Following recombination in the transformed domain, we obtain the final image by taking the inverse wavelet transform of the fused wavelet coefficients.

For the wavelet transform, we make use of the Daubechies (db5) wavelet.

3.4.5. Baseline

The baseline model is an SRCNN network applied to individual frames. It takes a motion-compensated LR frame as input and processes it through three convolutional layers, with ReLU applied to the two intermediate layers. The resulting output is the HR frame, which is combined with others to reconstruct the full video. To initialize the weights, a pre-trained SRCNN model trained on images is utilized.

3.5. Evaluation and Hypothesis Testing

3.5.1. Metrics

As discussed previously, mean-squared error is not an appropriate evaluation metric. The two metrics (predominantly used in literature) considered are:

- **PSNR**: measures the difference in pixels as - 10 $*$ $log_{10}(\frac{L^2}{MSE})$, where L = maximum range of colour value.
- **SSIM**: which measures the structural similarity between images. The SSIM index is calculated between various windows of an image. The measure between two windows x and y of size nxn is -

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

where μ and $sigma^2$ are the pixel mean and variance of the patches.

3.5.2. *Hypothesis Testing*

To compare the models, we sample from the test set to create n = 50 samples of k = 12 videos each, sampled with replacement from 25 test videos. By the Central Limit Theorem, taking sufficiently large random samples from the population with replacement, the distribution of the sample means will be approximately normally distributed. We will perform a one-tailed paired t-test on the observed metric to test our hypothesis with an alpha as 0.05, i.e. we seek to say with 95% confidence that the new approach outperforms the baseline. In other words:

- Null Hypothesis: The null hypothesis (H0) assumes that the true mean difference ($\mu_{A1} - \mu_{A2}$) - i.e., the mean of the metric of all the n samples is equal to zero
- Alternate Hypothesis: The one-tailed alternative hypothesis (H1) assumes that ($\mu_{A1} - \mu_{A2}$) is greater than zero

4. Experimental Results

For each experiment, the plot on the left compares the performance of the approaches on 25 test videos and plot the corresponding PSNRs for both the approaches.

The plot on the right gives the distribution of mean PSNRs (as computed for the 12 videos in each of the 50 samples).

4.1. *Experiment 1a*

In this experiment, we compare the multi-frame SRCNN approach based on MSE loss against the baseline.

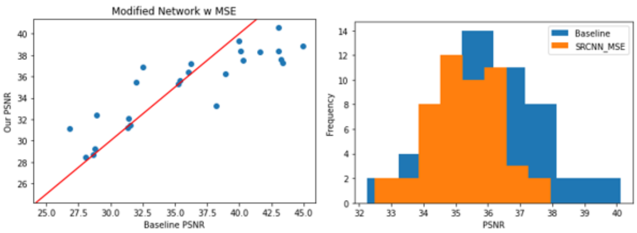


Figure 4: Multi-frame SRCNN with MSE Loss

We see that there are some videos for which our approach does better than the baseline while for others, it does not do as well. Performing a one-tailed paired t-test, we obtain a p-value of 0.99 and are not able to conclusively say that our model performs better than the baseline.

4.2. *Experiment 1b*

In this experiment, we compare the multi-frame SRCNN approach based on the feature space loss against the baseline.

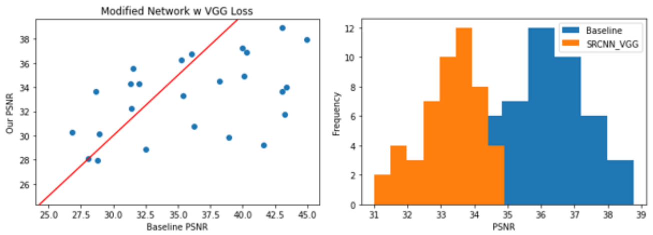


Figure 5: Multi-frame SRCNN with Feature Loss

In this case too, there are some videos for which our approach does better than the baseline. However, there are more videos for which the baseline performs better. Performing a one-tailed paired t-test, we obtain a p-value of 1 and cannot say that our approach performs improves upon the baseline.

4.3. *Experiment 1c*

In this experiment, we compare the multi-frame SRCNN approach based on the motion-aware loss against the baseline.

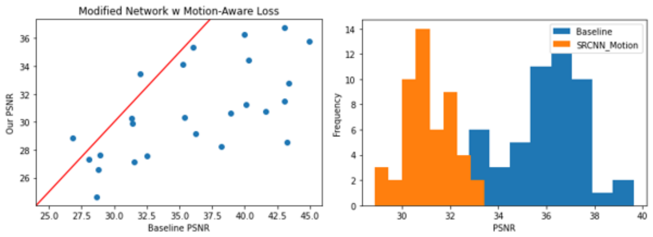


Figure 6: Multi-frame SRCNN with Motion Loss

In this case, we can see that the baseline performs better than our approach for most of the videos. For this approach, performing a one-tailed paired t-test gives us a p-value of 1 and our approach does not do better than the baseline.

4.4. *Experiment 2*

In this experiment, we compare the multi-sensor image fusion approach against the baseline.

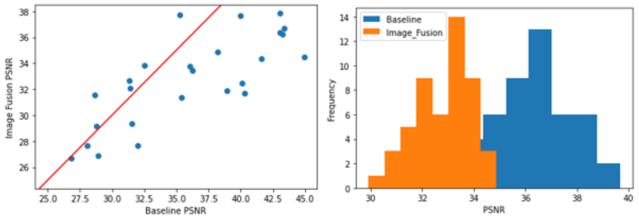


Figure 7: Multi-sensor Image Fusion

In this case too, our approach does not do better than the baseline.

4.5. Comparison

We compare the two approaches on two dimensions: performance (as measured by PSNR/SSIM) and efficiency (as measured by the training and runtimes).

4.5.1. Performance

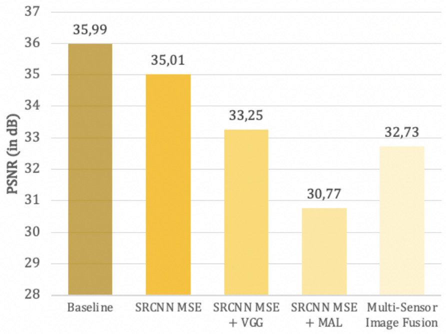


Figure 8: Average Test PSNR

Figure 8 compares the average PSNR values in dB for the 25 test videos for the approaches and the baseline. Despite coming close to the SRCNN and MSE loss, the proposed methods fail to surpass the baseline. However, it's worth noting that all approaches maintain a PSNR above 30, indicating the attainment of a satisfactory image quality. In Appendix A, we present the average SSIM values.

4.5.2. Efficiency

Results		
	Training Time	Runtime
SRCNN MSE	60s / epoch	7s
SRCNN VGG	130s / epoch	7s
SRCNN Motion	470s / epoch	7s
Image Fusion	—	180s

Table 1: Training and Run-times in Seconds

Table 1 shows the training times for the training dataset described above with a batch size of 256. The approximate training time per epoch is shown and is the lowest for the mean-squared error loss, increases for the feature space loss and is the highest for the motion-aware loss.

The runtime is shown for a sample test video with 14 frames. The network architectures take only about 7 seconds to run and are therefore much better suited for real-time applications than the Image Fusion technique which takes much longer (about 3 minutes) owing to the computation of the wavelet transforms.

4.5.3. Sample Results

Figure 9 shows the sample results taken from one of the test videos.

We also observe some motion artifacts with the motion-aware loss approach as seen in Figure 10. This is likely due to some lag in matching gradients and some smoothing may be able to alleviate the problem.

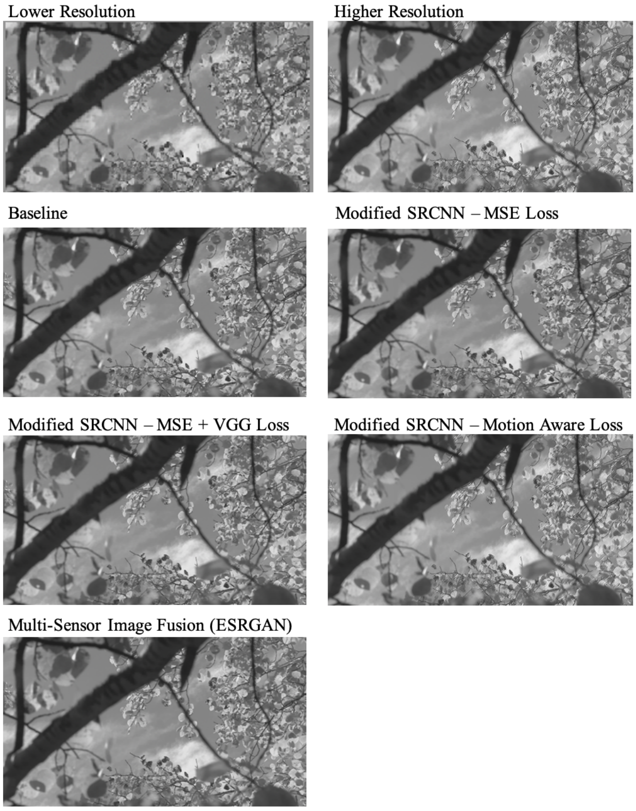

Lower Resolution
Higher Resolution
Baseline
Modified SRCNN – MSE Loss
Modified SRCNN – MSE + VGG Loss
Modified SRCNN – Motion Aware Loss
Multi-Sensor Image Fusion (ESRGAN)

Figure 9: Scene 40 comparison results

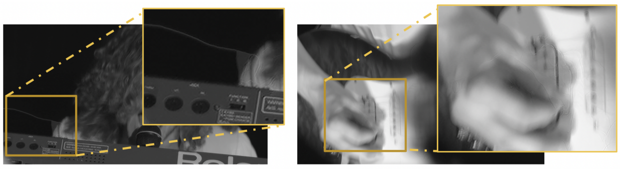


Figure 10: Motion Loss Artifacts

5. Conclusion

In this project, we have explored two approaches for video super-resolution - one based on a multi-frame SRCNN architecture and the other on wavelet based image fusion. We experiment with alternative loss functions more suited for the task as opposed to the traditional mean-squared error. We compare each of the approaches with a single-frame super-resolution baseline. One of our approaches is close to the baseline but the approaches do not conclusively outperform the baseline.

Some key areas of improvement would be to conduct a more extensive hyperparameter tuning on a larger dataset for more epochs. The motion-loss considered is local and incorporating a global loss could yield better results. Moreover, evaluating on alternate metrics (besides PSNR) which explicitly capture motion could be an important step to align better with human perception.

References

[1] Qiqin Dai Armin Kappeler, Seunghwan Yoo and Aggelos K. Katsaggelos. Video super-resolution with convolutional neural networks. 2016.

[2] Ferenc Husza r Jose Caballero Andrew Cunningham Alejandro Acosta Andrew Aitken Alykhan Tejani Johannes Totz Zehan Wang Wenzhe Shi Christian Ledig, Lucas Theis. Photo-realistic single image super-resolution using a generative adversarial network. 2017.

[3] S. K. Mitra H. Li, B. S. Manjunath. Multisensor image fusion using the wavelet transform. 1994.

[4] Peng Zhao Chao Dong Fanhua Shang Yuanyuan Liu Linlin Yang Radu Timofte Hongying Liu, Zhubo Ruan. Video super-resolution based on deep learning: a comprehensive survey. 2022.

[5] M. Irani Y. Wexler, E. Shechtman. Space-time video completion.

6. Contributions

Both students collaborated and worked together on all aspects and phases of the project. Abhranil assumed the primary role in exploring various loss functions, designing the multi-sensor image fusion approach, and performing hypothesis testing. On the other hand, Jan took the lead in experimenting with different architectures for the SRCNN and other unimplemented approaches. Abhranil has been primarily responsible for writing the report, while Jan has taken charge of the presentation.

GitHub code: `https://github.com/abhra001/Video_SuperResolution`

Appendix A. SSIM plot

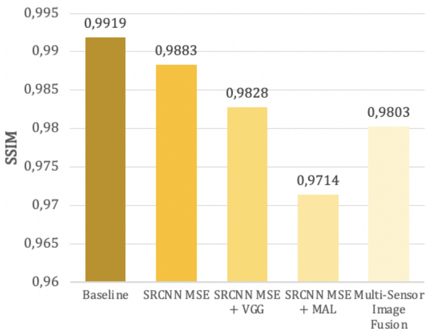
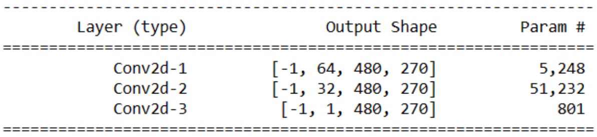


Figure A.11: Average Test SSIM

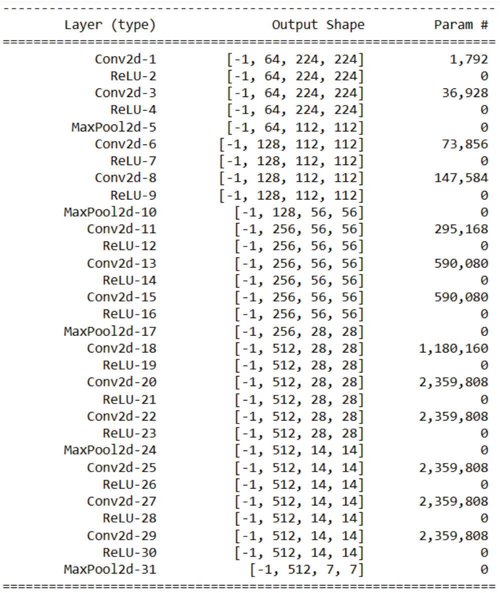
Appendix B. SRCNN Network



```
----------------------------------------------------------------
        Layer (type)               Output Shape         Param #
================================================================
            Conv2d-1         [-1, 64, 480, 270]           5,248
            Conv2d-2         [-1, 32, 480, 270]          51,232
            Conv2d-3          [-1, 1, 480, 270]             801
================================================================
```

Figure B.12: SRCNN Network

Appendix C. VGG Feature Network



```
----------------------------------------------------------------
        Layer (type)               Output Shape         Param #
================================================================
            Conv2d-1         [-1, 64, 224, 224]           1,792
              ReLU-2         [-1, 64, 224, 224]               0
            Conv2d-3         [-1, 64, 224, 224]          36,928
              ReLU-4         [-1, 64, 224, 224]               0
         MaxPool2d-5         [-1, 64, 112, 112]               0
            Conv2d-6        [-1, 128, 112, 112]          73,856
              ReLU-7        [-1, 128, 112, 112]               0
            Conv2d-8        [-1, 128, 112, 112]         147,584
              ReLU-9        [-1, 128, 112, 112]               0
        MaxPool2d-10          [-1, 128, 56, 56]               0
           Conv2d-11          [-1, 256, 56, 56]         295,168
             ReLU-12          [-1, 256, 56, 56]               0
           Conv2d-13          [-1, 256, 56, 56]         590,080
             ReLU-14          [-1, 256, 56, 56]               0
           Conv2d-15          [-1, 256, 56, 56]         590,080
             ReLU-16          [-1, 256, 56, 56]               0
        MaxPool2d-17          [-1, 256, 28, 28]               0
           Conv2d-18          [-1, 512, 28, 28]       1,180,160
             ReLU-19          [-1, 512, 28, 28]               0
           Conv2d-20          [-1, 512, 28, 28]       2,359,808
             ReLU-21          [-1, 512, 28, 28]               0
           Conv2d-22          [-1, 512, 28, 28]       2,359,808
             ReLU-23          [-1, 512, 28, 28]               0
        MaxPool2d-24          [-1, 512, 14, 14]               0
           Conv2d-25          [-1, 512, 14, 14]       2,359,808
             ReLU-26          [-1, 512, 14, 14]               0
           Conv2d-27          [-1, 512, 14, 14]       2,359,808
             ReLU-28          [-1, 512, 14, 14]               0
           Conv2d-29          [-1, 512, 14, 14]       2,359,808
             ReLU-30          [-1, 512, 14, 14]               0
        MaxPool2d-31            [-1, 512, 7, 7]               0
================================================================
```

Figure C.13: VGG Feature Network

Appendix D. Table of Results

Scene	Baseline	MSE	MSE+VGG	MSE+MAL	IF
3	28,09	28,45	28,08	27,36	27,67
5	26,85	31,12	30,29	28,87	26,66
6	31,99	35,44	34,27	33,43	27,64
8	28,89	32,40	30,11	27,64	26,87
13	32,50	36,86	28,84	27,54	33,84
15	43,03	40,57	38,93	36,75	36,35
18	39,98	39,34	37,27	36,28	37,64
21	44,92	38,88	37,93	35,76	34,49
23	40,33	37,50	36,93	34,46	31,67

Table D.2: PSNR values per scene

Scene	Baseline	MSE	MSE+VGG	MSE+MAL	IF
3	0,999	0,999	0,996	0,974	0,983
5	0,955	0,948	0,933	0,925	0,926
6	1,007	0,966	0,942	0,909	0,924
8	0,950	0,946	0,944	0,929	0,933
13	0,950	0,937	0,927	0,923	0,945
15	0,999	0,998	0,986	0,978	0,987
18	0,999	0,999	0,999	0,983	0,981
21	1,000	0,998	0,998	0,997	0,997
23	0,999	0,997	0,994	0,994	0,994

Table D.3: SSIM values per scene