**REPORT**

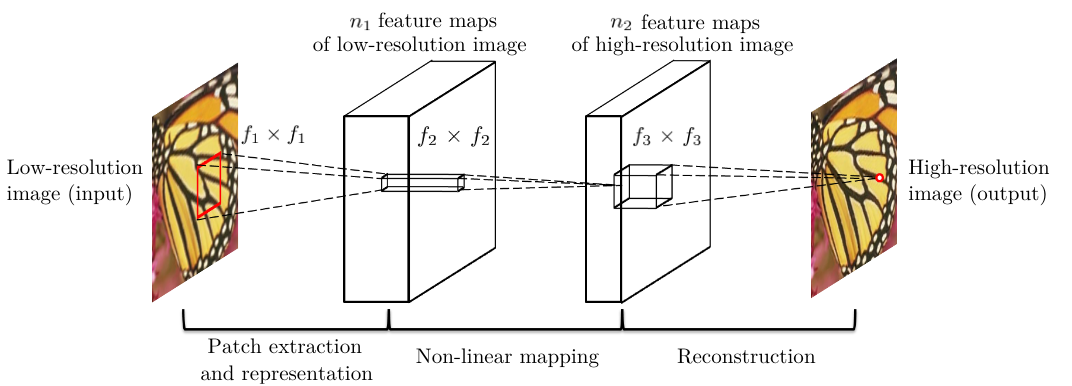
1. **ABSTRACT**

1. **INTRODUCTION**
2. **METHOD**

**3.1 Model Architecture:** As described in the paper, we designed a three-layer SRCNN model. The network takes a bicubic-upsampled image as input, learns to refine it by predicting a high-dimensional representation, and outputs a high-resolution image.

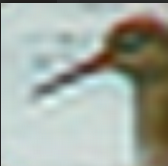
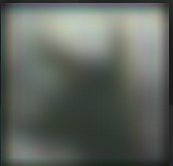
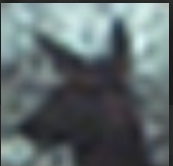
The first layer, called the **patch extraction and representation layer**, uses 128 channels and a 9 × 9 kernel to capture features such as edges, textures, and other fine details from the upsampled input. Its outputs are passed through a ReLU activation for non-linear feature representation.

The second layer, the **non-linear mapping layer**, applies a 5 × 5 convolution that reduces the feature depth from 128 to 64 channels. This stage introduces non-linear transformations, enabling the model to learn complex mappings between low- and high-frequency structures.

Finally, the **reconstruction layer** uses a 5 × 5 convolution to combine the processed features and produce the final three-channel RGB high-resolution image.

*Figure \_: Model Architecture.*

**3.2 Model Training:** The training script runs for 500 epochs and implements a complete training loop for three-layer SRCNN model. We used an STL-10 image dataset, downsampled it by a default factor of 2 to create a low resolution image and kept the original image as the high resolution image and created a low-resolution and high-resolution pair. The training scripts begins by loading paired low- and high-resolution image crops from the STL dataset using a custom loader. The low-resolution batch is then passed through the SuperResolution model, which outputs both a reconstructed high-resolution image and intermediate feature maps. Training optimizes the mean-squared error (MSE) between the predicted high-resolution image and the ground-truth using the Adam optimizer with a learning rate of 1 × 10⁻⁴. A second MSE loss is calculated between the upsampled low-resolution input and the ground-truth as a bicubic baseline for comparison. Gradients are back-propagated and the optimizer updates the model weights, with gradients zeroed after each step. This training procedure continues for 500 epochs, resulting in a fully trained SRCNN model capable of refining bicubic-interpolated images into higher-quality reconstructions.

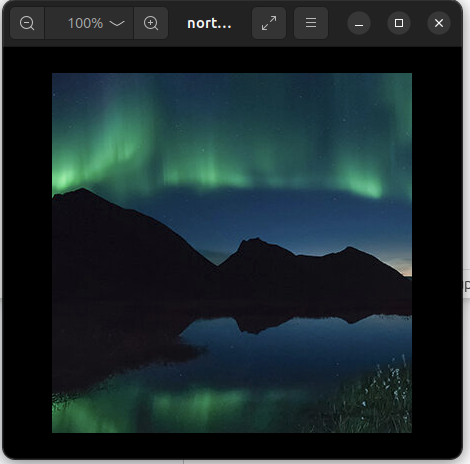


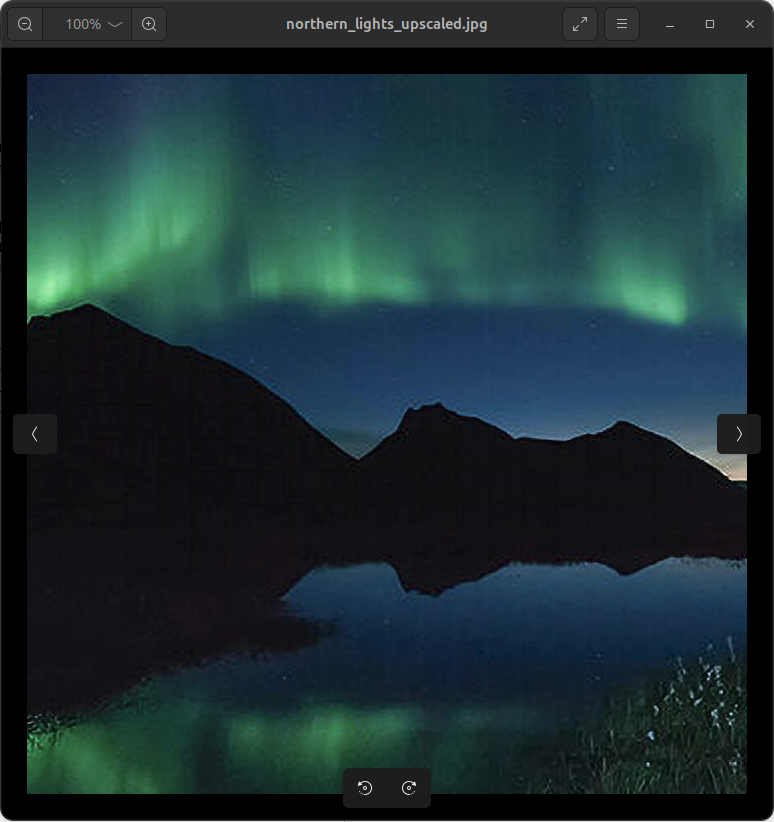
*Figure \_: Epoch 1 Figure \_: Epoch 33 Figure \_: Epoch 1 Figure \_: Epoch 33*

The above image shows the training process. We can see the difference in how patches are displayed at epoch 1 and how the model is learning to improve the image representation for high resolution images displayed at epoch 33.

**3.3 Model Inference:** The inference model implements the complete pipeline for single image super resolution using three-layer SuperResolution model. The inference script takes in a low resolution input image, upscales the input image using bicubic interpolation to match the desired output dimensions. This bicubic interpolation image is then processed to produce smaller patches of defined sizes and each patch is fed through trained SRCNN model, and the predicted high resolution patches are combined to reconstruct an entire high resolution image.

The run\_super\_resolution function acts as a high-level utility to perform end-to-end super-resolution on a given image. It first loads the pre-trained model weights and sets the network to evaluation mode. Then it transforms the input image into a tensor and calls the execute function to apply the super-resolution model. Finally, the resulting image tensor is converted back to a PIL image and saved to disk



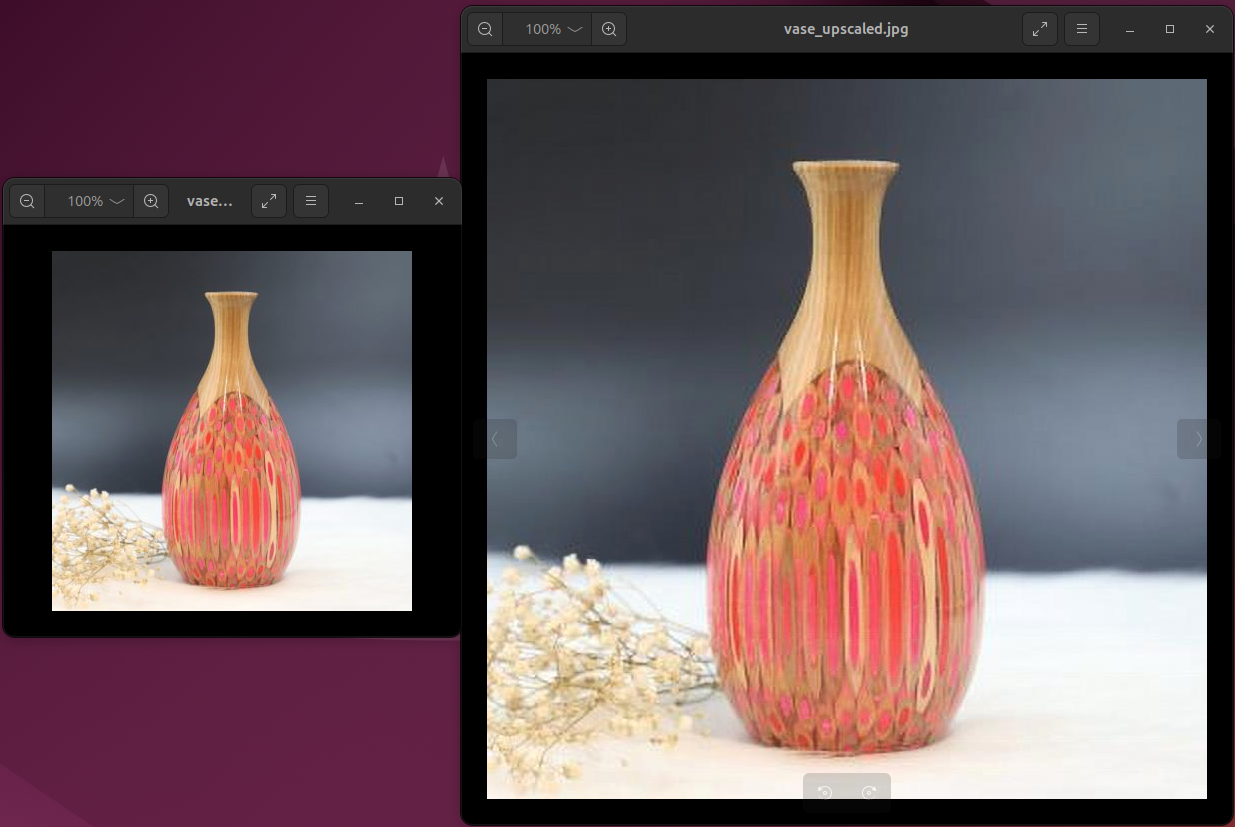
*Figure \_: Input Image*

*Figure \_: Output Image*

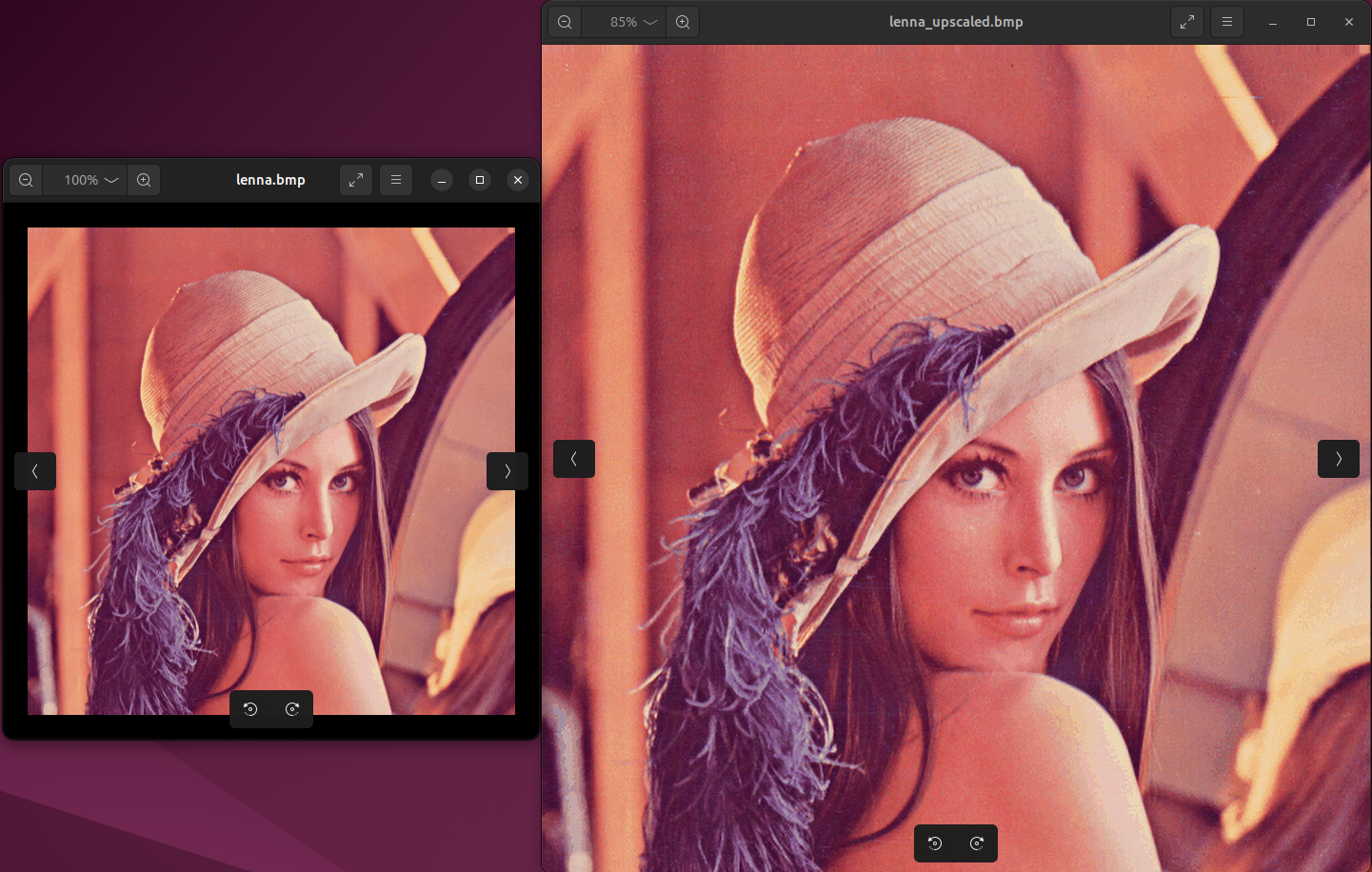
1. **RESULTS**



**Example 1:** The above image is an example of how our model achieved Super Resolution of an image. The smaller image is upscaled using bicubic interpolation and then passes through the SRCNN model to predict the ground truth.



**Example 2:** This is another example of single image super resolution. If we look closely, there are areas in the output image which appear to be a low resolution image and have not completely transformed into a full high resolution image. This might either be due to input image not being preprocessed correctly. The low resolution image when upscaled, did not fill in the correct details during interpolation where the model could predict the ground truth for the interpolated pixel value.



**Example 3:** This image predicts the high resolution image very accurately with fine details of the image such as the blue fur of the hat and the hair being visible enough to recognise. I assumed the low resolution image wouldn’t be able to show these details accurately but it maybe due to the trained model predicting the pixel values to its ground truth.

1. **REFLECTION AND ACKNOWLEDGEMENT**

**Reflection:** Implementing this paper, I am confident that a small three-layer super resolution CNN model is capable of learning and mapping a low-resolution image directly into a high resolution image.

A simple architecture emphasizing on 3 main processes like patch extraction, non-linear mapping and reconstruction can capture high dimensional details.

This paper showed me the potential of CNN models, how a simple 3 layer model can enhance a low resolution interpolated image into a high resolution output image simply by training the model of sufficient data.

**Acknowledgment:** We used the following dataset for model training: <https://cs.stanford.edu/~acoates/stl10/>

We used the idea to design the SRCNN model as described in the following link: <https://github.com/amanshenoy/image-super-resolution?tab=readme-ov-file>