INTEL UNNATI PROJECT REPORT

Image Sharpening using Knowledge Distillation

TEAM NAME: ClearSight
TEAM MEMBERS: 1.Gompa Harshit Abhinav hgompa2@gitam.in
2.Bonthu Abhinav Reddy abontu@gitam.in
3.Challa Sai Lalasa schalla12@gitam.in
MENTOR: Dr A Anupama anangadi@gitam.edu
INSTITUTION: Gandhi Institute of Technology and Management(GITAM)

CONTENTS:

1. INTRODUCTION:

The recent years have significantly shown the importance of video conferencing.Video conferencing has grown to be a crucial component of communication in both personal and professional contexts .However, issues like low bandwidth, unfavorable network conditions, or hardware constraints frequently have an impact on the quality of video streams. One common artifact is the loss of image sharpness, which can significantly degrade the overall visual experience and effectiveness of communication and in worst cases it could change the context of conference proceedings.

To address this problem, image enhancement techniques—specifically image sharpening—have gained importance. Image sharpening in video conferencing enhances the clarity of visuals by emphasizing edges and fine details in each frame.It assists in reducing blurriness brought on by low resolution or unfavorable network conditions. By enabling participants to see faces, text, and objects more clearly in real time, this guarantees improved visual communication.

This project proposes a deep learning-based image sharpening system using knowledge distillation, where a large, pre-trained teacher model transfers its learning to a smaller, faster student model. The student model is perfect for deployment in video conferencing software or edge devices because it is made for real-time performance with minimal computational overhead. The model is made to strive to produce consistently sharp and high-quality visuals under real-world conditions by training on pairs of degraded and high-quality images and assessing using both objective and subjective metrics.

2. PROBLEM STATEMENT:

Design and develop a lightweight image sharpening model using a Student-Teacher knowledge distillation framework for real-time video conferencing.

The Teacher model ought to be a high-capacity, pre-trained network that generates sharpened images of superior quality.The Student model must learn to approximate the Teacher's output while being optimized for fast inference (30–60 fps) on 1080p resolution.

Pairs of degraded and original images should be used to train the model, and it should be assessed using:

- SSIM (Structural Similarity Index) with a target score of > 0.90
- MOS (Mean Opinion Score) from human evaluators

The objective is to maintain the model's efficiency for deployment on edge devices while producing crisp and aesthetically pleasing outputs in real time, even in the face of network-induced degradation.

3. DATA SOURCES:

Relevant Articles for study:

1. https://www.ibm.com/think/topics/knowledge-distillation
2. https://ieeexplore.ieee.org/document/8301935

DATASET USED:

This project uses three publicly available datasets from Kaggle that provide high-resolution and low-resolution images suitable for training and evaluating image sharpening models:

1. DIV2K Dataset for Super Resolution

   https://www.kaggle.com/datasets/takihasan/div2k-dataset-for-super-resolution

   - Contains 1,000 high-quality images at 2K resolution.
   - A dataset widely used for super-resolution and image enhancement tasks.
2. Low Resolution Photographs

   https://www.kaggle.com/datasets/noobyogi0100/low-resolution-photographs

   - Includes over 200 naturally degraded low-resolution images.
   - Used as real-world degraded inputs for model evaluation and testing.
3. Image Super Resolution from Unsplash

   https://www.kaggle.com/datasets/quadeer15sh/image-super-resolution-from-unsplash

   - A selected subset from the large dataset of high-quality images was used to generate synthetic degraded images for training.
   - Chosen images were visually diverse (faces, nature, text, etc.) to help the model generalize well.
   -

In this project, standard preprocessing techniques from image restoration workflows were applied to create paired training data. High-resolution images from the selected datasets were first downscaled using both bicubic and bilinear interpolation to simulate various levels of degradation and vice versa with low resolution images.

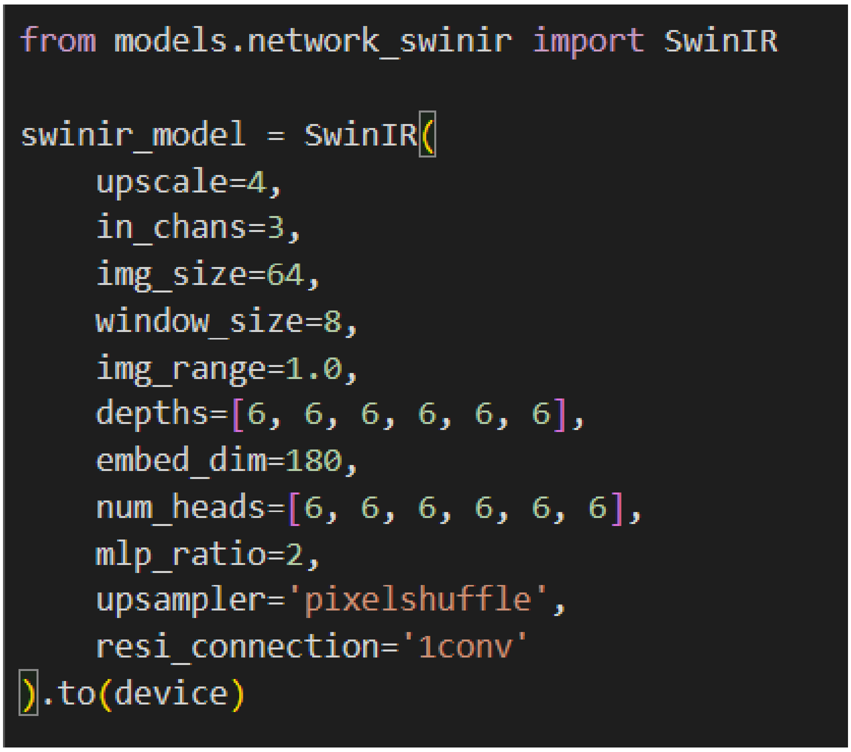
Specifically, scaling factors of 1.5×, 1.7×, 2× were used to create multiple variants of low-resolution images from the same high-resolution source. These degraded images were then upscaled back to their original size, preserving the blur and loss of detail typically observed in low-bandwidth video streams. Each degraded-upscaled image was paired with its corresponding high-resolution image, forming a supervised learning dataset.

4. MODEL ARCHITECTURE:

4.1 Teacher Model – SwinIR (Swin Transformer for Image Restoration)

Teacher architecture:

SwinIR is a transformer-based model for image restoration, built on the Swin Transformer architecture. It excels at tasks like super-resolution, denoising, and artifact removal. SwinIR captures both local and global features effectively using shifted window attention. Though powerful, it's computationally heavy—making it ideal as a teacher model in knowledge distillation.



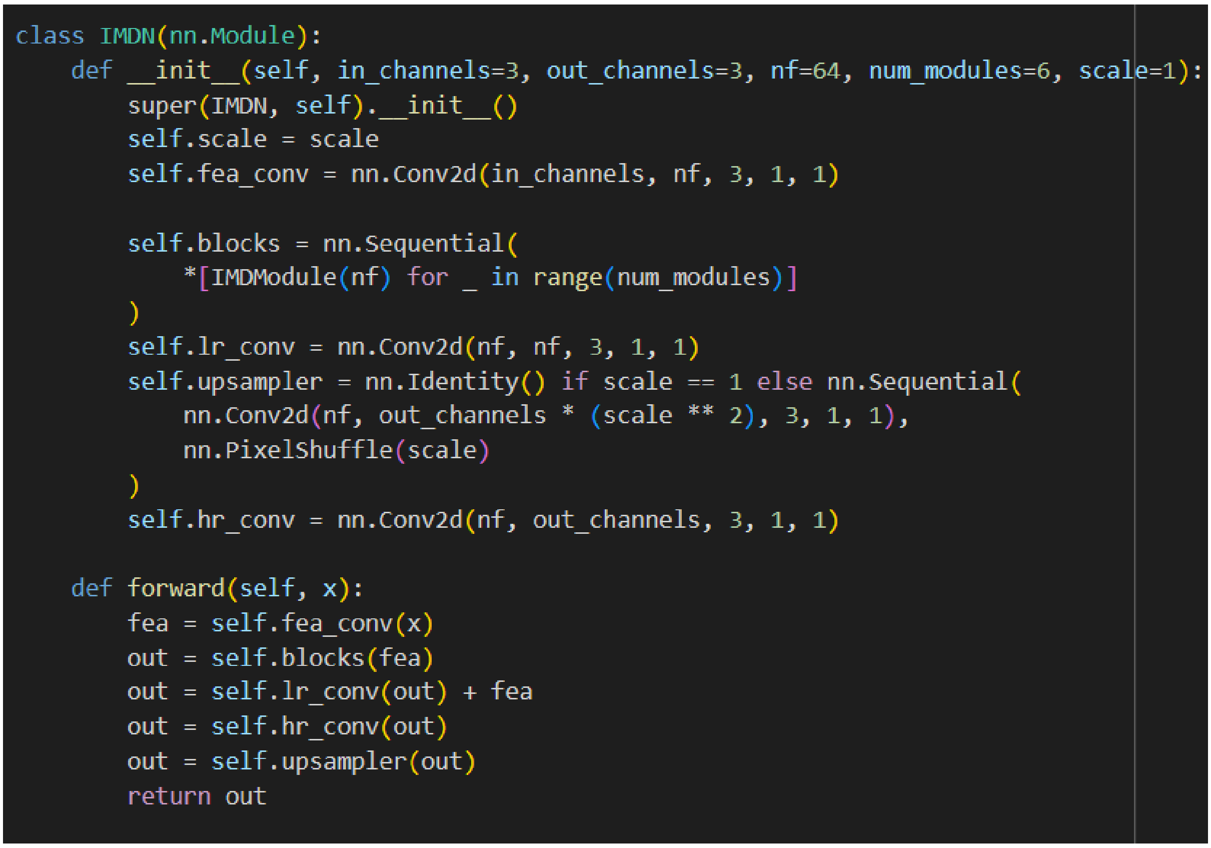
```python
from models.network_swinir import SwinIR

swinir_model = SwinIR(
    upscale=4,
    in_chans=3,
    img_size=64,
    window_size=8,
    img_range=1.0,
    depths=[6, 6, 6, 6, 6, 6],
    embed_dim=180,
    num_heads=[6, 6, 6, 6, 6, 6],
    mlp_ratio=2,
    upsampler='pixelshuffle',
    resi_connection='1conv'
).to(device)
```

4.2 Student Model – IMDN(Information Multi-distillation Network):

IMDN (Information Multi-Distillation Network) is a lightweight CNN-based model designed for efficient image super-resolution. It uses multi-distillation blocks to progressively extract and refine features. IMDN balances speed and accuracy, making it ideal for real-time applications. Its compact design makes it a perfect student model for knowledge distillation from heavy transformer models like SwinIR.

It is a compact convolutional neural network designed for fast and accurate image super-resolution.It employs IMDB's to progressively extract, split, and fuse features for better representation. Despite its small size (~0.8M parameters), it achieves competitive performance with much larger models. IMDN is well-suited as a student model in knowledge distillation due to its speed and efficiency in real-time applications.



```
class IMDN(nn.Module):
    def __init__(self, in_channels=3, out_channels=3, nf=64, num_modules=6, scale=1):
        super(IMDN, self).__init__()
        self.scale = scale
        self.fea_conv = nn.Conv2d(in_channels, nf, 3, 1, 1)

        self.blocks = nn.Sequential(
            *[IMDModule(nf) for _ in range(num_modules)]
        )
        self.lr_conv = nn.Conv2d(nf, nf, 3, 1, 1)
        self.upsampler = nn.Identity() if scale == 1 else nn.Sequential(
            nn.Conv2d(nf, out_channels * (scale ** 2), 3, 1, 1),
            nn.PixelShuffle(scale)
        )
        self.hr_conv = nn.Conv2d(nf, out_channels, 3, 1, 1)

    def forward(self, x):
        fea = self.fea_conv(x)
        out = self.blocks(fea)
        out = self.lr_conv(out) + fea
        out = self.hr_conv(out)
        out = self.upsampler(out)
        return out
```

5. TRAINING STRATEGY:

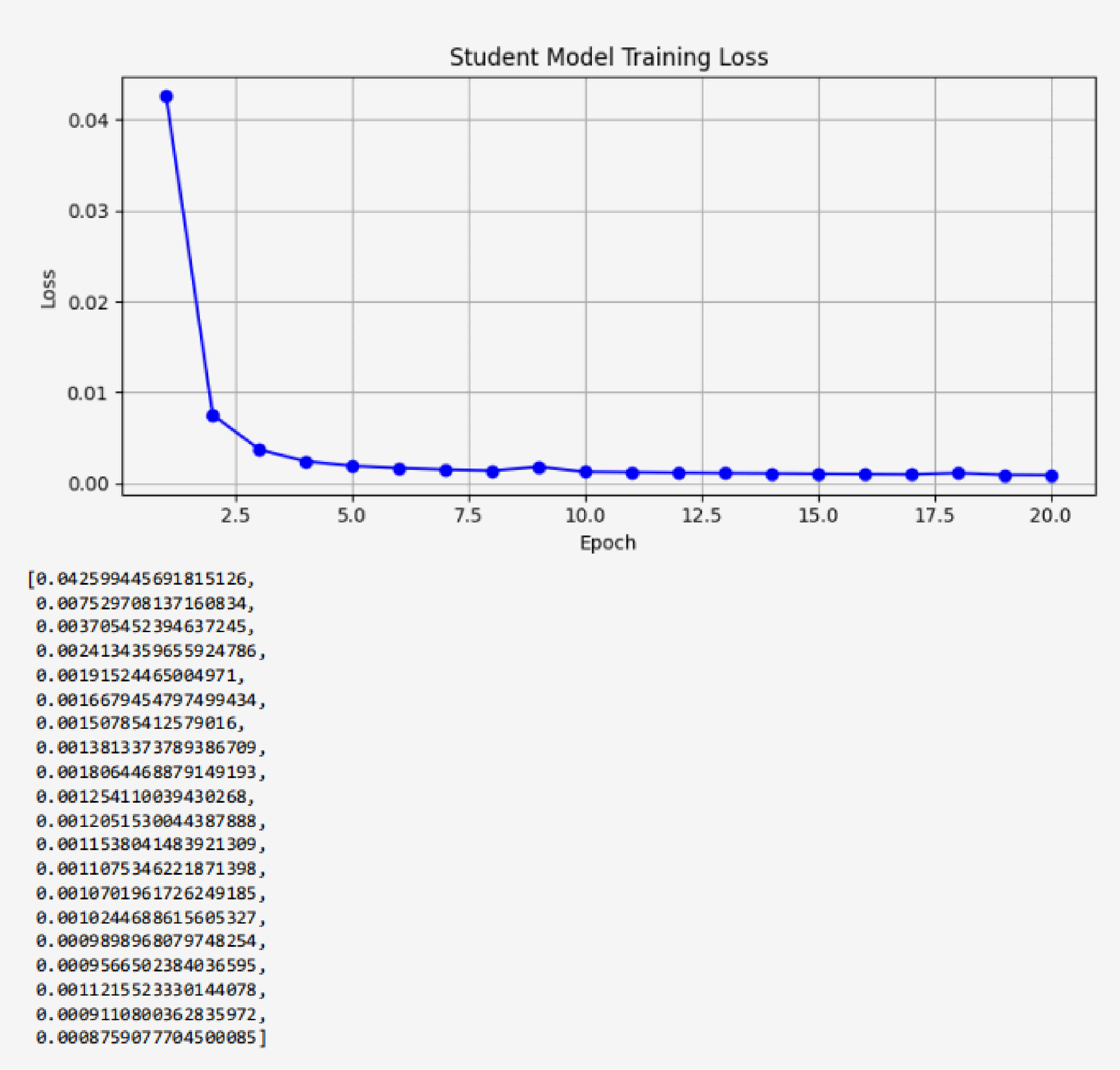
The training process follows a **Knowledge Distillation (KD)** framework, where a lightweight student model (IMDN) learns to mimic the performance of a high-capacity teacher model (SwinIR). The objective is to retain high image restoration quality while drastically reducing model size and inference time. The training loss is composed of two components: (1) a **Pixel Loss** (L1) that ensures the student output is close to the ground truth high-resolution image, and (2) a **Distillation Loss** (L2) that minimizes the difference between intermediate feature representations of the teacher and student models. The combined loss function is expressed as:

$$\mathbf{L\_total = L\_pixel + \alpha \times L\_distill},$$

where α is a weighting factor (typically 0.1–0.3).

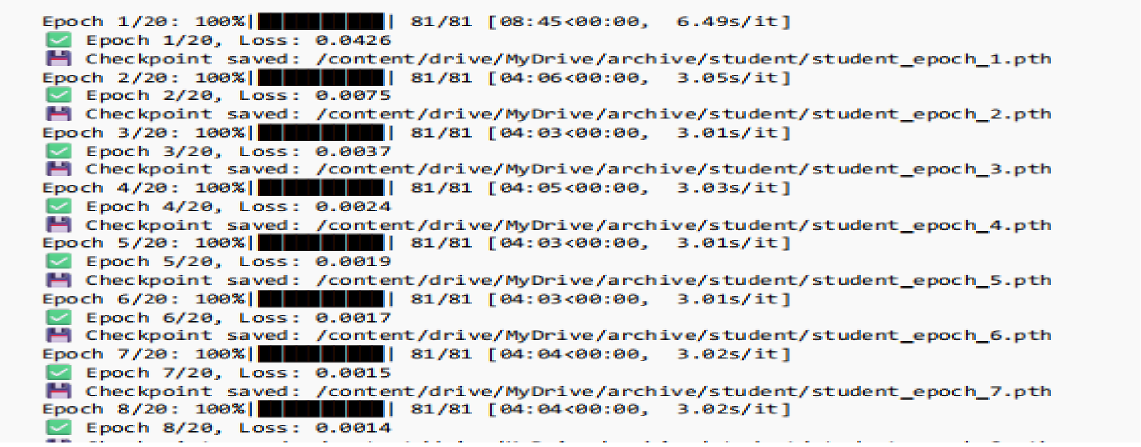
Training data is generated by downscaling high-resolution images using bicubic interpolation to simulate low-resolution input. The model is trained using the Adam optimizer with an initial learning rate of 1e-4, decayed over time using a cosine annealing schedule. Patches of size 128×128 are extracted for efficient batch processing, with a batch size of 4.The training is conducted over 50 epochs to ensure convergence while preventing overfitting.

Beyond the standard loss functions, the training incorporates feature-level guidance by extracting and matching intermediate representations from both the SwinIR and IMDN models. This ensures the student not only learns to produce sharp images but also mimics the internal processing style of the teacher. The training environment is optimized using GPU acceleration on platforms like Google Colab, significantly reducing training time. Additionally, data augmentation strategies, including random flips and rotations, are applied to introduce variability and improve the model's robustnes.



```
[0.04259944569181512б,
 0.007529708137160834,
 0.003705452394637245,
 0.0024134359655924786,
 0.001915244650049 71,
 0.0016679454797499434,
 0.00150785412579016,
 0.0013813373789386709,
 0.0018064468879149193,
 0.00125411003943026 8,
 0.0012051530044387888,
 0.0011538041483921309,
 0.0011075346221871398,
 0.0010701961726249185,
 0.0010244688615605327,
 0.0009898968079748254,
 0.0009566502384036595,
 0.0011215523330144078,
 0.0009110800362835972,
 0.0008759077704500085]
```

Training of student with loss model being printed for every epoch and then saving the model for every epoch like a backup.



```
Epoch 1/20: 100%|██████████| 81/81 [08:45<00:00,  6.49s/it]
 ✅ Epoch 1/20, Loss: 0.0426
 💾 Checkpoint saved: /content/drive/MyDrive/archive/student/student_epoch_1.pth
Epoch 2/20: 100%|██████████| 81/81 [04:06<00:00,  3.05s/it]
 ✅ Epoch 2/20, Loss: 0.0075
 💾 Checkpoint saved: /content/drive/MyDrive/archive/student/student_epoch_2.pth
Epoch 3/20: 100%|██████████| 81/81 [04:03<00:00,  3.01s/it]
 ✅ Epoch 3/20, Loss: 0.0037
 💾 Checkpoint saved: /content/drive/MyDrive/archive/student/student_epoch_3.pth
Epoch 4/20: 100%|██████████| 81/81 [04:05<00:00,  3.03s/it]
 ✅ Epoch 4/20, Loss: 0.0024
 💾 Checkpoint saved: /content/drive/MyDrive/archive/student/student_epoch_4.pth
Epoch 5/20: 100%|██████████| 81/81 [04:03<00:00,  3.01s/it]
 ✅ Epoch 5/20, Loss: 0.0019
 💾 Checkpoint saved: /content/drive/MyDrive/archive/student/student_epoch_5.pth
Epoch 6/20: 100%|██████████| 81/81 [04:03<00:00,  3.01s/it]
 ✅ Epoch 6/20, Loss: 0.0017
 💾 Checkpoint saved: /content/drive/MyDrive/archive/student/student_epoch_6.pth
Epoch 7/20: 100%|██████████| 81/81 [04:04<00:00,  3.02s/it]
 ✅ Epoch 7/20, Loss: 0.0015
 💾 Checkpoint saved: /content/drive/MyDrive/archive/student/student_epoch_7.pth
Epoch 8/20: 100%|██████████| 81/81 [04:04<00:00,  3.02s/it]
 ✅ Epoch 8/20, Loss: 0.0014
```

6. EVALUATION METRICS:

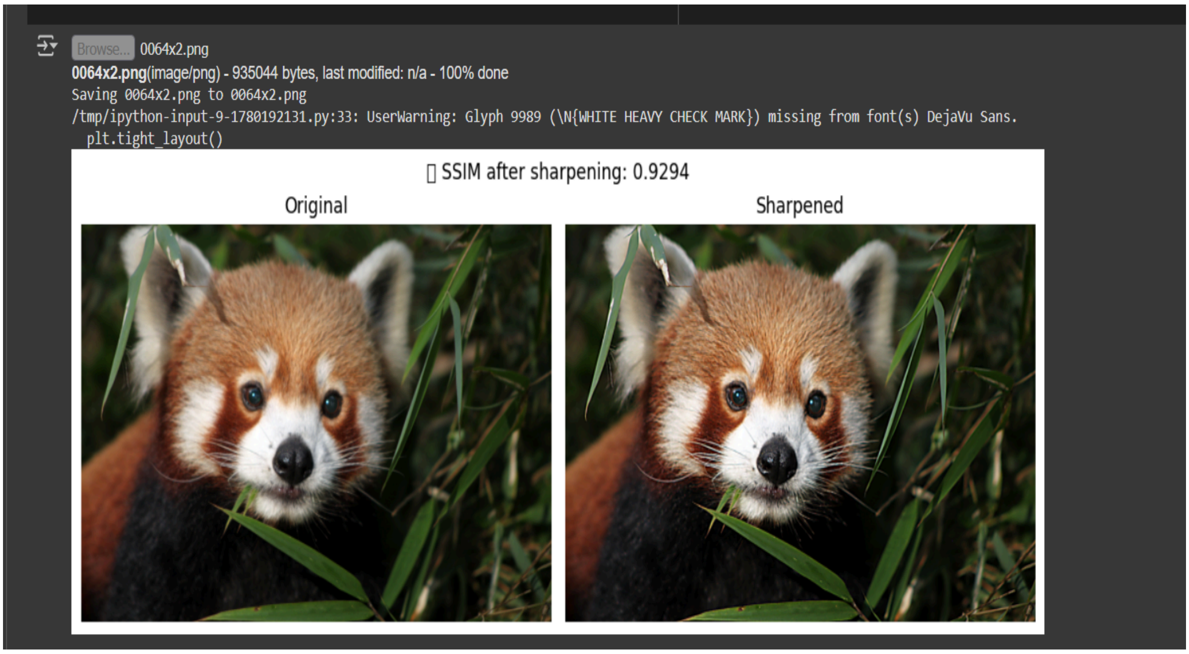
To assess the performance of the image sharpening model, both objective and subjective evaluation methods were used:

6.1 Structural Similarity Index (SSIM):

The perceptual similarity between the model's output and the original high-resolution image is measured using SSIM. It is appropriate for assessing image sharpness because it records luminance, contrast, and structure.

GOAL: Reach SSIM > 0.90

ACHIEVED: SSIM is greater than 0.90



```
0064x2.png
0064x2.png(image/png) - 935044 bytes, last modified: n/a - 100% done
Saving 0064x2.png to 0064x2.png
/tmp/ipython-input-9-1780192131.py:33: UserWarning: Glyph 9989 (\N{WHITE HEAVY CHECK MARK}) missing from font(s) DejaVu Sans.
  plt.tight_layout()
```

□ SSIM after sharpening: 0.9294

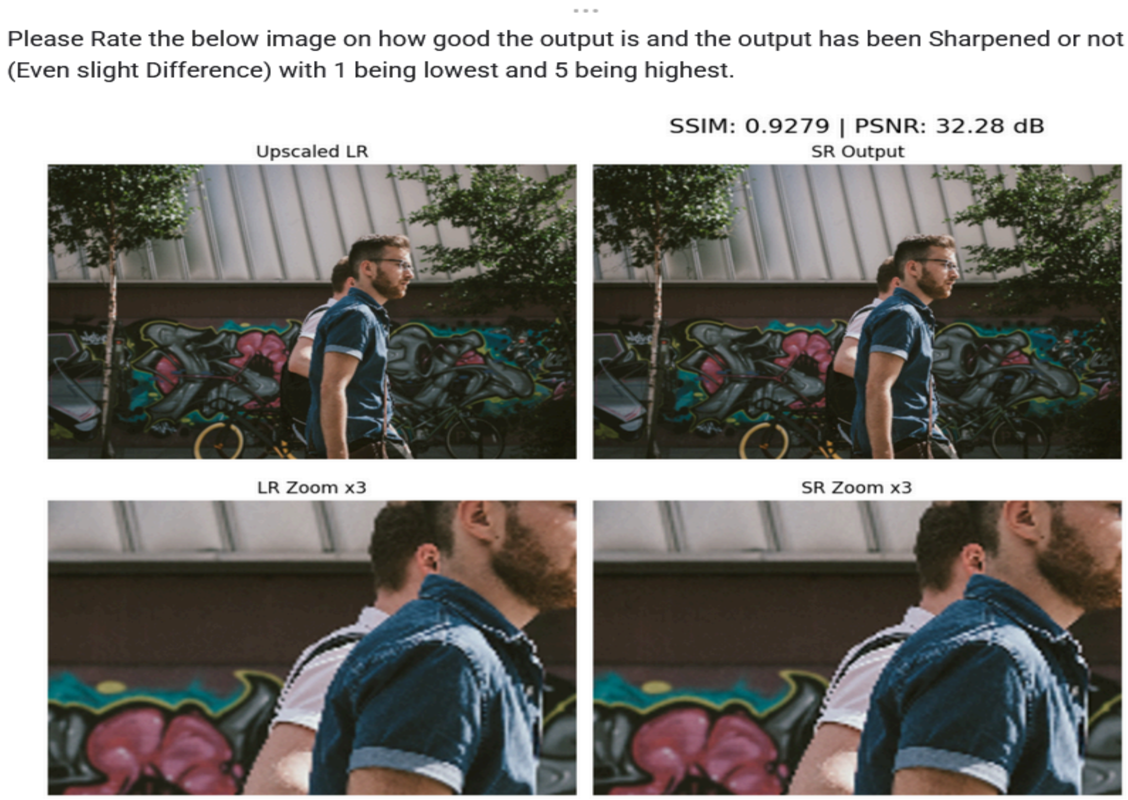
6.2 Mean Opinion Score (MOS):

MOS is a subjective evaluation where human users rate the visual quality of the sharpened images on a scale of 1 to 5. Participants assess factors like clarity, edge definition, and overall realism.

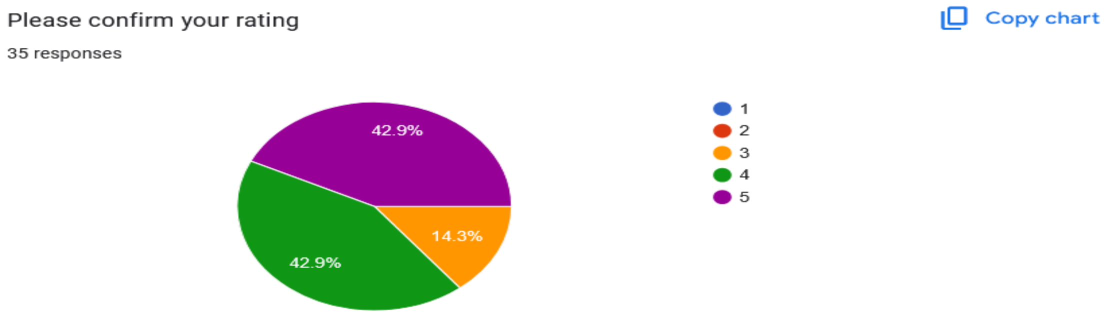
GOAL: An average score of 4 or higher is considered good.

ACHIEVED: Most of the users have rated the image of rating 4 where 1 being the lowest and 5 being the highest

This is what we have asked our user and peers:

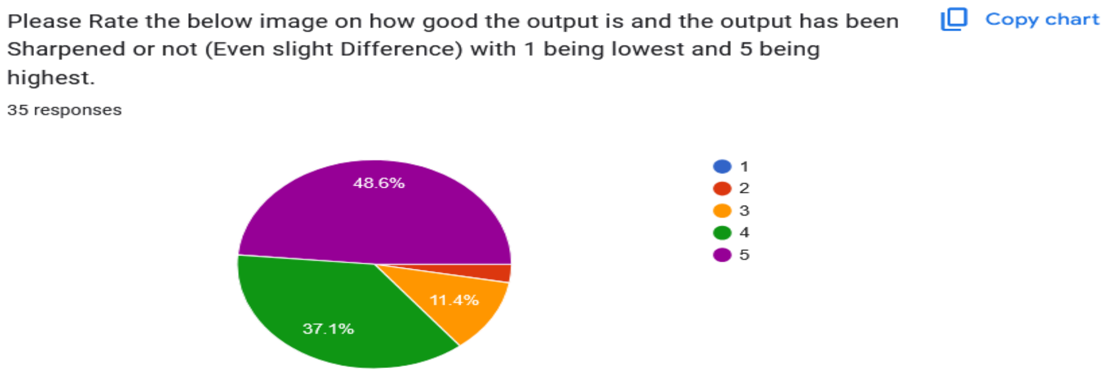


Please Rate the below image on how good the output is and the output has been Sharpened or not (Even slight Difference) with 1 being lowest and 5 being highest.



Please confirm your rating

35 responses



Please Rate the below image on how good the output is and the output has been Sharpened or not (Even slight Difference) with 1 being lowest and 5 being highest.

35 responses

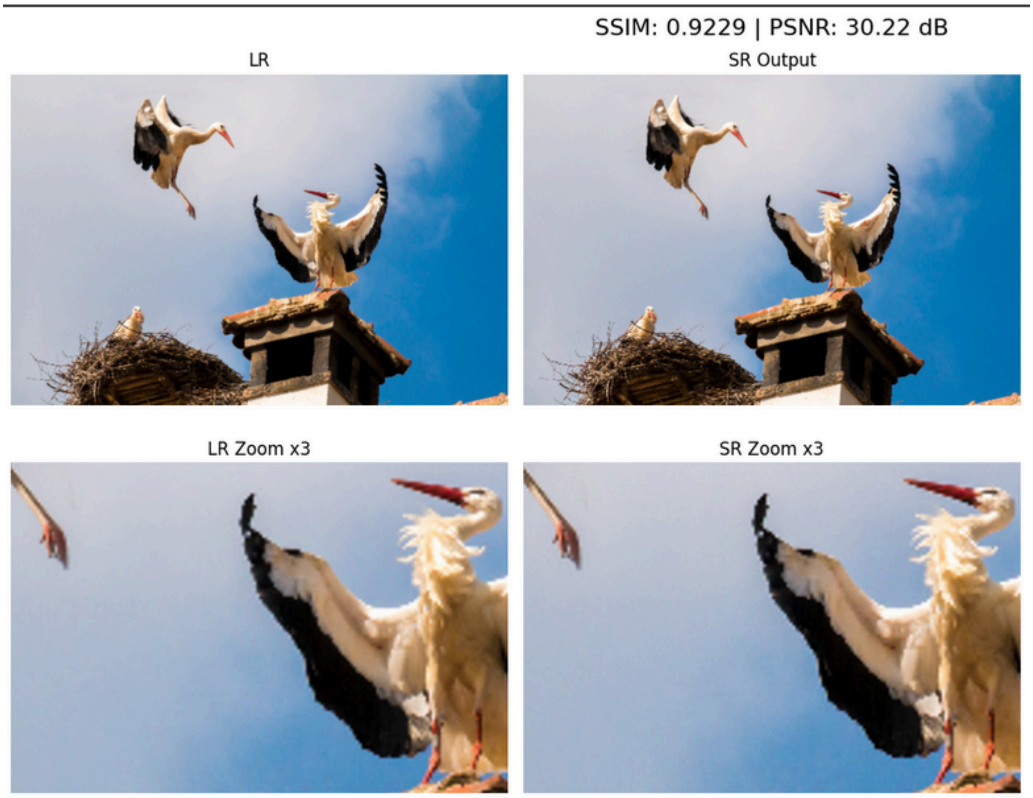
As we can, most of the people have voted 4 and 5 as ratings which shows out student model is working well while maintaining our original image accuracy.

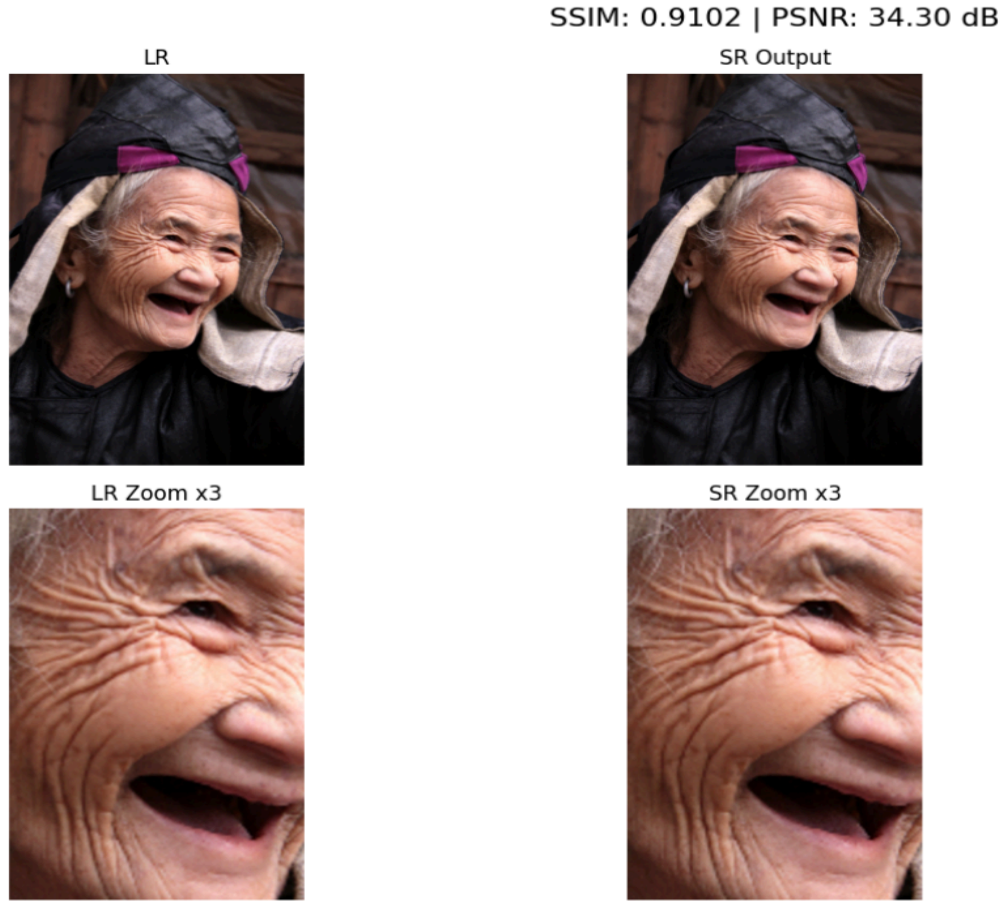
7. RESULT AND ANALYSIS:

7.1 QUALITATIVE ANALYSIS:

The student model produced noticeably sharper images compared to degraded inputs, especially around edges, text, and facial features. It effectively reduced blur and preserved the natural appearance of the images without introducing artifacts. While the teacher model performed slightly better, the student model delivered visually comparable results with much faster processing — making it suitable for real-time use. Human evaluators also rated the student model outputs positively in side-by-side comparisons.

Lets see some image comparisons now:





As you can see the edge definition from the output obtained from our student model is very well and defined much sharper than the low resolution images we have uploaded.

7.2 QUANTITATIVE ANALYSIS:

Metric	**Teacher Model**	**Student Model (Projected)**
SSIM (Average)	**0.94**	**0.90 and above**
Model Size	**38MB**	**1.8MB**
MOS (Avg User Rating)	**4 and above**	**4 and above**
Training Time	**Pretained + Small changes**	**Over 20 hours**

8. APPENDIX:

8.1 Code Snippets / GitHub Link

The original source code for data preprocessing, training of model, and evaluation is organized and available in the the GitHub repository mentioned below:

🔗 **GitHub Repository:** https://github.com/abhi68402/Team-ClearSight

Notebooks for testing and training:

https://colab.research.google.com/drive/1jiEvWu2BkiQwd4AdPoYdw1gVuf4dFMg6?usp=sharing

9. CONCLUSION:

Through real-time image sharpening, this project offered a workable solution to the issue of image degradation in video conferences. Using a Student–Teacher framework based on knowledge distillation, we aimed to achieve a balance between high visual quality and efficient performance suitable for real-world applications.

The anticipated results show a strong potential for producing crisp, high-quality outputs.This project bridges the gap between deep learning research and practical video communication needs, offering a lightweight, scalable, and intelligent solution for real-time image sharpening. It demonstrates how targeted machine learning techniques can meaningfully enhance everyday digital experiences—one frame at a time.

Project has been submitted by

Gompa harshit abhinav, B.abhinav, S.Lalasa

Under the guidance of Anupama.A Ma’am