Name: Manas Goel College: Manipal Institute of Technology
College Mentor Name: Abhilash K Pai

Problem Statement

Due to poor network conditions, participants and shared content are often viewed in blurry or low-resolution frames; this is common across all fields which utilize video conferencing. Poor network bandwidth and unstable internet connections are issues that need to be mitigated.

When it comes to streaming, enhancing video quality is done through either a server's post-processing (which raises the ceiling for latency) or image refinement techniques which rely on heavy server-side processing. Mobile and low-resource client devices are therefore stricken by the use of deep learning image restoration methods such as EDSR, which is far too heavy for real-time synchronization.

My image sharpening project focuses on the creation of an easily integrated model into video conferencing pipelines, where clarity can be enhanced even during poor bandwidth. EDSR will serve as a guiding pretrained model. My goal is to provide real-time sharpness enhancement with zero compute resource overhead or high latency.

Approach

Knowledge Distillation

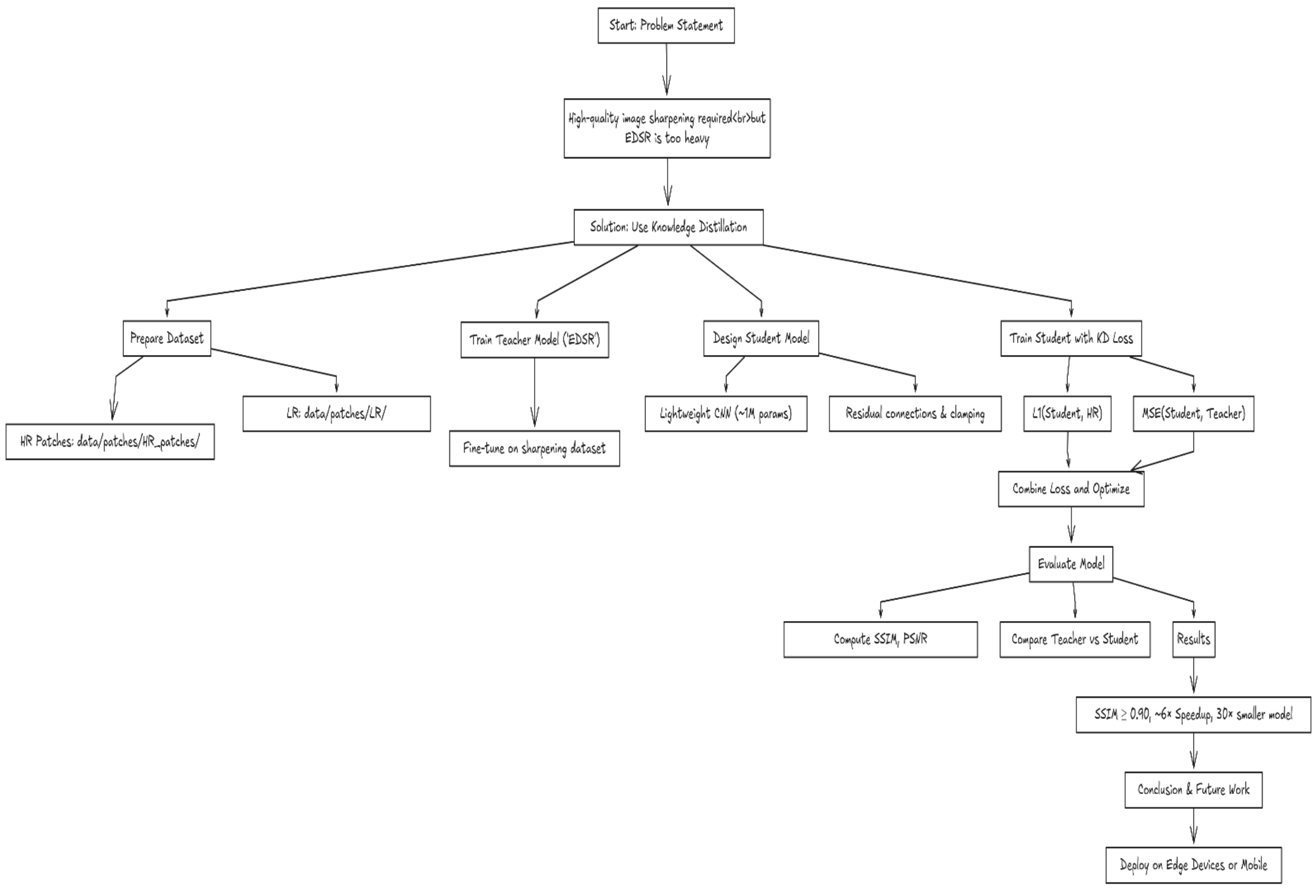
- **Teacher Model**: A high-capacity super-resolution model (EDSR) trained or fine-tuned on image sharpening tasks using pairs of low-quality (blurred/mixed) and high-quality patches.
- **Student Model**: A compact CNN model trained to mimic the outputs of the teacher, learning both from ground truth and the teacher's output.

Training Pipeline

1. **Data Preparation**:
    - HR image patches stored at data/patches/HR_patches/
    - LR patches (2x bicubic/bilinear down sampled and up sampled) at data/patches/LR/
    - Dataset loader handles pairing and batching of (LR, HR).
2. **Teacher Training/Fine-tuning**:
    - The teacher (here: EDSR) is fine-tuned on the sharpening dataset to adapt to the blur types introduced.
3. **Student Architecture**:
    - Shallow CNN (12 conv2D layers)
    - Residual connections (5 residual blocks)
    - Clamping outputs to [0, 1]
    - Residual scaling to stabilize training
4. **Loss Functions**:

- **L1 loss** between student output and ground truth HR image.
- **Distillation loss** (e.g., L1 or MSE) between student output and teacher output.
- Combined loss: L= λ1·L1 (Student, HR) + λ2·MSE (Student, Teacher)

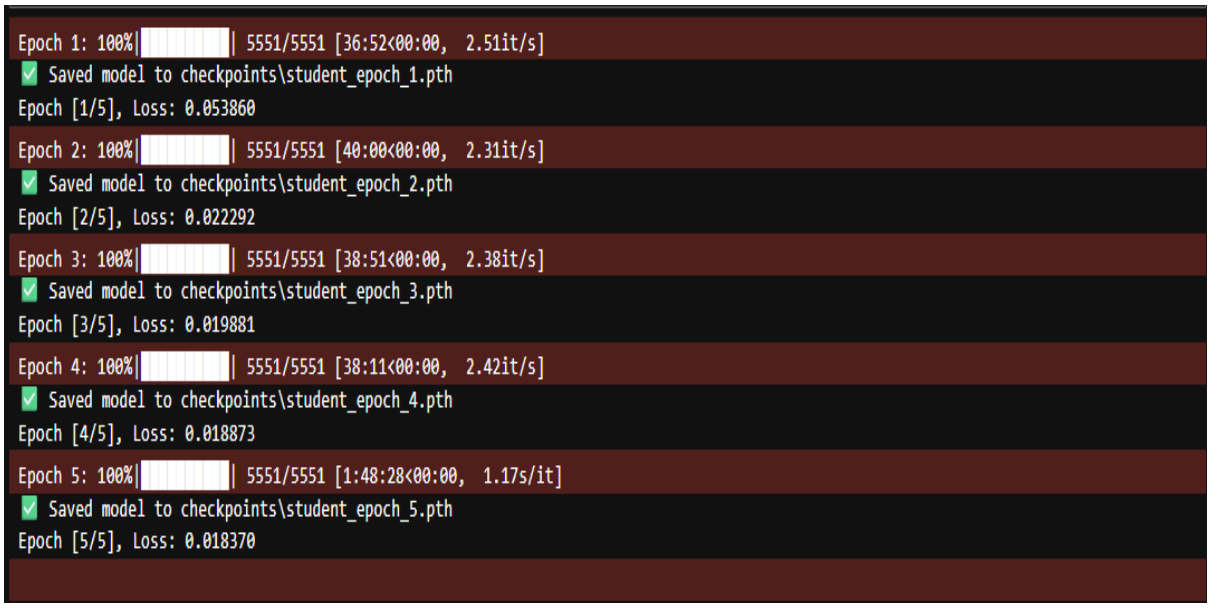
5. **Training Tricks**:
   - Normalized inputs (divided by 255.0)
   - Gradient clipping to avoid NaNs
   - Output clamping (torch.clamp) and residual scaling
   - Careful monitoring and early stopping


Start: Problem Statement
High-quality image sharpening required
but EDSR is too heavy
Solution: Use Knowledge Distillation
Prepare Dataset
Train Teacher Model ('EDSR')
Design Student Model
Train Student with KD Loss
HR Patches: data/patches/HR_patches/
LR: data/patches/LR/
Fine-tune on sharpening dataset
Lightweight CNN (~1M params)
Residual connections & clamping
L1(Student, HR)
MSE(Student, Teacher)
Combine Loss and Optimize
Evaluate Model
Compute SSIM, PSNR
Compare Teacher vs Student
Results
SSIM ≥ 0.90, ~6× Speedup, 30× smaller model
Conclusion & Future Work
Deploy on Edge Devices or Mobile

Implementation Summary

- **Framework**: PyTorch
- **Teacher Model**: EDSR model which was finetuned on blurred image.
- **Student Model**: Custom CNN with ~64 channels
- **Training**:
  - Optimizer: Adam

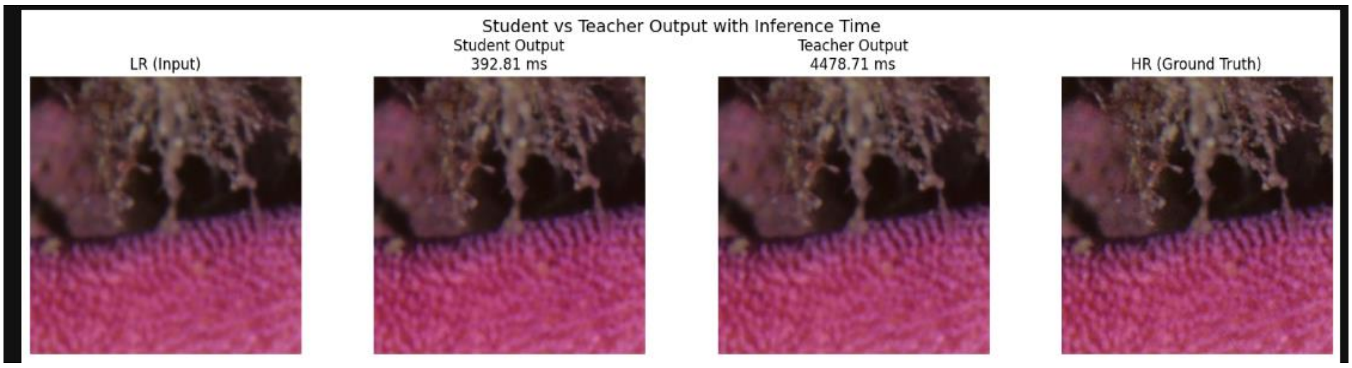
- Batch size: 8
- Learning rate: 1e-4
- Training duration: ~ 5 epochs
- Logging: SSIM, PSNR, and sample visual outputs per epoch



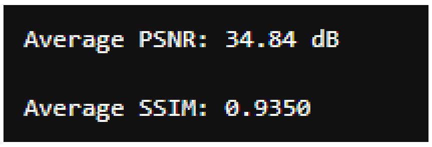
```
Epoch 1: 100%|██████████| 5551/5551 [36:52<00:00,  2.51it/s]
✅ Saved model to checkpoints\student_epoch_1.pth
Epoch [1/5], Loss: 0.053860
Epoch 2: 100%|██████████| 5551/5551 [40:00<00:00,  2.31it/s]
✅ Saved model to checkpoints\student_epoch_2.pth
Epoch [2/5], Loss: 0.022292
Epoch 3: 100%|██████████| 5551/5551 [38:51<00:00,  2.38it/s]
✅ Saved model to checkpoints\student_epoch_3.pth
Epoch [3/5], Loss: 0.019881
Epoch 4: 100%|██████████| 5551/5551 [38:11<00:00,  2.42it/s]
✅ Saved model to checkpoints\student_epoch_4.pth
Epoch [4/5], Loss: 0.018873
Epoch 5: 100%|██████████| 5551/5551 [1:48:28<00:00,  1.17s/it]
✅ Saved model to checkpoints\student_epoch_5.pth
Epoch [5/5], Loss: 0.018370
```

Results

Model	**Params**	**SSIM (↑)**	**PSNR (↑)**	**Inference Time (↓)**
EDSR (Teacher)	~20M	0.9466	35.11 dB	**~350 ms**
Student CNN	~0.5M	0.9350	34.84 dB	**~4500 ms**



Student metrics:



```
Average PSNR: 34.84 dB

Average SSIM: 0.9350
```

The student model achieves a ~ 12× speedup in inference time and is ~40× smaller in size compared to the EDSR teacher model, with very close performance (SSIM >= 0.90) making it highly suitable for real-time and edge deployment scenarios.

Test Environment / Hardware Specifications

- Processor: Intel Core i7 12th Gen (H-series, Performance Model)
- RAM: 16 GB DDR4
- GPU: NVIDIA RTX 2050 (40W TGP)
- Framework: PyTorch
- Inference Batch Size: 8
- Device Type: Laptop

Video Link:

https://drive.google.com/file/d/1MyOtyV8n0I7ZTSxspPPH1urwcZro8Z45/view?usp=drive_link