Problem Statement

Title: Image Sharpening using Knowledge Distillation

- Image sharpening is crucial in fields like medical imaging, surveillance, and photography.
- High-end deep learning models like Restormer offer excellent sharpening performance.
- However, these models are computationally heavy and slow.
- They require large memory, powerful GPUs, and high power consumption.
- Such models are impractical for real-time use on edge devices or mobile platforms.
- Deploying them in low-resource environments is not feasible.
- There's a clear gap between model performance and deployment efficiency.
- This creates a need for lightweight models that can still deliver good sharpening results.

Unique Idea Brief (Solution)

- We introduce a knowledge distillation-based approach for image sharpening
- A powerful model (Restormer) is used as the teacher to guide training
- A lightweight CNN acts as the student model to learn from the teacher’s output
- The student model is trained on blurred images using the teacher’s sharpened outputs
- This approach removes the need for high-resource inference at deployment time
- The student model is significantly faster and uses fewer resources
- It provides visually comparable results without depending on ground truth
- The final model is suitable for real-time, edge, or low-power device deployment

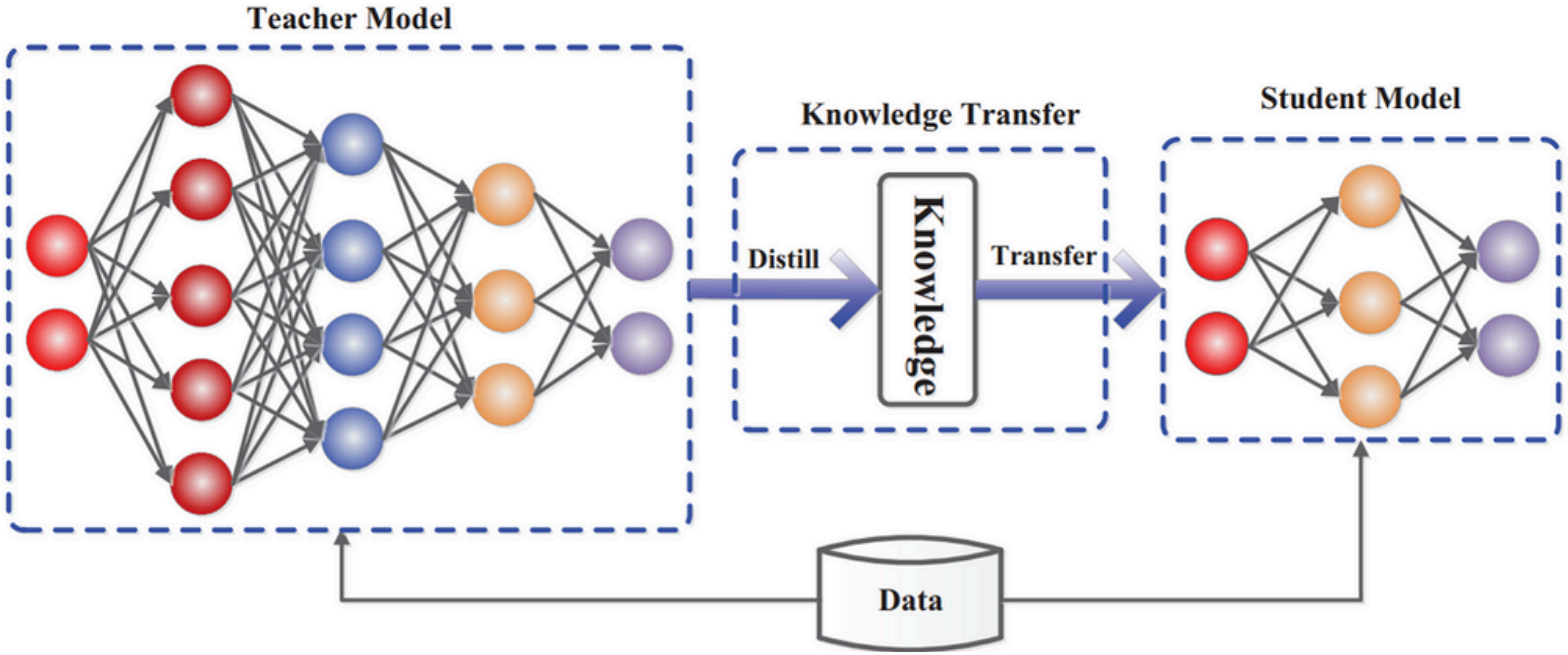
Features Offered

- Knowledge distillation technique used for training instead of traditional supervision
- Student model learns from teacher model outputs without needing ground-truth sharp images
- Lightweight CNN architecture designed for speed and efficiency
- Significantly reduced computational cost compared to large models like Restormer
- Trained on blurred-sharp image pairs with teacher-generated outputs
- Supports real-time or low-power deployment scenarios
- Produces sharpened images that are visually comparable to teacher output
- Performance evaluated using PSNR and SSIM metrics

Process flow

- Blurred images are collected and used as input to the pipeline
- These images are passed through the pretrained Restormer model acting as the teacher
- The teacher generates sharpened output images from the blurry inputs
- A lightweight student CNN model is given the same blurred inputs
- The student learns to mimic the output of the teacher through loss comparison
- No ground-truth sharp images are used during training
- Once trained, the student model alone is used for inference
- The final student model performs efficient image sharpening with reduced resources

Architecture Diagram



Technologies used

- Python : Core language used for development and scripting
- PyTorch : Used to build and train both teacher and student models
- Torchvision : Helped with image transformations and data loading utilities
- PIL : Used for reading, processing, and displaying image files
- OpenCV : Supported additional image preprocessing tasks if needed
- Skimage : Used to calculate PSNR and SSIM for performance evaluation
- Matplotlib : Used to visualize input and output images during testing
- VS Code : Primary code editor and development environment
- Restormer GitHub Repo : Source of pretrained teacher model and architecture
- Virtual Environment : Isolated environment used for managing project dependencies
- Frontend : Not included; this project is entirely backend-oriented

Team members and contribution:

Prathima :
Contributed to dataset preparation, model integration, and testing
Worked on organizing files, running evaluations, and visualizations
Likhitha :
Helped implement the student CNN model and load the Restormer teacher model
Managed training scripts, losses, and created visual outputs and GitHub uploads
Mounika :
Worked on image preprocessing, training loop debugging, and evaluation metrics
Supported report and idea slide preparation

Conclusion

- The project implemented image sharpening using knowledge distillation
- A lightweight student CNN was trained to mimic the output of the Restormer model
- The model was trained on 1029 blurred-sharp image pairs
- No ground-truth sharp images were used during training
- The student model produced visually comparable results to the teacher
- Achieved PSNR of 12.91 dB and SSIM of 0.4584
- The project is fully backend-oriented with no frontend implementation
- Final model is efficient and suitable for real-time deployment