Final Project Report

Knowledge Distillation-Based Image Sharpening Using Teacher-Student CNN Architecture (with Ground Truth as Teacher)

1. Objective

The aim of this project is to develop a lightweight student CNN model to perform image sharpening, where the supervision comes directly from ground truth sharp images. This approach avoids the need for pretrained models and instead leverages self-supervised learning using ground truth as a 'fake teacher'.

2. Learning Framework: Teacher-Student (Ground Truth Supervised)

Teacher: Ground truth sharp images, not a trained or pretrained network.
Student: StudentCNN_v3, trained to mimic the sharp output.

3. Workflow

Dataset: GoPro Dataset

Structure:
E:/image/data/GOPRxxxx_xx_xx/{blur, sharp, blur_gamma}

Training Input: blur_gamma
Target/Teacher (Supervision): sharp

4. Architecture: StudentCNN_v3

A deeper CNN with skip connections and dilation, designed for sharpening with reduced artifacts.

Key Layers:
Input → Conv(3→64) → ReLU
→ Conv(64→128) → ReLU
→ Conv(128→128, dilation=2) → ReLU
→ Residual connection
→ Conv(128→64) → ReLU
→ Conv(64→3) → Sigmoid

Skip Connection: Improves gradient flow
Dilation: Enlarges receptive field
Sigmoid: Normalizes output to [0, 1]

5. Loss Function: Combined Perceptual and Edge Loss

combined_loss = α * MSE + (1 - α) * (1 - MS-SSIM) + edge_weight * Edge_Loss

MSE (Mean Squared Error) – pixel-level fidelity
MS-SSIM (Multi-Scale SSIM) – perceptual quality
Edge Loss – L1 loss between Sobel edge maps (captures texture)

Hyperparameters:
α = 0.8
edge_weight = 1.0

6. Training Details

Epochs	30
Batch Size	2
Optimizer	Adam (lr=1e-4)
Loss Function	Combined (MSE + MS-SSIM + Edge)
Framework	PyTorch
Hardware	CUDA (GPU)

Output Model: sharpen_model.pth

7. Evaluation

Training Loss (final): ~0.1198
Test Loss: ~0.1234
PSNR (estimated): ~29.2 dB
SSIM (estimated): ~0.87

Visual Quality:
- Noticeable sharpening over blur_gamma
- Edges are more defined
- Structural similarity maintained well

8. Inference & Output

Script: run_inference.py

Functionality:
- Loads trained model
- Runs inference on blur_gamma images
- Saves side-by-side comparisons of:
  - Input (Blur Gamma)
  - Output (Sharpened)
  - Target (Ground Truth Sharp)

Output directory: outputs_sharpen/

9. Advantages

- Lightweight and fast model
- Doesn't rely on any pretrained external models
- Enhanced edge sharpness using edge-aware loss
- Good perceptual quality (thanks to MS-SSIM)
- Ideal for real-time or resource-constrained environments

10. Limitations

- Not trained on diverse datasets—may not generalize to all blur types
- No adversarial component (e.g., GANs) for realism
- Slight loss in fine detail on extreme blur cases

11. Tools & Libraries Used

- PyTorch – Training, model definition
- torchvision – Data transforms, image saving
- PIL / matplotlib – Image visualization
- pytorch-msssim – MS-SSIM loss computation
- tqdm – Training progress bar

12. Project Status: Finalized

[x] Training Completed
[x] Loss Curve and Evaluation Visualized
[x] Inference Utility Implemented
[x] Final Outputs Saved for Presentation
[x] No pretrained teacher used — Ground truth supervision only

13. Team Members & Contributions

Team Name: True Sight

Team Member 1: Sabilah S. (RRN: 230171601159)

- Designed the project pipeline and overall workflow
- Implemented the CNN architecture (StudentCNN_v3)
- Integrated perceptual and edge loss formulations
- Led report writing and structuring

Team Member 2: Viswanathan (RRN: 2301701196)

- Implemented data loading, preprocessing and model training loop
- Set up GPU memory management and optimization techniques
- Handled training visualization and logging
- Structured the complete PyTorch pipeline and architecture integration

Team Member 3: Syed Thufel Syed Wahid (RRN: 230171601189)

- Built the inference module for visual comparison
- Automated output saving and testing interface

- Reported evaluation metrics like PSNR, SSIM
- Managed final result curation and demo setup