Image Sharpening using a ResNet-based Architecture and Knowledge Distillation

Jerit Reji, Joel Mathew Samuel, Sebastian Abraham

Saintgits Group of Institutions, Kottayam, Kerala

Abstract

This report details the implementation of an image sharpening system using knowledge distillation. A deep ”teacher” model, based on a Residual Network (ResNet), is trained on the DIV2K dataset to restore degraded images. The training objective for this model is a composite loss function combining Structural Similarity (SSIM), Mean Squared Error (MSE), and a VGG-based Perceptual Loss. Subsequently, a smaller, computationally efficient ”student” Convolutional Neural Network (CNN) is trained to mimic the teacher's output using a distillation loss. The final student model's performance is evaluated against the teacher's, demonstrating the effectiveness of transferring knowledge to a compact architecture for practical applications.

Keywords: Image Sharpening, Deep Learning, Computer Vision, ResNet, Knowledge Distillation, SSIM, Perceptual Loss, PyTorch, Model Optimization, SwinIR, EDSR.

1 Introduction

Restoring detail in images degraded by blur, noise, or downscaling is a fundamental problem in computer vision. While deep learning models can learn complex restoration transforms, state-of-the-art architectures are often too large and computationally intensive for practical deployment on consumer devices. This creates a conflict between model performance and efficiency.

This project addresses this challenge by implementing a teacher-student framework based on knowledge distillation. The primary goal is to transfer the image restoration capability of a large, high-performance ”teacher” network into a much smaller and faster ”student” network. This approach aims to produce a final model that is both effective at image sharpening and suitable for resource-constrained environments.

2 Methodology for Selected Model

2.1 Dataset and Augmentation

The DIV2K dataset, containing high-resolution images, was used as the basis for training. A custom data pipeline was created to generate pairs of degraded and ground-truth images on-the-fly using the `albumentations` library. The degradation transform applies a random combination of the following:

- **Blur:** Gaussian or Motion blur, with a kernel size between 3 and 7.
- **Downscaling:** Resizing the image to between 60% and 80% of its original size.
- **Noise:** Adding Gaussian noise with a variance between 10 and 50.

2.2 Final Model Architecture

The core of the finalized project is a teacher-student framework using a ResNet-based teacher.

2.2.1 Teacher Model: ResNetSharpen

The teacher is a ResNet-inspired architecture designed for high-capacity learning. It features an input convolution layer mapping 3 to 64 channels, followed by 8 residual blocks, and an output convolution layer mapping 64 back to 3 channels. A global residual connection adds the input image to the network's output, allowing the model to focus on learning the sharpening details rather than reconstructing the entire image.

2.2.2 Student Model: StudentCNN

The student is a lightweight CNN with minimal layers, designed for fast inference. It consists of two 3x3 convolutional layers and also employs a global residual connection.

2.3 Final Loss Functions

2.3.1 Teacher Training Loss

The teacher model is trained using a composite loss function to balance pixel accuracy with perceptual quality. It is a weighted sum of an SSIM/MSE loss and a VGG-based Perceptual Loss, with weights of `ssim_w=0.8` and `perceptual_w=0.01`. The Perceptual Loss compares high-level features from a pre-trained VGG19 network to better align with human visual assessment.

```python
class TotalLoss(nn.Module):
    def __init__(self, ssim_w=0.8, perceptual_w=0.01):
        super(TotalLoss, self).__init__()
        self.ssim_w = ssim_w
        self.perceptual_w = perceptual_w
        self.mse = nn.MSELoss()
        self.ssim = lambda p, t: 1 - ssim_loss(p, t, data_range=1.0)
        self.perceptual = VGGPerceptualLoss()

    def forward(self, pred, target):
        ssim_mse_loss = self.ssim_w * self.ssim(pred, target) + \
                        (1 - self.ssim_w) * self.mse(pred, target)
        perceptual_loss = self.perceptual(pred, target)
        return ssim_mse_loss + self.perceptual_w * perceptual_loss
```

Listing 1: Composite Loss for Teacher Training

2.3.2 Knowledge Distillation Loss

The student model is trained using a distillation loss. This loss is a weighted average of two components: the mean squared error between the student's and the teacher's outputs, and the mean squared error between the student's output and the ground-truth image. The weighting factor `alpha` is set to 0.75, placing more emphasis on mimicking the teacher.

```
def distillation_loss(s_out, t_out, target, alpha=0.75):
    loss_teacher = nn.MSELoss()(s_out, t_out)
    loss_gt = nn.MSELoss()(s_out, target)
    return alpha * loss_teacher + (1 - alpha) * loss_gt
```

Listing 2: Knowledge Distillation Loss Function

3 Final Results and Discussion

The ResNet-based teacher and corresponding student model were trained using the Adam optimizer with a learning rate of 1e-4 and a Cosine Annealing scheduler. The final performance was evaluated on a held-out validation set using the Structural Similarity Index (SSIM) as the primary metric.

The final quantitative results for the selected models are as follows:

- **Final Teacher Model SSIM:** 0.6312
- **Final Distilled Student SSIM:** 0.6061

The distilled student model achieves over 96% of the teacher model's performance in terms of SSIM score. This demonstrates a highly successful knowledge transfer, as the significantly smaller student network retains the vast majority of the larger model's capability. Qualitative analysis, shown in Figure 1, supports this conclusion. The images produced by the student model show a clear visual improvement over the degraded inputs and are nearly indistinguishable from those produced by the much larger teacher model.

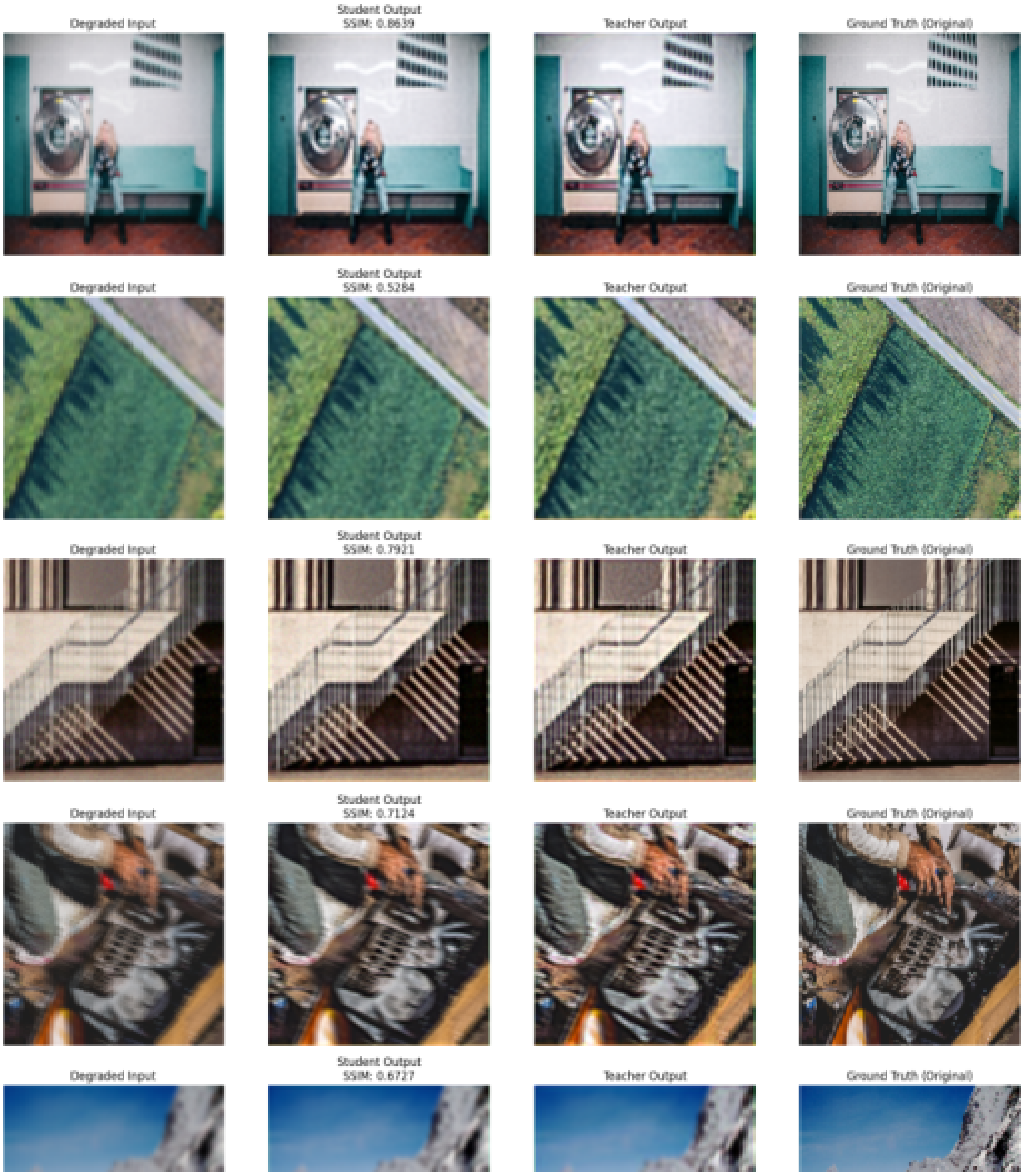

Figure 1: Visual comparison of model outputs on the validation set. Each row shows the degraded input, the student's sharpened output (with SSIM score), the teacher's output, and the original ground truth.

4 Alternate Model Evaluations

During development, several architectures were evaluated as potential teachers before finalizing the ResNet-based approach. This section summarizes the findings from those explorations.

4.1 Evaluation of DnCNN

An initial evaluation was performed using a DnCNN (Denoising Convolutional Neural Network) architecture as the teacher.[1] The model was trained using only Mean Squared Error (MSE) as the loss function. The DnCNN teacher achieved a final average SSIM of 0.7327, and its distilled student reached an even higher SSIM of 0.7386. Despite these strong quantitative scores, this approach was abandoned. MSE loss is known to produce overly smooth images that lack fine textural detail, and the high SSIM score did not correlate with superior visual quality. This highlighted the necessity of a loss function that better captures human perception.

4.2 Evaluation of EDSR as Teacher

The EDSR (Enhanced Deep Residual Networks) model, a popular architecture for super-resolution, was also evaluated as a teacher.[2] The `edsr_baseline_x4` pre-trained model was used. The distilled student achieved a final SSIM score of only 0.4514. This result was significantly lower than that of other tested approaches and was deemed insufficient for the project's goals. The primary challenges were attributed to GPU memory constraints and the difficulty of distilling knowledge from a model highly specialized for super-resolution to a more general sharpening task.

4.3 Evaluation of SwinIR as Teacher

Finally, we explored using SwinIR (Swin Transformer for Image Restoration), a state-of-the-art transformer-based model, as the teacher.[3] While this pipeline showed strong initial promise in terms of SSIM, it was ultimately impractical due to its significant computational demands. The process suffered from severe memory bottlenecks and prohibitively slow training cycles, which prevented the completion of a full training run. This experience solidified the decision to use a more balanced architecture.

[1] The code for this experiment is available at: github.com/jrtrj/ImageSharpening_KD/tree/dcnn

[2] The code for this experiment is available at: github.com/jrtrj/ImageSharpening_KD/blob/edsr

[3] The code for this experiment is available at: github.com/jrtrj/ImageSharpening_KD/tree/SwinIR

5 Conclusion

This project successfully developed and validated a knowledge distillation pipeline for image sharpening. The research journey involved a systematic evaluation of multiple advanced architectures—including DnCNN, EDSR, and SwinIR—as potential teacher models. This exploration revealed critical trade-offs between quantitative performance metrics (like SSIM), perceived visual quality, and computational feasibility. While architectures like SwinIR were powerful, they proved too resource-intensive for practical training. Conversely, a simple DnCNN trained on MSE produced high SSIM scores but lacked the visual fidelity essential for a high-quality restoration task.

The final architecture, a ResNet-based teacher trained with a composite perceptual loss, was chosen as the optimal solution. It provided the best balance of high-quality visual results and manageable training requirements. The knowledge from this teacher was successfully distilled into a lightweight student CNN, which recovered 96% of the teacher's performance (achieving a 0.6061 SSIM score) with a significantly smaller footprint. The project not only demonstrates a successful model compression but also underscores the importance of a pragmatic, iterative approach to model and loss function selection to achieve both high quality and practical utility in real-world applications.

Acknowledgements

We would like to express our sincere gratitude to the Intel® Unnati program for providing the opportunity and framework for this project. We also extend our thanks to our mentor Dr.Anju Pratap and institution, Saintgits Group of Institutions, for providing the necessary resources and academic support throughout the duration of this work.

References

[1] K. He, X. Zhang, S. Ren, and J. Sun, *Deep residual learning for image recognition*, in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016.

[2] G. Hinton, O. Vinyals, and J. Dean, *Distilling the knowledge in a neural network*, arXiv preprint arXiv:1503.02531, 2015.

[3] A. Paszke et al., *PyTorch: An Imperative Style, High-Performance Deep Learning Library*, in Advances in Neural Information Processing Systems, 2019.