"Image Sharpening using Knowledge Distillation"

An Internship Report submitted in partial fulfillment of the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

Submitted by

Team Name: VisionSharp AI

Chatti Ravindranath Tagore (VU22CSEN0600109)
K G S S Sampath (VU21CSEN500075)
Katuru Hrishikesh (VU22CSEN0500043)

@ Intel Unnati

May 2025-July 2025

Under the esteemed guidance of
Dr.Sesetti Anuradha
Associate Professor
Department of CSE

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
GITAM SCHOOL OF TECHNOLOGY
GITAM (Deemed to be University)

VISAKHAPATNAM
2025

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
GITAM SCHOOL OF TECHNOLOGY
GITAM (Deemed to be University)

DECLARATION

I hereby declare that the internship report entitled "**Image Sharpening using Knowledge Distillation**" is an original work done in the Department of Computer Science and Engineering, GITAM School of Technology, GITAM (Deemed to be University) submitted in partial fulfillment of the requirements for the award of the degree of B.Tech. in Computer Science and Engineering .The work has not been submitted to any other college or University for the award of any degree or diploma.

Date: 12/07/2025

Registration No.	**Student Name**
VU21CSEN0600109	Chatti Ravindranath Tagore
VU22CSEN0500075	K G S S Sampath
VU21CSEN0500043	Katuru Hrishikesh

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

GITAM SCHOOL OF TECHNOLOGY

GITAM (Deemed to be University)

CERTIFICATE

This is to certify that the internship report entitled "**Image Sharpening using Knowledge Distillation**" is a bonafide record of work carried out by Chatti Ravindranath Tagore (VU22CSEN0600109), K G S S Sampath, (VU21CSEN0500075), Katuru Hrishikesh(VU21CSEN0500043)

Date : 12/07/25

Internship Mentor
Dr. Sesetti Anuradha,
Associate Professor, CSE, GST,
GITAM, Visakhapatnam

Head of the Department
Dr. Gondi Lakshmeeswari,
Associate Professor, CSE, GST,
GITAM, Visakhapatnam

ACKNOWLEDGEMENT

This Internship report has been prepared as a part of Academic requirements of the Department of Computer Science and Engineering at the GITAM School of Technology. We would like to express our sincere gratitude to the department for the invaluable resources and infrastructure that facilitated our research and learning.

We are deeply thankful to **Dr. Sesetti Anuradha**, Associate Professor from the Department of CSE, GST, GITAM, Visakhapatnam for her constant support, insightful guidance, and motivation throughout this endeavour. Her expertise has been crucial to the successful completion of this work.

We wish to convey our heartfelt thanks to the **Intel Team** and **Dr. Katragadda Sharada** ,Associate Professor from the Department of CSE,GST,GITAM,Visakhapatnam for providing us the valuable opportunity to work on Intel Projects.

Our special thanks to **Dr. Gondi Lakshmeeswari**, Head of the Department,Computer Science and Engineering, GST, GITAM,Visakhapatnam for her encouragement and unwavering support, which played a vital role in helping us achieve our goals for this project.

Lastly, we would like to express our appreciation to the management of GITAM (Deemed to be University) for creating an academic environment that fosters innovation and learning, allowing us to work towards achieving the objectives of this project.

TABLE OF CONTENTS

1. PROBLEM STATEMENT

Image sharpening using knowledge distillation.

Loss of information or data is a significant issue in this data driven era. It occurs due to reasons such as poor internet connectivity and limited bandwidth which results in reduced image quality and underwhelming visual experience. This can cause content quality degradation especially in the department of education and IT. This ignites a need to come up with a solution that can enhance image sharpness and quality. The computational overhead that is required to train a model should also be taken into consideration.

2.OBJECTIVE

Considering the issues and problems discussed in the above problem statement there is one approach that is significantly more efficient than the classic CNN model approach which is Knowledge Distillation. In this approach a high performing trained teacher model guides the training of a lightweight student model. The objective of this student model is to restore sharpness of degraded images and process 1920x1080 images efficiently. This approach also aids in reducing excessive computational overhead as the parameters in the pre-trained teacher model are used to train the light-weight student model.

3.EXPECTED OUTCOMES:

- Usage for bilinear downscaling to simulate the degraded images that are encountered in real world scenarios.
- A light-weight CNN as the student model trained using Knowledge Distillation technique. The teacher model used in this case is the Restormer model.
- The primary metric to determine and assess the quality of the restored images is SSIM metric score(Structural Similarity Index).
- The solution that is to be built must demonstrate competitive SSIM scores with the average score of at least 85% to be a feasible option for real life scenario applications.

4.DELIVERABLES

The following are the main deliverables of this project, which focuses on image restoration using a lightweight student CNN model trained via knowledge distillation method(Teacher-Student model):

1.Main Source Code:

- A fully functional Python script that implements the following:
    1. A lightweight convolutional neural network(CNN) model for image restoration
    2. A Training pipeline using a teacher-student distillation framework, where the student learns to mimic a pretrained teacher model's outcome
    3. A restoration and evaluation function that:
        a. After the student model has been trained, restoration and evaluation is applied to validation dataset to measure the effectiveness of the student model
        b. The valid dataset and its corresponding degraded images are first loaded into the local directory.
        c. The function restoration and evaluation takes the student path, ground truth and blurred images as input which then restores the images
        d. We have the option to download the restored images
    4. Calculates SSIM scores between the ground truth images and restored images.

2.Trained Student Model

- A lightweight student convolutional neural network was trained for 10 epochs. The student model tries to mimic the outcomes of a teacher model(Restormer).

- After the training has been completed. The model was saved as student_model.pth

- The student_model.pth saves all the weight and other parameters and is the hearth of the model.

3.Performance Metrics:

- Achieved an average SSIM score of 0.9023 for the valid dataset which consists of 200 images. This score indicates a high structural similarity between the restored and ground truth images.

- Individual SSIM scores of each image have been printed during the evaluation for transparency

---

4.Restored Images Output:

- The restored images can be downloaded or viewed for qualitative comparison with ground truth images.

5.TEAM MEMBERS AND RESPONSIBILITIES

Team Name : VisionSharp AI

Team Member-1: Chatti Ravindranath Tagore

- Finding research papers on teacher models.
- Implementation and evaluation of Evaluation metrics.
- Data preprocessing(inputs) and documentation.
- Design and implementation of student model.

Team Member-2: K G S S Sampath

- Design and implementation of student model.
- Data preprocessing(Slicing)
- Implementation of Knowledge Distillation.
- Testing of the Teacher model.
- Backward propagation of student model training.

Team Member-3: Katuru Hrishikesh

- Dataset collection.
- Design and implementation of student model.
- Data preprocessing.(Bilinear)
- Testing of the Teacher model.
- Forward propagation of student model training.

All Team Together:

Report Writing.

Video Making.

6.TECH STACK

1.Programming Language:

Python is the programming language that is used for all data preprocessing techniques, model training, and for evaluation metrics.

2.Libraries and Frameworks:

a.Deep Learning:

PyTorch :

- PyTorch, an open source machine learning library used for building lightweight student CNN and to run interface with Restromer model.

torchvision :

- Provides the pretrained VGG16 model for the perceptual features extraction.

---

b.Image Processing:

PIL(Pillow):

- **PIL** is used mainly to load blurry images and resize them as input for the model.It is used to save the sharpened images after inference.

NumPy:

- Transforms PyTorch tensors into arrays for image evaluation and used to compute evaluation metrics such as SSIM.

3.**Evaluation:**

scikit-image:

- Used to Calculate SSIM scores during inferences that helps in understanding the similarity between ground truth images and student model's output.

Pytorch-msssim:

- Provides SSIM loss function during training to improve the perceptual quality of the student model.

4.Pretrained Model Teacher Model:

Restromer

- Restromer, cloned from Github provides high-quality and sharp images compared to other teacher models.
- Acts as a reference model to guide the light weight student model.

5.Data Handling:

Zipfile:

- Used to extract the dataset archives of ground truth images before training.

torch.utils.data.Dataset:

- Loads pairs of blurred images and ground truth sharp images.
- It is wrapped in a DataLoader for batching.

6.Evaluation and Loss:

L1 loss:

- Measures pixel difference between student output images and ground truth images.

SSIM loss:

- Measures and encourages structural similarity and perceptual quality.

Features Loss:

- VGG16 features are used to match perceptual content of student and teacher outputs.

7.Runtime Environment:

Google Colab:

- Provides an interactive environment for writing codes in cells and model training loops and for evaluation metrics and visualization codes.

GPU:

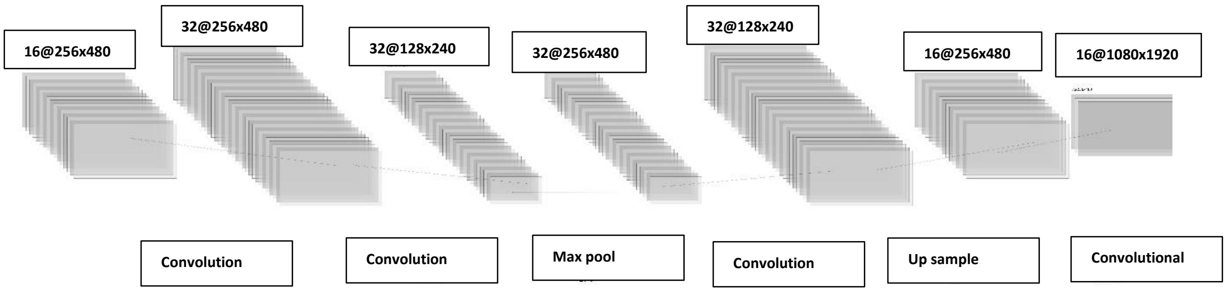
- Used for handling high resolution images and running deep networks such as Restromer and speeds up the model training.

8.Version Controls:

Git: The Restormer repository is cloned using Git and version control is managed.

Github: Used to host Restromer code and used as a repository for our project code.

7.SYSTEM ARCHITECTURE:



It is to be noted that in the above diagram both Convolutional layers and ReLU layers are combined.

This lightweight CNN specifically follows a encoder-middle-decoder architecture. Encoder extracts the feature representations and compresses spatial information. Middle layer adds a non linear transformation to the encoded result. This aids to increase expressive power. Decoder upscales the inputs back to original size for image reconstruction.The architecture consists of a total of 12 layers.

Layer 1:Convolutional layer(Encoder)

Inputs taken: B,3,64,64

Outputs delivered:B,16,64,64

The operation taking place here is 16 filters of size 3x3 applied on 3 channel input. Padding ensures that the spatial size remains the same. The role of this layer is to learn low level features like edges and textures.

Layer 2:ReLU layer(Activation function)

It is a non linear activation function. There is no change in shape in this layer. The purpose of this layer is to introduce non linear features and allow complex mappings.

Layer 3:Convolutional layer

Inputs taken:B,16,64,64

Outputs delivered:B,32,64,64

This layer doubles the feature channels. It learns more complex features from 16 channel input.

Layer 4:ReLU layer(Activation function)

It is a non linear activation function. There is no change in shape in this layer. The purpose of this layer is to introduce non linear features and allow complex mappings.

Layer 5: Pooling layer

Inputs taken:B,32,64,64

Outputs delivered:B,32,32,32

This layer reduces spatial resolution by half but preserves salient features.

Layer 6: Convolutional layer(Middle)

This layer does not change the shape of the input. It refines the compressed features.

Layer 7: ReLU layer(Activation function)

It is a non linear activation function. There is no change in shape in this layer. The purpose of this layer is to introduce non linear features and allow complex mappings.

Layer 8: Upsampling layer(Decoder)

This layer is used to upscale the spatial dimensions. No parameters like kernel_size and padding are not required in this step.

Layer 9:Convolutional layer

Input taken:B,32,64,64

Output delivered:B,16,64,64

This layer learns how to transform unsampled features into refined spatial features.

10: ReLU layer(Activation function)

It is a non linear activation function. There is no change in shape in this layer. The purpose of this layer is to introduce non linear features and allow complex mappings.

Layer 11:Convolutional layer

Inputs taken:B,16,64,64

Outputs delivered:B,3,64,64

This layer restores the inputs to the original channel dimension.

Layer 12: Sigmoid layer(Activation function)

This layer maps the outputs to be in the range of 0 to 1 which is suitable for normalized image pixels.

---

8. HOW THE CODE WORKS:

A.Preprocessing:

The preprocessing phase prepares the dataset before feeding it into the model. The steps are explained as follows:

1.Loading and Visualizing Raw Images:

- The code begins by collecting high-resolution(1920x1080) images from the dataset folder.A sample of the first few images is displayed using matplotlib.

2.Resizing Images:

- Each high-resolution image is resized to a fixed size of 464x256 pixels.This step ensures uniformity in input size. The resized images are saved in a separate directory for further preprocessing steps.

3.Slicing Each Image into crops using Numpy Slicing method:

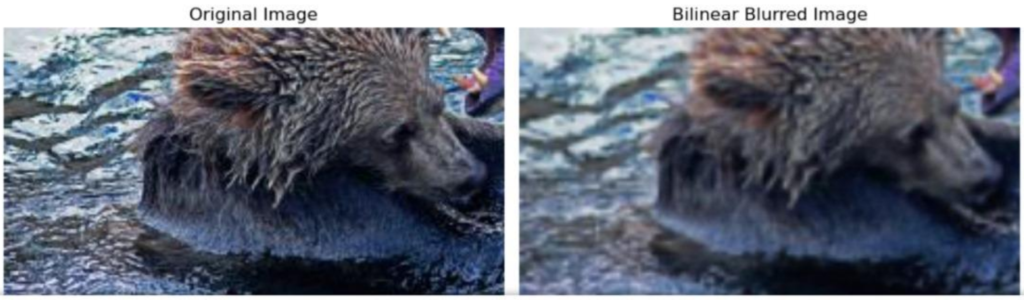
- To increase dataset variety and reduce the computational expenses,memory load , every resized image is split into four equal parts:
    - Top-left
    - Top-right
    - Bottom-left
    - Bottom-right

This is done by calculating the image sizes(height,width) and slicing the images accordingly. The resulting four crops are saved as separate images.

After the slicing procedure, there are 5200 images for training purposes.

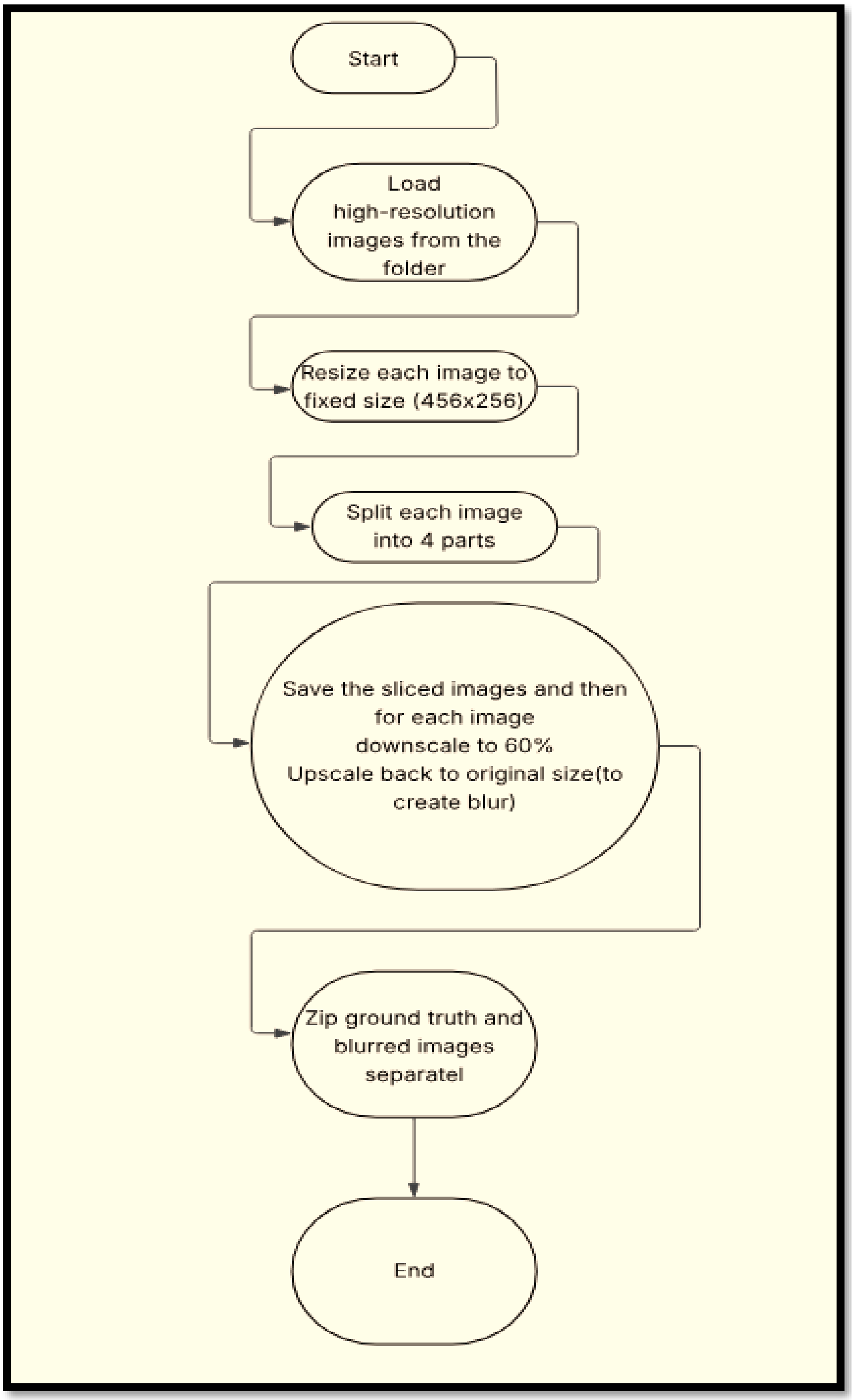
4.Generating blurred images using Bilinear Interpolation technique

- To simulate the degraded inputs of the original images, each original image is blurred using bilinear interpolation as follows:
- The image is first downscaled to 60% of its original size using bilinear interpolation.
- It is then upscaled back to the original image size, creating a blur effect.(As shown in the figure below):



- These blurred images are saved in a new folder.
- Ground truth images are the original images and blurred images are the training data.

Flowchart for Preprocessing:



B.Teacher Model:

For the teacher model in this project, we used the Restormer architecture, specifically the motion Deblurring variant. Restormer is a transform-based image restoration model known for its effectiveness in tasks such as deblurring, denoising, deraining and defocus deblurring.

The selected teacher mode(Restormer-Motion Deblurring) achieved an average SSIM score of 0.8838 on the training dataset.

C.Student Model:

The student model is a lightweight convolutional neural network. The training function takes ground truth images, blurred images, teacher model outputs and batch size as the input.

1.Forward propagation:

In the forward propagation, the blurred images are passed through three different layers. The layers are as follows:

1.Encoder
2.Middle layer
3.Decoder

The output of the student model are the restored images.

2.Backward propagation:

After generating the restored images, the model compares it with following:

- The teacher model's corresponding restored images which are used as soft labels
- The ground truth images (original images)

3.Loss function

The loss function which we named as distillation loss function in the code is a mixture of four different loss functions. These are as follows:

1. **L1 Loss function:**
   This loss function calculates the mean absolute error between the restored image generated by the student model and the corresponding ground truth image.

2.SSIM Loss function:

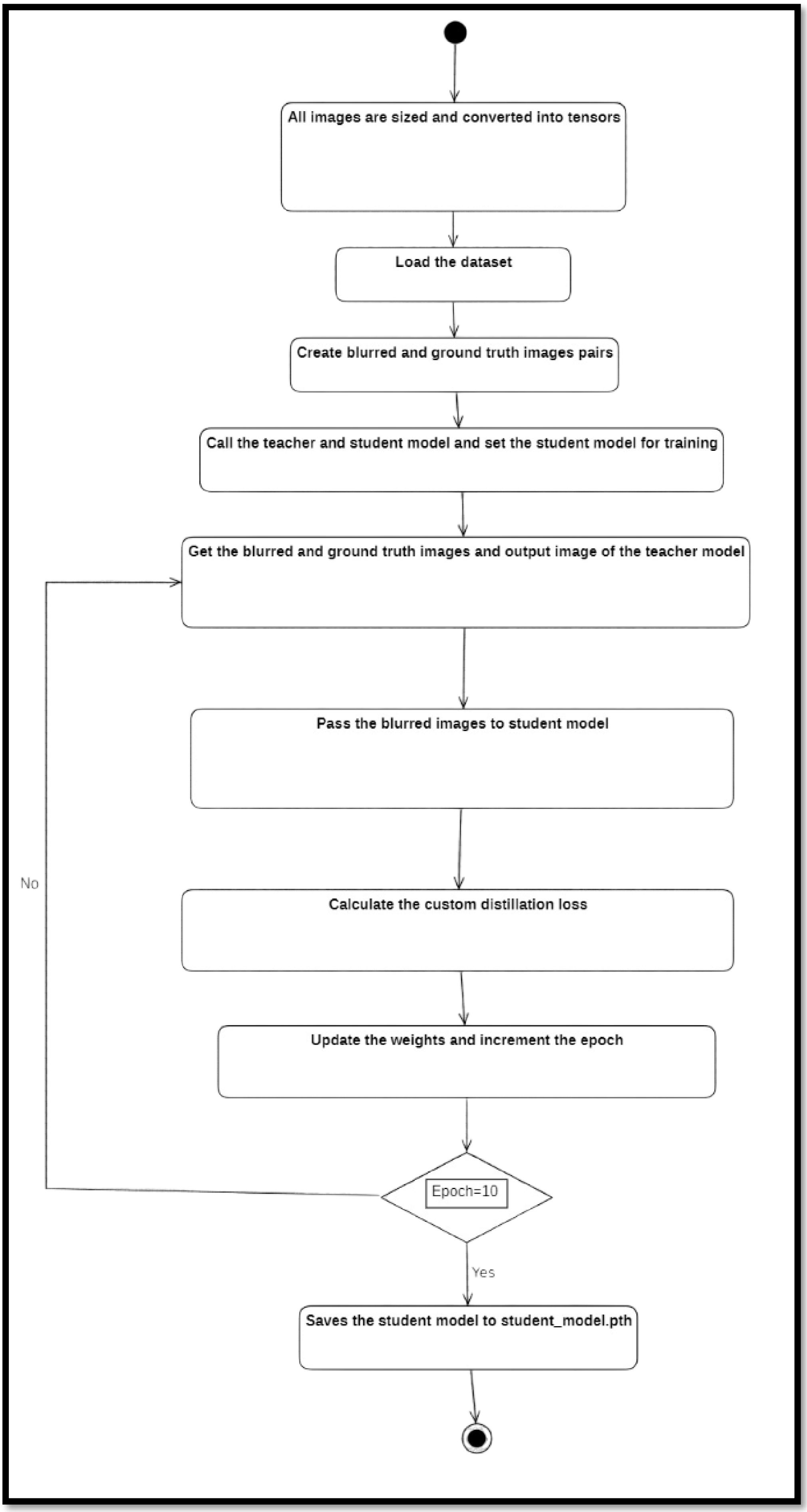
An unconventional way of using SSIM scores as a loss function is used here to increase the structural similarity between the restored images and ground truth images. It increases the perceptual quality of the image.

3.Feature Distillation Loss(VGG feature loss):

Feature maps are obtained from both the student and the teacher model outputs using a pre-trained VGG16 neural network.After the features maps has been obtained , Mean Squared Error is calculated between corresponding feature maps.This loss function helps the student model to mimic the outcomes of the teacher model.

4.Deep Feature loss(Lats VGG layer is used):

This loss focuses specifically on the deepest feature map. Again mse is calculated


All images are sized and converted into tensors
Load the dataset
Create blurred and ground truth images pairs
Call the teacher and student model and set the student model for training
Get the blurred and ground truth images and output image of the teacher model
Pass the blurred images to student model
Calculate the custom distillation loss
Update the weights and increment the epoch
Epoch=10
No
Yes
Saves the student model to student_model.pth

9.RESULTS SUMMARY

The student model achieved an SSIM score of 0.9023 outperforming the teacher model, which scored 0.8838. This shows that the student model has successfully learnt to restore images while maintaining perceptual accuracy,structural similarity.

10.APPLICATION DEMO VIDEO

https://drive.google.com/drive/folders/1ewjKEJROSeblKg2HJc5VQg4oKnw3RbEU?usp=sharing

12.CONCLUSION AND FUTURE SCOPE:

Conclusion:

By this project, our team successfully developed a light-weight image sharpening model for video conferencing using knowledge distillation.By employing a high-performance teacher model, Restromer and designing an effective student CNN, an average Structural Similarity Index (SSIM) score of 0.9023 is achieved.This indicates that our student model can restore the sharpness of the images and details in images degraded by conditions such as low bandwidth or poor network quality. This approach is utilized for balancing both visual and computational efficiency, making it suitable for real-time applications.

Future Scope:

Real-Time Deployment:

Deploy the model in real-time video conferencing applications.

Testing on Large Dataset:

Train and test the model on more diverse and large datasets.

Better Quality Assessment:

Explore new perceptual quality metrics beyond SSIM score.

User Customization:

Allow the users to adjust the level of sharpness according to their preferences.

Inference links:

https://drive.google.com/drive/folders/195dmOS4l2ltLfIncxxar9D3vs-FO0kas?usp=sharing (Our Dataset)

https://www.kaggle.com/datasets/ahmadahmadzada/images2000

https://drive.google.com/drive/folders/1ewjKEJROSeblKg2HJc5VQg4oKnw3RbEU?usp=sharing (Our Video)

Research Papers References:

https://arxiv.org/abs/1503.02531

https://github.com/Zhengchen1999/SwinFIR

https://openaccess.thecvf.com/content/CVPR2025/html/Long_Progressive_Focused_Transformer_for_Single_Image_Super-Resolution_CVPR_2025_paper.html

https://arxiv.org/html/2501.09268v1

https://www.sciencedirect.com/science/article/abs/pii/S0925231222013674

https://colab.research.google.com/drive/1C2818h7KnjNv4R1sabe14_AYL7lWhmu6?usp=sharing