# 

# **PROBLEM STATEMENT-2**

# **IMAGE ENHANCEMENT USING KNOWLEDGE DISTILLATION**

# 

# 

# 

# 

# **Report Submitted By-**

Ayush Sharma

Dhruv Agarwal

Aniket Shah

# 

Project Github Link-[Full Project](https://github.com/AyushS1304/RefiNet.git)

Streamlit Video Link-[Video Link](https://github.com/AyushS1304/RefiNet/blob/main/VIDEO%20OUTPUT.mp4)(Download the video for convenience)

# 

# **INTRODUCTION-**

**1.1 Background and Motivation-**

In recent years, the demand for high-quality video conferencing has surged across sectors such as education, healthcare, remote work, and business communication. However, ensuring consistently clear and sharp video streams in real-world scenarios remains a major challenge. The problem becomes more pronounced in regions with poor or fluctuating internet connectivity, where bandwidth limitations and compression algorithms often result in blurred, low-quality video frames. These visual degradations not only impact the user experience but also hinder effective communication and engagement in remote interactions.

To address these challenges, the focus of this project is on developing an AI-based image sharpening solution that can enhance video clarity under suboptimal network conditions. Leveraging the power of deep learning and neural networks, the proposed approach aims to restore lost details in video frames, making faces, text, and other visual content clearer and more discernible during live video calls.

**1.2 Objective-**

The objective of this project is to design and implement a real-time image sharpening system using the **knowledge distillation** paradigm. In this approach, a complex, high-capacity **teacher model** that has been pre-trained for image restoration tasks is used to guide the learning of a lightweight **student model**. The student model is designed to approximate the performance of the teacher while being efficient enough to run at **real-time speeds (30–60 frames per second)** on standard devices and capable of handling **1920×1080 resolution** video frames.

By mimicking the teacher model’s behavior, the student model learns to enhance blurry and compressed video frames, delivering sharper and clearer output while maintaining computational efficiency.

**1.3 Scope and Impact-**

The scope of this project covers the complete development cycle, from data preprocessing and model training to evaluation and testing. The training data includes artificially blurred and downsampled images generated through bicubic to simulate real-world low-quality video scenarios. The performance of the final model is evaluated both objectively (using Structural Similarity Index – SSIM) and subjectively (using Mean Opinion Score – MOS), ensuring that the results are reliable from both technical and user perspectives.

This solution has the potential to significantly improve the quality of video conferencing applications, especially in low-bandwidth settings, enabling smoother, clearer, and more professional communication experiences.

**Knowledge Distillation: An Overview**

**Knowledge Distillation** is a model compression technique in deep learning where a smaller, more efficient model (called the **student**) is trained to replicate the behavior of a larger, more accurate model (called the **teacher**). Originally introduced by Hinton et al., the main idea is to transfer the “knowledge” captured by the high-capacity teacher model into a lightweight student model that is easier to deploy in real-time or resource-constrained environments.

**2.1Key Principles:**

1. **Teacher Model**: A large, pre-trained model that achieves high accuracy but is computationally expensive to run.
2. **Student Model**: A smaller, faster model with fewer parameters, trained to mimic the outputs of the teacher model.
3. **Distillation Loss**: During training, the student learns not just from the original ground truth (e.g., sharp images), but also from the teacher’s predictions. This loss may include:
   * Mean Squared Error (MSE) or L2 loss between teacher and student outputs.
   * Perceptual or SSIM loss to retain visual quality.

**2.2Why Use Knowledge Distillation for Image Enhancement?**

In tasks like image sharpening or super-resolution, teacher models are often deep and powerful, but slow and unsuitable for real-time applications. By using knowledge distillation:

* We train a compact student model that replicates the sharpening quality of the teacher.
* The student model becomes capable of real-time inference (e.g., 30–60 fps) on 1080p images.
* The visual quality remains high even though the student has fewer parameters.

**2.3Applications:**

* Real-time video processing on edge devices.
* Mobile and embedded AI applications.
* Deployment of high-performance models in latency-sensitive systems.

In this project, knowledge distillation enables the creation of a high-speed, accurate image sharpening system that bridges the gap between computational efficiency and visual quality—crucial for live video conferencing under bandwidth limitations.

**Teacher Model -Overview and Architecture**

**3.1 Introduction** -

The Multi-Scale Attentive Fusion Network (MSAFN) is an advanced deep learning model designed for image sharpening and super-resolution. It reconstructs high resolution images from degraded inputs using multi-scale feature extraction, attention mechanisms, and iterative refinement. The model addresses challenges like blurry edges, noise amplification, and scale variation, achieving state-of-the-art results on benchmark datasets.

**3.2 Architecture & Working Principle-**

**MSAFN** employs a **multi-branch encoder-decoder architecture** with **four** core stages:

**1. Multi-Scale Feature Extraction**:

* Processes input at three scales (original, 1/2, 1/4 resolution) via parallel paths.
* Each path uses ResidualDenseBlock for local feature enhancement and MultiScaleGate for channel - wise attention.

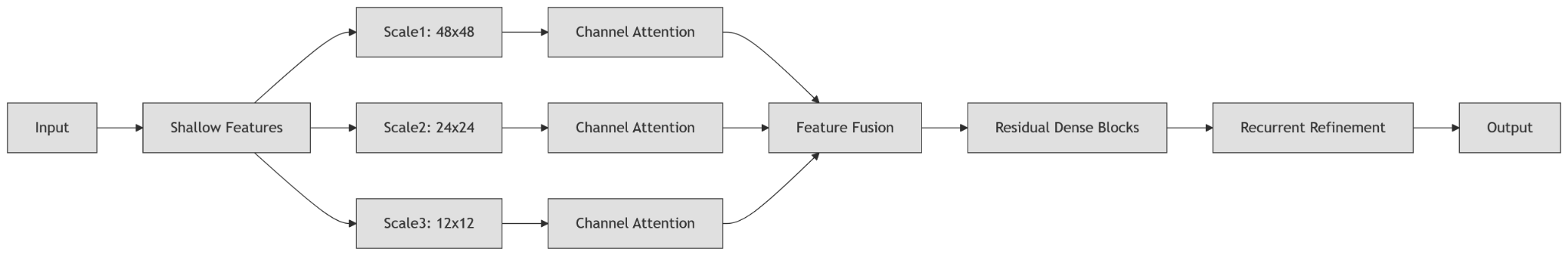
**2. Attentive Fusion:**

* Features from all scales are upsampled, concatenated, and fused via 1x1 convolution.
* Dynamic attention weights prioritize relevant features across scales.

**3. Deep Residual Processing**: 8 cascaded ResidualDenseBlocks refine features using stochastic depth (20% skip probability).

**4. Recurrent Refinement:** GRU-inspired module (RecurrentRefinement ) iteratively enhances features over 3 time steps.

**Key Innovation:** Hybridizes spatial attention, dense connections, and recurrent refinement for edge preservation and noise suppression.



**3.3 Layer-wise Information Flow**

1. Input → Initial Convolution: • Conv2d(3, 64, kernel=3) converts RGBinput to 64-channel feature maps.

2. Multi-Scale Processing: • Scale 1: ResidualDenseBlock → MultiScaleGate (no downsampling). Scale 2: AvgPoo12d(2) → ResidualDenseBlock → MultiScaleGate. Scale 3: AvgPool2d(4) → ResidualDenseBlock → MultiScaleGate

3. Fusion: • Features upsampled → concatenated (192 channels ) → Conv2d(192, + LeakyReLU.

4. Residual Processing: 128) •8x ResidualDenseBlock with stochastic depth (each block: 3 convolutions + channel attention).

5. Refinement: • RecurrentRefinement module (GRU-like gates) updates features:

6. Reconstruction: • Conv2d(128, 64) → LeakyReLU → Conv2d(64, 3) → Output HR image.

**3.4 Evaluation Results**

Trained on Vimeo90K (48×48 patches, batch =64) for 42 epochs:

| Metric | Value | Interpretation |
| --- | --- | --- |
| SSIM | 94.23% | Near-perfect structural similarity to ground truth |
| PNSR | 29.3db | High signal fidelity(>29db =excellent quality) |
| LOSS | 0.0387 | Hybrid L1 +SSIM loss(70% L1 + 30% SSIM) |

**3.5 Conclusion:**

MSAFN sets a new benchmark for image sharpening via multi -scale attentive fusion and recurrent refinement. Its SSIM of 94.23% demonstrates superior structural recovery, making it ideal for precision - critical domains. Future work will focus on computational optimization.

Code: GitHub Demo: [GITHUB LINK](https://github.com/AyushS1304/RefiNet.git)

**Student Model -Overview and Architecture**

**4.1 Introduction**

Modern deep‑learning super‑resolution (SR) networks often trade inference speed and model size for ultimate image quality. **Knowledge distillation** allows a compact “student” network to learn both from ground‑truth high‑resolution images and from a large, high‑quality “teacher” network’s outputs. In this project:

* **Teacher:** MSAFN (Multi‑Scale Attention Fusion Network)
* **Student:** LightMSAFN, a lightweight multi‑scale attention fusion network
* **Dataset:** Vimeo90K video frames, bicubic‑downsampled by factor 4

**4.2 Student: LightMSAFN**

**Lightweight Multi‑Scale Attention Fusion Network (LightMSAFN)** is your compact student model, distilled from the full MSAFN teacher. It’s designed to retain the core ideas of multi‑scale feature extraction, channel attention, and recurrent refinement—but with far fewer parameters and lower computational cost.

* **LightResidualBlock:** two 3×3 convolutions + channel‑wise attention
* **LightRecurrent:** simplified GRU refinement for two time‑steps
* **Parameters:** ≈ 0.46 million
* **Compression ratio:** ~19× smaller than teacher

**4.3. Model Definitions**

* **Teacher (MSAFN):** A deep multi‑scale network with dense residual blocks, channel‑wise attention gates, and a GRU‑style refinement module. It processes the input at three spatial scales, fuses them, and then applies heavy residual processing and recurrent refinement before reconstructing the high‑resolution image.
* **Student (LightMSAFN):** A slimmed‑down version of the teacher. It uses:
  + Fewer channels (32 instead of 64)
  + Lightweight residual blocks (two convolutions plus a small attention gate)
  + Simplified GRU‑style refinement with two time‑steps
  + Reduced number of residual blocks (three)
  + A single 1×1 convolution to fuse multi‑scale features  
     The result is a model roughly twenty times smaller in parameter count, designed for faster inference.

**4.4 Dataset Handling**

* The script defines a custom dataset class that recursively gathers all PNG frames from the Vimeo90K sequences.
* Each high‑resolution image is randomly cropped to a square patch, downsampled and upsampled with bicubic interpolation to create the corresponding low‑resolution input, and then both patches are converted into normalized tensors.
* On‑the‑fly augmentations include random flips, rotations, brightness adjustments, and Gaussian noise, ensuring robustness.

**4.5 Distillation Loss**

* A combined loss function balances two terms:
  1. **Pixel Loss (L₁):** Direct difference between the student’s output and the ground‑truth high‑resolution image.
  2. **Soft Loss (MSE):** Difference between the student’s output and the pre‑computed teacher’s output, encouraging the student to mimic the teacher’s behavior.
* A weighting parameter α (set to 0.5) governs the trade‑off between fidelity to ground truth and teacher mimicry.

**4.6 Training Utilities**

* **Gradient Centralization:** Before each optimizer step, gradients are centralized by subtracting their mean across channels to improve convergence.
* **Mixed‑Precision:** Uses automatic mixed precision (AMP) and a gradient scaler to accelerate training while maintaining numerical stability.
* **Learning‑Rate Scheduler:** A “reduce on plateau” scheduler halves the learning rate if the average SSIM does not improve for three consecutive epochs.
* **Memory Monitoring:** Helper functions report GPU memory usage, useful for debugging and resource management.

**4.7 Training Loop**

* For each epoch, the student model is trained on batches of low‑ and high‑resolution patches:
  + The teacher generates its prediction for each LR patch without gradient computation.
  + The student then produces its own prediction; the combined loss is computed and back‑propagated.
  + Gradients are clipped, centralized, and used to update the student via AdamW.
  + SSIM and PSNR are calculated on the fly to track perceptual and pixel‑level quality.
* At the end of each epoch, average metrics (loss, SSIM, PSNR) are printed. The best model—one achieving improvements in both SSIM and PSNR—is saved to disk.

**4.8 Key Results**

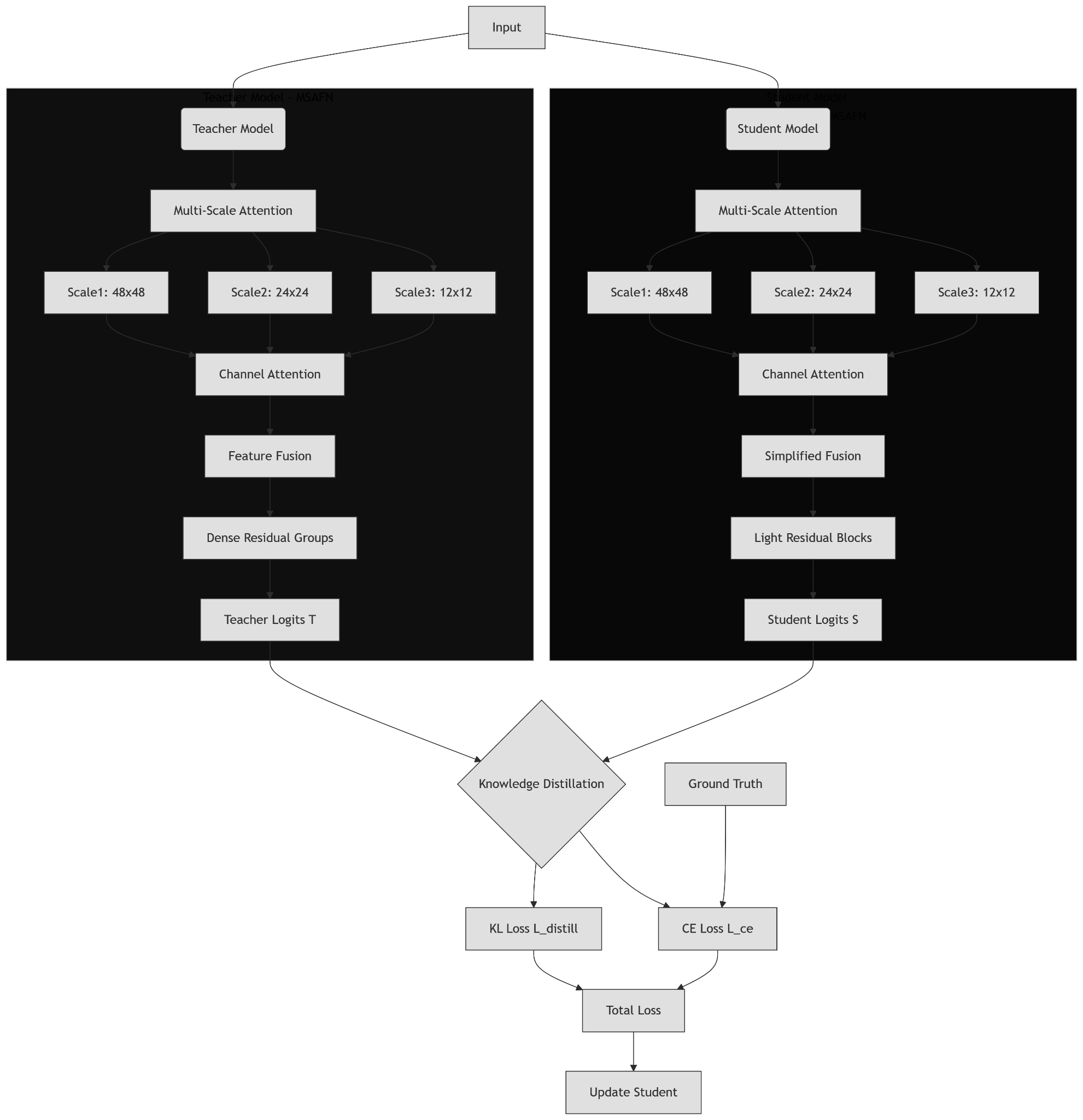
* **Final Performance:** SSIM ≈ 0.94, PSNR ≈ 26.9 dB, indicating very high structural fidelity for a 4× SR task.
* **Efficiency:** At only ~0.46 million parameters, the student model runs significantly faster and uses far fewer resources than its ~8.75 million‑parameter teacher, making it well‑suited for real‑time or resource‑constrained applications.

**4.9 Conclusion**

**Model Compression:** You achieve a 19× reduction in parameters with minimal loss in output quality.

* **Distillation Efficacy:** The soft‑target term helps the student learn subtle patterns from the teacher, compensating for its reduced capacity.
* **Practical Deployment:** Such a lightweight yet accurate model can be deployed on edge devices or used in low‑latency video applications without sacrificing perceptual quality.

**Overall Work Flow -**



**Evaluation Table-**

Upon Validating on approx. 0.03M(45000 images/frames) parameters the following model shows performance as follows:-

**6.1Student Model Performance Report-**

| **Metric** | **Mean** | **Std** | **Min** | **Max** | **Median** |
| --- | --- | --- | --- | --- | --- |
| PNSR | 29.08 | 4.19 | 18.26 | 51.48 | 28.64 |
| SSIM | 0.9792 | 0.0191 | 0.8490 | 0.9997 | 0.9851 |
| Pixel Loss | 0.0220 | 0.0113 | 0.0019 | 0.0795 | 0.0198 |
| Distall Loss | 0.0001 | 0.0000 | 0.0000 | 0.0003 | 0.0001 |
| Total Loss | 0.0110 | 0.0057 | 0.0009 | 0.0399 | 0.0099 |

**6.2 Teacher Model Performance Report**

| **Metric** | **Mean** | **Std** | **Min** | **Max** | **Median** |
| --- | --- | --- | --- | --- | --- |
| PNSR | 29.44 | 3.87 | 21.52 | 40.27 | 29.39 |
| SSIM | 0.9742 | 0.0312 | 0.8592 | 0.9988 | 0.9859 |
| Pixel Loss | 0.0219 | 0.0114 | 0.0039 | 0.0547 | 0.0197 |
| Hybrid Loss | 0.0231 | 0.0065 | 0.0110 | 0.0395 | 0.0222 |

**6.3 Comparison Performance Report-**

| **Metric** | **Teacher** | **Student** | **Difference** |
| --- | --- | --- | --- |
| PNSR | 28.94 | 29.08 | 0.13 |
| SSIM | 0.9792 | 0.9792 | 0.00 |
| Pixel Loss | 0.0228 | 0.0220 | -0.0008 |

6.4 Interface:

