National College of Ireland

Comparative Analysis of Attention-Based Deep Learning Architectures for Low-Light Image Enhancement

MSc Research Project

Artificial Intelligence

Likhita Kanikicherla

Student ID: 23319739

School of Computing

National College of Ireland

Supervisor: Taylou Maniganze

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name: ……. Likhita Kanikicherla……

Student ID: ……23319739……..……

Programme: ……MSc in AI…… **Year:** …2024-2025…..

Module: ……Thesis…….……

Supervisor: ……Taylou Maniganze…….……

Submission Due Date: 15-09-2025…….……

Project Title: Comparative Analysis of Attention based Deep Learning Architectures for Low-Light Image Enhancement

Word Count: …… **Page Count**……11……..……..

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: ……Likhita Kanikicherla……

Date: ……14-09-2025……

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

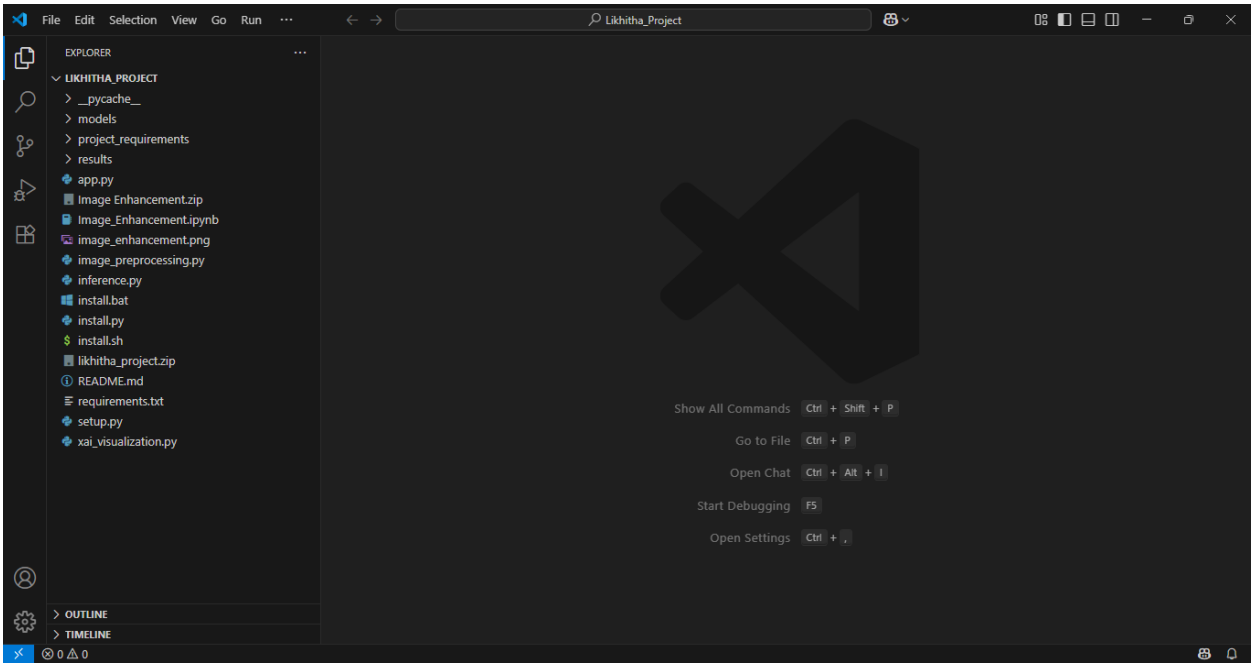
Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Comparative Analysis of Attention-Based Deep Learning Architectures for Low-Light Image Enhancement

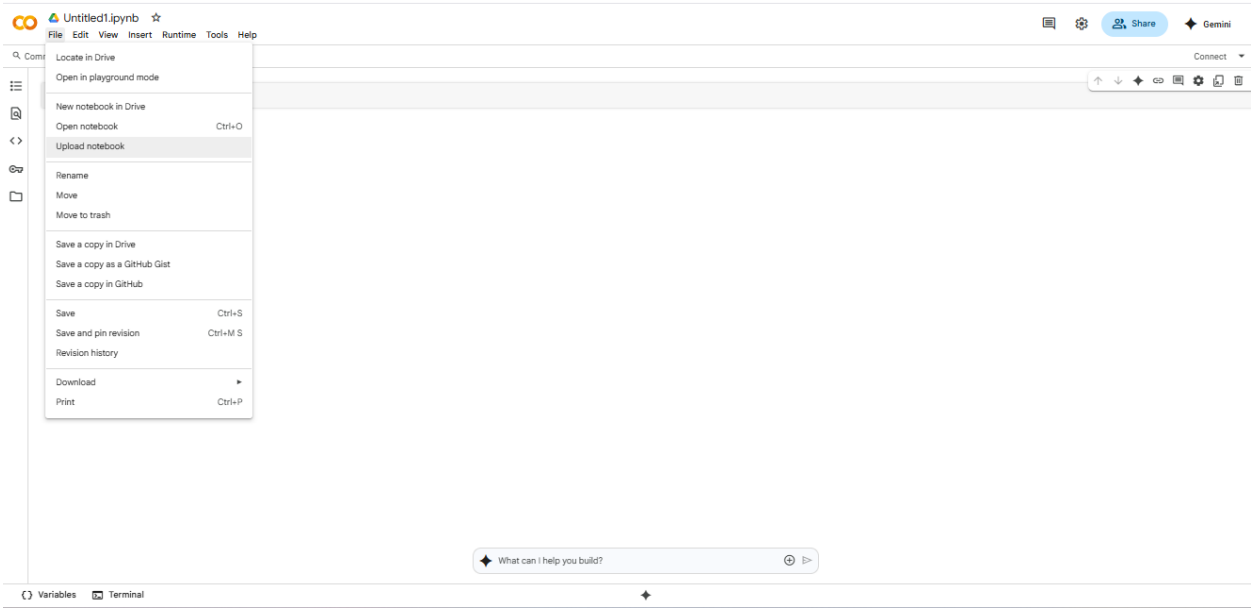
Likhita Kanikicherla

23319739

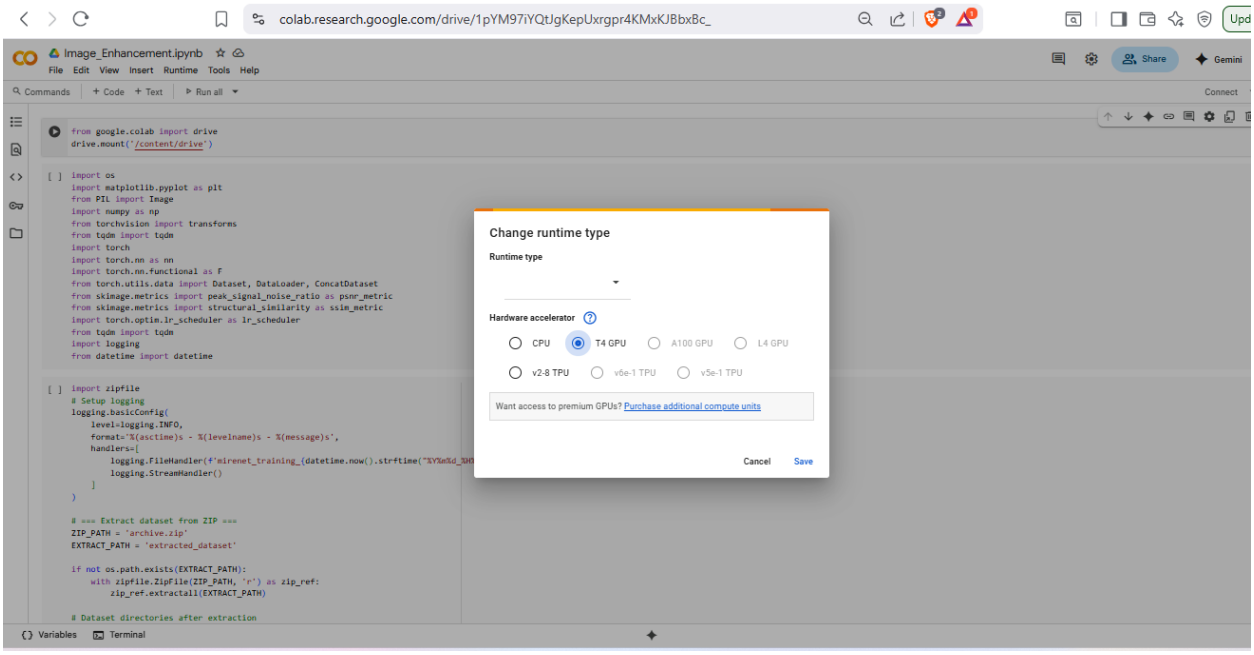
1. Unzip the downloaded folder and open extracted folder in VScode



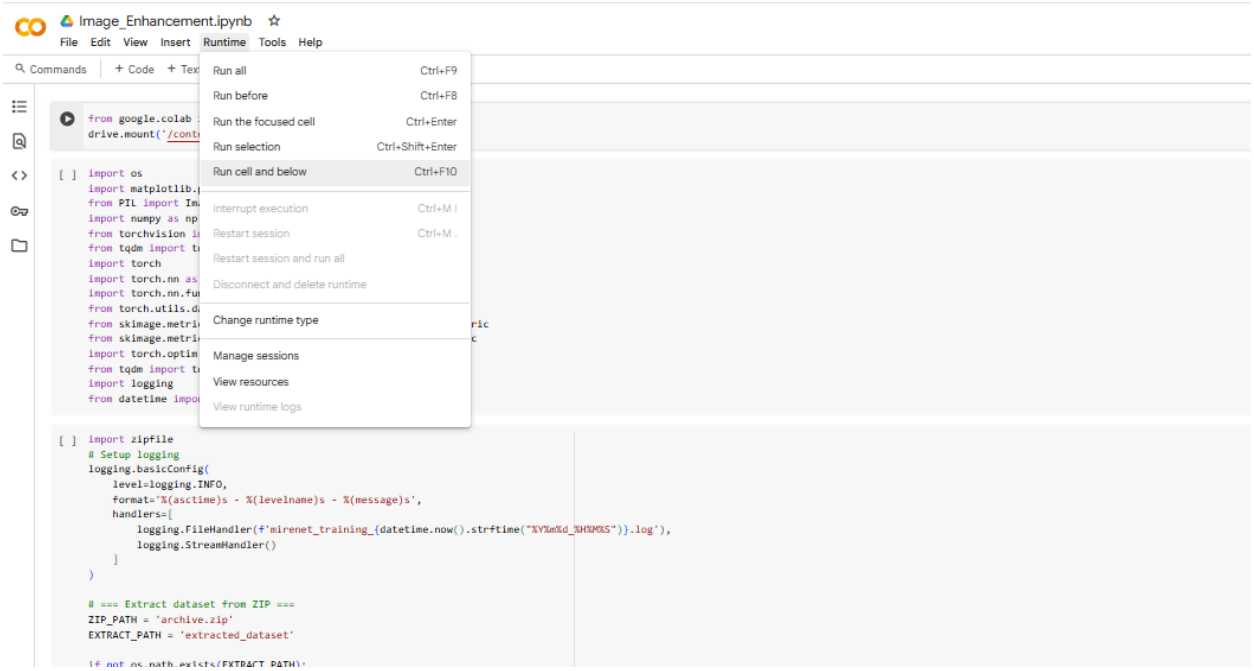
2. For training, login to google collab and upload “Image_Enhancement.ipynb” file as below using the upload notebook option



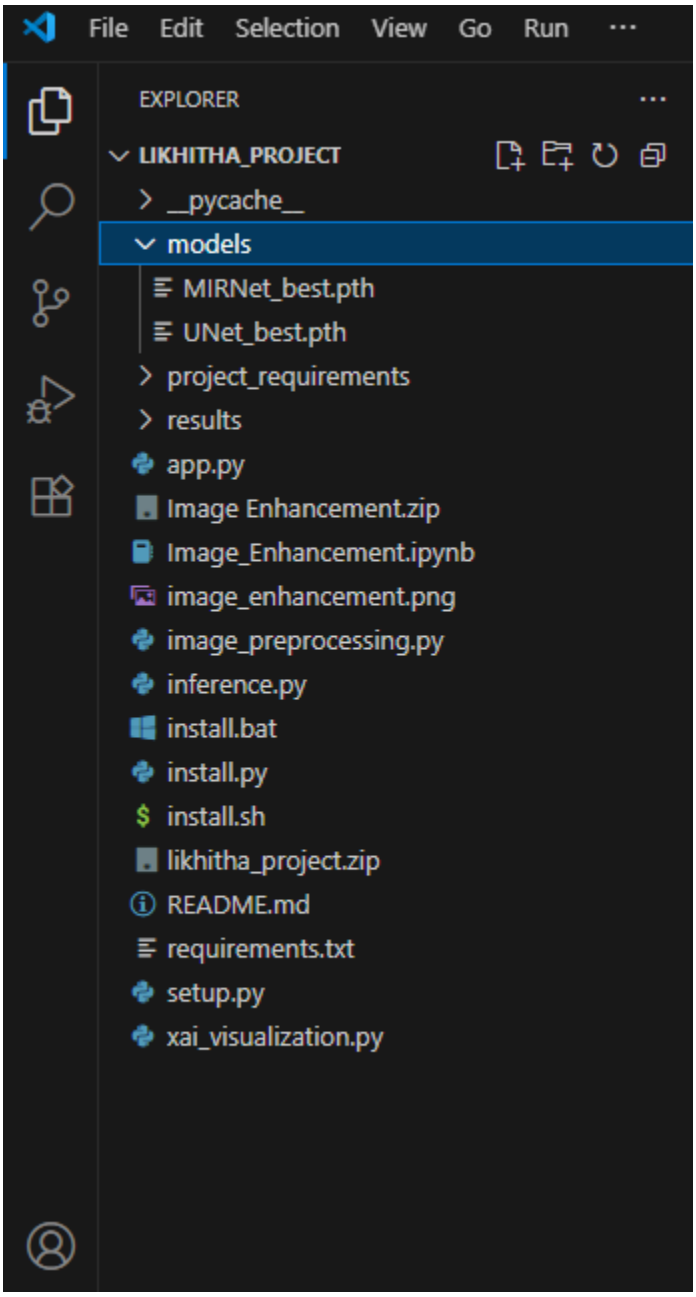
3. After uploading, change the runtime as follows



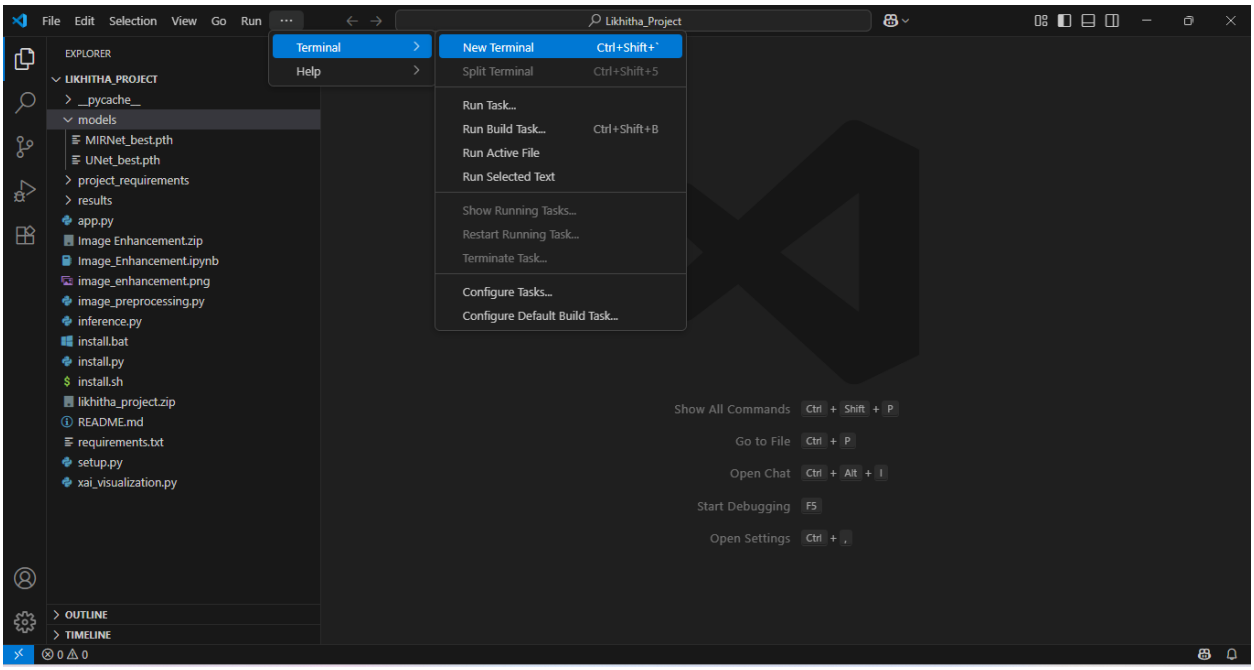
4. Later run all the cells as follows


Image_Enhancement.ipynb
File Edit View Insert Runtime Tools Help
Commands + Code + Tex
Run all Ctrl+F9
Run before Ctrl+F8
Run the focused cell Ctrl+Enter
Run selection Ctrl+Shift+Enter
Run cell and below Ctrl+F10
Interrupt execution Ctrl+M I
Restart session Ctrl+M .
Restart session and run all
Disconnect and delete runtime
Change runtime type
Manage sessions
View resources
View runtime logs
from google.colab
drive.mount('/cont
[ ] import os
import matplotlib.
from PIL import Im
import numpy as np
from torchvision i
from tqdm import t
import torch
import torch.nn as
import torch.nn.fu
from torch.utils.d
from skimage.metri
from skimage.metri
import torch.optim
from tqdm import t
import logging
from datetime impo
[ ] import zipfile
Setup logging
logging.basicConfig(
level=logging.INFO,
format='%(asctime)s - %(levelname)s - %(message)s',
handlers=[
logging.FileHandler(f'mirenet_training_{datetime.now().strftime("%Y%m%d_%H%M%S")}.log'),
logging.StreamHandler()
]
)
# === Extract dataset from ZIP ===
ZIP_PATH = 'archive.zip'
EXTRACT_PATH = 'extracted_dataset'
if not os.path.exists(EXTRACT_PATH):

5. Download the model, into the extracted folder

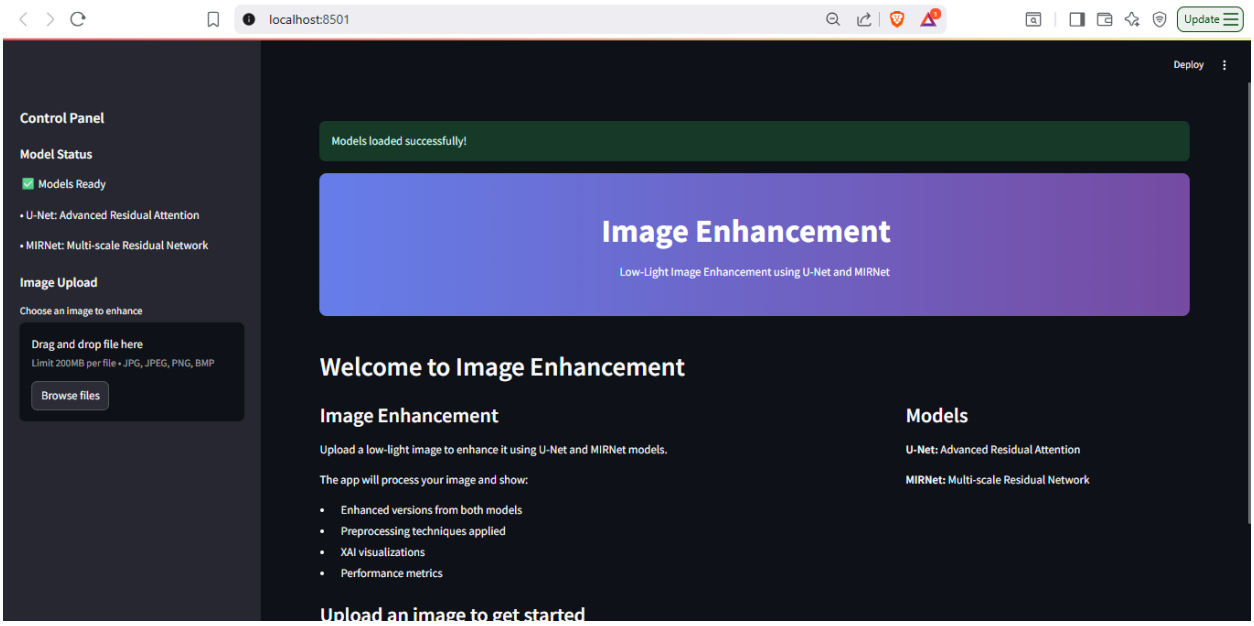


6. Open new terminal as follows

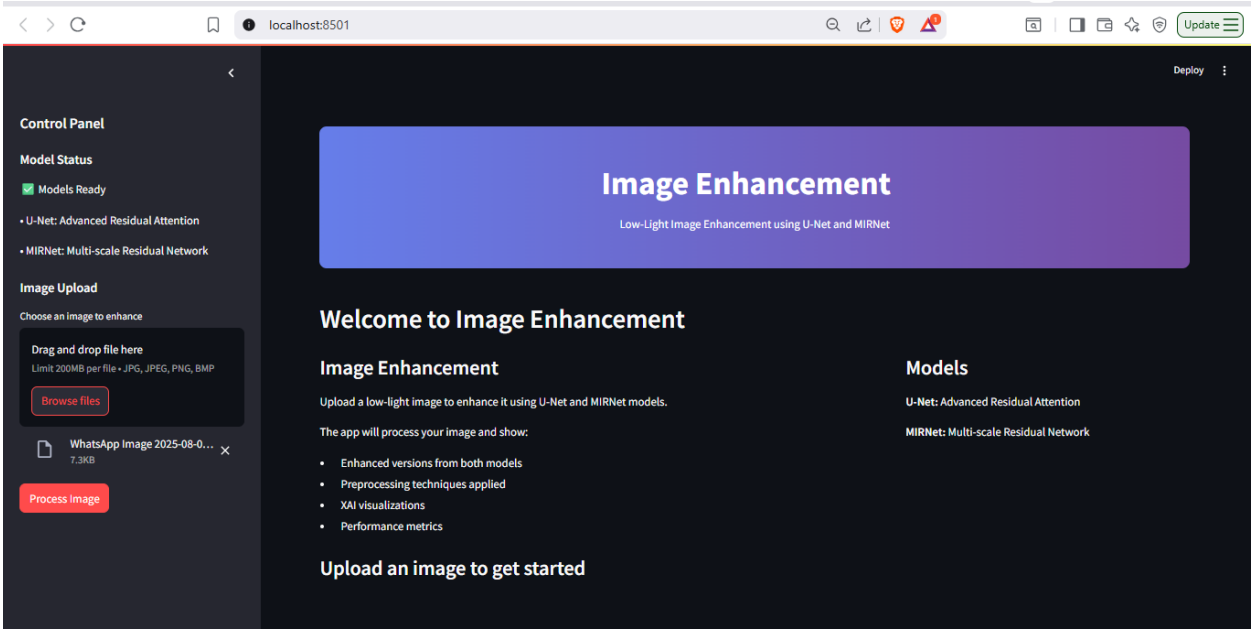


7. Run the following commands as follows
    a. conda create –name my_project python=3.10 -y
    b. conda activate my_project
    c. pip install -r requirements.txt
    d. streamlit run app.py

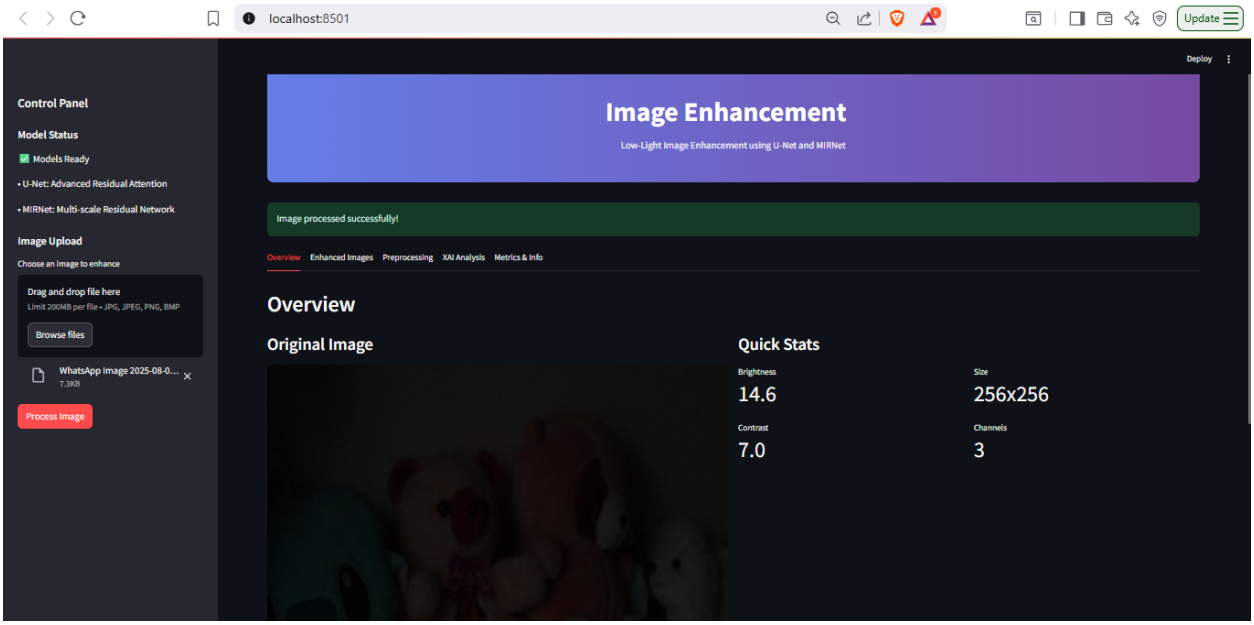
8. You can see the following UI after running previous commands

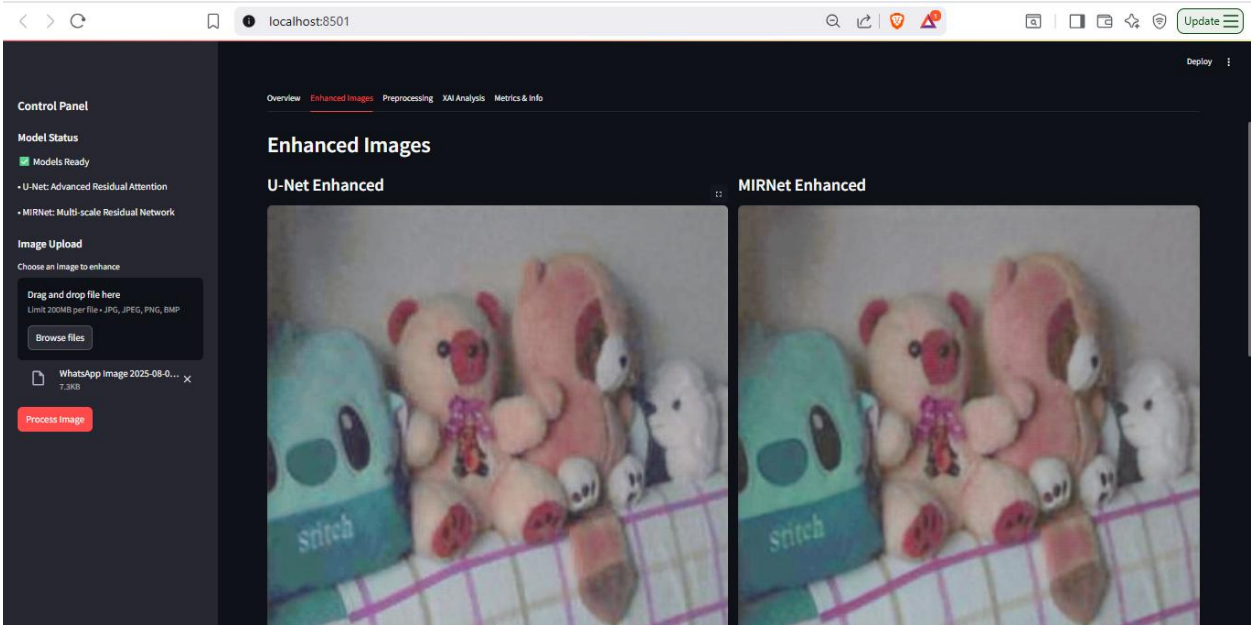


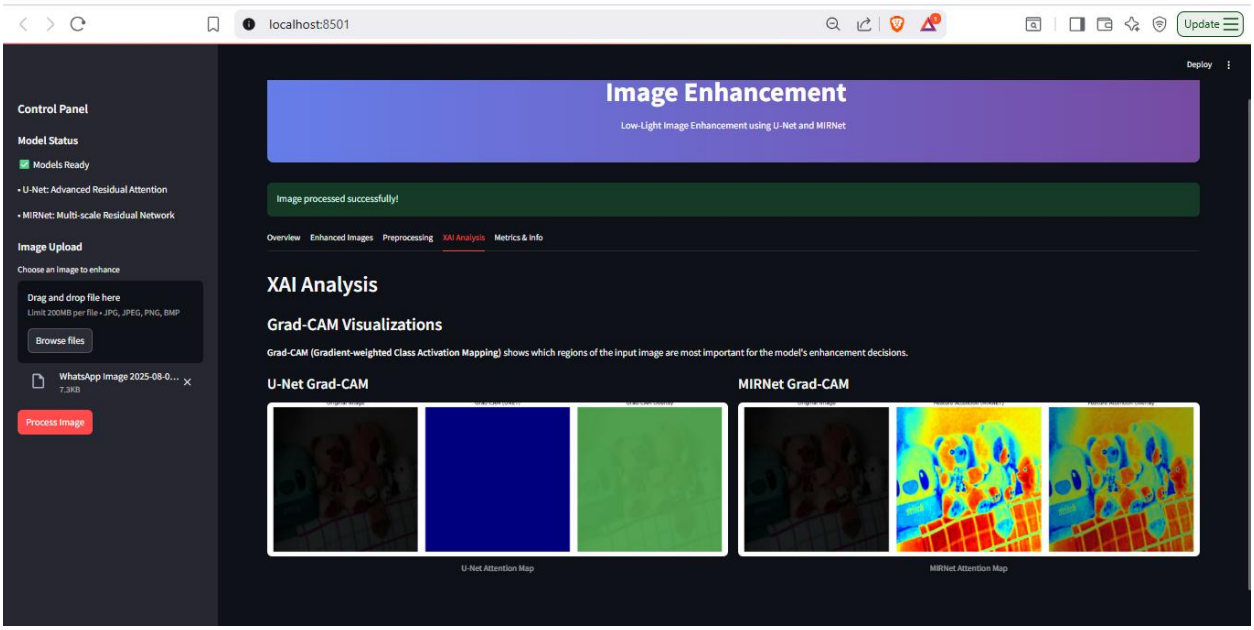
9. Click on Browse Files and upload an image

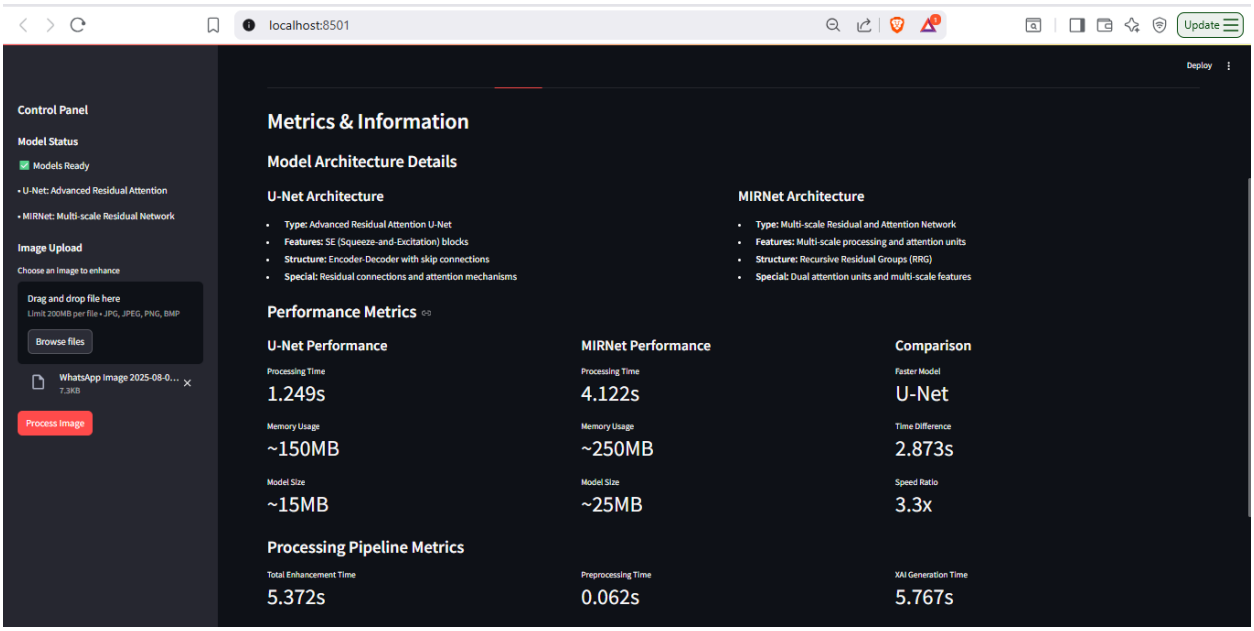


10. Click on process image to see the following results




localhost:8501
Update
Deploy
Control Panel
Model Status
Models Ready
• U-Net: Advanced Residual Attention
• MIRNet: Multi-scale Residual Network
Image Upload
Choose an image to enhance
Drag and drop file here
Limit 200MB per file • JPG, JPEG, PNG, BMP
Browse files
WhatsApp Image 2025-08-0...
7.3KB
Process Image
Overview
Enhanced Images
Preprocessing
XAI Analysis
Metrics & Info
Enhanced Images
U-Net Enhanced
MIRNet Enhanced


localhost:8501
Update
Deploy
Control Panel
Model Status
Models Ready
• U-Net: Advanced Residual Attention
• MIRNet: Multi-scale Residual Network
Image Upload
Choose an image to enhance
Drag and drop file here
Limit 200MB per file • JPG, JPEG, PNG, BMP
Browse files
WhatsApp Image 2025-08-0...
7.3KB
Process Image
Image Enhancement
Low-Light Image Enhancement using U-Net and MIRNet
Image processed successfully!
Overview
Enhanced Images
Preprocessing
XAI Analysis
Metrics & Info
XAI Analysis
Grad-CAM Visualizations
Grad-CAM (Gradient-weighted Class Activation Mapping) shows which regions of the input image are most important for the model's enhancement decisions.
U-Net Grad-CAM
MIRNet Grad-CAM
U-Net Attention Map
MIRNet Attention Map


localhost:8501
Update
Deploy
Control Panel
Model Status
Models Ready
• U-Net: Advanced Residual Attention
• MIRNet: Multi-scale Residual Network
Image Upload
Choose an image to enhance
Drag and drop file here
Limit 200MB per file • JPG, JPEG, PNG, BMP
Browse files
WhatsApp Image 2025-08-0...
7.3KB
Process Image
Metrics & Information
Model Architecture Details
U-Net Architecture
Type: Advanced Residual Attention U-Net
Features: SE (Squeeze-and-Excitation) blocks
Structure: Encoder-Decoder with skip connections
Special: Residual connections and attention mechanisms
MIRNet Architecture
Type: Multi-scale Residual and Attention Network
Features: Multi-scale processing and attention units
Structure: Recursive Residual Groups (RRG)
Special: Dual attention units and multi-scale features
Performance Metrics
U-Net Performance
Processing Time
1.249s
Memory Usage
~150MB
Model Size
~15MB
MIRNet Performance
Processing Time
4.122s
Memory Usage
~250MB
Model Size
~25MB
Comparison
Faster Model
U-Net
Time Difference
2.873s
Speed Ratio
3.3x
Processing Pipeline Metrics
Total Enhancement Time
5.372s
Preprocessing Time
0.062s
XAI Generation Time
5.767s