DETECTING AND RESTORATION OF PIXELATED IMAGE

INTRODUCTION:

Clear and high quality images are essential components of the fast growing technological era. However, pixelation, a common issue caused by excessive resizing or compression can significantly degrade an image's appearance making it look blurry and blocky. This loss of detail can be particularly problematic in fields such as digital media, medical imaging, and the preservation of historical documents. Therefore, detecting and restoring pixelated images is crucial to maintain the image quality.

This project focuses on identifying and enhancing pixelated images to restore their original clarity.The main objective of the project is to ensure that the image retains its sharpness and preserves the essential details. By employing various image processing techniques and machine learning algorithms, pixelated images are detected and their quality is improved.

PROBLEM STATEMENT:

The project addresses two main challenges in image processing:

a) **Detecting pixelated images:** Develop a lightweight, computationally efficient algorithm or machine learning model to accurately detect pixelated images. The goal is to achieve over 90% accuracy with minimal false positives,while processing 1080p resolution inputs at 60 frames per second.

b) **Enhancing pixelated images:** Design an algorithm to restore pixelated images, significantly improving their quality while maintaining efficiency. The algorithm should be capable of distinguishing between pixelated and non-pixelated images, enhancing only those that are pixelated.

UNIQUE IDEA(Solution):

Our solution combines efficient image classification and generation techniques to address both aspects of the problem:

a) For pixelation detection, we implemented a lightweight pre-trained Transfer Learning model called **YOLO-V8** , optimised for speed and accuracy, with a focus on rare class scenarios.

b) For Image Enhancement we had to use another pre-trained model called **MIRNETv2** which enables us to reduce the noise from the pixelated images.

FEATURES OFFERED:

- High-speed pixelation detection for 1920 x 1080 images
- Accurate classification with satisfactory F1 score.
- The detection algorithm can process images at a high frame rate.
- The model maintains a high accuracy rate with minimal false positives.
- Restoration of images upto a suitable PSNR.

DATASET:

Data Availability and Creation

There was no open source dataset that satisfied all the requirements of the project. In order to meet the project expectation, a custom dataset was created using the **Pixaby** API. Pixabay is a popular platform that provides high quality images and videos, free to use for various purposes. Using the Pixabay API, a diverse range of high resolution images were obtained. This ensures variety in the training of the model.

Resolution Adjustment

The high resolution images obtained from Pixabay API were of a different resolution. In order to meet the specific requirement of **1920x1080** pixel resolution for all images, the **Lanczos Resampling Method** was employed. Lanczos resampling is a high quality filtering technique that uses a sinc function to interpolate pixel values. This method is known for its ability to resize images while preserving edge details and minimising artefacts. By applying Lanczos resampling method, all the images in the dataset were resized to 1920x1080 pixels without introducing pixelation, thereby maintaining image quality.

Pixelation Techniques

To train the model to detect pixelation, the dataset was divided into pixelated and non-pixelated images in 0.6 ratio (60% pixelated and 40% non pixelated). Various pixelation techniques were used to pixelate the images in order to introduce variety.

The 4 main pixelation techniques used were:

- Block Averaging: It involves dividing the image into various non-overlapping blocks , averaging the colour values and applying it to all pixels within the block.
- JPEG Compression: This involves reducing the compression quality of JPEG images to introduce compression artefacts that simulate pixelation.
- Mosaic Effect: It involves segmenting the image into blocks and replacing each block with a single colour which produces a distinct pixelated appearance.
- Noise: Noise was added to some of the images resulting in random variation of pixel values.

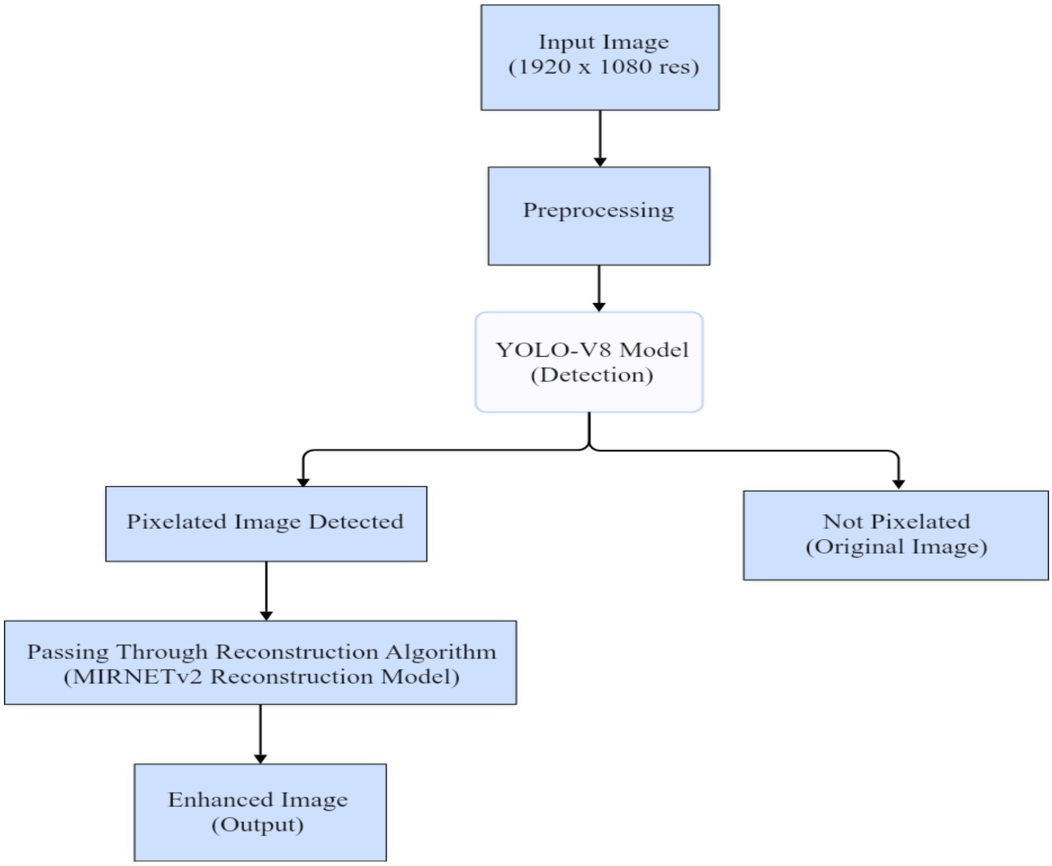
By combining these three pixelation techniques and adding noise, a variety of pixelated images were generated that provided a robust training dataset for the model.

PROCESS FLOW:

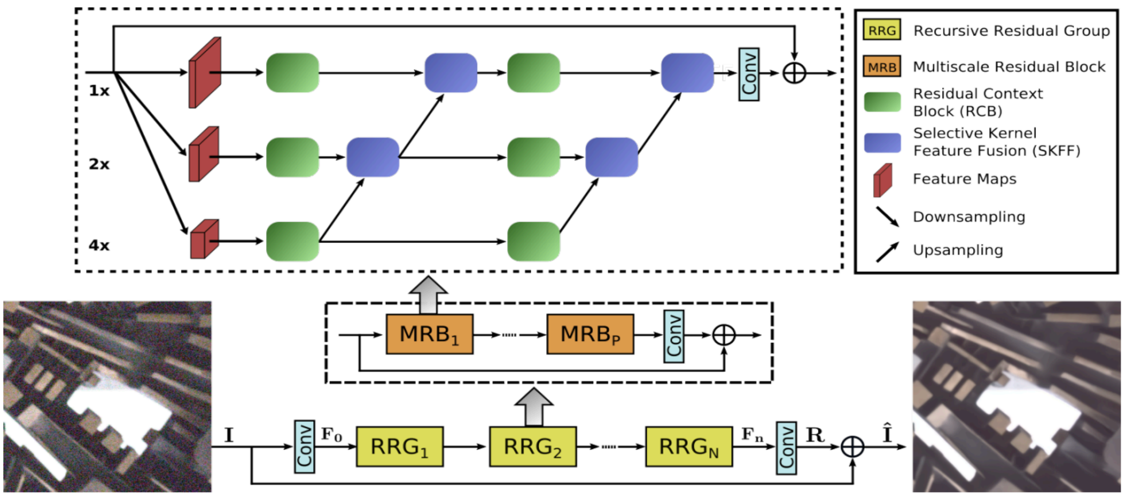
1. Image Acquisition
   - Receive 1920x1080 input image
2. Preprocessing
   - Downscale image if necessary (to optimise processing speed)
   - Normalise pixel values
3. Pixelation Detection
   - Pass preprocessed image through CNN
   - CNN outputs probability of pixelation
   - Apply threshold to determine if image is pixelated
4. Decision Point
   - If image is not pixelated:
     - Skip to step 7 (Output)
   - If image is pixelated:
     - Continue to step 5 (Image Restoration)
5. Image Restoration
   - Pass pixelated image through MIRNetv2 restoration model
   - Generate high-quality restored image
6. Quality Assessment
   - Compare restored image to input image using metrics (LPIPS, PSNR)
   - Ensure restoration improved image quality
7. Output
   - If no restoration : output original image
   - If restoration required: output enhanced image
8. Performance Measures
   - Record processing time to ensure speed requirements are met.
   - F1 score is used to measure the accuracy of detection model
   - Peak Signal to Noise Ratio (PSNR) and Mean Squared Error (MSE) to measure the performance of restoration model

This process flow covers the entire pipeline from input to output, including the decision points and quality checks. It aligns with the project requirements of high-speed processing and accurate detection/restoration.

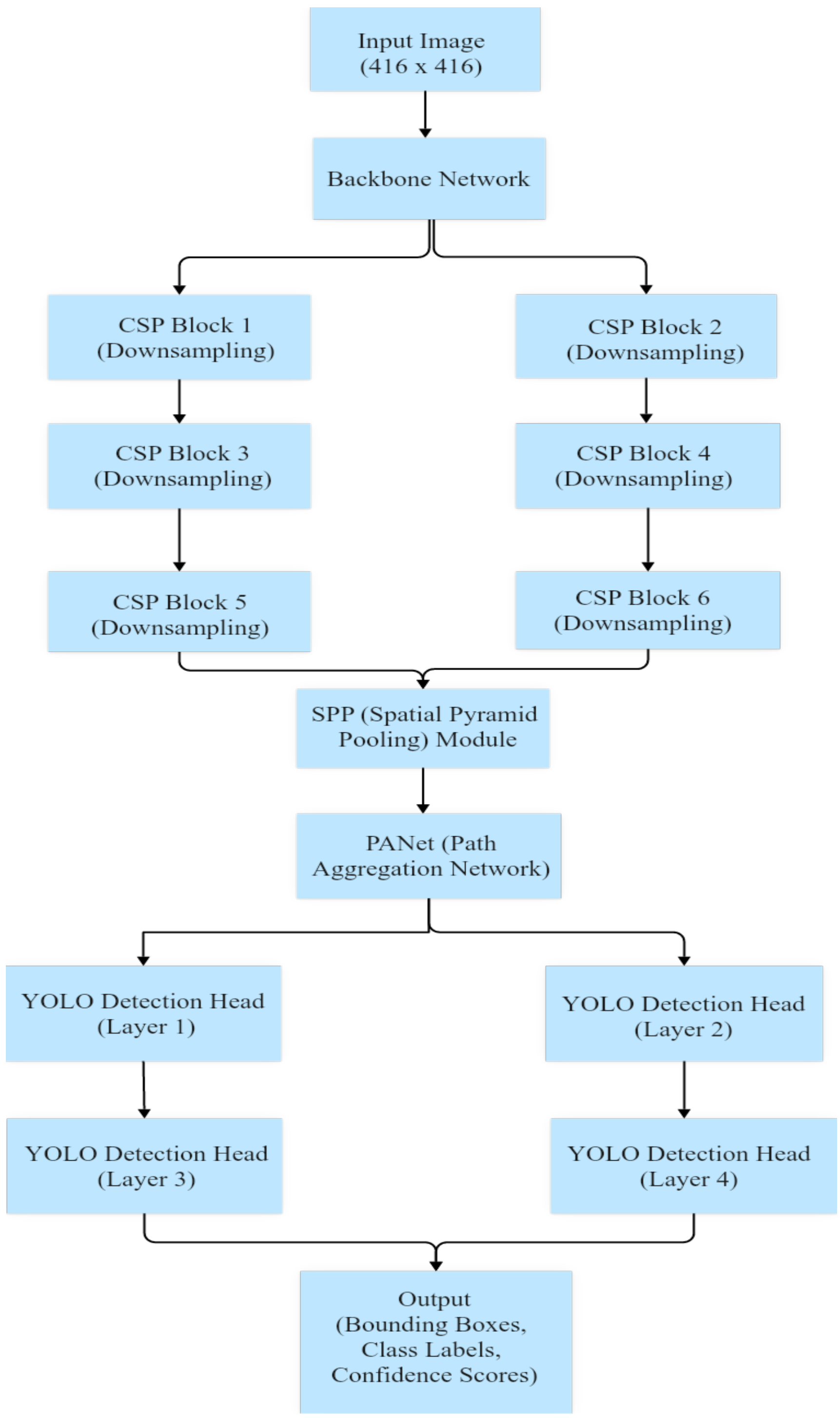
ARCHITECTURE DIAGRAM



- **MIRNETv2 Reconstruction Model Architecture**



- **YOLO-V8 Detection Model Architecture**



TECHNOLOGIES USED:

- Programming Language: **Python**
- Deep Learning Framework: **TensorFlow**
- Libraries Used: **OpenCV, NumPy, Matplotlib, TensorFlow, skimage, Pillow, random, os**
- Platform: **Jupyter Notebook**
- Machine Learning Model Used:
    - Detection: **YOLO-V8**
    - Reconstruction: **MIRNETv2**
- CPU Used: **Intel i9-12900K**

TEAM MEMBERS AND CONTRIBUTIONS:

TEAM MEMBER 1:

Name: Mohammed Faisal

Email: 1ms22ai030@msrit.edu

Contribution: Training and testing of model

TEAM MEMBER 2:

Name: Mohamed Sahil

Email: 1ms22ai028@msrit.edu

Contribution: Validation of model and data pre processing

TEAM MEMBER 3:

Name: Devika P

Email: 1ms22ai013@msrit.edu

Contribution: Data generation and pre processing

CONCLUSION:

The project has made a significant advancement in addressing the issue of pixelated images, both in detection and restoration. By integrating deep learning models and advanced image processing techniques, a robust system capable of identifying and enhancing pixelated images was developed. The results demonstrate that this approach can significantly improve image clarity and details, making it useful in fields such as digital media , medical imaging etc.

The project adheres to the constraints specified by the Intel Unnati Programme. We thank the Intel Unnati team and our college for providing us with this unique opportunity to improve our technical skills and making us industry ready.