INTEL UNNATI INDUSTRIAL TRAINING PROGRAM 2024

Detect Pixelated Image and Correct It

Presented by **Sujal Pareek**
RA2311033010169

Date of Submission: 15/07/2024

Name: Sujal Pareek

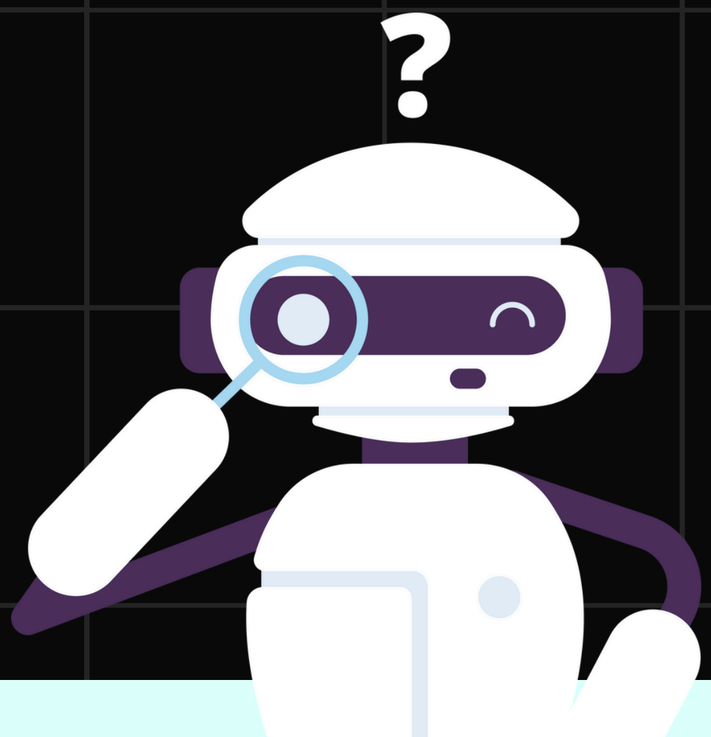
Institute: SRM Institute of Science and Technology Kattankulathur

Branch/Specialization: B.Tech. CSE-Software Engineering

Registration Number: RA2311033010169

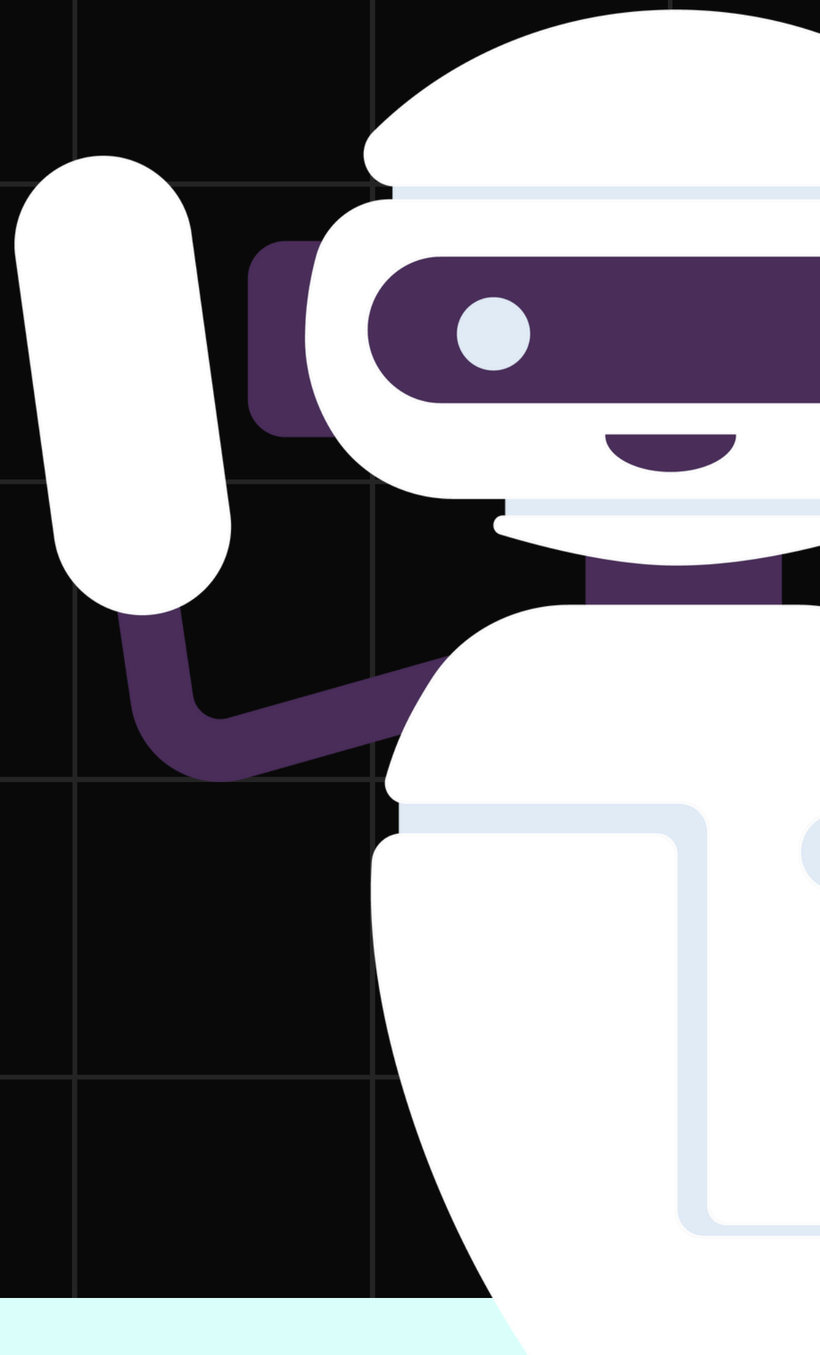
Internal Mentor: Dr. S Raguvaran

TABLE OF CONTENTS



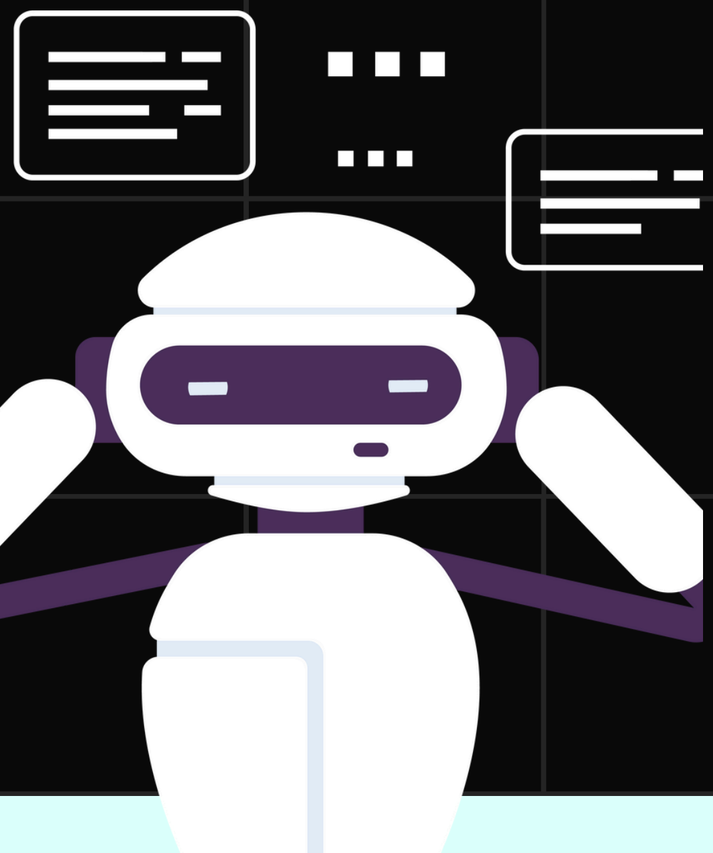
Unique Idea

Pixelated images, often resulting from resizing, compression, or privacy preservation, lose critical detail and clarity. Correcting pixelation involves restoring the lost high-frequency information to enhance image quality. Traditional interpolation methods fall short in generating realistic textures and details. Using Generative Adversarial Networks (GANs), specifically a deep-learning model designed for super-resolution tasks, can adaptively restore pixelated images by learning complex patterns and textures



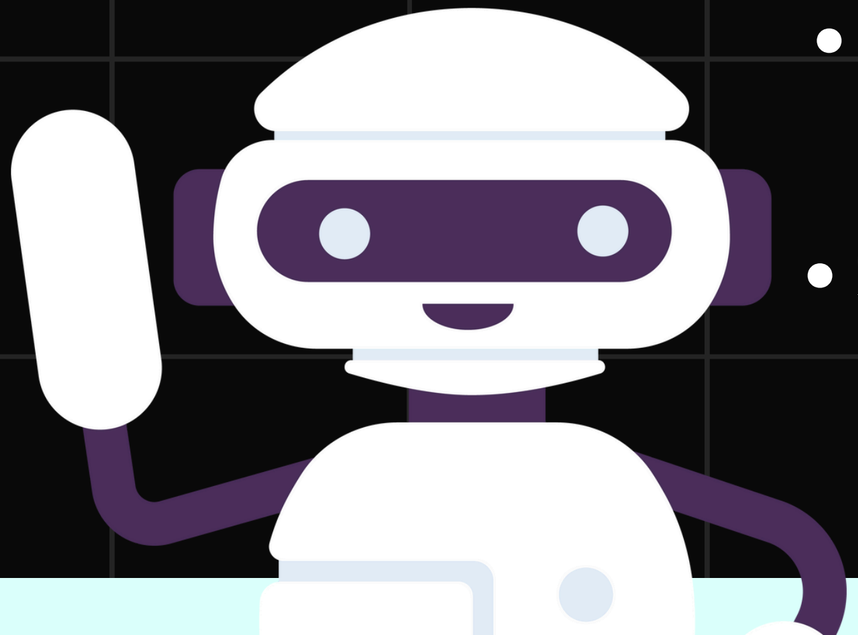
Brief Solution

- Step 1: Data Collection and Preprocessing
- Step 2: Model Architecture
- Step 3: Training with GANs
- Step 4: Evaluation and Fine-Tuning
- Step 5: Deployment and Use Cases



Features Offered

- Automatic Pixelation Detection
- Adaptive Correction Capability
- End-to-End Solution
- Training with Realistic Data
- Performance Metrics
- Scalability and Deployment



Process flow

Step 1:

Image Preprocessing Function: preprocess image

Purpose: Downscale the image to reduce computational complexity and noise.

Process: Resize the image to half its original size using linear interpolation.

Step 2:

Edge Detection Function: detect edges

Purpose: Detect edges in the image to help identify pixelated regions.

Process: Convert the image to grayscale and apply the Canny edge detection algorithm.

Step 3:

Block Analysis Function: analyze blocks

Purpose: Analyze small blocks of the edge-detected image to extract features that indicate pixelation.

Process: Divide the edge-detected image into blocks, calculate variance and edge density for each block.

Step 4:

Feature Extraction Function: extract features

Purpose: Extract features from a set of images and associate them with corresponding labels.

Process: For each image, preprocess, detect edges, analyze blocks, and compile features and labels.

Process flow

Step 5:

Model Training Main Script: train.py

Purpose: Train a Logistic Regression model to classify pixelated images.

Process : Collect image paths and label. Extract features from images. Split the dataset into training and testing sets. Compute class weights to handle class imbalance. Train the Logistic Regression model. Calculate optimal classification threshold using precision-recall curve. Save the trained model to disk.

Step 6:

Model Evaluation Function: process single image

Purpose: Use the trained model to classify a new image as pixelated or non-pixelated.

Process : Preprocess the new image. Detect edges and analyze blocks. Predict the probability of pixelation for each block. Calculate the average prediction for the entire image. Classify the image based on the optimal threshold.

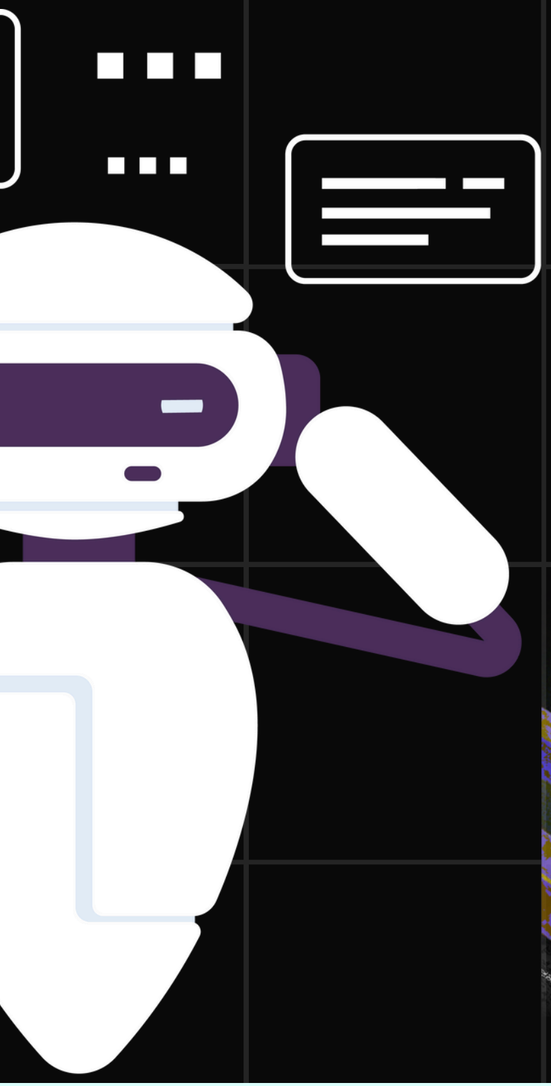
Step 7:

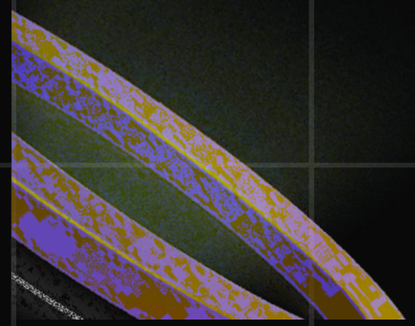
Inference Script Main Script: main.py

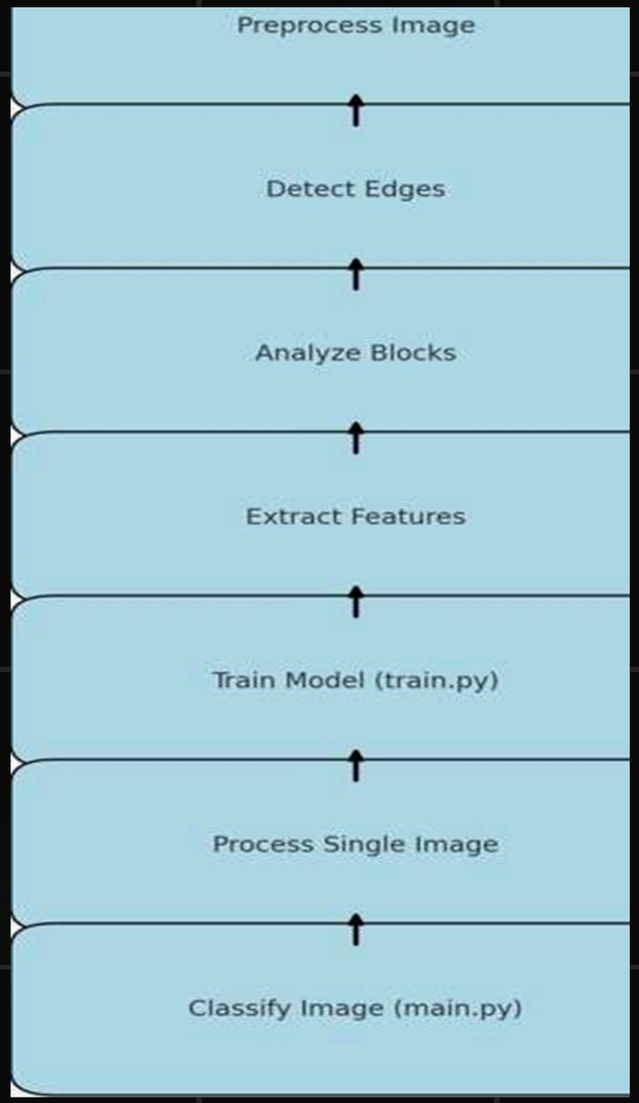
Purpose: Load the trained model and classify a new image.

Process : Load the saved model. Set the optimal threshold for classification. Process a new image to determine if it is pixelated or non-pixelated. Print the classification result and confidence

Architecture Diagram







Technologies used

- Python
- OpenCV (cv2)
- NumPy (np)
- Scikit-learn (sklearn)
- Joblib
- OS (os)
- Glob
- TQDM

conclusion

The challenge of detecting and correcting pixelated images can be effectively addressed using advanced machine learning and deep learning techniques. By leveraging a combination of convolutional neural networks (CNNs) for detection and generative models like GANs or autoencoders for correction, we can create an efficient and automated solution to enhance image quality.

the proposed model offers a robust and comprehensive solution to the problem of pixelated images, combining state-of-the-art machine learning and deep learning techniques to deliver significant improvements in image quality

Thankyou

Sujal Pareek