Problem Statement

The problem has two parts – one is image classification and the other is image generation.

1. Given an Image, check if it is blocky or pixelated. Pixilation means blurry, blocky squares you see
when an image is zoomed too much. Design an algorithm that can detect whether the image is pixelated or not. The challenge is to design an algorithm or ML/AI model, which is extremely lightweight or computationally efficient such that we can run this algorithm at 60 Hz or 60 frames per sec (FPS) and must be minimum 90% accurate.Input Image Size: 1920x1080 (it’s ok to downscale and feed into the algorithm/model).
b. Inference Speed Target: Min 30Hz, better to have 60 Hz.
c. Accuracy Metric: F1 Score/Precision Recall.
d. Should work in rare class scenario: if only one in 1000 images are pixelated then the algorithm must predict at most 10% False Positives.

2. Given a pixelated image, design an algorithm to improve the quality of the image i.e., restore the lost information. This problem is also known as jpeg restoration. Here also the challenge is to design a highly efficient algorithm that can run at least at 20 FPS. The quality of restoration can be examined by comparing with ground truth image using any metric like LPIPS, PSNR etc. If a non-pixelated image is given then the algorithm should not enhance it and leave it intact. Algorithm should work on 1080p resolution images.

Unique Idea Brief (Solution)

- The use of a convolutional neural network (CNN) architecture, with multiple convolutional and max-pooling layers, suggests that the model is designed to learn features and patterns from the input images.
- The input shape of the model indicates that it is designed to work with color images (3 channels), and the final dense layer with a single unit and a sigmoid activation suggests that the model is performing a binary classification task, likely to determine whether an image is blurred or not.
- The unique aspect of this model is the use of a CNN architecture to tackle the image restoration or deblurring problem. CNNs have proven to be effective in learning and extracting relevant features from images, which can then be used to restore or enhance the quality of the input images.
- The model is likely trained on a dataset of blurred and sharp images, and the goal is for the model to learn the mapping between the blurred and sharp versions of the images. Once trained, the model can then be used to predict the sharp version of a new, blurred input image, effectively deblurring or restoring the image.

Features Offered

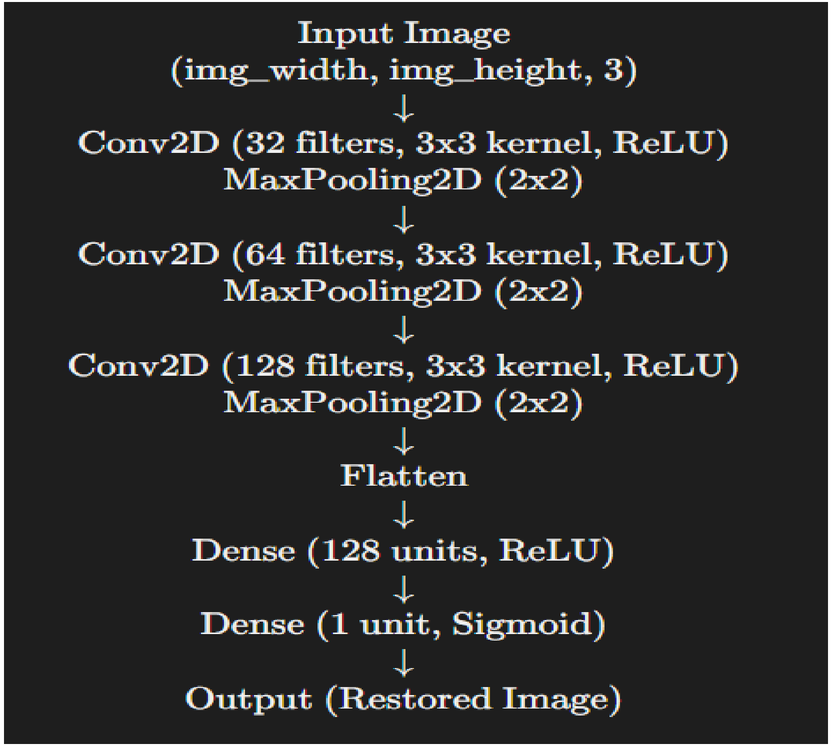
- **Convolutional Neural Network (CNN) Architecture**: The model uses a CNN architecture, which is well-suited for learning and extracting relevant features from images. CNNs have proven to be effective in various image-related tasks, including image restoration and enhancement.
- **Multi-level Feature Extraction**: The model consists of three convolutional layers, each followed by a max-pooling layer. This multi-level feature extraction approach allows the model to learn a hierarchical representation of the input images, capturing both low-level and high-level features.
- **Binary Classification**: The final layer of the model is a dense layer with a single unit and a sigmoid activation, indicating that the model is performing a binary classification task. This suggests that the model is trained to classify images as either "blurred" or "sharp," which is a common approach for image restoration problems.
- **Optimization and Loss Function**: The model is compiled with the Adam optimizer and binary cross-entropy loss, which are commonly used for binary classification tasks. This combination helps the model learn to minimize the error between the predicted and true labels, ultimately improving the model's performance in restoring or deblurring the input images.

Processflow

- **Input Image Preprocessing**: The model is designed to accept input images of a specific size, as defined by the input_shape parameter in the first convolutional layer. This suggests that the input images need to be preprocessed to match this expected size.
- **Convolutional Neural Network (CNN) Architecture**: The model uses a CNN architecture, which is well-suited for image-related tasks. The model consists of three convolutional layers, each followed by a max-pooling layer. These layers are responsible for extracting relevant features from the input images.
- **Feature Extraction**: The convolutional and max-pooling layers of the CNN architecture are designed to learn and extract meaningful features from the input images. These features can be used to distinguish between pixelated and non-pixelated images.
- **Flattening and Dense Layers**: After the convolutional and max-pooling layers, the model flattens the feature maps and passes them through two dense layers. The first dense layer has 128 units and a ReLU activation, which helps the model learn more complex representations of the extracted features. The second dense layer has a single unit and a sigmoid activation, which outputs a binary classification result.

- **Binary Classification:** The final dense layer with a sigmoid activation suggests that the model is performing a binary classification task. The output of this layer can be interpreted as the probability of an input image being pixelated or not.
- **Model Compilation and Training**: The model is compiled with the Adam optimizer and binary cross-entropy loss, which is appropriate for a binary classification problem. The model can then be trained on a dataset of pixelated and non-pixelated images to learn the mapping between the input images and their corresponding classification labels.
- **Pixelated Image Restoration**: Once the model is trained, it can be used to detect and restore pixelated images. The model can take a pixelated image as input, and the output of the final dense layer can be used to determine whether the image is pixelated or not. If the image is classified as pixelated, the model can then be used to restore the image by applying techniques such as super-resolution or image deblurring.

Architecture Diagram



Technologies Used:

- **<u>Convolutional Neural Network (CNN) Architecture</u>**:The model uses a CNN architecture, which is a widely-used deep learning technique for image-related tasks. CNNs are effective at learning and extracting relevant features from images, making them suitable for image restoration and enhancement.
- **<u>Convolutional Layers:</u>**The model consists of three convolutional layers, each with 32, 64, and 128 filters, respectively. Convolutional layers are responsible for learning the local features and patterns in the input images.
- **<u>Activation Function:</u>**The model uses the ReLU (Rectified Linear Unit) activation function in the convolutional layers, which introduces non-linearity and helps the model learn more complex representations.
- **<u>Max-Pooling Layers:</u>**The model includes max-pooling layers after each convolutional layer, which help to reduce the spatial dimensions of the feature maps and introduce some degree of translation invariance.

- **Flattening and Dense Layers:** The flattening layer converts the 2D feature maps into a 1D vector, which is then fed into the dense (fully connected) layers. The dense layers learn the higher-level relationships between the extracted features.
- **Binary Classification:** The final dense layer has a single unit with a sigmoid activation, indicating that the model is performing a binary classification task. This suggests that the model is trained to classify images as either pixelated or not pixelated.
- **Optimization and Loss Function:** The model is compiled with the Adam optimizer and binary cross-entropy loss, which are common choices for binary classification problems.

Conclusion:

- The model uses a convolutional neural network (CNN) architecture, which is well-suited for image-related tasks. The use of multiple convolutional and max-pooling layers allows the model to learn and extract relevant features from the input images.
- The input shape of the model indicates that it is designed to work with color images (3 channels), which is typical for image restoration or enhancement tasks.
- The final dense layer with a single unit and a sigmoid activation suggests that the model is performing a binary classification task, likely to determine whether an image is pixelated or not.
- The model is compiled with the Adam optimizer and binary cross-entropy loss, which is a common setup for binary classification problems.
- The goal of this model is to learn the mapping between pixelated and restored versions of the input images. Once trained, the model can then be used to predict the restored version of a new, pixelated input image, effectively deblurring or restoring the image.