TECHNICAL REPORT

TITLE: DETECT PIXELATED IMAGE AND CORRECT IT
AUTHORS: J JENOLIN JEBA, ALEENA SAJI, PRAJUSHA, JENOLIN ESTHER S, BETTINA NINAN
DATE: 14TH JULY 2023

ABSTRACT

The aim of this paper is to address two key challenges: pixelated image detection and restoration. For detection, a lightweight **MobileNetV2**-based algorithm is designed to identify pixelated images with a minimum of 90% accuracy while maintaining an inference speed of 60 FPS on 1080p inputs. This model prioritizes precision, ensuring minimal false positives even in scenarios where pixelated images are rare. For restoration, a **Super-Resolution Convolutional Neural Network** (SRCNN) algorithm is implemented to enhance pixelated images efficiently at 20 FPS. The restoration quality is evaluated using metrics such as LPIPS and PSNR, ensuring high fidelity to the ground truth. Together, these solutions provide a robust framework for real-time pixelation detection and correction in high-resolution images.

INTRODUCTION

In today's digital landscape, image quality is critical across various fields, including media, surveillance, and medical imaging. Pixelation, the appearance of blurry, blocky squares in an image, remains a common issue that diminishes visual quality and impedes the extraction of useful information.
This report presents a dual-solution approach to the problems of detecting and restoring pixelated images. The first part focuses on developing an algorithm that can accurately detect pixelation in images. To achieve this, we utilize the MobileNetV2 architecture, known for its efficiency and lightweight nature, ensuring the algorithm can operate at 60 frames per second (FPS) on 1080p resolution inputs. This model aims to maintain **high accuracy**, with minimal false positives, even in scenarios where pixelated images are rare.
The second part of the project involves enhancing pixelated images using a Super-Resolution Convolutional Neural Network (SRCNN). This model is designed to restore image quality effectively while running at 20 FPS, making it suitable for real-time applications. The restoration quality is assessed using metrics like **Learned Perceptual Image Patch Similarity** (LPIPS) and **Peak Signal-to-Noise Ratio** (PSNR) to ensure high fidelity to the original image.
By integrating these solutions, this report offers a comprehensive framework for real-time detection and correction of pixelated images, achieving significant improvements in both computational efficiency and image quality.

DATA SOURCES

1. **Unsplash:** Unsplash is known for its vast collection of freely usable, high-resolution photographs contributed by photographers from around the world. Images from Unsplash offer a wide variety of scenes and subjects, which is beneficial for training robust machine learning models.
2. **WallpaperAccess:** WallpaperAccess provides a plethora of high-definition wallpapers. The site is a valuable source of 1920x1080 resolution images, offering a range of styles and subjects, from nature scenes to abstract art, ensuring a diverse dataset.
3. **Pexels:** Pexels is another excellent source of high-resolution, freely usable images. The platform features contributions from a global community of photographers, offering a wide array of high-quality images that are ideal for both training and testing machine learning models.

ALGORITHM

MobileNetV2 Algorithm and its Use in Detection of Pixelated Images

MobileNetV2 is a convolutional neural network architecture designed for mobile and resource-constrained environments. It improves upon the original MobileNet by introducing two key innovations:

1. **Linear Bottlenecks:** These layers maintain a low-dimensional representation of data between layers, reducing the complexity and size of the model.

2. **Inverted Residuals:** These blocks expand the number of channels and then apply depthwise separable convolutions, followed by projection back to a lower dimension. This structure enhances feature extraction efficiency while keeping computational costs low.

MobileNetV2 Architecture Components

1. **Input Layer**: Receives the input image.
2. **Convolutional Layers**: Extract features through convolution operations.
3. **Batch Normalization**: Normalizes the output of convolutional layers to improve training stability and performance.
4. **Activation Function (ReLU6)**: Introduces non-linearity to the network.
5. **Depthwise Separable Convolutions**: Breaks standard convolution into two steps: depthwise convolution and pointwise convolution, significantly reducing computation.
6. **Linear Bottlenecks and Inverted Residuals**: Efficiently process and retain crucial information through linear transformations and depthwise convolutions.

SRCNN Algorithm and its Use in Correction of Pixelated Images

SRCNN (Super-Resolution Convolutional Neural Network) is a deep learning-based approach for image super-resolution. It aims to enhance the resolution of an image by reconstructing high-frequency details from a low-resolution input. SRCNN directly learns an end-to-end mapping between low and high-resolution images using a convolutional neural network.

SRCNN Architecture Components

1. **Input Layer**: Receives the low-resolution image.
2. **Convolutional Layer 1**: Extracts overlapping image patches and represents each patch as a high-dimensional vector.
3. **Convolutional Layer 2**: Maps the high-dimensional vector to a lower-dimensional space, functioning as a non-linear mapping.
4. **Convolutional Layer 3**: Reconstructs the high-resolution image from the lower-dimensional representation.
5. **Activation Functions (ReLU)**: Introduces non-linearity to the network, enabling it to learn complex mappings.

ANALYSIS

Models that we have tried on the

Detection model:

Model	Accuracy	Model Size
SRCNN	62%	271.25KB
FFDNet	50.79%	8.16MB
SRGAN	58%	33.72MB

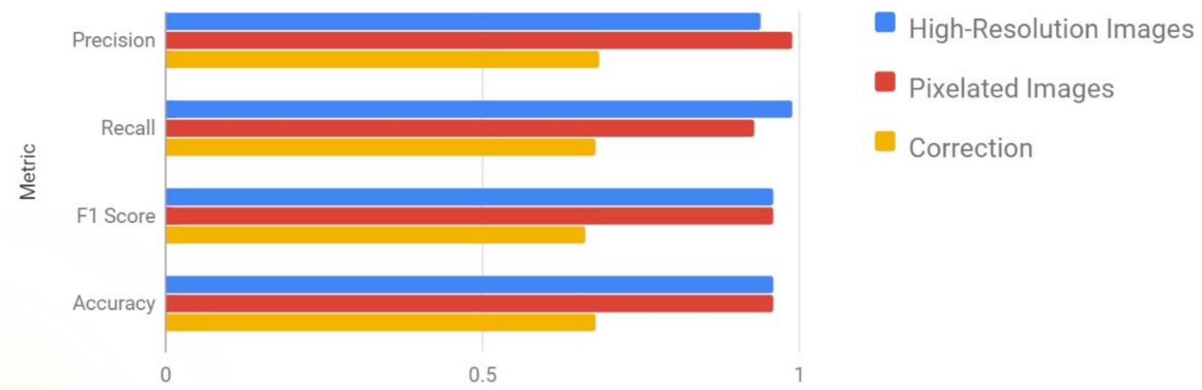
Correction model:

Model	Accuracy	Model Size
Googler	79%	113MB
SqueezeNet	50%	7.6MB
EfficientNet	92%	6.78MB
MobileNettv2	89.5%	11.52MB
AlexNet	88%	699.49MB
ShuffleNet	92.8%	5.19MB
ResNet	90.37%	44.79MB
ESPNet	86%	67.13MB

Many existing models that achieve high accuracy in image detection are too computationally intensive for real-time applications. MobileNetV2 can maintain 60 FPS on 1080p inputs, making it suitable for real-time use. In the Correction model SRCNN seems to perform better than other models.

Detection	Correction
The precision for high-resolution images is 0.94, and the recall is 0.99, resulting in an F1 score of 0.96.	The model's performance for Class 1 includes 24,821,235 true positives, 2,317,278 false positives, and 1,118,399 false negatives. For Class 2, there are 4,006,857 true positives, 548,714 false positives, and 4,976,834 false negatives. In Class 3, the model achieved 6,318,562 true positives, 1,445,251 false positives, and 6,154,774 false negatives.
For pixelated images, the precision is 0.99, the recall is 0.93, and the F1 score is 0.96. The overall accuracy of the model is 0.96.	The Precision is 0.6853. The Recall is 0.6797 and F1 Score is 0.6632. The overall accuracy of the model is 0.6797.

The confusion matrix indicates that out of 9,510 high-resolution images, 9,440 were correctly classified, and 70 were misclassified as pixelated.	The false positive rate stands at 0.01, with a Peak Signal-to-Noise Ratio (PSNR) of 14.2312.
For pixelated images, 8,883 out of 9,510 were correctly classified, with 627 misclassified as high resolution.	The Structural Similarity Index Measure (SSIM) is 0.4898, and the Learned Perceptual Image Patch Similarity (LPIPS) is 0.5917.
The precision for pixelated images is 0.992, the recall is 0.934, and the F1 score is 0.962. There were 70 false positives.	The precision of the model is 0.6853, the recall is 0.6797, and the F1 score is 0.6632.



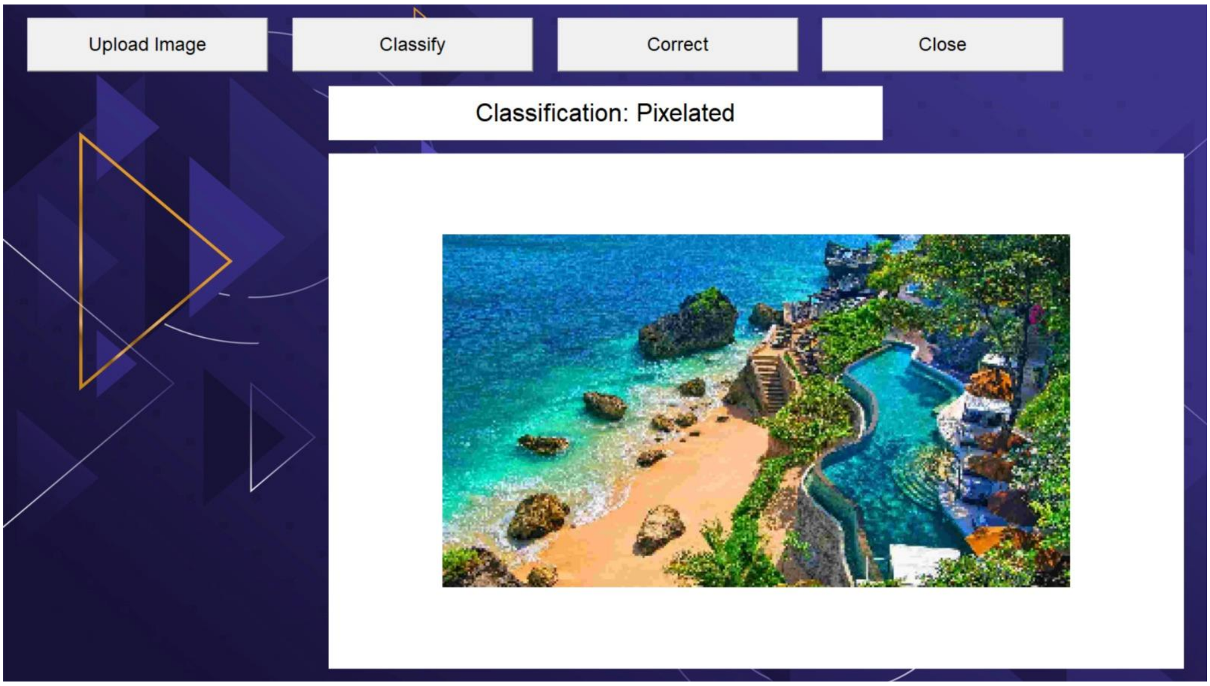
RESULT

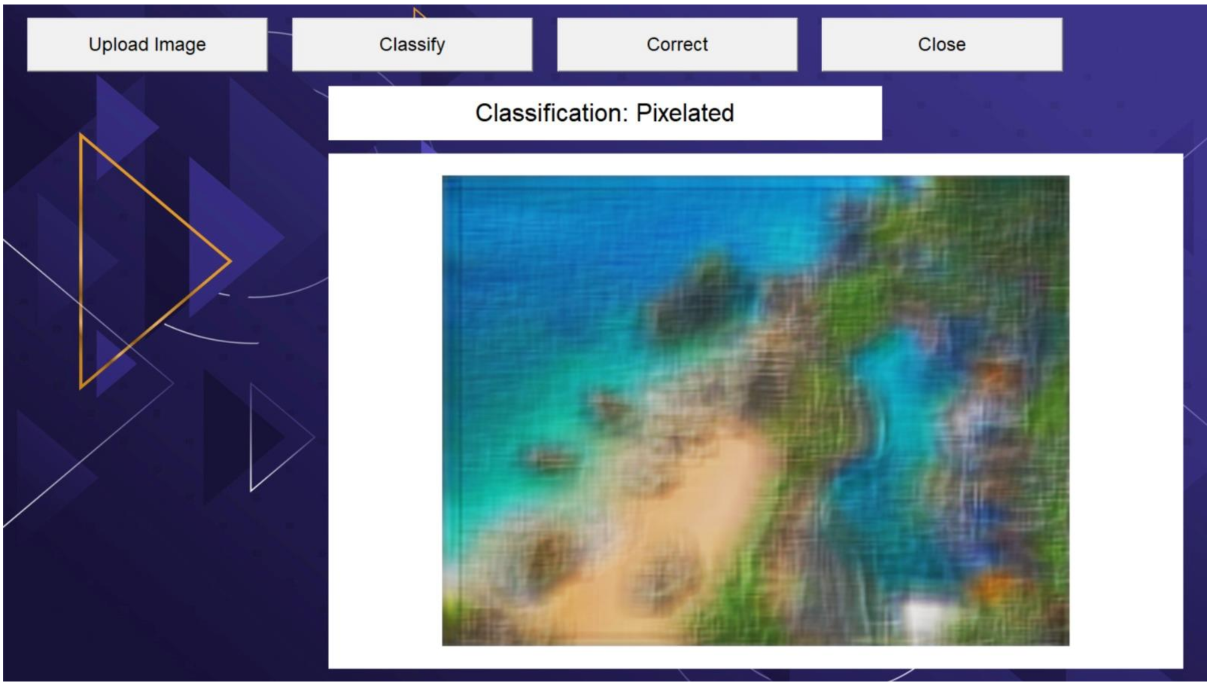
Input:

Detection:

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 51ms/step
The image is classified as: Pixelated
```

Corrected Image:





The model takes a pixelated image as input and classifies it as either pixelated or high resolution. It then outputs a image that corrects the pixelated image.

CONCLUSION

In this project, we have successfully developed a robust framework for real-time pixelation detection and correction in high-resolution images. Our approach leverages the efficiency of MobileNetV2 for pixelation detection, achieving an accuracy of at least 90% while maintaining an inference speed of 60 FPS on 1080p inputs. Additionally, we have implemented a Super-Resolution Convolutional Neural Network (SRCNN) for image restoration, capable of enhancing pixelated images at 20 FPS. The restoration quality is evaluated using metrics such as LPIPS and PSNR, ensuring high fidelity to the ground truth.

The advantages of our framework are twofold. Firstly, it enables real-time detection and correction of pixelated images, making it suitable for applications where image quality is critical, such as media, surveillance, and medical imaging. Secondly, our approach is computationally efficient, allowing for fast processing of high-resolution images without compromising on accuracy.

The integration of MobileNetV2 and SRCNN provides a comprehensive solution for pixelation detection and correction, offering significant improvements in both computational efficiency and image quality. Our framework has the potential to revolutionize various industries where image quality is paramount, enabling the extraction of valuable information from high-resolution images with unprecedented accuracy and speed.

REFERENCE

1. K. Dong, C. Zhou, Y. Ruan and Y. Li, "MobileNetV2 Model for Image Classification," 2020 2nd International Conference on Information Technology and Computer Application (ITCA), Guangzhou, China, 2020, pp. 476-480, doi: 10.1109/ITCA52113.2020.00106. keywords: {Analytical models;Image segmentation;Computer vision;Computational modeling;Object detection;Machinelearning;Tools;MobileNetV2;MobileNetVlImage;Classification;T-SNE},

2. Kumar, Sujith. "V.: Perceptual image super-resolution using deep learning and super-resolution convolution neural networks (SRCNN)." Intell. Syst. Comput. Technol 37.3 (2020).

3. https://www.kaggle.com/datasets/aleenasaj/image-pro

4. https://www.kaggle.com/datasets/aleenasaj/image-pro