DETECT PIXELATED IMAGES AND CORRECT IT

PROBLEM STATEMENT

- The task involves both image classification (detecting pixelation) and image generation (restoring pixelated images).
- The detection algorithm must run at 60 FPS, while the restoration algorithm should achieve 20 FPS.
- Both algorithms should be highly accurate, with detection having a focus on minimizing false positives
- Algorithms should work seamlessly with 1080p images, potentially downscaling for processing.
- Evaluation based on F1-score, precision-recall for detection, and LPIPS, PSNR for restoration

UNIQUE IDEA BRIEF (SOLUTION)

Detection model

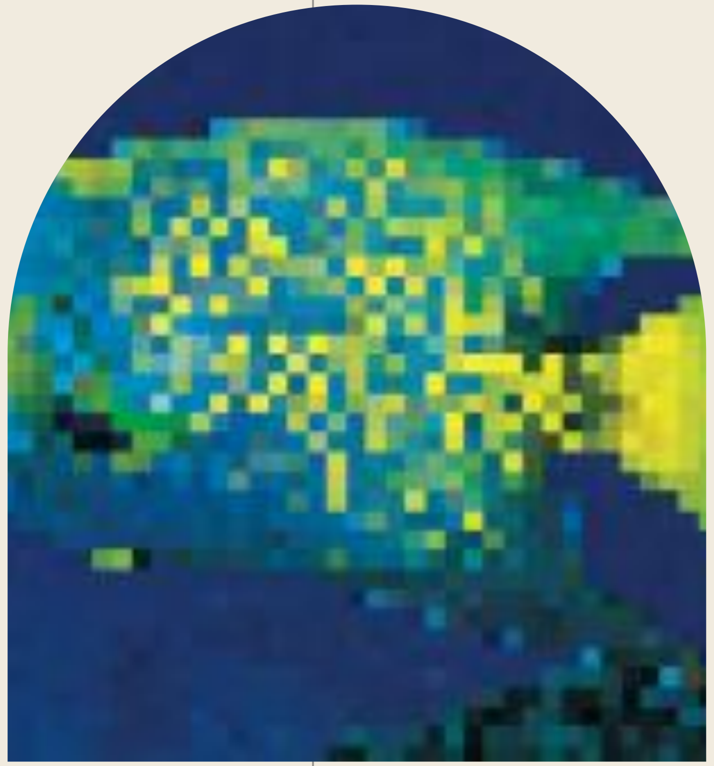
- MobileNet is a deep learning model designed specifically for efficient image classification and other vision tasks on mobile and embedded devices.
- This model uses depthwise seperable convolution approach which reduces the number of parameters and computational cost with higher accuracy
- MobileNet is designed to have a small model size, which is crucial for deployment on devices with limited storage and memory and comes with pre trained weights on large datasets like imagenet
- Lower memory consumption

High resolution

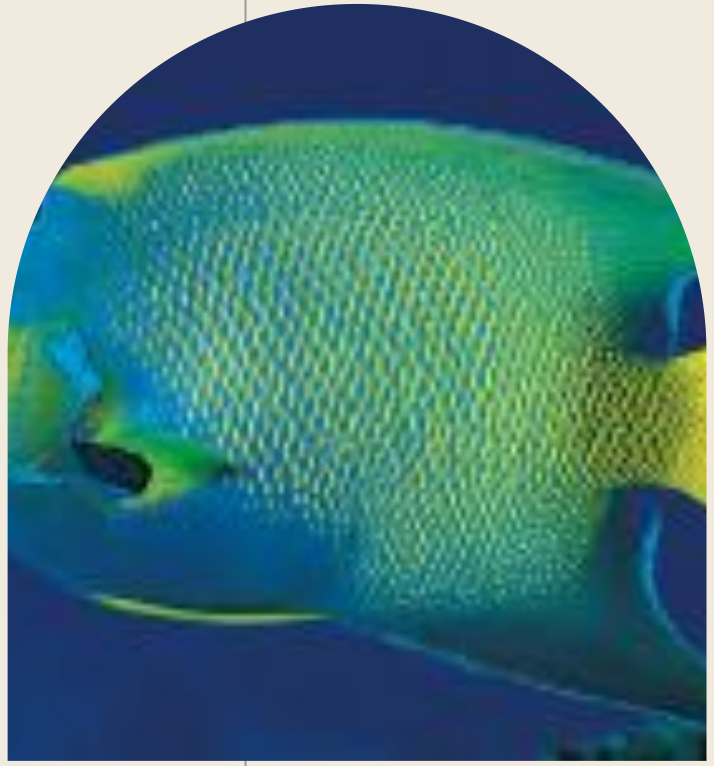
Pixelated

UNIQUE IDEA BRIEF (SOLUTION)
Correction model

- The use of SRCNN(Super-Resolution Convolutional Neural Network) model is a deep learning-based approach for image super-resolution. It consists of a series of convolutional layers designed to learn the mapping between low-resolution (pixelated) and high-resolution images.
- The use of a deeper network architecture with varying filter sizes for different layers.
- A learning rate scheduler to improve model training efficiency.
- Handling and preprocessing of RGB images to ensure uniform input for the model.



pixelated



corrected

FEATURES OFFERED

Detection model

- Utilizes depthwise seperable convolutions to reduce the computational cost and number of parameters compared to traditional CNN's.
- Width and resolution multiplier helps in decreasing the computational complexity and allowing for a balance between speed and accuracy
- Small model size making it suitable for real time applications
- Due to its lightweight and efficient nature, MobileNet can be deployed on mobile phones, tablets and embedded devices, extending its use beyond powerful desktop environments and can be converted and used with other frameworks such as PyTorch, enabling flexibility in model deployment..

FEATURES OFFERED

Correction model

- Allowing flexibility in input dimensions
- Adam optimizer, known for its efficient handling of sparse gradients _and adaptive learning rate, which speeds up training and ensures _better convergence.
- Consists of multiple convolutional layers with varying filter sizes to capture different levels of image features
- The trained model is saved in HDF5 format, allowing for easy loading and deployment in future applications.

PROCESS FLOW

MAY 25

Dataset Preparation

Each members of our team started to collect images with 1920 x 1080 resolution. Then processing the images to 10%,20% and upscaling and downscaling of images were done to get a variety in the dataset.

JUNE 5

ML Algorithm for Detection

Tried with SqueezeNet, EfficientNet, MobileNetV2, AlexNet, ShuffleNet, CNN, SVM, ResNet, ESPNet and from these models MobileNetV2 was finalized

JUNE 18

ML Algorithm for Correction

Tried with SRCNN, FFDNet, FBCNN, EDSR, VDSR, SRGAN, ESRGAN and from these models MobileNetV2 was finalized

JULY 1

Preparation of GUI

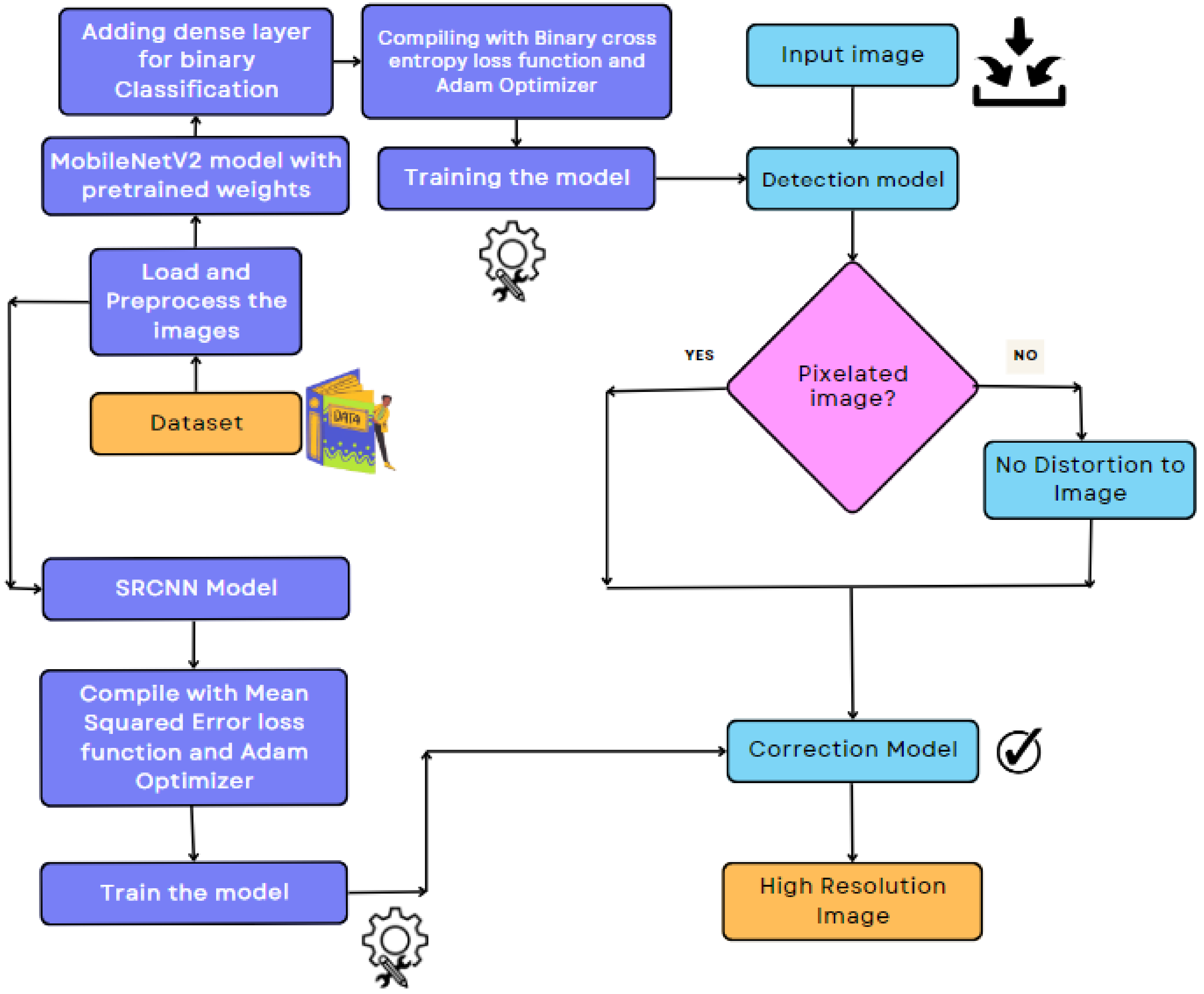
Preparing a Simple GUI to test our models and performing Subjective Analysis and Metric Evaluation

JULY 4

Documentation

Preparing the results, inferences obtained from the project has been noted down down and documented.

ARCHITECHTURE DIAGRAM



SOURCE CODE LINKS

https://github.com/jenojeba/Detection-and-Correction-of-Pixelated-Images

Datasets:

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

Output with metric evaluation

Detection model

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

```
Classification Report:
                 precision    recall  f1-score   support

High resolution       0.94      0.99      0.96      9510
      Pixelated       0.99      0.93      0.96      9510

       accuracy                           0.96     19020
      macro avg       0.96      0.96      0.96     19020
   weighted avg       0.96      0.96      0.96     19020

Confusion Matrix:
[[9440   70]
 [ 627 8883]]
Precision: 0.9921813917122753
Recall: 0.9340694006309148
F1 Score: 0.9622488219682608
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 2s 2s/step
Image classified as Pixelated with FPS: 0.55
```

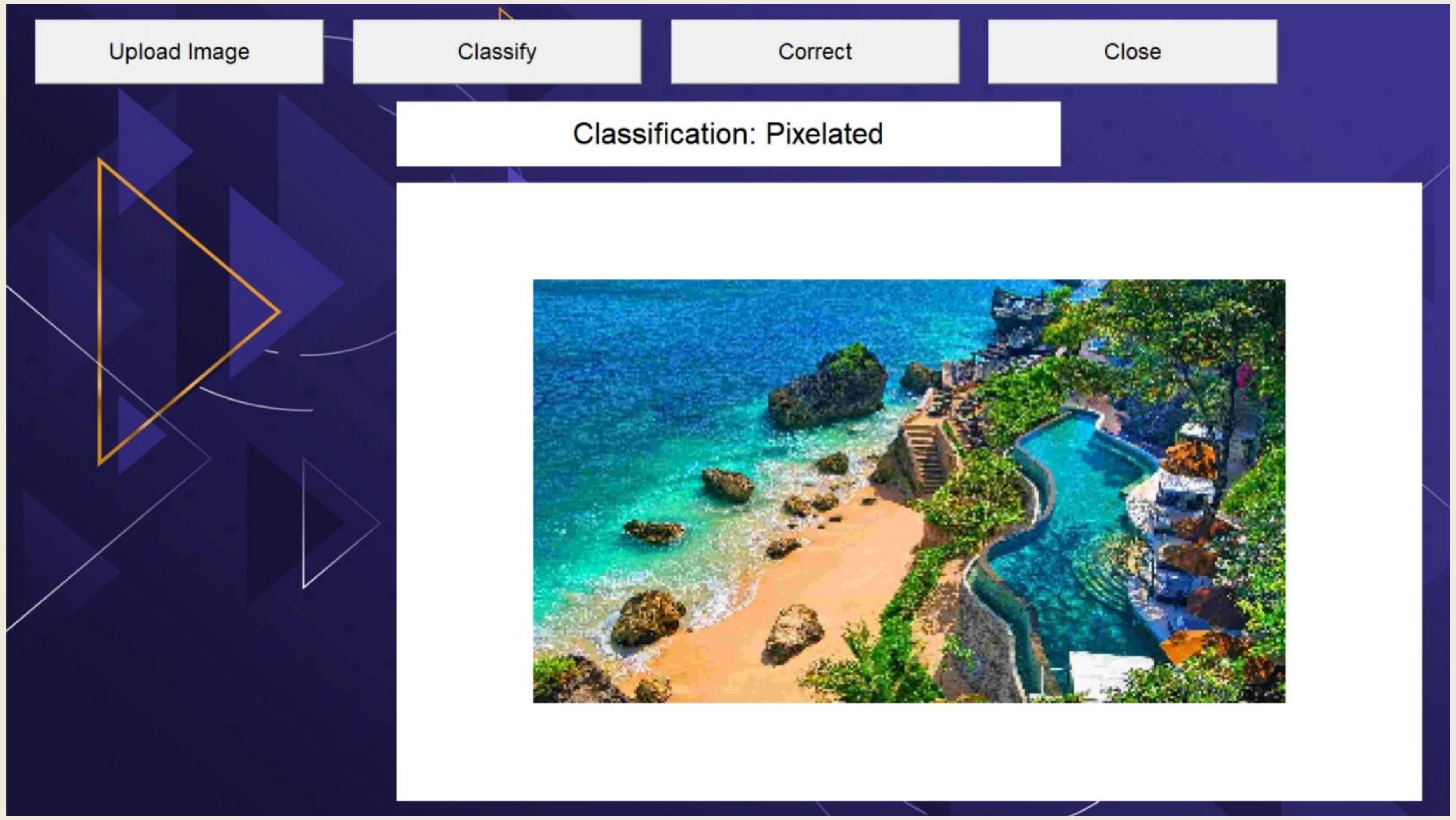
Output with metric evaluation -Correction model

Pixelated Image

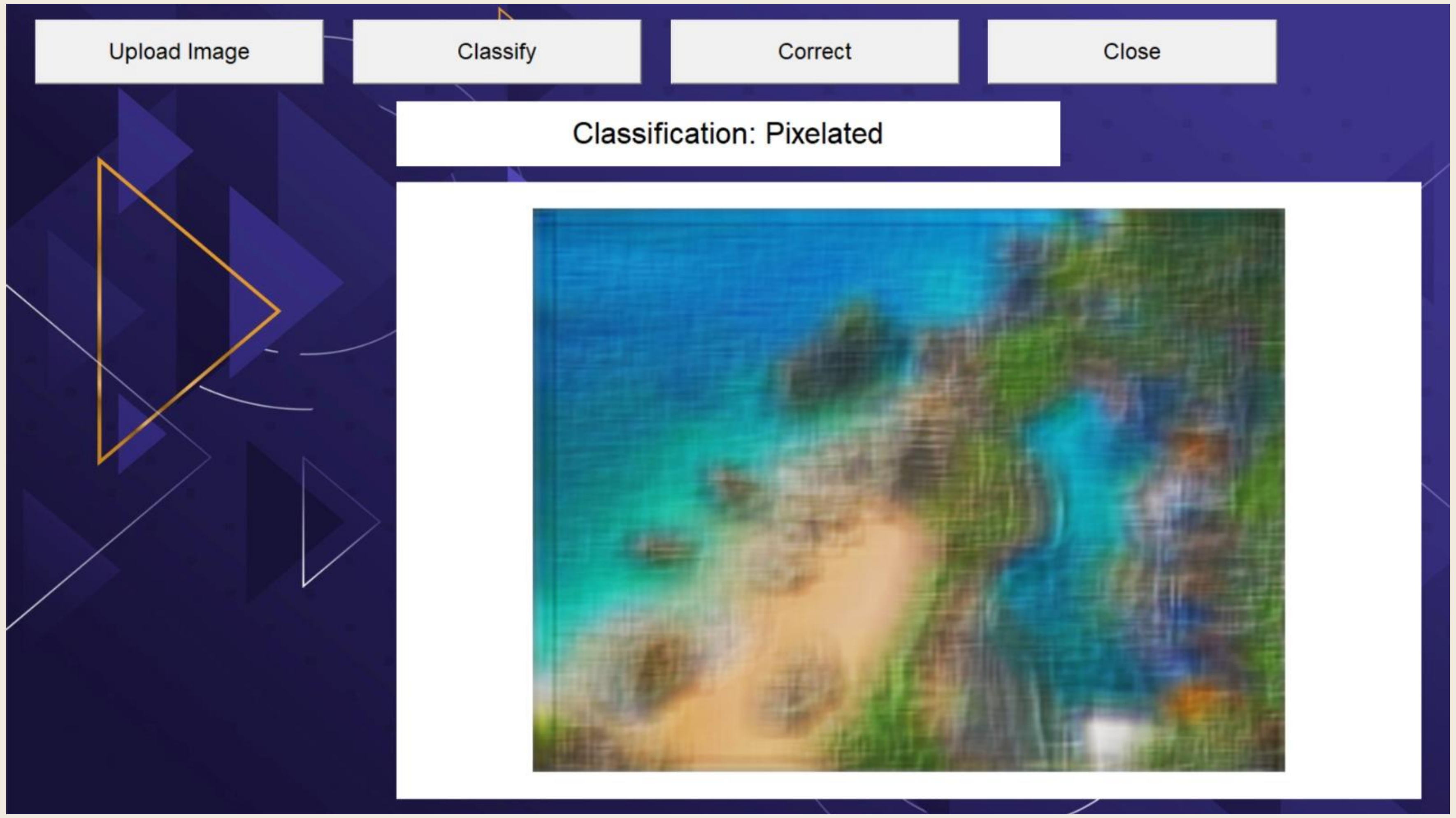
Corrected Image

```
Confusion Matrix:
[[24821235  2317278  1118399]
 [ 4976834  4006857   548714]
 [ 6154774  1445251  6318562]]
Precision: 0.6853
Recall: 0.6797
F1 Score: 0.6632
Accuracy: 0.6797
FPS: 0.01
PSNR: 14.2312
SSIM: 0.4898
LPIPS: 0.5917
```

GRAPHICAL USER INTERFACE



Detection



Correction

Technologies used

The Models uses:

- Python as it core programming language.
- TensorFlow and Keras as frameworks for building and training deep learning models
- NumPy for numerical operations and handling arrays.
- Pillow (PIL) for image manipulation
- Scikit-Learn for data splitting and evaluation.
- OS and Shutil for file and directory operations.
- CNNs (Convolutional Neural Networks) for processing and enhancing images.

- The use of Kaggle Notebooks provide an integrated environment for data science and machine learning, offering several features that facilitate the development, testing, and sharing of machine learning models.

Subjective Analysis

- The Correction model has a mean opinion score of 3.9/5
  (100 images were obtained from the model and the images were rated out of 5)

- The Detection model has a mean opinion score of 4.39/5
  (100 images were obtained from the model and the images were rated out of 5)

Team members and contribution

J Jenolin Jeba (Leader) : Collected and curated dataset images, developed multiple machine learning models utilizing various algorithms like SqueezeNet, EfficientNet, MobileNet, AlexNet , ShuffleNet, ResNet, ESPNet, SRCNN, FFDNet, SRGAN, ESRGAN etc., and independently completed the final ML model notebook, and conducted metric evaluation. The work of documentation was also finished.

Aleena Saji : Contributed a significant portion of the dataset, developed a GUI model for the project, debugged errors in multiple models, prepared ML model with RNN,SVM,SRCNN and assisted with the subjective analysis and documentation.

Team members and contribution

Jenolin Esther S : Contributed to dataset creation , tried making ML model and conducted the subjective analysis.

Prajusha R : Contributed to dataset creation , tried making ML model and conducted the subjective analysis.

Bettina Ninan : Contributed to dataset creation , tried making ML model.

CONCLUSION

- The model demonstrates a strong capability to classify pixelated and high-resolution images effectively.
- The SRCNN model serves as a foundational approach to super-resolution, demonstrating the feasibility of correcting pixelated images.
- By building upon this foundation with further refinements and advanced techniques, the model's capabilities can be significantly enhanced for various high-resolution image reconstruction tasks

REFERENCES

https://wallpaperaccess.com

https://unsplash.com/

https://pexels.com/