Problem Statement

09: Detect Pixelated Image and Correct It

Unique Idea Brief (Solution)

Pixelated Image Detection

We developed a pixelated image detection model using the MobileNetV2 architecture fine-tuned for binary classification. By generating pixelated versions of high-resolution images through downscaling and upscaling techniques, we created a balanced dataset. Cropping, instead of resizing, preserved essential textures, ensuring high detection accuracy. The model, trained with an optimized learning rate and early stopping

Pixelated Image Restoration

We utilized the ESPCN (Efficient Sub-Pixel Convolutional Neural Network) model for image super-resolution. Downscaling pixelated images by a factor of 4 reduced pixelation artifacts, and subsequent upscaling with the ESPCN model enhanced image clarity. The ESPCN model's efficiency and high-quality upscaling capabilities make it ideal for image restoration, preserving original details and textures while reducing pixelation.

Features Offered

Pixelated Image Detection

- **High Accuracy**: Utilizes MobileNetV2 for precise pixelation detection.
- **Balanced Dataset**: Created pixelated versions of high-res images for effective training.
- **Efficient Preprocessing**: Crops instead of resizing to preserve textures.
- **Optimized Training**: Uses Adam optimizer with early stopping for robust performance.
- **Real-World Ready**: Effective for practical pixelation detection tasks.

Pixelated Image Restoration

- **Super-Resolution**: Employs ESPCN model for high-quality image upscaling.
- **Pixelation Reduction**: Downscales to reduce pixelation artifacts, then upscales for clarity.
- **Efficient Processing**: High-quality results with fewer parameters.
- **Quality Preservation**: Maintains original details and enhances clarity.

Process flow

Pixelated Image Detection

Dataset Preparation:

Collect high-resolution images from Div2k and ‘widescreen Images’ dataset from Kaggle.

Create pixelated versions by downscaling (5x or 6x) and upscaling using nearest neighbor or bilinear interpolation.

Ensure a balanced dataset with both pixelated and non-pixelated images.

Preprocessing:

Crop images to 224x224 pixels to preserve texture and detail.

Rescale pixel values for normalization.

Model Training:

Utilize MobileNetV2 architecture pre-trained on ImageNet.

Fine-tune the model with an Adam optimizer (learning rate of 1e-4).

Split dataset into training (80%) and validation (20%) sets with stratification.

Train model with early stopping over 13 epochs and a batch size of 32.

Evaluation:

Assess model performance using accuracy metrics.

Achieve peak validation accuracy of 98.25%.

Process flow

Pixelated Image Restoration

Dataset Preparation:

Collect pixelated images for restoration.

Preprocessing:

Downscale images by a factor of 4 to reduce pixelation artifacts.

Model Application:

Use the ESPCN model to upscale the downscaled images.

ESPCN reorganizes low-resolution feature maps into high-resolution outputs.

Post-Processing:

Evaluate restored images using PSNR and SSIM metrics.

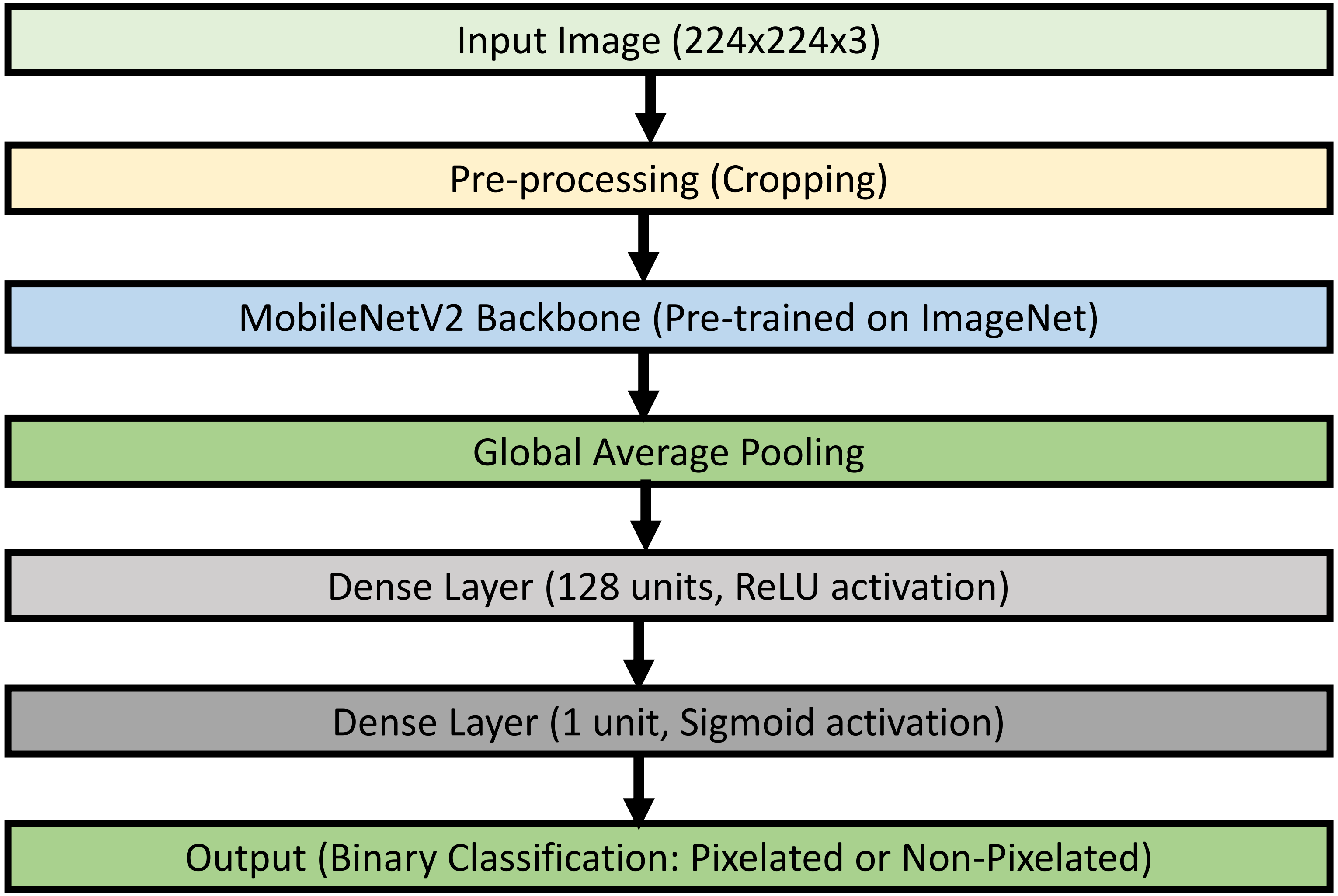
Ensure high visual quality and clarity.

Performance Assessment:

Measure processing speed (FPS) for real-time application feasibility.

Architecture Diagram

Pixelated Image Detection: MobileNetV2-based Binary Classifier



Technologies used

Pixelated Image Detection:

Machine Learning Framework: TensorFlow/Keras for model development and training.

Preprocessing: PIL (Python Imaging Library) for image manipulation tasks.

Model Architecture: MobileNetV2 architecture pre-trained on ImageNet for feature extraction and transfer learning.

Optimization: Adam optimizer for gradient descent optimization during training.

Evaluation Metrics: Accuracy, precision, recall, F1-score for model evaluation.

Dataset Handling: pandas for data manipulation, scikit-learn for dataset splitting and evaluation metrics computation.

Technologies used

Pixelated Image Restoration:

Deep Learning Framework: TensorFlow/Keras for model development and implementation.

Model Architecture: ESPCN (Efficient Sub-Pixel Convolutional Neural Network) for image super-resolution.

Image Processing: OpenCV for image handling and processing tasks.

Evaluation Metrics: PSNR (Peak Signal-to-Noise Ratio), SSIM (Structural Similarity Index) for image quality assessment.

Visualization: Matplotlib for plotting and visual representation of metrics and images.

Team members and contribution:

Pixelated Image Detection:

David Rosario Selvaraj (URK22CS1007)

Shannandhirshath T (URK22CS1045)

Melvin Matthew N (URK22CS1157)

Pixelated Image Correction:

Finney Rajan S (URK22CS1056)

Beneesh S (URK22CS1190)

Conclusion

Pixelated Image Detection:

Our project developed a MobileNetV2-based model for accurately detecting pixelated images. By creating a balanced dataset and using effective preprocessing techniques, we achieved high detection accuracy, essential for applications needing precise identification of pixelation.

Pixelated Image Restoration:

Using the ESPCN model, our project focused on restoring pixelated images by downscaling to reduce artifacts and then upscaling for enhanced clarity. This approach demonstrates ESPCN's efficiency in preserving details while improving overall image quality.