Problem Statement

Detect Pixelated Image and Correct It

Unique Idea Brief (Solution)

This project addresses the problem of pixelated images with a two-part system: Pixelated Image Detection and Pixelated Image Correction.

Pixelated Image Detection:
- **Model:** MobileNetV2 for efficient and accurate pixelation detection.
- **Function:** Determines if an image is pixelated.

Pixelated Image Correction:
- **Model:** DnCNN for restoring pixelated images to high quality.
- **Training:** Uses pairs of original and pixelated images.
- **Output:** Evaluated with PSNR and SSIM for quality assurance.

Benefits:
- **High-Quality Restoration:** Ensures images are restored effectively.
- **Efficiency:** Capable of real-time processing.
- **User-Friendly:** Simple web interface for image correction.

Features Offered

Real-Time Processing:

•**Efficiency:** Capable of running at least 30 frames per second (FPS).
•**Model Size:** Lightweight models under 10 MB for faster processing and deployment.

User-Friendly Web Interface:

•**Upload & Display:** Allows users to upload images, view them on the screen, and check for pixelation.
•**Correction:** Provides a simple button to correct pixelated images.
•**Quality Metrics Display:** Shows PSNR and SSIM values for the corrected image.

Process flow

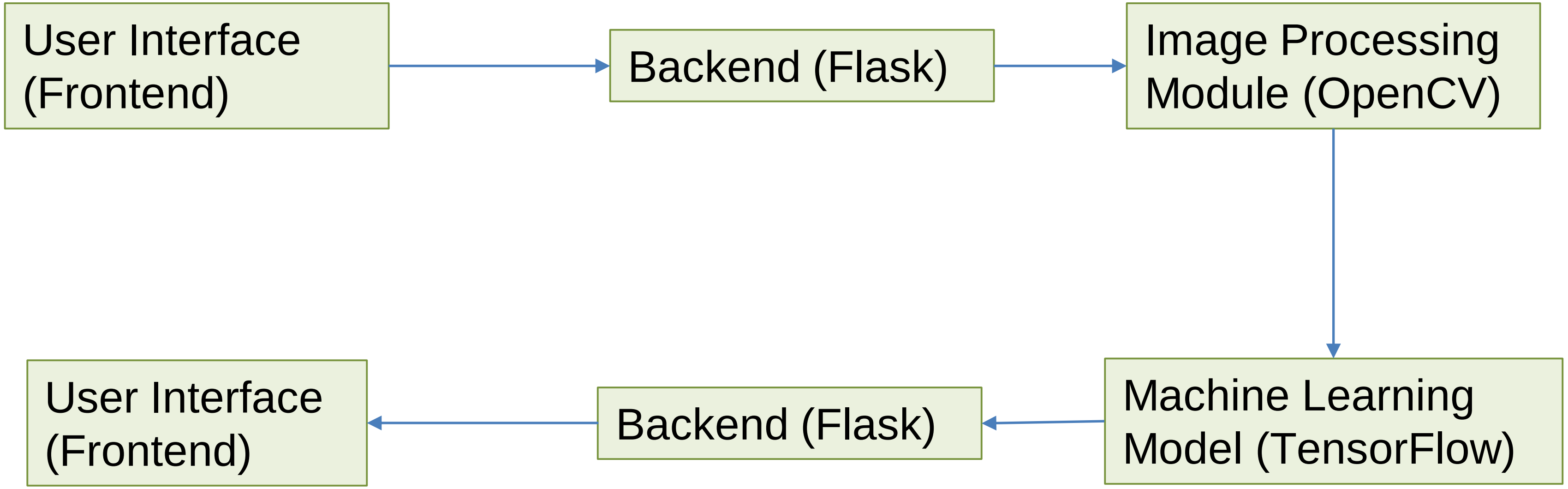
Image Upload:
•**User Action:** User uploads an image via the web interface.
•**Backend Action:** Image is received and preprocessed for model input.

Pixelation Detection:
- MobileNetV2 Convert the image to the required input shape (224x224).
- Normalize the image data.
- Predict if the image is pixelated using the detection model.

Pixelated Image Correction:
- Image is fed into the DnCNN model for correction.
- The corrected image is generated and processed for display.
- Compute PSNR (Peak Signal-to-Noise Ratio).
- Compute SSIM (Structural Similarity Index).

Display Results:
•**Web Interface:**
- Shows the original and corrected images side-by-side.
- Displays PSNR and SSIM metrics.

Architecture Diagram


User Interface (Frontend)
Backend (Flask)
Image Processing Module (OpenCV)
Machine Learning Model (TensorFlow)
Backend (Flask)
User Interface (Frontend)

Technologies used

Backend Framework:
•**Flask:** Utilized as the web framework for handling API requests and responses.

Machine Learning Framework:
•**TensorFlow:** Chosen for its robustness in building and deploying machine learning models.

Development Tools:
•**Python:** The primary programming language used for backend development, machine learning implementation, and image processing

Image Processing and Evaluation:
•**OpenCV:** Employed for various image processing tasks such as pixelated image detection and preprocessing.

Additional Libraries:
•**NumPy, Pandas:** Used for data manipulation and numerical computations.
Matplotlib, Seaborn: Employed for visualizing data and model outputs during development and evaluation phases.

Deployment:
AWS (Amazon Web Services), Google Cloud Platform: For deploying the application, leveraging cloud services for scalability and reliability.

Team member and contribution:

Upendra

•**Role:** Project Lead, Backend Developer, Frontend Developer

•**Responsibilities:**

- Designed and developed the Flask-based backend architecture.
- Integrated TensorFlow for machine learning model loading and inference.
- Implemented image processing techniques for pixelated image detection and correction.
- Designed simple user interface with HTML, CSS, Javascript.

Conclusion

This project addresses the challenge of pixelated image detection and correction using advanced machine learning and image processing techniques. By leveraging Flask for API integration, TensorFlow for model deployment, and OpenCV for image processing, a scalable solution has been developed that showcases expertise in both software engineering and AI technologies.

The project involved overcoming various technical challenges, leading to improved model accuracy and efficiency. Moving forward, there is potential to expand the application's capabilities and explore new AI-driven image enhancement techniques.

Thank you for your attention and interest. Any questions or feedback are welcome.