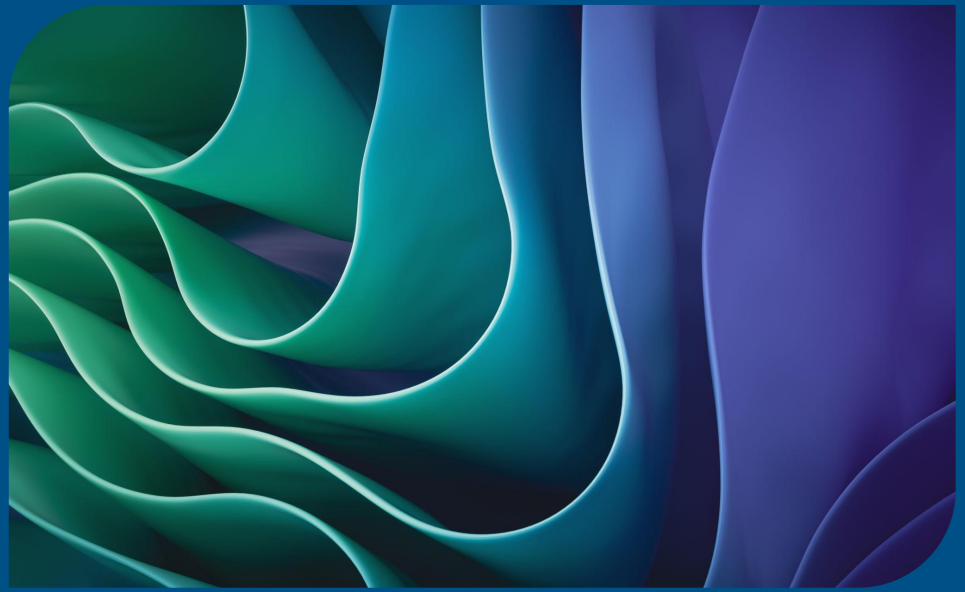
Deep Learning Project Demo



Kranthi Chaithanya Thota
Vishal Singarapu

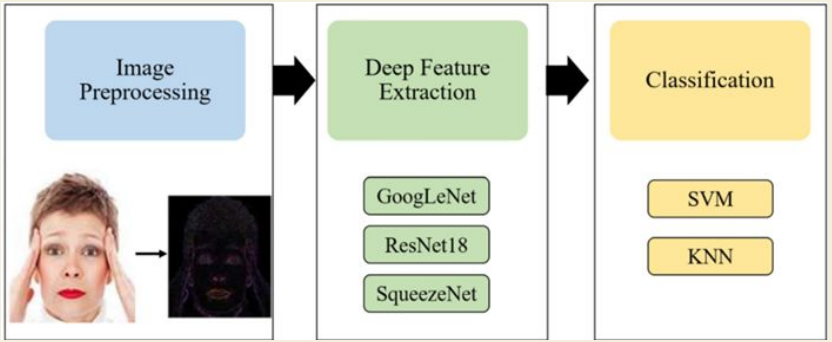
- Introduction
- Methods Used in the Research Paper
- Disadvantages and Scope of Improvement
- Proposed Improvements
- Methodology of Proposed
- Current Implementations
- Results and Comparison
- Next Steps

Introduction

- **Objective:** Enhance deepfake detection through improved feature extraction and model optimization.
- **Background:** Deepfake technologies pose challenges in misinformation and disinformation.
- **Key Focus:**
  - Use advanced CNN architectures.
  - Optimize models for real-time applications.
  - Incorporate ensemble learning.

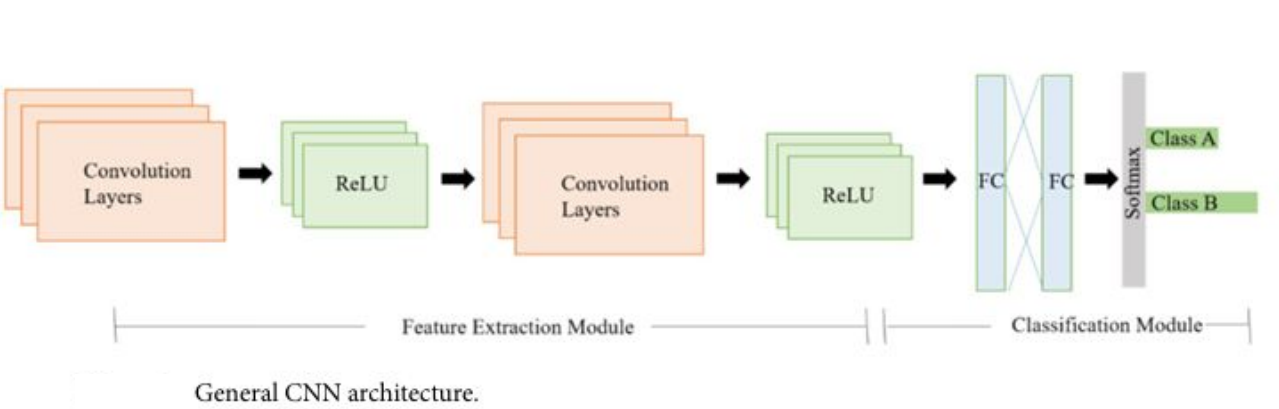
Methods Used in Research

- **Preprocessing:** Error Level Analysis (ELA) for detecting pixel-level manipulations.
- **Feature Extraction:**
  - **Pre-trained CNNs:** ResNet18, GoogLeNet, SqueezeNet.
- **Classification:**
  - **SVM:** Hyperparameter-tuned Gaussian kernel.
  - **KNN:** Optimized neighbors and distance metrics.
- **Evaluation Metrics:** Accuracy, Precision, Recall, F1-score.



Disadvantages and Scope of Improvement

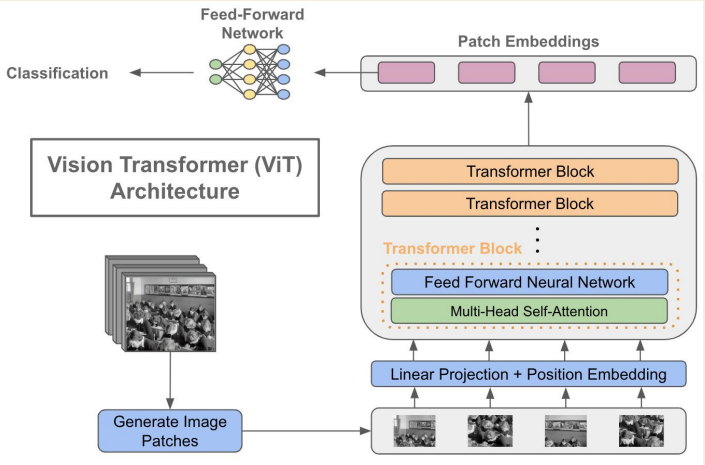
- **Disadvantages:**
  - **Limited architectures:** No exploration of modern CNNs or Vision Transformers.
  - **Computational inefficiencies:** Lack of optimization for real-time use.
  - Simplistic ELA preprocessing.
- **Scope of Improvement:**
  - Incorporate advanced architectures like EfficientNet and ViTs.
  - Implement model optimization techniques like pruning and quantization.
  - Enhance preprocessing for better robustness.



General CNN architecture.

Proposed Improvements

- **Improved Preprocessing:**
  - Adaptive ELA with dynamic scaling and resizing.
- **Modern Architectures:**
  - Use EfficientNet for scalability and efficiency.
  - Explore Vision Transformers for attention-based feature extraction.
- **Real-Time Applications:**
  - Apply pruning and quantization for deployment.
- **Ensemble Learning:**
  - Combine CNN features with classifiers like SVM and KNN.

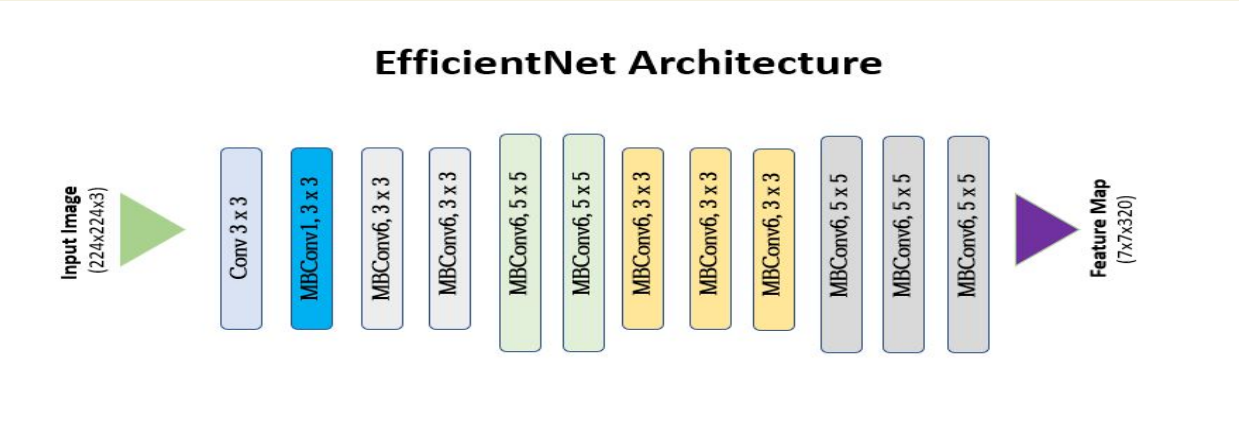


Methodology of Proposed

- **Data Pipeline:**
  - Adaptive ELA preprocessing.
  - Data augmentation for robustness.
- **Modeling:**
  - Implement EfficientNet for deep feature extraction.
  - Integrate Vision Transformers for advanced patterns.
- **Classification:**
  - Train SVM and KNN with optimized hyperparameters.
- **Evaluation:**
  - Measure accuracy, F1-score, precision, and recall.

Current Implementations

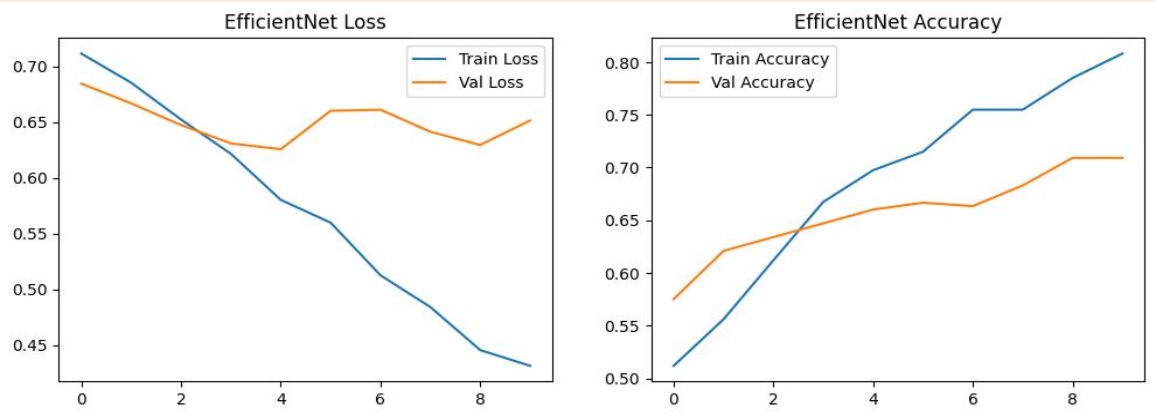
- **Preprocessing:**
  - Adaptive ELA with quality-controlled JPEG resave and resizing.
- **Model:**
  - Pre-trained EfficientNet for feature extraction.
- **Data Handling:**
  - Augmentation via ImageDataGenerator.
  - Train-validation-test split for reproducibility.
- **Metrics:**
  - Setup for performance evaluation (confusion matrix, F1-score).

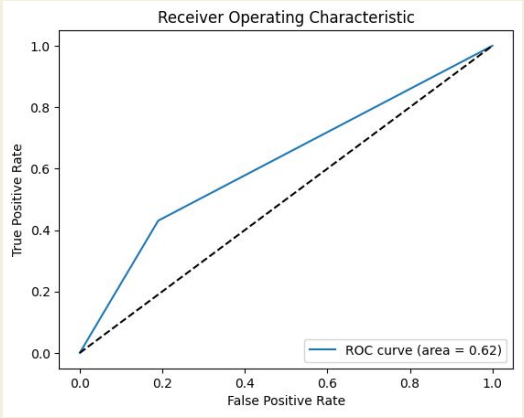


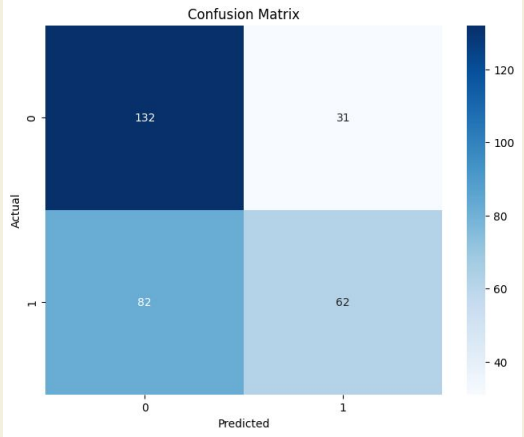
Results and Comparison

- **Current Model:**
  - Adaptive ELA preprocessing.
  - EfficientNet: Faster and scalable.
- **Research Paper:**
  - Multiple CNN architectures evaluated (ResNet18, GoogLeNet, SqueezeNet).
- **Comparison:**
  - Current model excels in preprocessing and computational efficiency.
  - Lacks classifier integration and metric evaluation.

```
Accuracy: 0.63, Precision: 0.67, Recall: 0.43, F1-Score: 0.52
```







Next Steps

- Integrate SVM classifiers for feature classification.
- Apply pruning and quantization for real time use.
- Increase the training time.
- Hyper Parameter tuning.