Deep Fake Detection Using Modern Techniques

Kranthi Chaithanya Thota
Vishal Singarapu

Inspired From : Deep fake detection and classification using error-level analysis **A paper from 2023**

Overview

The rise of deep fakes has introduced significant challenges, including misinformation, fraud, and reputational harm. These synthetic images and videos, generated using sophisticated AI algorithms, blur the line between reality and manipulation.

To address these risks and safeguard trust in digital media, this project leverages a robust approach combining **Error-Level Analysis (ELA)** with advanced deep learning techniques.

By analyzing compression inconsistencies at the pixel level and utilizing modern techniques like EfficientNet and Vision Transformers (ViT), this system provides an effective framework for detecting and classifying deep fake images with high accuracy and reliability.

Dataset

The dataset contains two main folders labeled **training_real** with 1081 images and **training_fake** with 960 images.

The **training_fake** folder mainly contains 3 levels of fake images labelled as easy, mid and hard with different levels of difficulty in recognizing those images.

All images are of 600x600 pixels in size.

Dataset Link :
Real and Fake Face Detection

Proposed Methodology

Error-Level Analysis (ELA):

- Forensic technique to highlight tampered image areas.
- Identifies pixel-level inconsistencies by analyzing compression levels.

Deep Learning Models:

- **EfficientNet**: Lightweight and efficient CNN for feature extraction.
- **Vision Transformer (ViT)**: Uses attention mechanisms for fine-grained feature detection.

Machine Learning Classifiers:

- **K-Nearest Neighbors (KNN)**: Simplifies classification with dimensionality reduction.

Preprocessing

Error Level Analysis (ELA) and Enhancements:

- The adaptive_ela function computes the difference between the original image and a resaved version with a 90% quality. This image is then enhanced to better view and detect the differences.

Image Resizing:

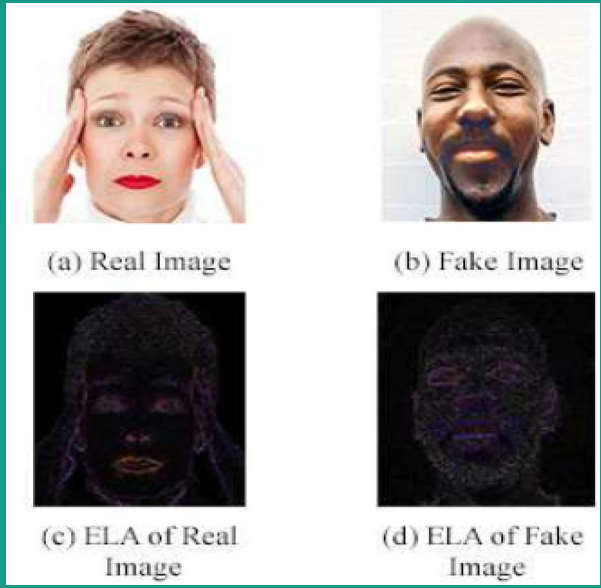
- Resizing the processed ELA image to a fixed dimension of **224x224 pixels**, ensuring compatibility with deep learning models.

Multiprocessing:

- Using multiprocessing to efficiently process a large number of images concurrently.

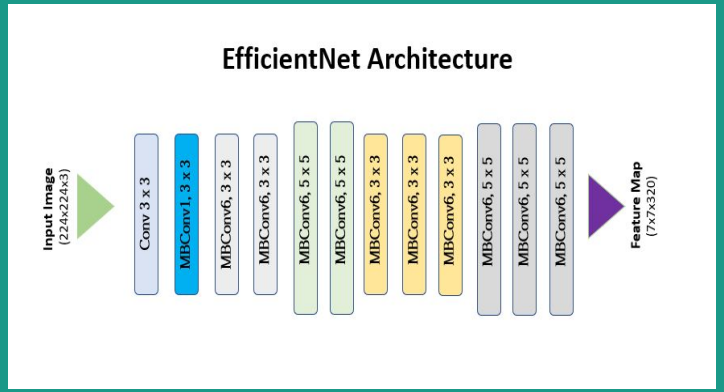
Output:

- The final preprocessed dataset is a NumPy array of resized ELA images, ready for input into the feature extraction models.



EfficientNetV2B0

- A lightweight convolutional neural network optimized for efficiency and accuracy.
- Pre-trained on ImageNet for transfer learning.



Parameters :

Input Shape: **(224, 224, 3)**

Optimizer: **Adam** with a learning rate of **1e-4**.

Loss Function: **Binary Crossentropy** (suitable for binary classification).

Metrics: **Accuracy**.

Regularization:

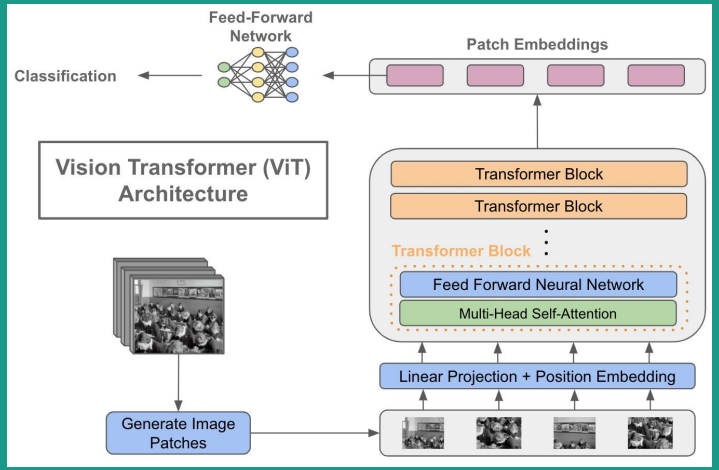
- Dropout: **0.5** for regularizing fully connected layers.
- Fine-tuning: First **100 layers frozen**, allowing transfer learning on deeper layers.

Callbacks:

- **EarlyStopping**: Patience of 7 epochs to prevent overfitting.
- **ReduceLROnPlateau**: Reduces learning rate when validation loss plateaus.
- **ModelCheckpoint**: Saves the model with the highest validation accuracy.

Vision Transformers (ViT)

- Uses self-attention mechanisms to capture global image features, making it ideal for detailed image analysis.
- Pre-trained on ImageNet weights (vit_b_16 model in PyTorch).



Parameters:

- Optimizer: **Adam** with a learning rate of **1e-4**.
- Loss Function: **Binary Crossentropy with Logits**.
- Data Augmentation:
  - Resize to **(224, 224)**.
  - Random Flips, Rotations.
- Batch Size: **32**.
- Epochs: **10**.
- Best Model Saved: Using **torch.save** with validation loss monitoring.

K-Nearest Neighbours (KNN)

- A non-parametric machine learning algorithm that classifies based on proximity to training samples in feature space.
- Applied to PCA-reduced features from EfficientNet and ELA images.

Parameters:

- Number of Neighbors: **5**.
- Distance Metric: **Euclidean distance**.
- Feature Reduction: **Principal Component Analysis (PCA)** reduces dimensions to **200** components, improving computational efficiency.

Ensemble Model

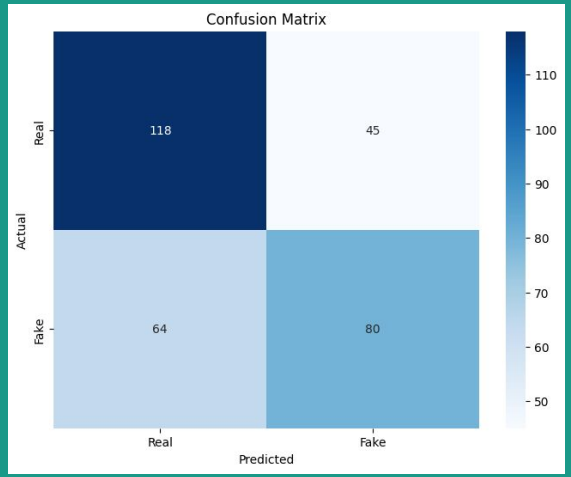
- Combines predictions from EfficientNet, ViT, and KNN using a **majority voting** strategy.
- Enhances robustness by leveraging the strengths of diverse models.

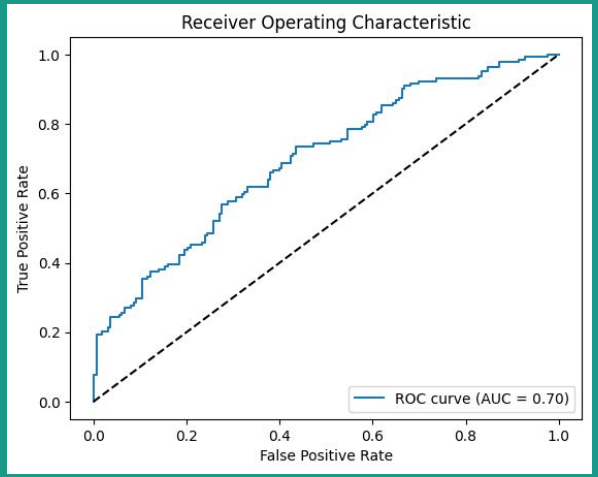
Implementation:

- Probabilities from each model are aggregated.
- Classification is based on the majority vote of predictions.

Results - EfficientNet

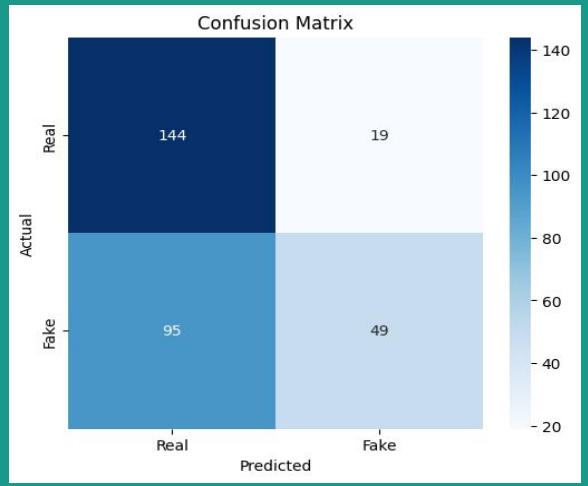
- The AUC value is **0.70**, indicating a moderately effective model in distinguishing between real and fake images.
- Accuracy : 64%
- Precision: 72.4%
- Recall: 64.8%
- F1-Score: 68.4%

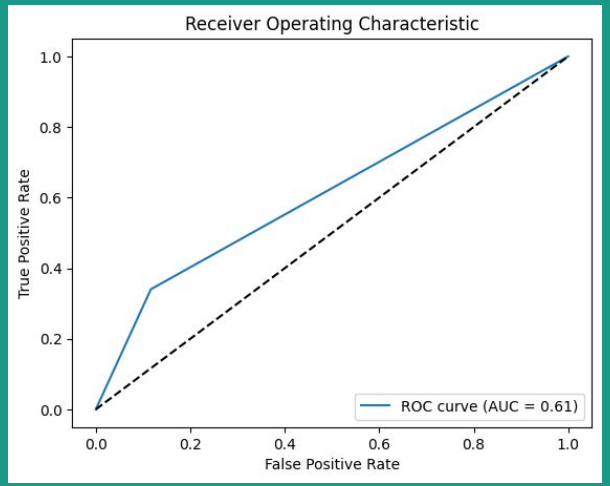




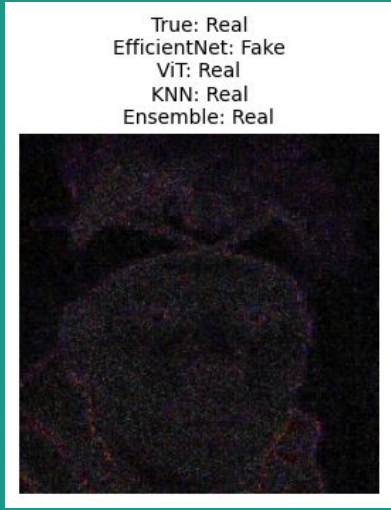
Results - EnsembleNet

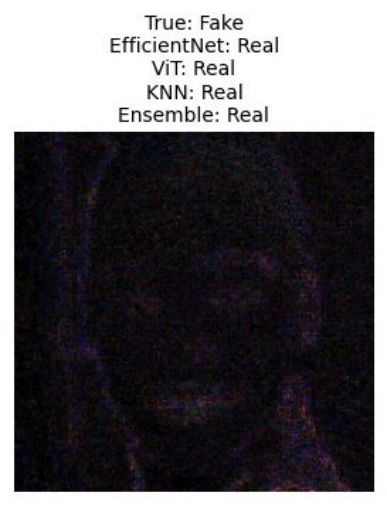
- The AUC value is **0.61**, indicating a moderately effective model in distinguishing between real and fake images.
- Accuracy : 62.9%
- Precision: 88.4%
- Recall: 60.28%
- F1-Score: 71.8%

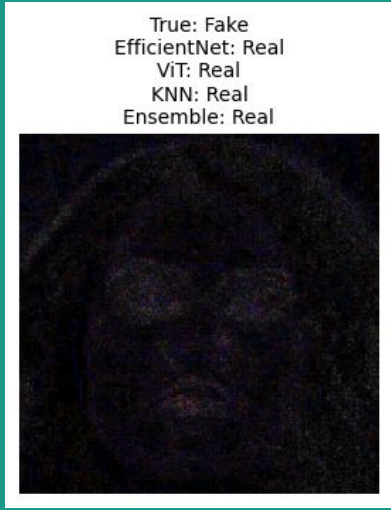


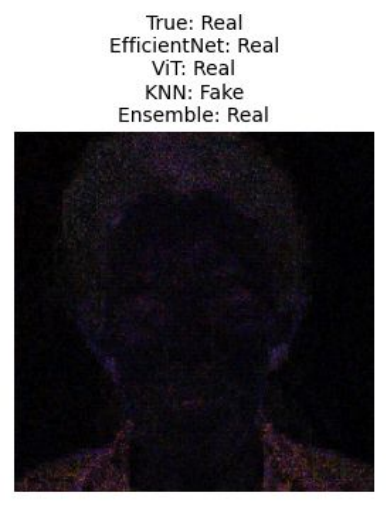


Predictions









Conclusions

Overall, The EfficientNet Model on its own can detect the deepfake images accurately but has a significantly low precision when compared to EnsembleNet Model which has better precision and moderate recall. However the balance of EfficientNet Model is much better. Which makes it a suitable model for further improvements.

The EfficientNet model used in current scenario has only 6 million parameters and it still performs well. The performance of this model can be improved by increasing the training time and the number of parameters.

Q&A