50.021
AI Project

Deepfake detection

By: Sihan
Natthan
Sean
Lee Jing
Maathavan

Overview

Problem Statement

In the era of advanced digital manipulation techniques, the proliferation of deepfake technology poses significant challenges to the authenticity and credibility of digital media.

The ability to create highly realistic fake videos and images using artificial intelligence with ease has raised concerns regarding misinformation, fraud, and privacy breaches. As a result, there is an urgent need for robust and reliable methods to detect deepfakes and distinguish them from genuine content

As such, our team aims to detect videos that have been manipulated using methods such as Deepfakes, Face2Face, FaceSwap and NeuralTextures. We explored various Convolutional Neural Network (CNN) models such as Residual Network (ResNet) and EfficientNet.

Dataset: FaceForensics++

- consists of 1000 video sequences from youtube that were edited using advanced face manipulation techniques such as:
  - Facial reenactment
    - Face2Face
    - NeuralTextures
  - Facial reconstructions
    - Face Swaps
    - Deepfakes

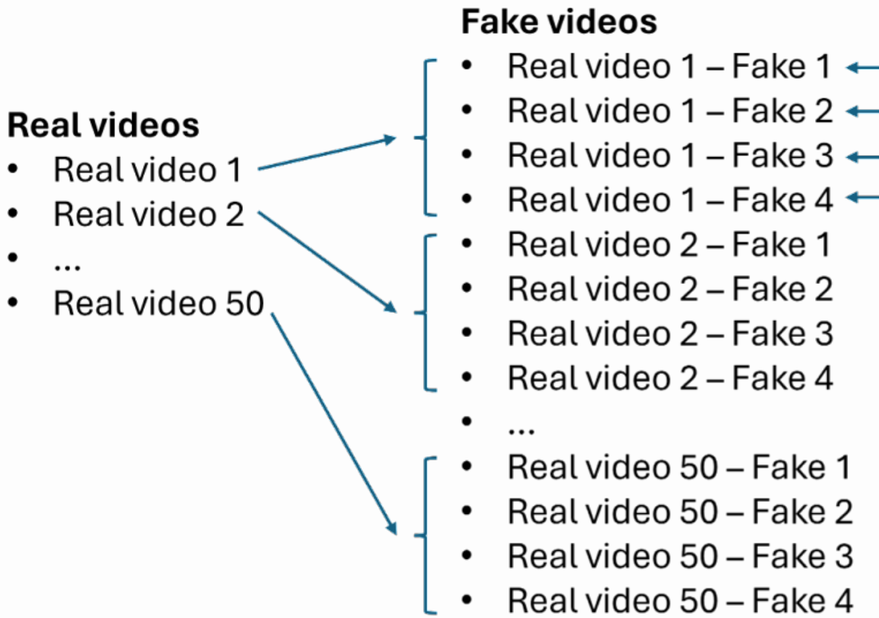
Data Split

- **Authentic Videos**: 50
- **Manipulated Videos**: 200
- **Total Faces Extracted**: 1250

Data Distribution

- **Training**: 1000 faces (80%)
- **Validation**: 125 faces (10%)
- **Testing**: 125 faces (10%)

Dataset: Overview



Methods of Deepfake Generation

1. Deepfakes
2. Face2Face
3. FaceSwap
4. NeuralTextures

Extraction of frames and faces for each video

- frames_per_vid = 5
- faces_per_frame = 1

Dataset composition

- Real to Fake ratio = 1:4
  - 50 real + 200 fake = 250 videos
- Total number of faces = 250 videos x frames_per_vid x faces_per_frame = 1250 faces
- Train to Validate to Test ratio: 8:1:1
  - Train: 1000 faces
  - Validate: 125 faces
  - Test: 125 faces

Pre-processing Steps

The team conducted **preprocessing** on the training data to **prepare** the **image dataset** for **model training**.

After **video segmentation** into frames and **face extraction** from frames using face_recognition, some **incorrectly recognised images** were identified and removed, **retaining** only faces **larger than 75x75 pixels**.

All images were subsequently **resized to 224x224 pixels** to suit the requirements of the pre-trained ResNet models and strike **a balance** between **computational efficiency** and **information capture.**

01

Video segmentation

Extracted 5 frames from each video.

02

Face Extraction

Utilized face_recognition library.

03

Image Filtering

Removed incorrectly recognized faces.
Retained images >75x75 pixels.

04

Image Resizing

Resized all images to 224x224 pixels.

Model

ResNet

- **Introduction:**
  - Proposed in the “Deep Residual Learning for Image Recognition” paper
- **Objective:**
  - Address degradation problem in training deep neural networks
- **Key Features:**
  - **Residual Functions:** Learn with reference to layer inputs
  - **Residual Blocks:** Multiple convolution layers with shortcut connections
  - **Shortcut Connections:** Perform identity mapping, skipping layers
- **Training Method:**
  - Utilizes stochastic gradient descent (SGD) with backpropagation

EfficientNet

- **Introduction:**
  - Proposed in “EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks” paper
- **Objective:**
  - effectively scale up convolutional neural network models
- **Scaling Method:**
  - **Compound Scaling:** Scales width, depth, and resolution uniformly
  - **Fixed Scaling Coefficients:** Used to maintain uniform scaling
- **EfficientNet Variants:**
  - **EfficientNet-B0:** Baseline model
  - **EfficientNet-B1 to EfficientNet-B7:** Scaled versions using different compound coefficients

Using Cross - Entropy Loss function as the default loss function across all models

Training

- **Experiment Setup**
  - **Models:** ResNet and EfficientNet pre-trained on ImageNet
  - **Epochs:** 15
  - **Batch Size:** 32
  - **Loss Function:** Cross-entropy

- **Variables for Exploration**

Optimizer	Learning Rate	L2 Regularisation	Dropout Rate
• Adam	• 5e - 3	• 0	• 0
• Stochastic Gradient Descent (SGD)	• 1e - 3	• 1e - 3	• 0.1
• Adadelta	• 5e - 4	• 1e - 4	• 0.2
• RMSprop		• 1e - 5	• 0.3
		• 1e -6	• 0.4
		• 1e - 7	• 0.5

Training: Model Exploration

The aim of model exploration was finding the best model for each of the model architecture. The five ResNet models vary in the numbers of layers in the network, while the six EfficientNet models are of various different scales. We were able to narrow down all the ResNet options to ResNet50 and EfficientNet options to EfficientNetB2

Architecture Exploration

- **ResNet Variants**
  - ResNet18
  - ResNet34
  - ResNet50
  - ResNet101
  - ResNet152
- **EfficientNet Variants**
  - EfficientNetB0 to EfficientNetB5

Focused Exploration

- **Selected Models**
  - ResNet50
  - EfficientNet B2

Evaluation Metrics

- **Accuracy**
- **Precision**
- **Recall**
- **F1 Score**
- **Score ranges from 0 to 1; closer to 1 indicates better performance**

Performance

ResNet Architecture Exploration

ResNet Architecture	Test Accuracy	Test Precision	Test Recall	Test F1 Score
ResNet18	0.7521	0.8113	0.8958	0.8515
ResNet34	0.7686	0.8091	0.9271	0.8641
ResNet50	0.8099	0.8476	0.9271	0.8856
ResNet101	0.6116	0.8657	0.6042	0.7117
ResNet152	0.7025	0.7885	0.8542	0.8200

Performance

EfficientNet Architecture Exploration

EfficientNet Architecture	Test Accuracy	Test Precision	Test Recall	Test F1 Score
EfficientNetB0	0.8264	0.8571	0.9375	0.8955
EfficientNetB1	0.7934	0.8198	0.9479	0.8792
EfficientNetB2	0.8264	0.8571	0.9375	0.8955
EfficientNetB3	0.8250	0.8319	0.9792	0.8995
EfficientNetB4	0.7686	0.8208	0.9062	0.8614
EfficientNetB5	0.7851	0.7966	0.9792	0.8785

Performance

Optimiser Exploration (ResNet50)

Optimiser	Test Accuracy	Test Precision	Test Recall	Test F1 Score
Adam	0.8099	0.8476	0.9271	0.8856
SGD	0.7934	0.7934	1.0000	0.8848
Adadelta	0.7934	0.7934	1.0000	0.8848
RMSprop	0.7934	0.7934	1.0000	0.8848

Performance

Optimiser Exploration (EfficientNetB2)

Optimiser	Test Accuracy	Test Precision	Test Recall	Test F1 Score
Adam	0.8264	0.8571	0.9375	0.8955
SGD	0.7934	0.7934	1.0000	0.8848
Adadelta	0.7851	0.7966	0.9792	0.8785
RMSprop	0.7851	0.8017	0.9688	0.8774

Performance

Learning Rate Exploration (ResNet50)

Learning Rate	Test Accuracy	Test Precision	Test Recall	Test F1 Score
5e-3	0.7934	0.7934	1.0000	0.8848
1e-3	0.8099	0.8476	0.9271	0.8856
5e-4	0.8099	0.8476	0.9271	0.8856

Performance

Learning Rate Exploration (EfficientNetB2)

Learning Rate	Test Accuracy	Test Precision	Test Recall	Test F1 Score
5e-3	0.7769	0.8000	0.9583	0.8720
1e-3	0.8264	0.8571	0.9375	0.8955
5e-4	0.8264	0.8319	0.9792	0.8995

Performance

Weight Decay Exploration (ResNet50)

Weight Decay	Test Accuracy	Test Precision	Test Recall	Test F1 Score
0	0.8099	0.8476	0.9271	0.8856
1e-3	0.6694	0.7745	0.8229	0.7980
1e-4	0.8264	0.8261	0.9896	0.9005
1e-5	0.8347	0.8276	1.0000	0.9057
1e-6	0.7107	0.8144	0.8229	0.8187
1e-7	0.7934	0.8034	0.9792	0.8826

Performance

Weight Decay Exploration (EfficientNetB2)

Weight Decay	Test Accuracy	Test Precision	Test Recall	Test F1 Score
0	0.8264	0.8571	0.9375	0.8955
1e-3	0.7686	0.8148	0.9167	0.8627
1e-4	0.8264	0.8713	0.9167	0.8934
1e-5	0.8182	0.8491	0.9375	0.8911
1e-6	0.8512	0.8421	1.0000	0.9143
1e-7	0.8182	0.8364	0.9583	0.8932

Performance

Dropout Probability Exploration (ResNet50)

Dropout Probability	Test Accuracy	Test Precision	Test Recall	Test F1 Score
0	0.8347	0.8276	1.0000	0.9057
0.1	0.7686	0.7931	0.9583	0.8679
0.2	0.7934	0.8142	0.9583	0.8804
0.3	0.7686	0.8091	0.9271	0.8641
0.4	0.7934	0.8034	0.9792	0.8826
0.5	0.8017	0.8000	1.0000	0.8889

Performance

Dropout Probability Exploration (EfficientNetB2)

Dropout Probability	Test Accuracy	Test Precision	Test Recall	Test F1 Score
0	0.8512	0.8421	1.0000	0.9143
0.1	0.8182	0.8426	0.9479	0.8922
0.2	0.7769	0.8000	0.9583	0.8720
0.3	0.8099	0.8120	0.9896	0.8920
0.4	0.7851	0.8182	0.9375	0.8738
0.5	0.8017	0.8214	0.9583	0.8846

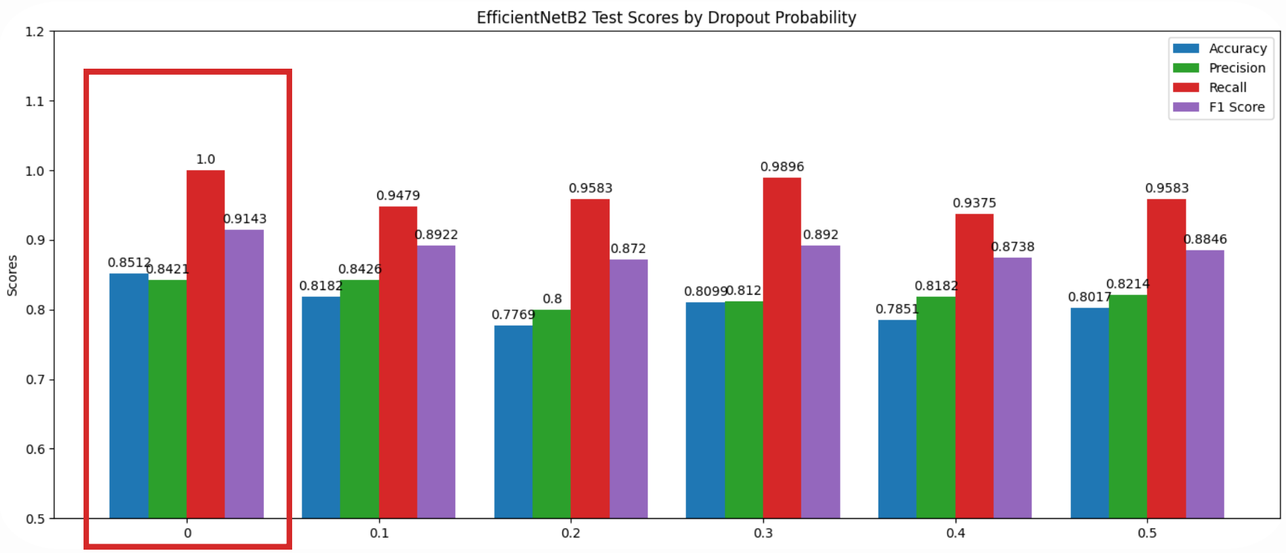
Best Performing Model

Setup

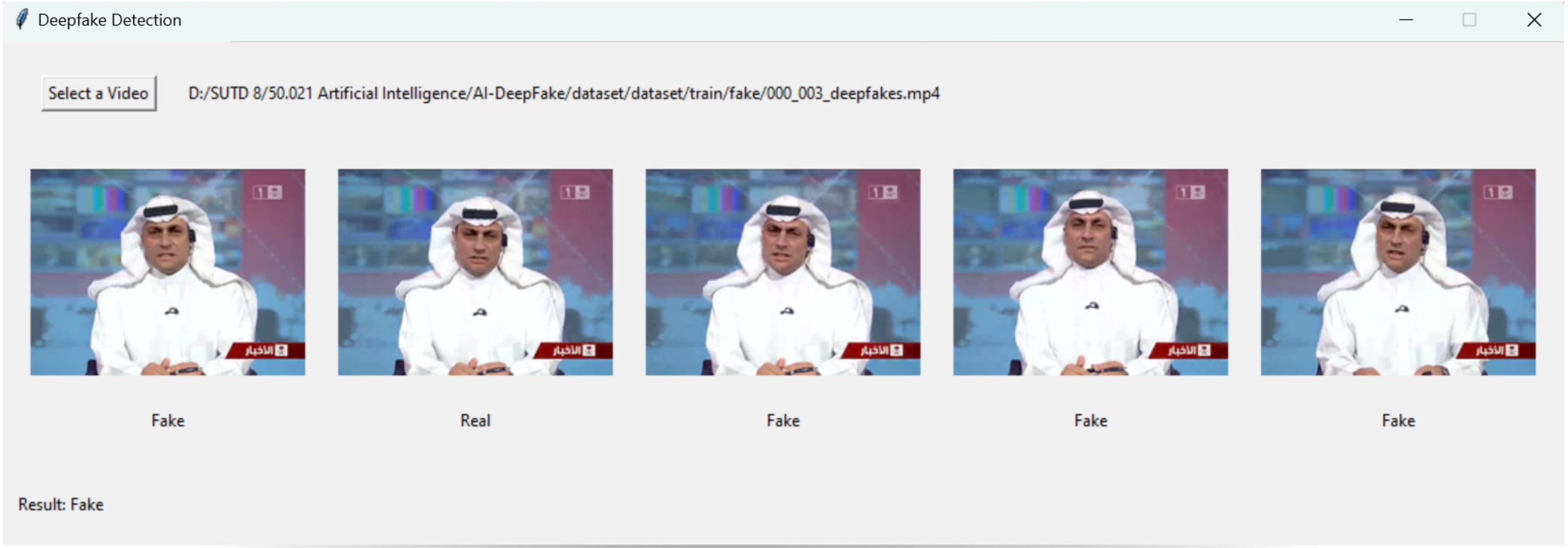
- **Models:** EfficientNetB2
- **Opitimiser:** Adam
- **Learning Rate:** 1e-3
- **Weight Decay:** 1e-6
- **Dropout:** no dropout
- **Epochs:** 15
- **Batch Size:** 32
- **Loss Function:** Cross-entropy

Performance

- **Accuracy:** 0.8512
- **Precision:** 0.8421
- **Recall:** 1
- **F1 score:** 0.9143



GUI Demo


Deepfake Detection
Select a Video
D:/SUTD 8/50.021 Artificial Intelligence/AI-DeepFake/dataset/dataset/train/fake/000_003_deepfakes.mp4
Fake
Real
Fake
Fake
Fake
Result: Fake

Future Improvements

- Explore different parameter exploration methods
- Repeated searches for extensive configuration exploration
- Use Larger and more diverse datasets that consists of different facial features to make model more generalisable
- Developing capabilities to detect temporal changes in a video so as to increase detection accuracy.

Alternate model explorations

- Architectures
  - DenseNet
- Transformers
  - MaxViT
  - ViT-B

50.021
AI Project

THANK YOU!