Evaluation Metrics, Completed and Future Work

for

DeepFake Face Identification (CNN and XAI Models)

Dhruvi Sachin Shah (106122038)

Sanjana Gummuluru (106122106)

Introduction

DeepFake, which is a portmanteau of the terms ‘deep learning’ and ‘fake’, is a new vein of AI generated fake videos synthesized using generative ML models. They can achieve high degrees of realism and have thus been used in malignant ways, manipulating people into believing something is real when it is not.

The proposed model for DeepFake Face Identification uses a CNN-based approach to detect manipulated media, given that CNNs are particularly well suited for image and video analysis tasks.

The incorporation of Explainable AI (XAI) allows for the detection process to be more interpretable by the user, by highlighting the features that led to the model’s classification as real/fake.

Literature Review

Deepfake detection has gained significant attention in recent years due to the increasing realism and potential misuse of deepfake technologies. Various approaches have been developed to address this problem, with Convolutional Neural Networks (CNNs) being the most widely used model due to their effectiveness in image-related tasks. Early models, such as VGG16 and VGG19, have been successful in detecting manipulated images, but newer architectures, such as ResNet, EfficientNet, and DenseNet, have shown superior performance due to their deeper architectures and more efficient feature extraction capabilities.

However, one major limitation in current deepfake detection methods is that they operate as ‘black boxes.’ Although these models achieve high accuracy, they offer little insight into how decisions are made, which can be problematic in real-world applications where interpretability and trust are essential. The usage of XAI techniques would make the decision-making process of AI systems transparent and understandable to humans. Additionally, we can ensure that the model is detecting relevant artifacts in deepfake images rather than unrelated patterns in the data, making the model more robust and reliable in real-world applications.

Base CNN Model

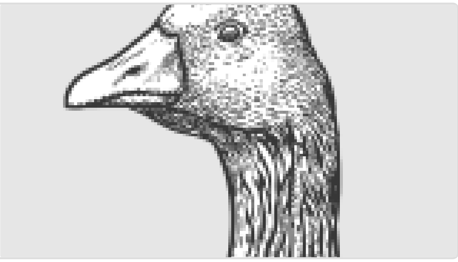
We have utilized the OpenForensics Dataset, which contains over 190k images (both real and fake) split into train, test, and validation sets.

The Convolutional Neural Network (CNN) used in this project consists of multiple convolutional layers followed by max-pooling layers, fully connected layers, and a final softmax activation for classification. The network architecture is designed to extract features from input images and classify them as either real or fake. We used ReLU activations for each convolution layer and applied dropout to prevent overfitting.

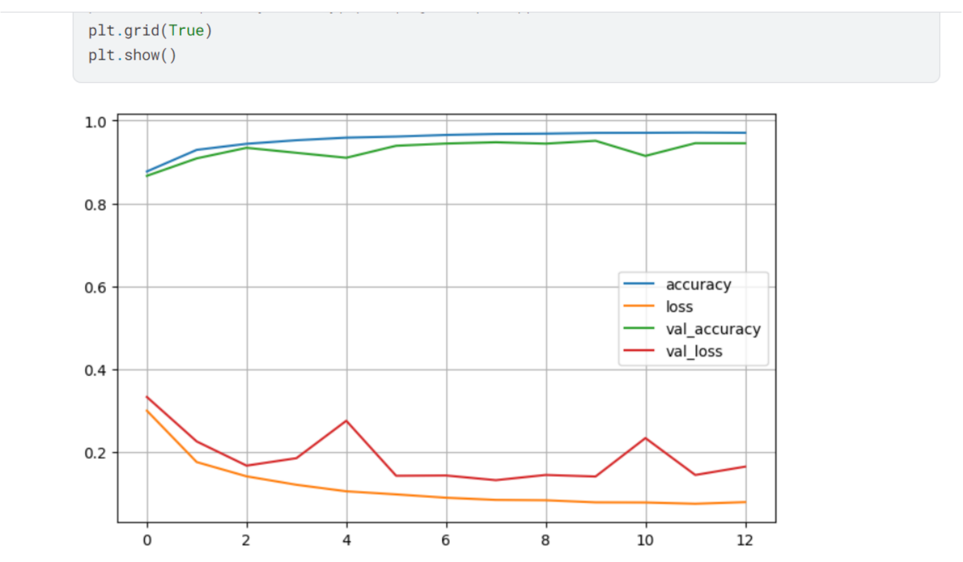
cnn-deepfake

Explore and run machine learning code with Kaggle Notebooks | Using data from deepfake and real images

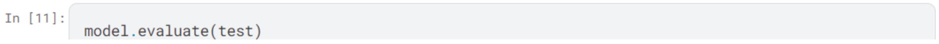
k https://www.kaggle.com/code/sanjanagummnitt/cnn-deepfake



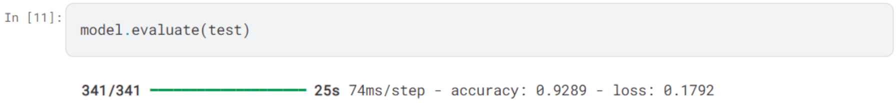
Images



```
plt.grid(True)
plt.show()
```

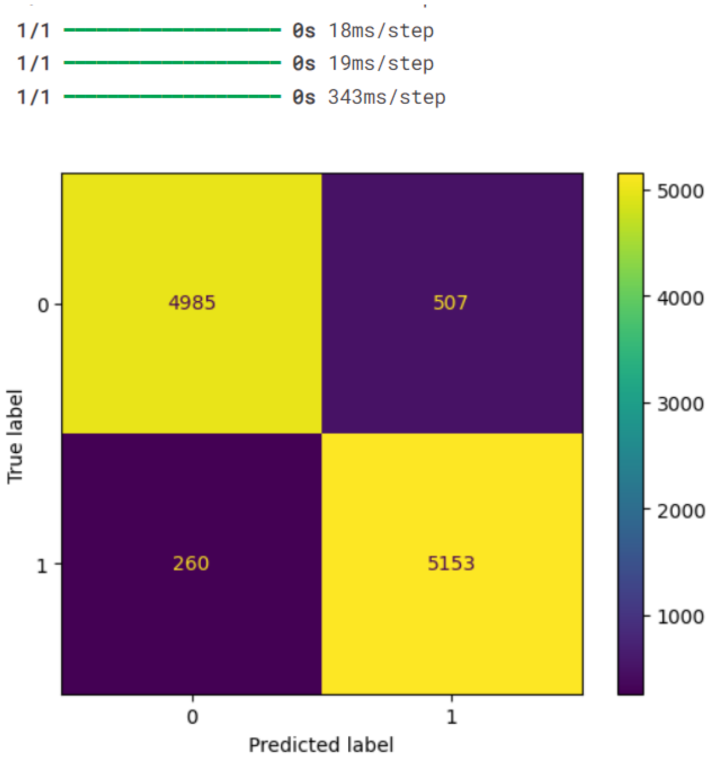


```
In [11]: model.evaluate(test)
```



```
In [11]: model.evaluate(test)
```

```
341/341 ━━━━━━━━━━━━━━━━━━━━ 25s 74ms/step - accuracy: 0.9289 - loss: 0.1792
```



```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 18ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 19ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 343ms/step
```

Proposed XAI Model

We plan to use 2 techniques in order to help with visualization and pixel-level insights:

- Grad-CAM for visualizing CNN's focus by highlighting the regions in the image that led to this classification
- SHAP for getting the mathematical contribution of each input feature (or pixel) to the model's decision

These XAI techniques are integrated into a pipeline where first, the model is trained, then an explanation is generated using the XAI technique and then finally, it is visualized. For Grad-CAM, a heatmap is generated that overlays on the face showing what contributes to the decision and for SHAP we get a visual breakdown of how the model weighted different parts of the image.

Finally, we interpret the generated results to:

- Ensure the model validation by checking if it focuses on relevant regions like eye or mouth and not on the background of the image. If it is focusing on the wrong areas, we debug it.
- Use the explanations to justify why the decision was made the way it was.

Evaluation Metrics

The chosen metrics are accuracy, precision, recall, F1-score, loss, confusion matrix, detection time, and memory usage. Each of these metrics highlights different aspects of the model's capabilities and limitations.

Terms used:

- TP (True Positives): Deepfake images correctly classified as fake.
- TN (True Negatives): Real images correctly classified as real.
- FP (False Positives): Real images incorrectly classified as fake.
- FN (False Negatives): Fake images incorrectly classified as real.

Accuracy

Accuracy measures the overall effectiveness of the model by calculating the proportion of correct predictions (both real and fake images) to the total number of predictions. It is a simple yet important metric that gives a broad view of the model's performance.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Precision

Precision, also known as the positive predictive value, measures the proportion of correct positive (fake) classifications out of all predicted positives (fake images). A high precision indicates that the model has a low false positive rate, meaning it rarely misclassifies real images as fake.

$$Precision = \frac{TP}{TP + FP}$$

Recall

Recall, also known as sensitivity or true positive rate, measures the proportion of actual positives (fake images) that are correctly identified by the model. A high recall means the model effectively captures most of the fake images, minimizing the number of missed deepfakes.

$$Recall = \frac{TP}{TP + FN}$$

F1-Score

The F1-score is the harmonic mean of precision and recall. It provides a balanced view of the model's performance, especially when there is a trade-off between precision and recall. This metric is particularly useful when the dataset is imbalanced, ensuring that neither false positives nor false negatives dominate the evaluation.

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

Loss

Loss is a key metric used to evaluate the model's optimization during training. It quantifies how far the predicted probabilities are from the actual labels. In this project, we use binary cross-entropy as the loss function since we are dealing with a binary classification problem (real vs. fake).

$$Loss = -\frac{1}{N} \sum_{i=1}^{N} [y_i \log(p_i) + (1 - y_i) \log(1 - p_i)]$$

Where:

- y_i is the true label (1 for fake, 0 for real).
- p_i is the predicted probability for the positive class (fake).
- N is the number of samples.

Confusion Matrix

A confusion matrix is a tabular representation that summarizes the performance of the model in terms of its correct and incorrect classifications for each class (real and fake). It provides insights into the types of errors the model is making, which can help in fine-tuning the model.

Detection Time

Detection time refers to the time taken by the model to classify an image as either real or fake. It is a crucial metric for applications requiring real-time performance, such as video authentication on social media platforms or news outlets. Faster detection times make the model more practical for deployment in real-world environments.

In this project, detection time will be measured as the average time taken to classify an image during the inference phase. Optimizing the model for faster detection times is important, especially when processing large volumes of media in time-sensitive applications.

Memory Usage

Memory usage refers to the amount of computational resources (RAM or GPU memory) consumed by the model during the inference phase. Memory usage is critical, especially when deploying the model in environments with limited resources, such as mobile devices or cloud-based systems.

In this project, memory usage will be measured by monitoring the resource consumption during model inference. Efficient memory usage ensures that the model can scale and handle larger datasets without causing system slowdowns or crashes.

Future Work

The full implementation of the XAI Model is expected to be submitted on 22/10/2024. We aim to complete our implementation at least a week prior (15/10/2024) to allow for challenges, improvement, and the final report.