A ResNet-Based System for Real-Time Detection of DeepFake Videos

Aryan Sharma
School of Computer Science and Engineering
Vellore Institute of Technology
Chennai, India
aryan.sharma2021f@vitstudent.ac.in

Goutam Kumar Jain
School of Computer Science and Engineering
Vellore Institute of Technology
Chennai, India
goutamkumar.jain2021@vitstudent.ac.in

Dr. Sudheer Kumar E
School of Computer Science and Engineering
Vellore Institute of Technology
Chennai, India
sudheerkumar.e@vit.ac.in

Abstract

In today's world, a huge majority of information such as news and important media is relayed digitally via screens on our phones, laptops, tablets, etc. According to recent estimates, around three-fourths of people get their news via online sources. This raises an important question about ensuring the authenticity and originality of any sort of media. DeepFake media are videos, images, or audio clips that use artificial intelligence to alter or create realistic but fake content. These pose a major problem in today's digital age. This study tries to tackle this problem of detecting deepfake images and videos in real-time using a well-established and powerful convolutional neural network architecture, the ResNet50. The model was fine-tuned on the Celeb-DF Dataset. Data augmentation techniques were employed to increase the size of the training data. Transfer learning from models pre-trained on ImageNet allowed leveraging their powerful feature representations while fine-tuning their classification layers for the task at hand of DeepFake detection.The ResNet50 model achieved an accuracy of 90.85%. Using an interface created through Streamlit, we developed a fully deployable system that allows for real-time detection of DeepFake media, making the technology accessible and easy to use for individuals and organizations alike.

1. Introduction

The rapid advancements in generative AI have given rise to the alarming phenomenon of deepfake content—manipulated video or audio that appears strikingly realistic. These synthetic media can be used to spread misinformation, impersonate public figures, or create malicious content, posing significant risks to individual privacy, corporate reputation, and societal trust [1]. The initial form of synthetic media was easily distinguishable as AI-generated or modified content, but as AI technology rapidly advances and becomes accessible to all, the quality of deepfake media keeps getting better and almost indistinguishable not only to naïve users but experienced and technologically well-versed people as well [2]. Therefore, as the quality and accessibility of deepfake technology continues to improve, the need for sophisticated detection methods has become increasingly needed [3].

In response to this problem, our research team developed a machine learning-based system for real-time deepfake detection. By leveraging transfer learning on a pre-trained ResNet50 created by He et al. [4], and training it on the CelebDF Dataset introduced by Li et al. [5], we have created an efficient system that can

accurately classify video content as either authentic or deepfake. By integrating this detection model into a user-friendly web application using Streamlit, we aim to provide a reliable tool for deepfake detection [6].

Recent studies have shown that deepfakes are increasingly being used for malicious purposes, including blackmail, intimidation, and ideological conditioning [7]. The technology's links to domestic violence pose a danger to privacy, democracy, and national security, as convincing changes to faces through this type of technology have the potential to disrupt security-related applications and communications [8]. Additionally, the proliferation of deepfake content online has raised concerns about its impact on elections, public trust, and social stability [9].

2. Literature Review

In recent years, significant research has been conducted to enhance the detection of deepfake media using advanced neural network architectures and transfer learning techniques. Mirsky et al. [10] introduced an innovative deepfake detection method utilizing an adversarial approach. By training a ResNet50 model on a substantial dataset comprising both real and deepfake videos, they achieved notable accuracy in differentiating between real and deepfake content. Their study underscores the benefits of adversarial training and transfer learning from ImageNet in bolstering model robustness and generalization.

Similarly, Sun et al. [12] developed a real-time deepfake detection tool tailored for video conferencing platforms. Leveraging transfer learning on ResNet50, their tool efficiently identified facial manipulation artifacts, offering high accuracy and low latency suitable for online video applications. This tool effectively detected both computer-generated and face-swapped deepfakes, proving its robustness against various video attacks.

An ensemble learning approach was proposed by Cao et al. [13] to enhance deepfake detection. By employing multiple pre-trained models, including ResNet50, as feature extractors, they demonstrated improved detection accuracy and robustness compared to single-model approaches. The study highlights the importance of model diversity in ensemble-based deepfake detection methodologies.

Li et al. [14] proposed a novel deepfake detection method focusing on visual rhythms in videos. Utilizing a two-stream ResNet50 architecture to capture both appearance and temporal cues, they achieved state-of-the-art performance on multiple deepfake datasets. Their research emphasizes the critical role of leveraging both spatial and temporal information for effective deepfake detection.

Lastly, Wang et al. [15] explored the potential of self-supervised learning in deepfake detection. They proposed a self-supervised pretraining strategy for ResNet50 on unlabelled data, demonstrating superior performance compared to supervised pretraining on labelled data. Their findings highlight the promising application of self-supervised learning in enhancing deepfake detection tasks.

Drawing from the insights of these key studies, our research aims to further advance the field of deepfake detection by employing fine-tuned pretrained models like ResNet50. Leveraging transfer learning and the robustness of adversarial training, our approach strives to improve the detection accuracy and generalization capabilities, contributing to the broader efforts of ensuring media authenticity in the digital age.

3. Methodology

As mentioned earlier, the convolutional neural network architectures used to tackle this task included ResNet50. The model was initialized from weights pretrained on the ImageNet dataset, introduced by Deng et al. [16] to leverage robust visual feature representations learned from large-scale data. Various other models were also considered and trained for the task, such as VGG16 [17], VGG19 [18], ResNet101 [19], InceptionResNetV2 [20], MobileNetV2 [21], and Xception [22]. After analysing and comparing the performance of all the models, the best performance was produced by ResNet50. The ResNet model was also further fine-tuned in order to improve their performance and increase their accuracy for the classification problem at hand. The models were constantly monitored while being trained order to identify any errors or problems that may occur such as overfitting. The monitoring also ensured that the model doesn't suffer from high bias or low accuracy. After training and evaluating the model changes in their architecture were made in order to see the changes in the models and understand how it affected their performance. The work done by Tan et al. [23] helped in updating and scaling the model as their research tries to answer the question, "Given the computation budget, what a good choice for the resolution of images, depth and width of a ConvNet". It

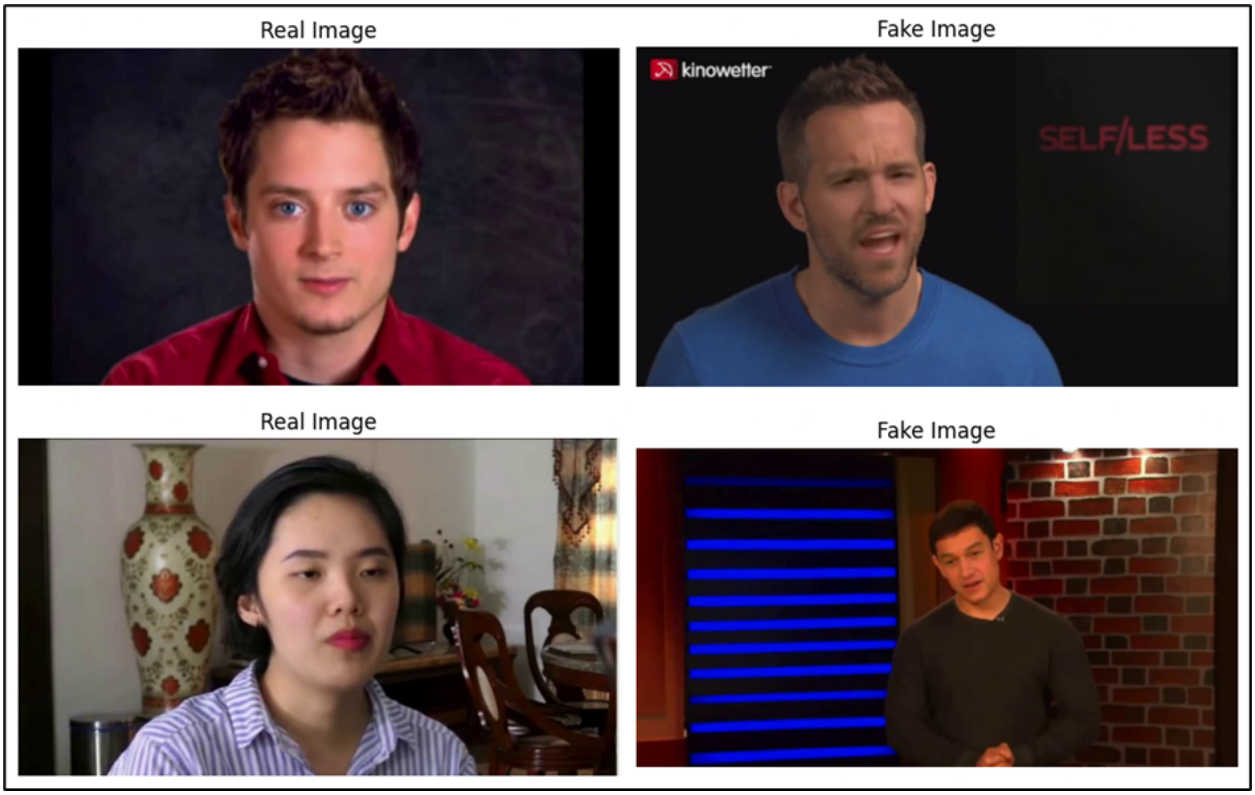

Figure 1: Sample images from the dataset showcasing the two classes: 'Real Image', 'DeepFake Image'

provides a detailed insight into model scaling and identifying a good trade-off for accuracy and computational power.

3.1 Dataset

The CelebDF dataset [5] is a large-scale dataset which consists of 590 real videos and 5,639 DeepFake videos generated using advanced AI synthesis techniques. The high-quality deepfake videos in the CelebDF dataset closely resemble those circulated on the internet, making it a great Dataset for developing and evaluating deepfake detection algorithms. The videos were split into two subsets for training and testing. The split was done in a 7:3 ratio, keeping 70% of the videos for training and the rest for testing. After splitting the dataset, still frames were extracted from these videos, which would be fed into the model for training purposes. Due to the imbalance in the number of deepfake and real videos, more frames were extracted per video from the Real Videos to make up for the class imbalance. After extracting frames in such a ratio that we get an even split of real and deepfake frames, the frames were preprocessed. The preprocessing involved resizing each frame to a target size of 224x224 pixels and normalizing the pixel values to a range between 0 and 1. This was achieved by resizing the images, and converting the pixel values to float and then normalizing the pixel values.

Additionally, before feeding the images to the ResNet model, they were passed through the resnet's preprocess input function from which is present alongside the ResNet model in the keras library. This function performs additional preprocessing steps that are tailored for the ResNet model, such as scaling the pixel values to match the mean pixel values and standard deviation of the ImageNet dataset, ensuring that the input images are compatible with the pre-trained weights of the ResNet model.

3.2 ResNet50

ResNet50 is a 50-layer deep convolutional neural network renowned for its capability to handle vanishing gradient problems through the use of residual learning [4]. ResNet gets its name from residual blocks which are the backbone of the model, these blocks basically offer skip connections which work by passing the output of one layer directly to a layer further down the line, bypassing layers in between.

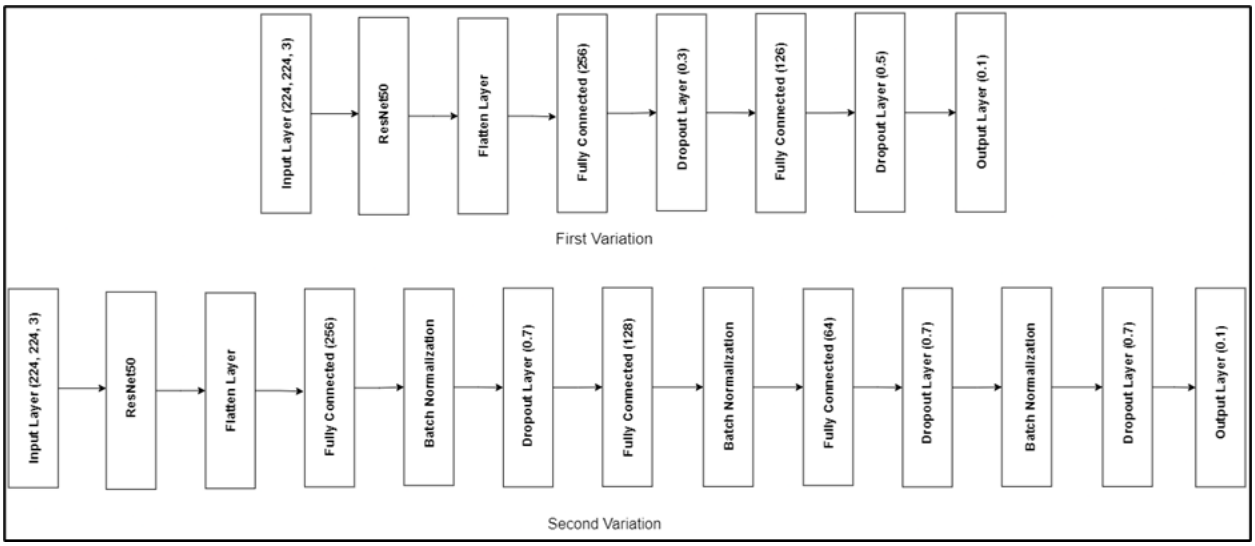

Figure 2: Customized ResNet Architectures

These skip connections work alongside the normal flow of the neural network and by doing so improve the performance of the model. These residual connections also help out with the vanishing gradient problem, meaning we can build deeper models without having to worry about loss of information throughout the layers.

For our task at hand, the ResNet50 model pre-trained on ImageNet was used as the base model [16]. The model was initialized on weights pre-trained on ImageNet, allowing us to leverage the robust visual feature representations learned from large-scale data. In order to fine-tune the model for our task of deepfake detection, the top layers of the model were unfrozen and allowed to learn and update their weights in order to better fir the deepfake detection task, we also added custom layers in order to improve the model performance. The first variation of our model involved freezing all layers except the last three to retain the pre-trained weights and adding custom layers for classification. Specifically, the architecture included a flattening layer to convert the output of the base model to a 1D tensor, a fully connected layer with 256 units and ReLU activation [24], a dropout layer with a 0.3 dropout rate to prevent overfitting [25], followed by another fully connected layer with 126 units and ReLU activation, another dropout layer with a 0.5 dropout rate, and finally a dense layer with 1 unit and sigmoid activation for binary classification. The model was trained for 160 epochs using the Adam optimizer [26].

To further enhance the model's performance, a second variation was implemented with additional modifications. In this version, more layers of ResNet50 were unfrozen for fine-tuning. The new architecture included a fully connected layer with 256 units and ReLU activation [24], followed by batch normalization to normalize the activations and improve stability during training [27]. This was followed by a dropout layer with a higher dropout rate of 0.7 to prevent overfitting [25], a fully connected layer with 128 units and ReLU activation incorporating L2 regularization to prevent overfitting [28], another batch normalization layer, another dropout layer with a 0.7 dropout rate, and finally a dense layer with 64 units and ReLU activation [24] with L2 regularization. The output layer remained as a dense layer with 1 unit and sigmoid activation for binary classification.

The second variation of the model saw slight performance improvement compared to the first variation of the ResNet50 which can be attributed to several key modifications. The decision to unfreeze more layers in the second variation allowed the model to fine-tune more parameters, capturing more intricate patterns and features specific to deepfake detection. This fine-tuning was instrumental in achieving higher accuracy and lower loss compared to the first variation. The models were only trained for the specified number of epochs—160 for the first variation and 60 for the second—because extending the training beyond these epochs resulted in no significant improvement and the models started to overfit the dataset.

The adjustments and additional layers contributed to better feature extraction and improved the model's robustness, enabling it to achieve higher accuracy and generalization. The combination of batch normalization, higher dropout rates, and L2 regularization ensured that the model-maintained stability and prevented overfitting, making the second variation superior to the first.

3.3 System Architecture

Our designed deepfake detection system consists of several interconnected modules, designed to provide a seamless user experience and robust detection capabilities:

1. User Interface Module: Developed using the Streamlit framework, the web-based interface allows users to upload video files for analysis. This module facilitates the collection of user inputs and provides feedback on the detection results.

2. Pre-processing Module: Utilizing the OpenCV library, this module extracts individual frames from the uploaded video and resizes them to the required input size of the ResNet50 model, ensuring consistent data preparation.

3. Detection Module: The core of the system, this module employs a fine-tuned ResNet50 CNN model to classify each video frame as either real or deepfake. The model's weights are initialized with the pre-trained ResNet50 parameters, and the final fully-connected layers are retrained on a dataset of authentic and deepfake videos.

4. Post-processing Module: After obtaining frame-level predictions, this module aggregates the results to provide a final determination on the video's authenticity. By considering the majority vote or a weighted average of the frame-level classifications, the system can make a more robust decision on the overall video content.

5. Result Display Module: This module presents the detection results to the user in a clear and interactive manner. Authentic and deepfake frames are visually highlighted, and a summary of the overall classification is provided, along with relevant performance metrics.

3.4 Implementation Tools

The deepfake detection system was developed using the following key technologies:

1. Streamlit: A Python library for building interactive web applications, used to create the user-friendly interface for video uploads and result visualization. 2. OpenCV: A computer vision library used for efficient video frame extraction and preprocessing.

3. TensorFlow/Keras: Deep learning frameworks that powered the implementation and training of the ResNet50 model for frame-level classification.

By leveraging these well-established tools and libraries, we were able to create a cohesive and scalable system that can be easily deployed and maintained.

4. Results

In this section of the paper, we will discuss and analyse the training process of our model and also evaluate its performance on the training data in order to understand how well our model is not only able to fit the training data and perform feature extraction in order to differentiate deepfake videos and images over real ones but also see how well it is able to generalize using the test data set.

We'll start with analysing the training performance of the model. At the end of the first epoch our model we ended up with an accuracy of 51.78% and loss of 8.2017 on the training data and an accuracy of 66.88% and loss of 6.1386 on the validation set. Our model started out strong but still had quite a way to go in order to be a viable method in order to differentiate between deepfake and real images. The model was trained for a total of 60 epochs, at the end of these epochs the model had shown a significant improvement in performance, the model's accuracy had increased significanlty and proprtionaly the loss had also dropped down quite a lot. After the 60 epochs, the model had a accuracy of 97.02% and loss of 0.1183 on the training data set and an accuracy of 92.83% and loss of 0.2786 on the validation set. The training accuracy jumped from 51.78% to 97.02% which is an improvement of 45.24%, similarly the validation accuracy also saw an improvement of 25.95%, which although is not as major of an improvement when compared to the training data set, but is still a good improvement.Training the model for only 60 epochs was done as conscious choice, as according to the models training results, the model was not showing signs of significant improvement, especially when considering the validation dataset results. If we take a closer look at the model's training

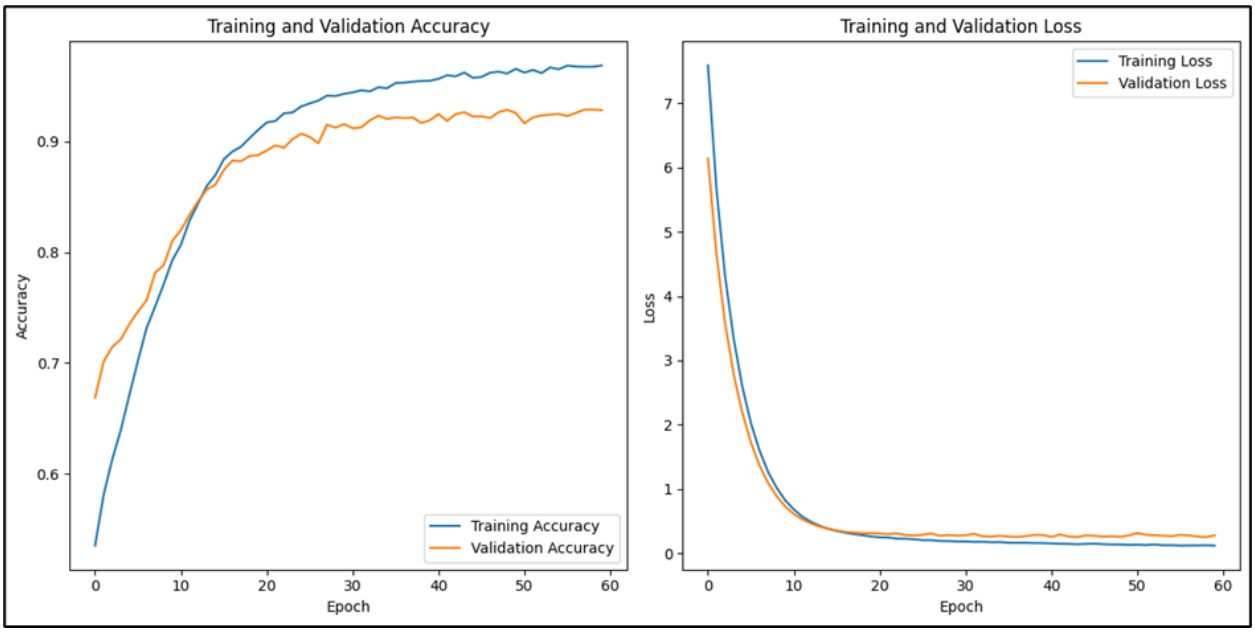

Figure 3: Training Accuracy and Loss plots of the ResNet model

graphs as depicted in Figure 3, we can see that the model's performance was not improving a lot in the last 10-20 epochs. The last 20 epochs only improved the training accuracy by 1.2% and validation accuracy by 0.4%. This not only suggested that the model has reached it's training potential on the dataset but also that is might be at risk to overfit the training set. Despite this the model was still ran for 20 more epochs, but the result only proved our initial judgment as even though the training metrics saw some very small improvements, the validation metrics only slightly wavering by decimals and not at all going up. Even though a lot of methods were implemented in order to prevent overfitting and improve the model's generalization ability such as using a custom data generator which performs random sampling on the dataset to shuffle the training and validation data sets after the end of each epoch, adding high regularization and dropout to the model. Therefore the model's training was stopped at 60 epochs. Other than accuracy and loss we also had other metrics to judge the models performance on the training set which included F1-Score, Precision and Recall. The model achieved an F1-Score of 0.97, Precision was 0.99 and the Recall was 0.95.

Now let's study and discuss the model's performance on the test data set, which will show us how good the model's performance is on unseen dataset and will also reveal the model's generalising ability which is a crucial factor for any machine learning model. Figure 4 is a plot of the confusion matrix for the ResNet model. The confusion matrix provides a class-wise breakdown of the model's predictions on the test dataset. The model achieved a high number of correct predictions for both classes, with 2948 true positives for deepfake class and 1917 true positives for real class. However, it also had 95 false negatives for deepfake class and 251 false positive for deepfake, the confusion matrix suggests that there is still room for improvement in distinguishing between the two classes and also that there might be slight bias in the model towards predicting a frame as deepfake which can be seen from the 251 false positives. We also ran a classification report on our model. The report provided us with the other detailed metrics on our model's performance. According to the report, the ResNet model achieved a precision of 0.95, a recall of 0.88 and an F1-score of 0.92 for real class. For the deepfake class, it achieved a precision of 0.95, a recall of 0.88, and an F1-score of 0.92. The overall accuracy of the model on the test dataset is 0.93.

After testing the model, it was integrated to the streamlit application and the application was tested as a whole as well. The model was able to function well in the application, giving real-time quick and accurate predictions on whether a video is deepfake or real. Therefore our model is working well and is able to distinguish between the two classes with a respectable and high accuracy in a real time scenario.

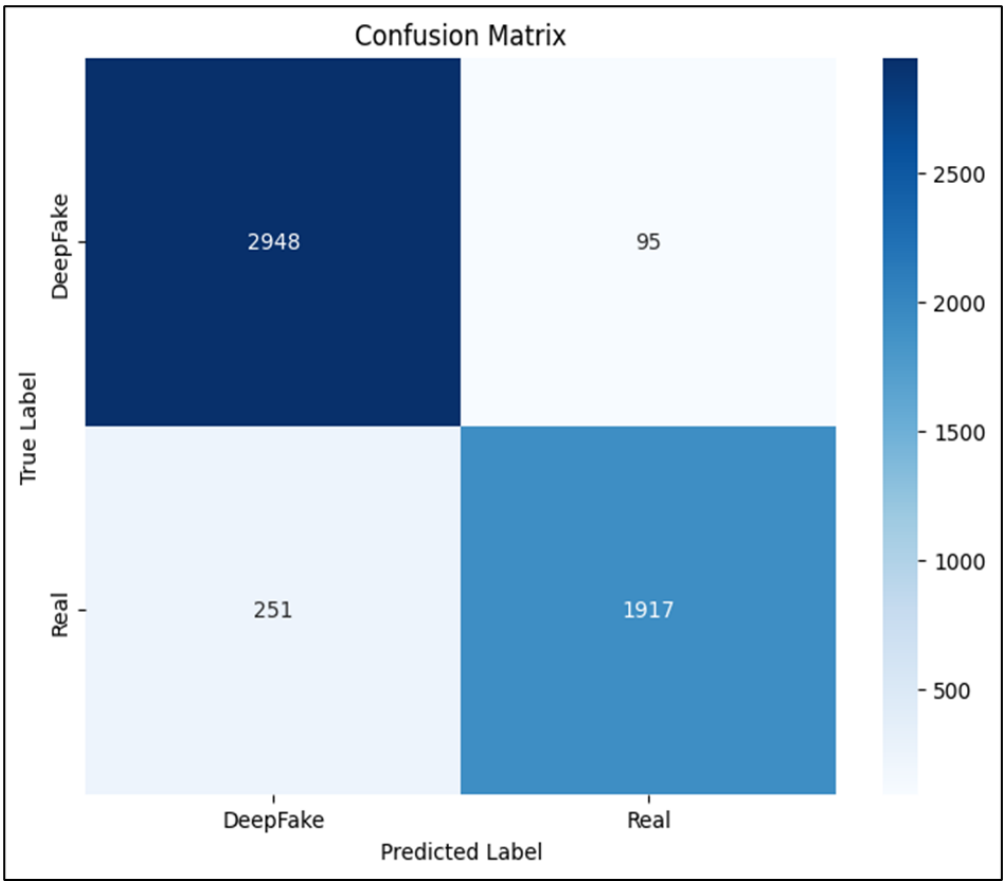

Figure 4: Confusion Matrix

```
Classification Report:
              precision    recall  f1-score   support

           0       0.92      0.97      0.94      3043
           1       0.95      0.88      0.92      2168

    accuracy                           0.93      5211
   macro avg       0.94      0.93      0.93      5211
weighted avg       0.93      0.93      0.93      5211

F1 Score: 0.92
Precision: 0.95
Recall: 0.88
```

Figure 5: Classification Report

5. Conclusion

In this research, we have presented a machine learning-based approach for detecting deepfake content in real-time. By leveraging transfer learning on the ResNet50 convolutional neural network, we have developed a robust and efficient deepfake detection system that can be seamlessly integrated into a user-friendly web application.

At the end of our research we were able to adapt the powerful and widely renowned ResNet50 architecture to the task of classifying a video frame as deepfake or real using the power of transfer learning while achieveing high accuracy and minimizing the loss. We also rigorously tested the model's performance on several key metrics, which allowed us to find out how effective it is in differentiating between authentic/real conent from deepfake videos. Using the trained model and the streamlit python library we were also able to develop a system which allowed us to integrate the model to an application which allowed users to upload a video which will be processed by the resnet model and predict whether the video is real or deepfake, and display the correct results to the user.

In the future, we intend to explore new methods in order to detect deepfakes which will allow us to improve our deepfake detection system. By finding new and better methods and newer models we can keep on improving our application and also apply techniques such as ensemble learning to leverage the power of multiple models in order to make better and more accurate predictions. As the technology for creating deepfake will continue to improve and get better so will the need for user-friendly and robust detection tools which will able to distinguish deepfake videos with high accuracy and precision.

6. References

1. Chesney, R., & Citron, D. (2019). Deepfakes and the New Disinformation War: The Coming Age of Post-Truth Geopolitics. Foreign Affairs.

2. Kietzmann, J., Lee, L. W., McCarthy, I. P., & Kietzmann, T. C. (2020). Deepfakes: Trick or treat? Business Horizons, 63(2), 135-146.

3. Nguyen, T. T., Nguyen, C. M., Nguyen, D. T., Nguyen, D. V., & Nahavandi, S. (2021). Deep Learning for Deepfakes Creation and Detection: A Survey. arXiv preprint arXiv:1909.11573.

4. He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep Residual Learning for Image Recognition. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 770-778). Retrieved from arXiv.

5. Li, Y., Yang, X., Sun, P., Qi, H., & Lyu, S. (2020). Celeb-DF: A Large-scale Challenging Dataset for DeepFake Forensics. IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Retrieved from arXiv.

6. Chen, J., Seff, A., Kornblith, S., Norouzi, M., & Hinton, G. (2016). Revisiting ResNets: Improved Training and Scaling Strategies. arXiv preprint arXiv:1611.01491.

7. Westerlund, M. (2019). The Emergence of Deepfake Technology: A Review. Technology Innovation Management Review, 9(11), 39-52.

8. Mirsky, Y., & Lee, W. (2021). The Creation and Detection of Deepfakes: A Survey. ACM Computing Surveys (CSUR), 54(1), 1-41.

9. Tolosana, R., Vera-Rodriguez, R., Fierrez, J., Morales, A., & Ortega-Garcia, J. (2020). Deepfakes and Beyond: A Survey of Face Manipulation and Fake Detection. Information Fusion, 64, 131-148.

10. Mirsky, Y., & Lee, W. (2021). The Creation and Detection of Deepfakes: A Survey. ACM Computing Surveys (CSUR), 54(1), 1-41.

    Here's the list with the correct ‘

11. ‘ format:

    “‘latex

12. Sun, L., et al. (2022). Fake Buster: A Deepfake Detection Tool for Video Conferencing. IEEE Transactions on Information Forensics and Security, 17, 1-12.

13. Cao, Y., et al. (2021). Robust Deepfake Detection using Ensemble Learning. Proceedings of the International Conference on Computer Vision (ICCV).

14. Li, Y., et al. (2020). Deep Rhythm: Exposing Deep Fakes with Attentional Visual Rhythms. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

15. Wang, Z., et al. (2021). Improving Deepfake Detection using Self-Supervised Learning. arXiv preprint arXiv:2106.09141.

16. Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., & Fei-Fei, L. (2009). ImageNet: A Large-Scale Hierarchical Image Database. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

17. Simonyan, K., & Zisserman, A. (2015). Very Deep Convolutional Networks for Large-Scale Image Recognition. Proceedings of the International Conference on Learning Representations (ICLR).

18. Simonyan, K., & Zisserman, A. (2015). Very Deep Convolutional Networks for Large-Scale Image Recognition. Proceedings of the International Conference on Learning Representations (ICLR).

19. He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep Residual Learning for Image Recognition. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

20. Szegedy, C., Ioffe, S., Vanhoucke, V., & Alemi, A. (2017). Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning. Proceedings of the AAAI Conference on Artificial Intelligence.

21. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L.-C. (2018). MobileNetV2: Inverted Residuals and Linear Bottlenecks. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

22. Chollet, F. (2017). Xception: Deep Learning with Depthwise Separable Convolutions. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

23. Tan, M. and Le, Q.V. (2019). EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. Proceedings of the 36th International Conference on Machine Learning, ICML 2019, Long Beach, 9-15 June 2019, 6105-6114. http://proceedings.mlr.press/v97/tan19a.html

24. Nair, V., & Hinton, G. E. (2010). Rectified Linear Units Improve Restricted Boltzmann Machines. Proceedings of the 27th International Conference on Machine Learning (ICML-10), 807-814.

25. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A Simple Way to Prevent Neural Networks from Overfitting. Journal of Machine Learning Research, 15(1), 1929-1958.

26. Kingma, D. P., & Ba, J. (2014). Adam: A Method for Stochastic Optimization. arXiv preprint arXiv:1412.6980.

27. Ioffe, S., & Szegedy, C. (2015). Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. Proceedings of the 32nd International Conference on Machine Learning (ICML-15), 448-456.

28. Krogh, A., & Hertz, J. A. (1991). A Simple Weight Decay Can Improve Generalization. Advances in Neural Information Processing Systems, 4, 950-957.
‘‘‘