Facial DeepFake Detection

Deshpande, Sanket
Department of Computer Science
Georgia State University
Atlanta, US
sdeshpande8@student.gsu.edu

***Abstract*—In the era of rapid technological advancements, deepfake technology has become a significant concern due to its potential misuse in creating deceptive digital content, particularly in manipulating facial images and videos. This paper presents a comprehensive approach for detecting deepfakes, specifically focusing on facial alterations, using the CelebDFv2 dataset, a prominent benchmark in the field. We will study 2 approaches,one a feature vector based distance metric regression and the other a direct face crop EfficientNet Classifier. The efficacy of our approach is validated through rigorous testing against various metrics, including accuracy, losses, and Grad-CAM to show the exact portion of the image dictating the performance over existing methods. In addition to contributing to digital media forensics, this research addresses the urgent need to mitigate the personal losses associated with deepfakes, such as identity theft, reputational damage, and emotional distress, thereby enhancing individual security in digital interactions.**

Keywords—Keywords: Deepfake Detection, Digital Forensics, Convolutional Neural Networks, Image Processing, Adversarial Training, CelebDFv2 Dataset, Facial Manipulation, Media Integrity, Grad-CAM

1. Introduction

The emergence of deepfaketechnology has catalyzed significant concerns across various sectors due to its ability to generate convincing fake videos and images that can deceive individuals, manipulate public opinion, and undermine trust in media. This technology leverages sophisticated machine learning and artificial intelligence tools, particularly generative adversarial networks (GANs), to create hyper-realistic media content. Additionally, autoencoders and StyleGAN have also been extensively used for generating deepfakes, where autoencoders efficiently recreate facial images, and StyleGAN allows for the generation of high-quality, customized facial features on demand. As the fidelity and accessibility of deepfake generation increase, the imperative to develop robust detection methods becomes paramount[4]. In the realm of deepfake detection, researchers have explored a multitude of techniques aimed at identifying and mitigating the threats posed by these manipulations. These methods often utilize advanced machine learning algorithms, including convolutional neural networks (CNNs), recurrent neural networks (RNNs), and more recently, transformer models which offer promising results due to their ability to process sequential data more effectively. Techniques such as digital watermarking, signal integrity analysis, and biometric cues analysis also play crucial roles.

Furthermore, the development and enhancement of detection methods are greatly aided by the availability of diverse datasets. Datasets such as [3] DeepFake Detection Challenge (DFDC), FaceForensics++[2], and CelebDFv2 provide a wide range of examples covering various deepfake generation methods, which are essential for training and benchmarking detection models. Each dataset presents unique characteristics and challenges, ranging from varied video quality and compression levels to the sophistication of the face-swapping techniques used.

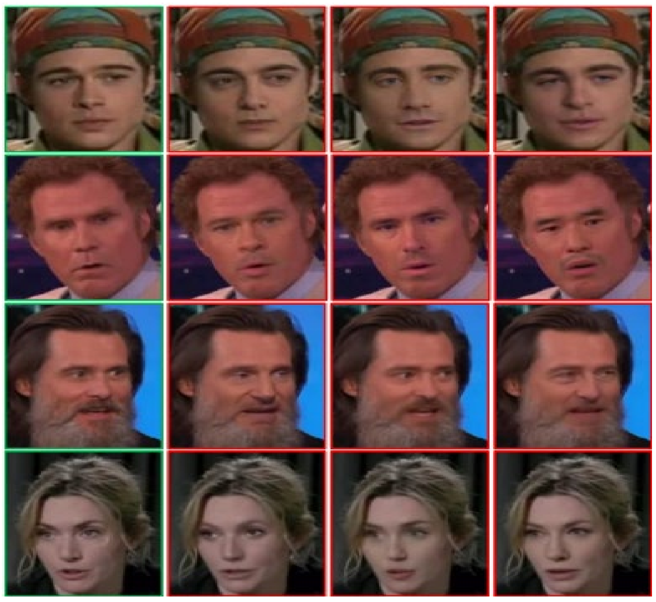


Fig 1: Celeb Dfv2, Left Column is the Pristine Samples

2. Literature Survey

2.1 Deepfake Generation techniques

The process of generating deepfakes involves sophisticated AI technologies that manipulate or generate visual and audio content with a high potential to deceive. The development of deepfake technology has primarily been driven by the advancements in machine learning and neural network architectures.

A) Generative Adversarial networks(GANs)

Generative Adversarial Networks (GANs) [6] are at the forefront of deepfake generation. A GAN consists of two neural networks, the generator, and the discriminator, which are trained simultaneously. The generator creates images or videos that look like the authentic ones, while the discriminator evaluates their authenticity. The continuous adversarial process enhances the quality of the generated fakes, making them increasingly difficult to detect. This technique has been popularized by its ability to produce high-resolution, realistic images and videos.

B) Autoencoders

Autoencoders[5] are another fundamental technology used in deepfake creation. They work by compressing an input (such as an image of a face) into a smaller, dense representation and then decompressing it back into the original format. In the context of deepfakes, autoencoders are typically used in pairs—one for the source face and one for the target face. This allows the system to swap faces by encoding the source face, transferring the encoded data to the target autoencoder, and then decoding it to produce a manipulated output that maintains the target's facial expressions and orientations.

C) Face2Face and FaceSwap

Face2Face[7] is a real-time face capture and reenactment technique that modifies the facial expressions of a target video by using a source actor's expressions. It captures the facial expressions of the source actor with a standard webcam and then transfers them to the target video in real time. This method can be used to alter facial expressions and lip movements in existing videos, thus enabling the manipulation of footage of public figures or anyone else appearing in digital media. FaceSwap is a technique that involves swapping the face of one person with another in a video. This method uses machine learning algorithms to match the skin tone and lighting conditions, making the swap appear more realistic. FaceSwap has been widely used for both benign and malicious purposes, and its accessibility through open-source platforms has made it a popular tool for creating deepfakes.

2.2 Face Extraction techniques

In the realm of deepfake creation and detection, precise face extraction is essential. Two prominent methods used for this purpose are Multi-task Cascaded Convolutional Networks (MTCNN) and MediaPipe. MTCNN[8] is a robust framework designed for efficient face detection, facial landmark localization, and face alignment. It uses a cascaded structure with three stages of deep convolutional networks to refine predictions from coarse to fine, excelling in handling various face positions and occlusions. On the other hand, MediaPipe,[9] developed by Google, offers a versatile ML solution for real-time face detection and landmarks detection across platforms. It features a lightweight, high-performance model that operates effectively on mobile devices, making it ideal for applications requiring fast and reliable face tracking. Both methods serve critical roles in handling face data, with MTCNN providing high accuracy in feature extraction and Media Pipe offering flexibility and ease of integration for real-time applications.

2.3 Feature Extraction

Effective feature extraction is pivotal for both generating and detecting deepfakes, as it significantly impacts the performance of the underlying models. This section reviews three sophisticated feature extraction methods that have proven to be highly efficient in handling deep learning tasks related to image and video processing: InceptionNet, VGG-Face2, and EfficientNetB0.

A. InceptionNet

InceptionNet architectures, particularly those used in versions like Inception V3, are renowned for their depth and width optimization, which helps in efficient computation and robust feature extraction at various scales. The design of these networks includes modules with parallel convolutions of different sizes that allow the network to capture information at various resolutions. This capability makes InceptionNet particularly useful for tasks where contextual information from different scales is crucial.

B. VGG-Face2

VGG-Face2 is an improvement over the original VGG model, specifically tailored for face recognition. It uses deep convolutional networks to process images through a sequence of convolutional layers that capture an increasingly complex hierarchy of facial features. This method is known for its effectiveness in capturing minute and detailed facial characteristics, making it a popular choice for applications that require accurate face verification and identification.

C. EfficientNet B0

EfficientNet B0[1] introduces a scalable architecture that uses a compound coefficient to uniformly scale the depth, width, and resolution of the network. This approach allows for efficient optimization of accuracy and computational resources, making EfficientNet B0 adaptable to various computational constraints while maintaining high performance. Its balanced scaling enhances the model's ability to perform feature extraction tasks across different domains efficiently.

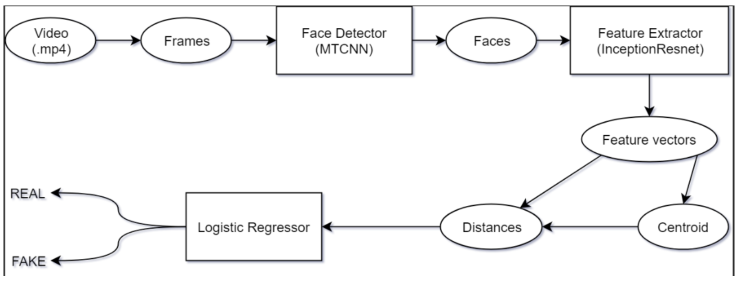


Fig 2.1 Approach1: Proposed Pipeline to extract faces and Classify Deepfakes Vs Pristine Using feature Vector Regression

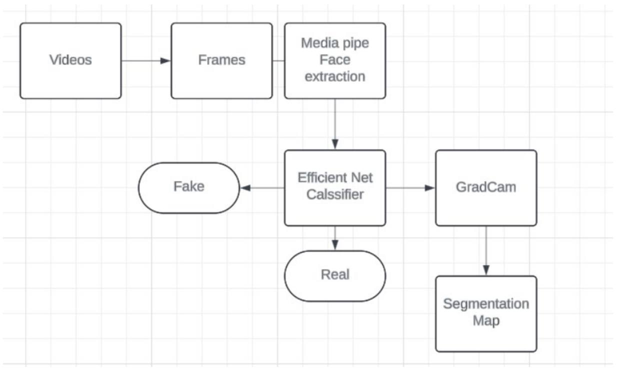


Fig 2.2 Approach2: Proposed Pipeline to Classify Deepfakes vs Pristine and Generation of Grad Cam Segmentation Map

3. METHODS

3.1 Dataset

The CelebDFv2 dataset is a prominent resource in the study of deepfake detection, developed to provide a realistic benchmark for evaluating the performance of deepfake detection algorithms. This dataset is an enhanced version of the original CelebDFv1, featuring significant improvements in the quality and the variability of the deepfake videos it contains. CelebDFv2 was specifically designed to address the challenges posed by the high quality deepfake generation methods.

3.2 Approach 1: Feature Vector Regression

The proposed methodology for deepfake detection is encapsulated in a streamlined pipeline [See Fig 2.1] that integrates various stages from video preprocessing to classification. The workflow begins with raw video data in the .mp4 format, which is then decomposed into individual frames. These frames serve as the input for the face detection stage, where Multi-task Cascaded Convolutional Networks (MTCNN) are employed to identify and extract facial regions with high precision. Once the faces are detected, the next phase involves feature extraction using the InceptionResNet architecture. This network is particularly adept at distilling rich and informative feature vectors from the extracted faces, capturing the nuances that may indicate whether the video is real or fabricated.

The extracted feature vectors then undergo further analysis to compute distances between them, effectively quantifying the divergence in facial features that can be indicative of deepfake manipulation. Additionally, the centroid of the feature vectors is calculated, providing a reference point that represents the 'average' face in terms of the extracted features. This centroid aids in discerning anomalies and inconsistencies in the data that are characteristic of deepfakes.

The final step in the pipeline employs a logistic regressor, a statistical model chosen for its efficacy in binary classification tasks. By inputting the calculated distances and centroid information, the regressor is trained to output a binary decision, classifying the video as either REAL or FAKE. This end-to-end approach ensures a comprehensive analysis of each video frame, facilitating accurate and reliable deepfake detection.

3.3 Approach 2: Efficient Net Classification and Grad Cam

This approach to deepfake detection commences with video files, which are systematically broken down into individual frames. These frames then undergo face extraction via Media Pipe, a versatile machine learning framework that provides real-time face detection . . Media Pipe's ability to detect facial features in various environmental conditions ensures that face extraction is both accurate and robust, providing a strong foundation for the subsequent steps in the detection process. Post face extraction, an Efficient Net B0 classifier is employed to analyze the extracted facial features. The model processes the facial data, outputting a binary classification as either 'Fake' or 'Real'. To further refine the detection process and provide interpretability, Grad-CAM (Gradient-weighted Class Activation Mapping) is integrated into the pipeline. Grad-CAM uses the gradients of any target concept, flowing into the final convolutional layer to produce a coarse localization map highlighting the important regions in the image for predicting the concept. This is particularly useful for identifying the specific areas in the face that led to the model's decision, providing visual explanations for the classifier's determinations. Finally, a segmentation map is generated from the Grad-CAM outputs, which visualizes the areas within the frames that most significantly contribute to the model's classification decision. This segmentation map serves as an intuitive tool for visualizing the specific features and regions that the classifier deems as manipulations indicative of a deepfake, thereby enhancing the transparency and trust in the detection system. This approach not only provides a means for high accuracy deepfake detection but also offers a layer of interpretability to the decision-making process, enabling users to understand and trust the results of the model.

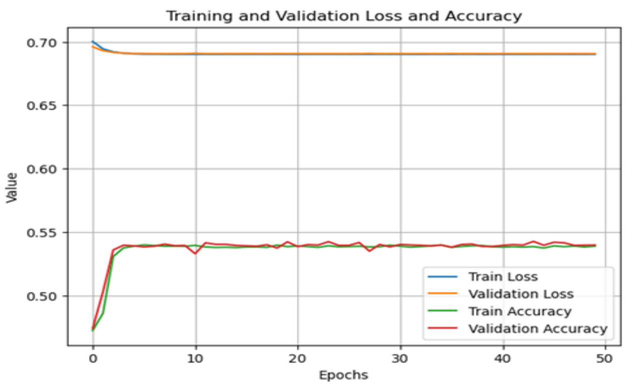


Fig 3.1Approach1 Training and Validation Accuracy/Loss

4. RESULTS

4.1 Approach 1 Results

This result was obvious and showed by the scatter plot in [See Fig 3.2] as the distances did not show any separability for both the labels FAKE and REAL.As a result the Validation and even the Training Accuracy could not cross 54% as shown in Fig3.1 Possible reason for this could be that the deep fakes , when introducing an artifact, the distance created by the artifact would be hidden behind the distance introduced by a low resolution image that has ever changing pixels in a frame to frame setting. Hence its hypothesized that this approach would would for high resolution images which can be a future contribution.

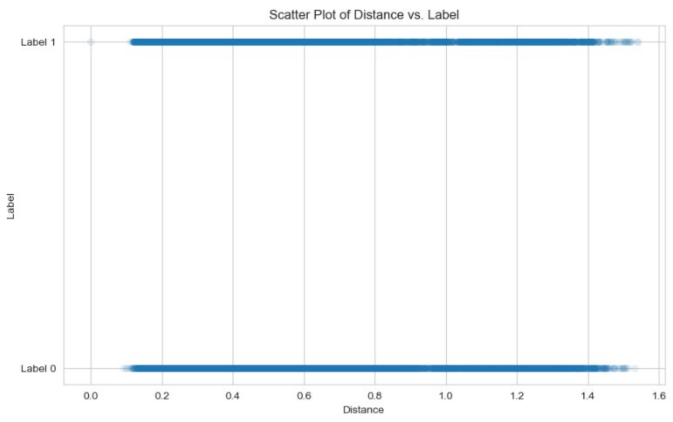
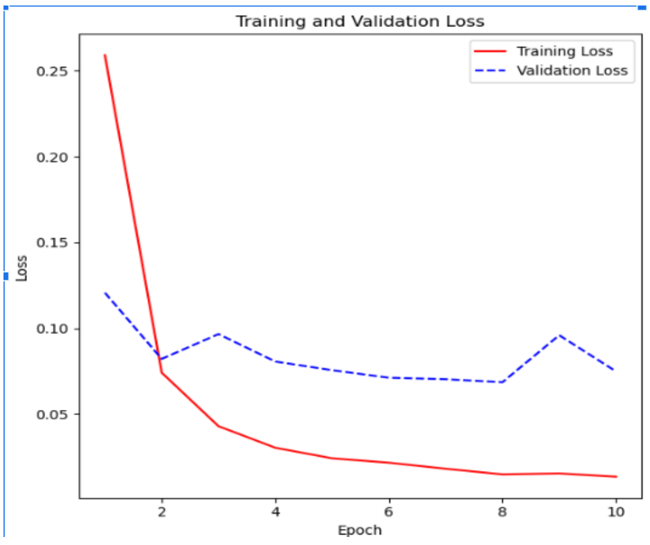
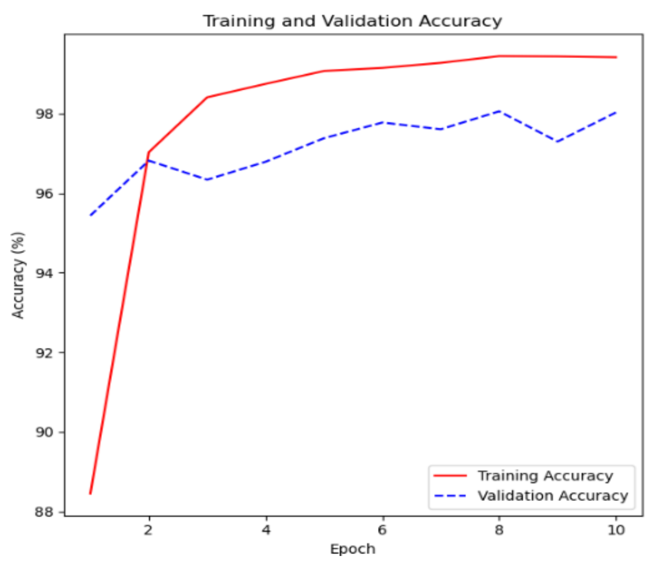


Fig 3.2 Approach 1: Scatter Plot of Feature Vector Distance and Labels

4.2 Approach 2 Results

The approach 2 showed wildly different results with near instant training of 4 epochs to converge at a validation accuracy of 97% and a testing accuracy of 93%. These results can be verified and further analysed by looking at the GradCam segmentation heatmap that sums all gradients in the last layer of the EfficientNet and applies a ReLU to find out contributions of individual pixels in the classification. It can be seen that a strong red circle around the face indicating the area where the new face has been stitched on the new one. It seems to be much better separable using a CNN like Efficient Net directly than sepearting labels over distances of feature vectors. Rather than the artifacts hiding in the data compression , the stich area seems to be a better metric for the network to differentiate and classify the labels.





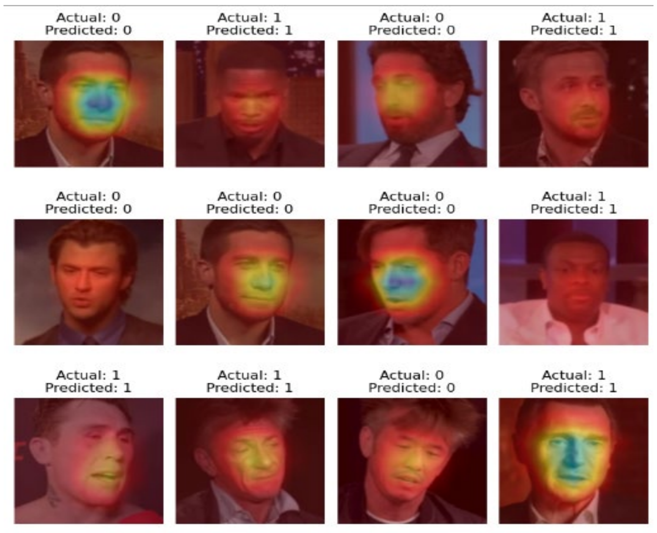


Fig 3.3 A) Approach 2 Training vs Validation Loss

B)Approach 2 Training vs Validation Accuracy

C) Approach 2 GradCam Segmentation Heatmap of Test Data

5. CONCLUSION AND DISCUSSION

The key takeaway from these experiments seems to be the following:

Low quality images will not be seperable with mere distance from the mean frame of the video.The stich area between the outer and inner face is a key area to find artifacts created by the deep-fake techniques. See Fig 3.3 C The face at 2nd row and 3rd column is clearly showing red all along. This clearly means that the background around the face , if considered will introduce a bias in the classification process. It needs to be further studied if the margin/background reduction give enhance the segmentation map results of the particular candidate.The face extarction procedure could be more robust as seen in Fig 4.1, some on the non -faces have been selected as face candidates thus resulting in poor classification.In low resolution, it seems to be futile to further improve a classification accuray of 97% validation as most generated images will be on par at the pristine ones.

This method of classification will work only as long as the pristine images are not contaminated with deepfakes as theron we will have to rely on more preventative measures than deepfake detection.

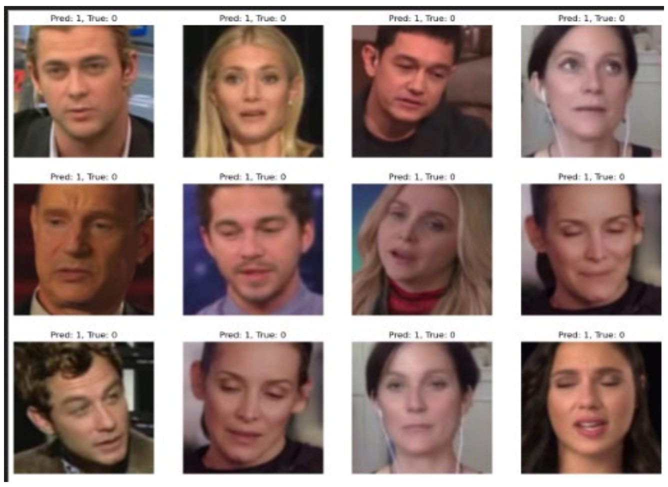


Fig 4.1 Wrongly classified predictions 1 for REAL, 0 for FAKE

6. REFERENCES

[1] Tan, M., & Le, Q. (2019). EfficientNet: Rethinking Model Scalingfor Convolutional Neural Networks. Proceedings of the 36th International Conference on Machine Learning, 6105-6114.

1. **[2] Rossler, A., Cozzolino, D., Verdoliva, L., Riess, C., Thies, J., & Nießner, M. (2019).** FaceForensics++: Learning to Detect Manipulated Facial Images. *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 1-11. DOI: 10.1109/ICCV.2019.00009.

2. **[3] Dolhansky, B., Howes, R., Pflaum, B., Baram, N., & Ferrer, C. (2020).** The DeepFake Detection Challenge (DFDC) Preview Dataset. *arXiv preprint arXiv:2006.07397.*

3. **[4] Agarwal, S., Farid, H., Gu, Y., He, M., Nagano, K., & Li, H. (2019).** Protecting World Leaders Against Deep Fakes. *Proceedings of the CVPR Workshops on Media Forensics*, 50-59.

[5] Kingma, D. P., & Welling, M. (2013). Auto-Encoding Variational Bayes. arXiv preprint arXiv:1312.6114.

[1] **[6] Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2014).** Generative Adversarial Nets. Proceedings of the International Conference on Neural Information Processing Systems (NIPS), 2672-2680.

[7] Thies, J., Zollhöfer, M., Stamminger, M., Theobalt, C., & Nießner, M. (2016). Face2Face: Real-time Face Capture and Reenactment of RGB Videos. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2387-2395. DOI: 10.1109/CVPR.2016.262.

[8] **Zhang, K., Zhang, Z., Li, Z., & Qiao, Y. (2016).** Joint Face Detection and Alignment Using Multitask Cascaded Convolutional Networks. IEEE Signal Processing Letters, 23(10), 1499-1503. DOI: 10.1109/LSP.2016.2603342.

[9] **Lugaresi, C., Tang, J., Nash, J., McClanahan, C., Uboweja, E., Hays, M., Zhang, F., Chang, C.-L., Yong, M. H., Lee, J., Chang, W., Hua, W., Georg, M., & Grundmann, M. (2019).** MediaPipe: A Framework for Building Perception Pipelines. arXiv preprint arXiv:1906.0817