DeepFake Detection

Anonymous CVPR submission

Paper ID 1234

Abstract

In recent years, advances in deep learning have facilitated the creation of realistic deepfake images and videos, presenting a significant challenge in detecting and preventing malicious usage. This report discusses the methodologies and neural network architectures applied in our deepfake detection project, including ResNet50, VGG16, DenseNet, and EfficientNet models, as well as improvements made through data augmentation and transfer learning. Our approach achieved notable accuracy levels, with DenseNet and VGG16 models outperforming others in identifying manipulated content from real media. The results illustrate the effectiveness of CNN-based architectures for this purpose and suggest future areas of improvement. The tests demonstrate a very successful detection rate showing an accuracy of 52.96%, 69.27%, 67.37%, 100%, and 100% for ViT, EfficientNet, ResNet50, VGG16, and DenseNet, respectively.

Keywords— Deep-fakes, face exploitation, AI, Deep Learning, autoencoders, generative adversarial network, forensics, review, convolutional Neural network (CNN), recurrent neural network (RNN)

1. Introduction

With advancements in artificial intelligence, deep generative models like GANs (Generative Adversarial Networks) have created highly realistic synthetic media, commonly known as ”deepfakes.” Deepfake technology can produce images and videos nearly indistinguishable from accurate content by blending authentic elements with manipulated features. Although this technology holds potential for entertainment and creative applications, it raises significant ethical and security concerns. Deepfakes have been used to spread misinformation, impersonate individuals, and influence public opinion, especially in political and social media contexts.

Given these societal risks, there is an urgent need for automated detection systems to keep pace with evolving deepfake generation techniques. Manual detection is impractical due to the high volume of digital media and the increasing sophistication of fake content. Existing approaches to image forgery detection [8, 19] often analyze inconsistencies related to typical camera processes or focus on specific image alterations. For Example, image noise [11] can indicate splicing, while compression artefacts [2] provide clues about manipulation.

Today, the threat of fake news and falsified video content is widely recognized. As over 100 million hours of video are consumed daily on social networks, detecting falsified videos has become increasingly critical. While progress has been made in detecting image forgeries, identifying manipulations in video remains challenging due to frame degradation after compression. Current video forensic methods [16] primarily address video re-encoding [28] and recapture [29, 15], but detecting edited content remains difficult.

This project addresses these challenges by implementing deep learning models for scalable deepfake detection. Leveraging CNNs and transfer learning, we experimented with various architectures to detect fake images effectively. Our work evaluated multiple models to identify the most suitable architecture and training strategy, contributing a valuable tool for distinguishing natural from manipulated media. The project objectives included:

- Model Selection and Testing: Experiment with CNN-based architectures like ResNet50, VGG16, DenseNet, and EfficientNet to assess their accuracy in detecting fake media.
- Data Augmentation and Preprocessing: Implementing preprocessing and augmentation strategies to improve model robustness, reduce overfitting, and manage dataset variations effectively.
- Optimization and Validation: Fine-tuning hyperparameters, optimizing training, and evaluating models across metrics to achieve high accuracy with minimal false positives.

1.1. DeepFake

Deepfake technology aims to swap a person’s face in a video with another’s, initially emerging in 2017 to generate face-swapped media. It uses a pair of autoencoders trained in parallel, each with an encoder and a decoder. The encoder compresses input images into a reduced variable set, while the decoder reconstructs an approximation of the original image by minimizing the difference between the input and the output.

In Deepfake generation, two separate autoencoders E_A and $E_B$are trained on facial images of individuals A and B, respectively. The encoder weights are shared between both autoencoders, but each has a dedicated decoder. This setup encodes general facial features—like illumination, position, and expression—while reconstructing distinct details of each individual's face. The final step applies the trained encoder and decoder to each target video frame, aligning and merging the swapped face.

While this method creates compelling results, it has flaws: facial reenactment can fail if occlusions lead to blurred areas or double contours. Autoencoders struggle with fine details due to input compression and slightly blurting reconstructed faces. Increasing encoding space may improve detail but can reduce realism by passing unwanted morphological information to the decoder.

1.2. Related Work

Deepfake technology, designed to superimpose one person's face onto another's in a video, emerged in 2017 primarily for generating face-swapped media. Facilitated initially by autoencoders, this technique relies on a dual autoencoder setup to encode and decode faces. The encoder compresses facial data while decoders reconstruct faces with specific characteristics, enabling realistic face-swapping with attributes like illumination and facial expression intact [1].

The rapid spread of deepfake content can be attributed to the rise of smartphones and social media platforms [2]. Initially, visual artefacts such as pixel collapse made deepfakes discernible, yet advancements in deep learning and GANs (Generative Adversarial Networks) [3] have made them nearly indistinguishable from authentic media. The application of deepfake technology has expanded across fields, benefiting e-commerce and entertainment by generating realistic virtual experiences. It has also shown potential in medical contexts, such as helping Alzheimer's patients preserve memories or detecting anomalies in X-ray images [4].

The misuse of deepfakes, however, poses significant social risks. Public figures are especially vulnerable, with manipulated content of notable personalities, such as Barack Obama and Joe Biden, and celebrities like Taylor Swift and Emma Watson, circulated online [5,6]. This content can be used for misinformation, political manipulation, and defamation, exacerbating issues like cyberbullying and social distrust. Scholars are responding by developing models to detect deepfakes and mitigate their harmful impacts. Google, Facebook, and Microsoft have created datasets to support the development of new deepfake detection models [7].

Machine learning models have become crucial for detecting deepfake content, employing classifiers such as Support Vector Machines (SVM) and Naive Bayes. Recent advancements include using CNNs (Convolutional Neural Networks) and recurrent neural networks to identify GAN-generated images. Researchers have implemented a range of CNN architectures, including DenseNet, VGG-Face, and ResNet50, which have shown promising results for distinguishing deepfakes from natural images. Researchers also developed custom models using datasets from sources like Kaggle to evaluate model performance based on accuracy, precision, recall, F1-score, and AUC [8–10].

Deepfake creation techniques have continued to evolve, with autoencoders playing a pivotal role. The initial approach to deepfake creation utilized encoder-decoder frameworks, where paired encoder-decoder networks swap characteristics between faces by sharing encoder parameters. These methods are employed in frameworks such as Deepfake [13], Faker [14], and Deep FaceLab, which is based on TensorFlow [15]. Enhanced methods, such as the adaptive GAN architecture [16], attempt to improve face-swapping quality by combining adversarial losses with perceptual quality metrics, as seen in VGG-Face [17].

Other CNN-based methods have advanced face recognition for deepfakes. For instance, FaceNet [18] improves face recognition by optimizing facial orientation and consistency, while CycleGAN [19] allows transformations between facial images without direct supervision, further challenging detection. A recent study by Korshunov and Marcel [23,24] generated a dataset with low- and high-quality deepfake videos to analyze temporal discrepancies and visual irregularities in facial movements [25]. Research has indicated that standard facial detection techniques fail to identify deepfakes due to the advanced features generated by GANs [27].

Efforts to detect deepfakes focus on temporal inconsistencies across frames and interframe coherence. For instance, time-aware CNNs with LSTM layers have been proposed to capture temporal anomalies, as evidenced by Guera and Delp's work [45]. These models focus on phenomena like irregular blinking patterns in deepfakes, as noted by Li et al. [46], who used eye landmarks to detect facial landmark inconsistencies in videos. Additionally, Nguyen et al. [48] introduced container nets, addressing CNN limitations in specific deepfake detection tasks and improving feature routing and information hierarchy [49,50].

In summary, extensive research and innovative deep-learning models are being developed to identify deepfake content. As deepfake techniques become more sophisticated, detection approaches must leverage spatial and temporal features, combining techniques like CNNs, RNNs, and GANs to improve reliability in distinguishing manipulated content from genuine media.

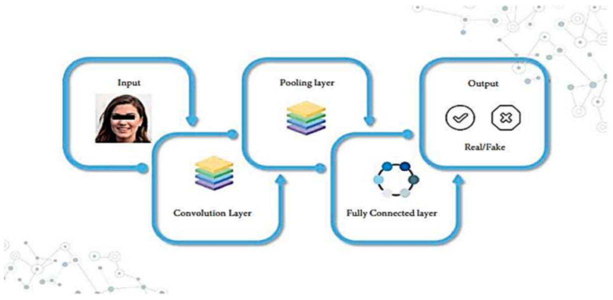

Figure 1. Employment Process Chart.

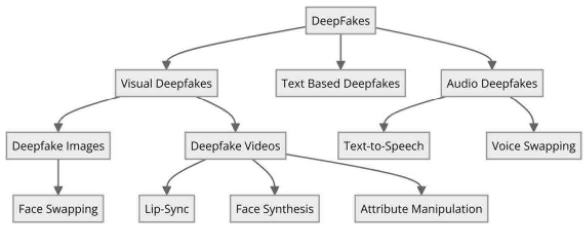

Figure 2. Hierarchical classification of deepfake content available on social media platforms. The image also shows the methods used to create visual and audio deepfakes. Deepfake images and videos are frequently used on social media platforms.

2. Methodology

2.1. Experimental Setup

Python programming language [28] was used to evaluate all experiments. PyTorch module was utilized to build the neural network techniques employed. The scientific evaluations of the research study based on performance metrics were examined. Testing and validation loss, accuracy, precision, and f1-score were the performance metrics utilized for performance evaluations.

2.2. Dataset

The benchmark deepfake dataset used for training and testing our models is publicly available on Kaggle by the Department of Computer Science, Yonsei University. The dataset contains expert-generated photoshopped face images which combine numerous faces separated by nose, eyes, mouth, and whole face. It contains 1081 images of real and 960 images of fake faces. The sample images from the deepfake dataset are analyzed with the target label. https://www.kaggle.com/datasets/ciplab/real-and-fake-face-detection

2.3. Preprocessing

Initially, the images were analyzed for blurriness using the variance of the Laplacian method, with a set threshold to identify low-quality (blurry) images. Blurry images were then sharpened using an edge detection technique, specifically the Canny method, combined with a weighted addition to enhance the edges and overall clarity. Following this, face detection was conducted using the MTCNN (Multi-task Cascaded Convolutional Networks) algorithm to identify facial regions. Detected faces were cropped and saved, replacing the original images to focus solely on the relevant facial data, ensuring consistency and enhancing the dataset's suitability for machine learning tasks related to facial analysis.

2.4. Model

We used multiple deep learning architectures for training and testing data, including EfficientNet, ViT (Vision Transformer), ResNet-50 (32x4d), DenseNet, and VGG16, each chosen for their unique feature extraction and generalization strengths. Initially, we prepared the dataset by resizing images to match the input requirements (e.g., 224x224 pixels) and applied data augmentation techniques, like random cropping, horizontal flipping, rotation, and colour jittering were applied to introduce variety and prevent overfitting. Each model was fine-tuned with pre-trained weights on ImageNet, with training parameters set to optimize performance, including a small learning rate (e.g., 1e-4), batch size of 32, and 30-50 epochs to avoid overfitting. Using Adam or SGD optimizers, we trained the models and evaluated them on the test data, analyzing metrics like accuracy, F1 score, and training loss to determine the best-performing architecture. This approach allowed us to leverage the strengths of each model, comparing results to identify the most effective solution for our dataset.

2.5. ResNet CNN 50

Instead of rewriting the classifier, we utilized the pre-trained **ResNet50** model, which served as a powerful feature extractor for detecting deepfake images. We fine-tuned the network by adding custom layers specific to our task and adjusting the learning rate to ensure proper convergence during gradient descent optimization. This approach allowed us to leverage the pre-trained weights from ImageNet while adapting the model to our deepfake detection task.

The deep layers of **ResNet50** enable the model to capture both high-level and low-level features, making it particularly effective at distinguishing between real and fake faces. By analyzing subtle artefacts introduced during face-swapping, such as irregularities in facial contours, eye blinking inconsistencies, and lighting issues, ResNet50 effectively detects these discrepancies.

After fine-tuning and training the model on our dataset, we achieved an accuracy of **67.37%**, demonstrating the model's capability to identify manipulated content.

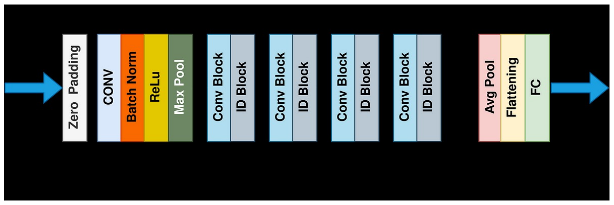

Figure 3. ResNet50 flow structure

2.6. EfficientNet technique

We utilized pre-trained EfficientNet models (b0, b1, b4, and b7) as feature extractors for DeepFake detection, benefiting from their scalable architecture that optimizes accuracy and efficiency. By fine-tuning these models with task-specific layers and adjusting the learning rate, we adapted the models for our deepfake detection needs. EfficientNet's compound scaling approach enables it to capture a wide range of image details, from fine textures to global patterns, which helps identify inconsistencies introduced during face-swapping, such as unnatural lighting, distorted facial shapes, and subtle motion anomalies. The model's versatility across different sizes (b0 to b7) allowed us to explore trade-offs between computational efficiency and detection performance

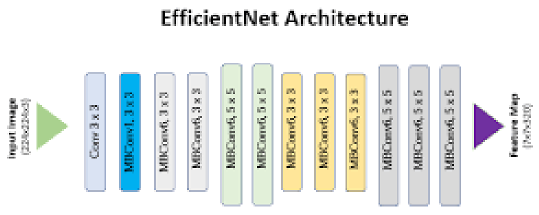

Figure 4. EfficientNet flow structure

2.7. ViT technique

For DeepFake detection, we employed the Vision Transformer (ViT) architecture, leveraging its ability to capture global image context by treating the input image as a sequence of patches. We fine-tuned a pre-trained ViT model on our dataset, adjusting key hyperparameters for optimal performance detecting DeepFake anomalies. Unlike traditional convolutional approaches, ViT processes image patches in parallel, which helps it learn spatial relationships more effectively, making it adept at identifying irregularities introduced during face-swapping, such as mismatched facial textures and unnatural skin details. By focusing on long-range dependencies within the image, the ViT model enhances the detection of subtle inconsistencies in facial features, lighting, and artefacts that typically arise in DeepFake videos.

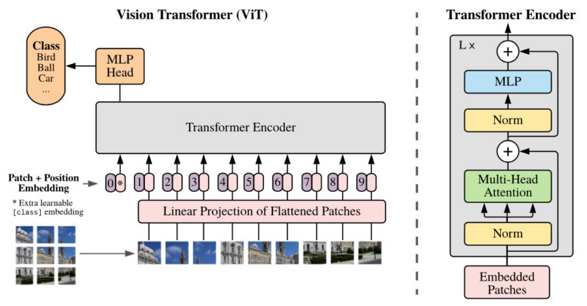

Figure 5. ViT flow structure

2.8. VGG16 technique

We employed the VGG16 model, a CNN architecture known for its success in feature extraction, for deepfake detection. We effectively adapted it by fine-tuning VGG16's pre-trained convolutional layers and replacing the top layers with a custom classification block to distinguish between real and fake faces.

Configuration Overview: Input and Convolutional Layers: Input images were resized to (224, 224, 3). The pre-trained convolutional layers extract hierarchical features, identifying patterns from simple edges to complex structures like facial shapes and lighting irregularities. Classification Block: A dropout layer after flattening prevents overfitting. Dense layers with ReLU activation refine feature combinations, and a sigmoid output layer provides a binary probability for real vs. fake classification. Training Parameters: Approximately 14 million parameters were trained using Adam optimizer and cross-entropy loss over 30 epochs, with a learning rate scheduler adjusting every 10 epochs. Performance: The model achieved 100% accuracy in identifying deepfakes, leveraging VGG16's deep layers to capture subtle artefacts like lighting inconsistencies and facial irregularities often present in manipulated images.

This adaptation of VGG16 underscores its strength in detecting complex image manipulations characteristic of deepfakes.

2.9. DenseNet technique

The DenseNet model, initially proposed by Huang et al., is another powerful pre-trained neural network widely used for image recognition tasks. DenseNet is built on Dense

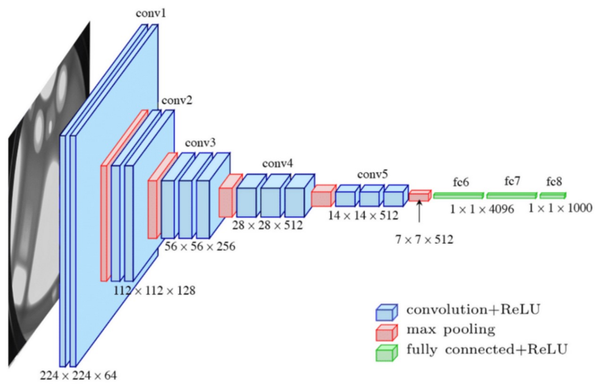

Figure 6. VGG16 flow structure

Convolutional Network (DenseNet) architecture, which enhances information flow between layers by connecting each layer to every other layer in a feed-forward manner. This architecture was first utilized in the 2017 ILSVRC competition. In our study, we adapted DenseNet for deepfake detection by replacing its top layers with a Multi-Layer Perceptron (MLP) block.

We detail the configuration parameters and layer structure of DenseNet. The input layer of the model has an output shape of (100, 256, 256, 3). DenseNet incorporates 512 units and a high parameter count of approximately 12 million to handle complex patterns within images. To reduce overfitting, a dropout layer is included after the convolutional blocks. Flattening layers are then employed to transform the pixel data into a one-dimensional array. Dense layers are added to aid in deepfake prediction, with an eight-unit dense layer using ReLU activation. Finally, a sigmoid-activated output layer is applied for binary classification in deepfake detection.

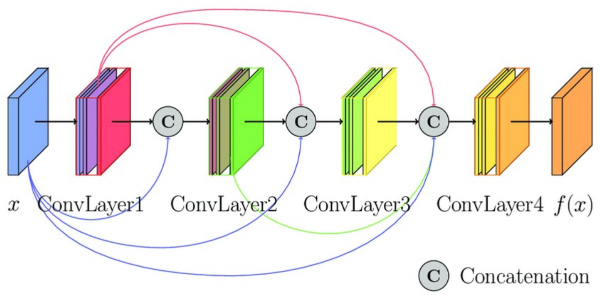

Figure 7. DenseNet flow structure

3. Conclusion

In conclusion, this project aimed to explore various deep learning architectures for DeepFake detection, assessing their performance based on key metrics such as loss, accuracy, F1 score, and ROC-AUC. Among the models tested,

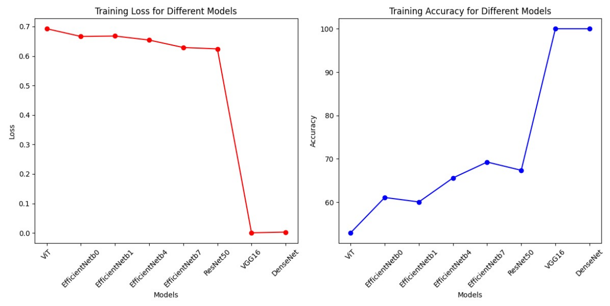

Figure 8. Accuracy and Loss Plot for Different Models

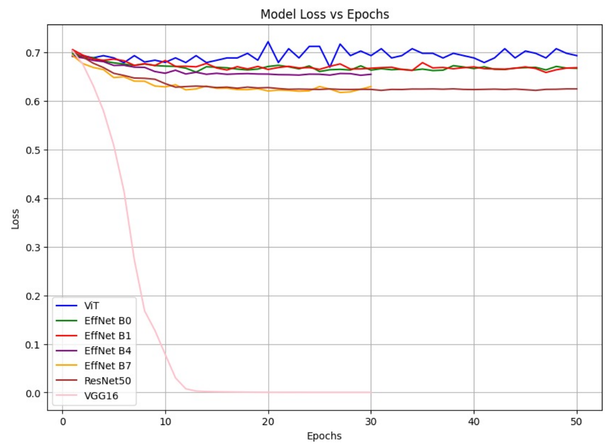

Figure 9. Model Loss and Epochs

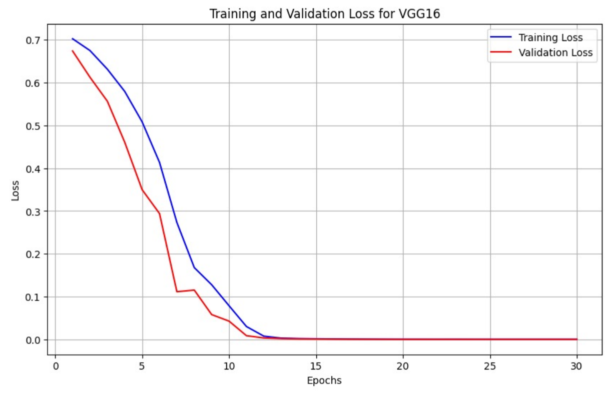

Figure 10. Validation Loss for VGG16

VGG16 and DenseNet achieved perfect accuracy of 100%, with extremely low loss values, indicating their strong ability to detect DeepFake content effectively. ResNet50 also showed strong performance, with an accuracy of 67.37% and relatively low loss, making it a competitive model for this task.

In contrast, the **Vision Transformer (ViT)**, while innovative, showed the lowest accuracy at **52.96%**, with a loss of 69.25% and relatively poor ROC-AUC (0.5128), indicat-

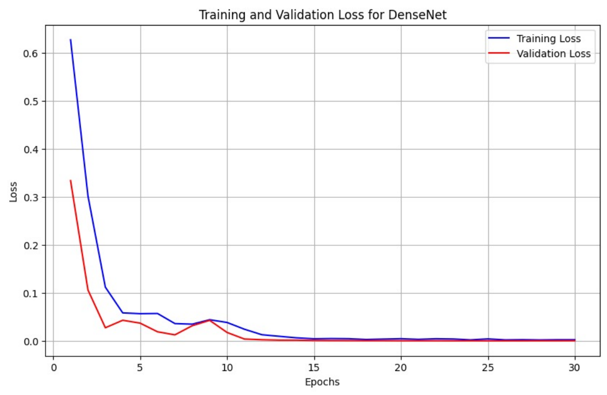

Figure 11. Validation Loss for DenseNet

ing that this architecture struggles with capturing the fine details necessary for DeepFake detection in this context. However, ViT's ability to process global image context may still hold potential with further tuning and optimization.

The EfficientNet family demonstrated promising results, with EfficientNetb7 achieving the highest accuracy within the **EfficientNet models at 69.27%**, followed closely by EfficientNetb4 at 65.61%. These models outperformed ViT, though they did not match the performance of **100% accuracy of VGG16 or DenseNet**.

While traditional CNN-based models like VGG16 and DenseNet delivered superior results, future work could improve ViT and EfficientNet architectures by fine-tuning their parameters further or incorporating hybrid approaches to enhance DeepFake detection. This work can also be extended to deepfake detection in videos by isolating various frames.

4. References

[1] B. Balas and C. Tonsager. Face animacy is not all in the eyes: Evidence from contrast chimeras. Perception, 43(5):355– 367, 2014. 3

[2] M. Barni, L. Bondi, N. Bonettini, P. Bestagini, A. Costanzo, M. Maggini, B. Tondi, and S. Tubaro. Aligned and nonaligned double jpeg detection using convolutional neural networks. Journal of Visual Communication and Image Representation, 49:153–163, 2017. 1

[3] B. Bayar and M. C. Stamm. A deep learning approach to universal image manipulation detection using a new convolutional layer. In Proceedings of the 4th ACM Workshop on Information Hiding and Multimedia Security, pages 5–10. ACM, 2016. 1

[4] F. Chollet. Xception: Deep learning with depthwise separable convolutions. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 1800–1807, 2017. 5

[5] F. Chollet et al. Keras. https://keras.io, 2015. 5

[6] D. Erhan, Y. Bengio, A. Courville, and P. Vincent. Visualizing higher-layer features of a deep network. University of Montreal, 1341(3):1, 2009. 6

[7] S. Fan, R. Wang, T.-T. Ng, C. Y.-C. Tan, J. S. Herberg, and B. L. Koenig. Human perception of visual realism for photo and computer-generated face images. ACM Transactions on Applied Perception (TAP), 11(2):7, 2014. 3

[8] H. Farid. A Survey Of Image Forgery Detection. IEEE Signal Processing Magazine, 26(2):26–25, 2009. 1

[9] P. Garrido, L. Valgaerts, O. Rehmsen, T. Thormahlen, P. Perez, and C. Theobalt. Automatic face reenactment. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 4217–4224, 2014. 2

[10] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. arXiv preprint arXiv:1502.03167, 2015. 3

[11] T. Julliand, V. Nozick, and H. Talbot. Image noise and digital image forensics. In Y.-Q. Shi, J. H. Kim, F. Perez-Gonz ´ alez, ´ and I. Echizen, editors, Digital-Forensics and Watermarking: 14th International Workshop (IWDW 2015), volume 9569, pages 3–17, Tokyo, Japan, October 2015. Lecture Notes in Computer Science (LNCS), Springer International Publishing. 1

[12] D. E. King. Dlib-ml: A machine learning toolkit. Journal of Machine Learning Research, 10:1755–1758, 2009. 4

[13] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980, 2014. 5

[14] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In Advances in neural information processing systems, pages 1097–1105, 2012. 3

[15] J.-W. Lee, M.-J. Lee, T.-W. Oh, S.-J. Ryu, and H.-K. Lee. Screenshot identification using combing artifact from interlaced video. In Proceedings of the 12th ACM workshop on Multimedia and security, pages 49–54. ACM, 2010. 1

[16] S. Milani, M. Fontani, P. Bestagini, M. Barni, A. Piva, M. Tagliasacchi, and S. Tubaro. An overview on video forensics. APSIPA Transactions on Signal and Information Processing, 1, 2012. 1

[17] N. Rahmouni, V. Nozick, J. Yamagishi, and I. Echizen. Distinguishing computer graphics from natural images using convolution neural networks. In IEEE Workshop on Information Forensics and Security, WIFS 2017, Rennes, France, December 2017. IEEE. 1

[18] Y. Rao and J. Ni. A deep learning approach to detection of splicing and copy-move forgeries in images. In Information Forensics and Security (WIFS), 2016 IEEE International Workshop on, pages 1–6. IEEE, 2016. 1

[19] J. A. Redi, W. Taktak, and J.-L. Dugelay. Digital

image forensics: a booklet for beginners. Multimedia Tools and Applications, 51(1):133–162, 2011. 1

[20] A. Rossler, D. Cozzolino, L. Verdoliva, C. Riess, J. Thies, ¨ and M. Nießner. Faceforensics: A large-scale video dataset for forgery detection in human faces. arXiv preprint arXiv:1803.09179, 2018. 4, 5

[21] V. Schetinger, M. M. Oliveira, R. da Silva, and T. J. Carvalho. Humans are easily fooled by digital images. arXiv preprint arXiv:1509.05301, 2015. 3

[22] W. Shi, F. Jiang, and D. Zhao. Single image superresolution with dilated convolution based multi-scale information learning inception module. arXiv preprint arXiv:1707.07128, 2017. 3

[23] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556, 2014. 3

[24] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. The Journal of Machine Learning Research, 15(1):1929–1958, 2014. 3

[25] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, et al. Going deeper with convolutions. Cvpr, 2015. 3

[26] J. Thies, M. Zollhofer, M. Stamminger, C. Theobalt, and M. Nießner. Face2face: Real-time face capture and reenactment of rgb videos. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 2387– 2395, 2016. 1, 2, 3

[27] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In Computer Vision and Pattern Recognition, 2001. CVPR 2001., volume 1, pages I–I. IEEE, 2001. 4

[28] W. Wang and H. Farid. Exposing digital forgeries in video by detecting double mpeg compression. In Proceedings of the 8th workshop on Multimedia and security, pages 37–47. ACM, 2006. 1

[29] W. Wang and H. Farid. Detecting re-projected video. In International Workshop on Information Hiding, pages 72– 86. Springer, 2008. 1

[30] F. Yu and V. Koltun. Multi-scale context aggregation by dilated convolutions. preprint arXiv:1511.07122, 2015. 3

5. Appendix

https : / / colab . research . google . com / drive / 11o1rob9eOMz6 - T1rhyjqtNGEjsZQOzv6 ? usp = sharing This Google Colab link provides the code for training and testing the dataset.

https : / / www . kaggle . com / datasets / ciplab/real-and-fake-face-detection This is the dataset used for training purpose.