Evaluating Deep Learning Techniques for Detecting Synthetic Media Manipulations

Gayathry Chandramana Krishnan Nampoothiry  
Computing and Information Sciences  
*University of North Florida*Jacksonville, Florida  
gayathryck@gmail.com

*Abstract*— The rapid advancement of Artificial Intelligence, machine learning and deep learning technologies during recent decades brought forth innovative techniques and various tools for multimedia manipulation. The malicious use of deep learning algorithms has enabled the creation of incredibly realistic digital content such as videos, images, and audio, known as Deepfakes. By manipulating facial expressions, voices, or even entire scenes, deepfakes can blur the line between real and fake content, making it increasingly difficult to differentiate between what is genuine and what has been artificially created. This situation demands the need of strong detection systems to identity and halt the proliferation of fake content on social media and digital platforms. This research paper examines deep learning models like InceptionResnetV2, VGG19, CNN, and Xception for identifying deepfake images and videos. These architectures specialize in detecting small anomalies that humans cannot notice to strengthen protection against AI-generated media manipulations. For training and evaluation, data has been taken from the Kaggle deep fake dataset, and the results show high accuracy for the Xception algorithm achieving an accuracy of 99%.

Keywords— *Deepfake Detection, Deep Learning, Generative Adversarial Networks (GAN), Kaggle, InceptionResnetV2, VGG19, Convolution Neural Network (CNN), Xception*

# Introduction

The name ‘Deepfake’ results from combining the concepts of Deep Learning (DL) and Fake. It describes specific photo-realistic video, image or audio contents created with the support of DL. The unethical use of deepfakes have been aimed at to spread misinformation, propaganda and fake news, foment political discord and hate, or even harass and blackmail people. Public figures and celebrities are often a target for deepfakes as a large number of publicly available media is readily accessible for training deepfake models, and subsequent deepfake videos have the power to influence a large number of people.

AI-enabled software tools like FaceApp, FaceSwap and DeepFaceLab have been used for generating realistic-looking face swapping images and videos. All these applications are built on a deep neural network. The best model for the creation of DeepFakes are Generative Adversarial Networks (GAN) which were proposed based on zero-sum game theory and introduced in 2014 by Ian Goodfellow [1]. GAN combines two neural networks, a generator and a discriminator that learn simultaneously. The generator tries to capture the potential distribution of real samples, and generates new data samples. The discriminator is often a binary classifier, discriminating real samples from the generated samples as accurately as possible. The optimization process of GANs is a minimax game process. The min represents 0 and max represents 1. The 0 represents fake output and 1 represent the authentic output. Discriminator tries to get closer to 1 in order to create a realistic Deepfake and the optimization goal is to reach Nash equilibrium [2], where the generator can be considered to have accurately captured the distribution of real samples. It is becoming more and more important and challenging to study how to distinguish whether a picture or video is true.

With the ongoing evolution of deepfake technologies detection becomes increasingly difficult. Detecting synthetic content requires systems that identify subtle digital traces and temporal anomalies which humans typically cannot see. Detection efforts become more difficult when faced with diverse lighting conditions as well as varied backgrounds and changes in video resolution. Detection accuracy experiences substantial improvement when powerful model architectures that capture spatial and temporal features are applied. The widespread availability of deepfake creation tools demands stronger and more adaptable detection models than ever before.

Most current algorithms for detecting Deepfakes rely on manually crafted features or Convolutional Neural Networks (CNNs) for processing. This paper examines how multiple pre-trained leading neural networks including InceptionResNetV2, VGG-19, CNN and Xception perform when faced with real-world variances and explores their effectiveness for practical applications. This research involves detailed analysis and testing with the Deepfake Challenge Dataset which includes a wide range of genuine and altered video samples obtained from Kaggle.

# Releated Works

Deepfake information is spreading more quickly than ever before because of the increasing use of cell phones and AI software tools. At present, techniques for detecting deepfakes concentrate on tackling the binary classification of images or frames relying on deep learning. These methods identify inconsistencies in videos or images based on eye blinking rate [3], noting the difference between head pose [4] of an original video and fake video, and detecting the artifacts of eyes, teeth and face [5], face warping artifacts, lack of self-consistency etc. The MesoNet [6] algorithm is introduced as a shallow CNN designed for detecting fake faces by examining each frame individually through the utilization of CNNs. A general capsule-network based method has been proposed to detect manipulated images and videos [7].

Other Deep Learning (DL) techniques used to identify deepfakes includes MobileNetV2 [8], InceptionV3 [9], DensNet121 [10],EfficientNetB0 [11]**,** A detailed study has been conducted summarizing 112 relevant articles on a variety of methodologies [12] like deep learning-based techniques, classical machine learning-based methods, statistical techniques, and blockchain-based techniques. They researchers evaluated the performance of various methods and concluded deep learning method outperform other methods in deepfake detection.

Guera and Delp [13] proposed end to end trainable recurrent systems: a CNN that extracts most critical features and an LSTM for sequential analysis. The model they developed accurately predicts if the fragment being analyzed comes from a deepfake video or not with an accuracy greater than 97%.

A lot of advanced research is being conducted to improve the performance and accuracy of deepfake detection and prevent the spread of deepfakes as a permanent solution. A recent study on deepfake detection using block chain technology [14] provides an effective solution to protect the integrity and trustworthiness of visual media. The approach combines SegCaps and CNN methods for improved image feature extraction, followed by capsule network (CN) training to enhance generalization. A novel data normalization technique is introduced to tackle data heterogeneity stemming from diverse global data sources. Moreover, transfer learning (TL) and preprocessing methods are deployed to elevate DL performance. The results show a remarkable accuracy improvement of 6.6% compared to six benchmark models.

Research efforts have explored an innovative model using a combination of Quantum Transfer Learning (QTL) and Class-Attention Vision Transformer (CaiT) architectures [15]. The advantages of quantum computing offered by QTL were merged with the global feature extraction capabilities of the CaiT. The results demonstrated that the proposed method achieved a remarkable performance in detecting deepfake videos, with an accuracy of 90% and ROC AUC score of 0.94 achieved.

# Proposed Approach

The proposed methodology contains several essential steps needed for this work. The initial stage involved collecting data which was then followed by choosing appropriate deep learning models for assessment. The main aim of my research is to explore different deep learning algorithms and evaluate each one's performance in detecting deepfakes using metrics like accuracy, support, recall, F1 score, and precision. This research examines models like InceptionResnetV2 [16], VGG19 [17], CNN [18], and Xception [19]. The comparison and results of the above deepfake detection techniques are summarized at the end of this paper. Figure 1 shows the entire architecture of proposed system.

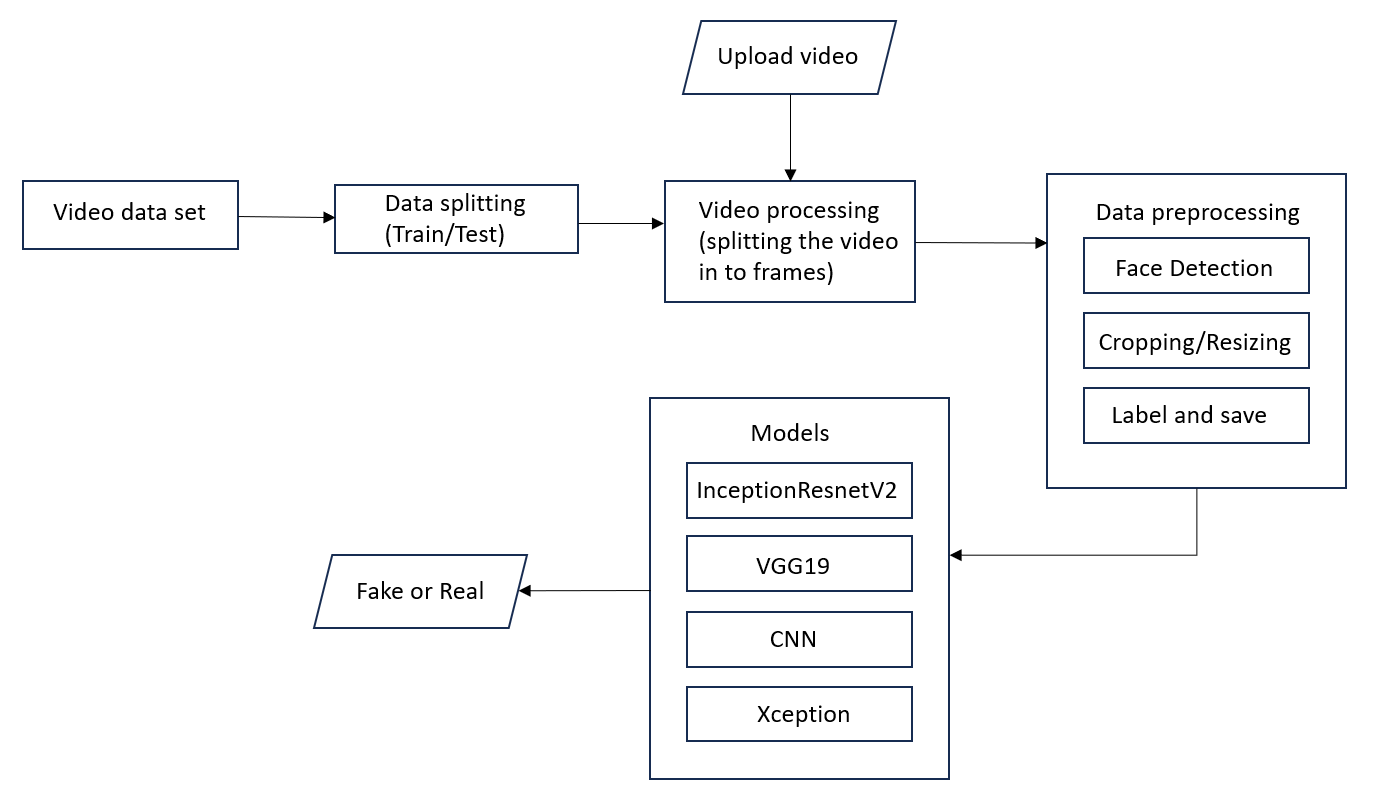


Figure 1: System Architecture

## Data Set

The dataset used in this study is from Deepfake Detection Challenge of Kaggle [20]. The size of the downloaded dataset is 2.14 GB, comprising both real and fake video files. The training dataset consists of 400 videos, with metadata. Json file consists 323 fake videos and 77 real videos, along with 400 testing videos and a CSV file. This dataset provides multiple situations where people stand or sit and face towards or away from the camera along with various backgrounds, lighting combinations, and video quality levels. Training videos use a resolution of 1920 × 1080 pixels and vertical videos use 1080 × 1920 pixels. The dataset also includes 400 private videos for testing purposes. A total of 80% of the dataset was used for training and 20% for testing purpose.

## Data Preprocessing

This stage includes Data preparation, feature extraction and classification. Firstly, the video input is divided into frames and perform face recognition trimming those that contain detected faces. Algorithm performance improves through image flattening and noise reduction since cleaner images demonstrate significant accuracy gains for models. Pre-processing techniques including image resizing and data normalization optimize the model's input layer to boost performance. Applying these operations helps remove distortions which could affect network performance while simultaneously improving features that should be emphasized.

In our experiment, we utilize the dlib and OpenCV packages for this task. It extracts frames from each video. From these frames, images containing human faces are extracted, labelled and stored as Real or Fake along with video file name and frame number in .png format. Now resolution of each image is standardized and adjusted to 128\*128 pixels to serve as input for the classification algorithm. Training involves both feature extraction and category label assignment. The dataset divides into training and testing parts where learning occurs through labeled frames from both real and fake videos. The testing phase does not include labeling operations because its purpose is to assess how well the model can differentiate real content from fake content.

## Data Modelling

In my research, the performance of 4 different deep learning models is compared for detecting deep fakes. Complete code and architecture diagrams used in this study can be accessed from my public repository at “<https://github.com/GayathryCK/Information-Assurance>”

### InceptionResnetV2:

InceptionResNetV2[21] Architecture is a 164 layer deep neural network based on Inception V3 and ResNet framework and was trained using more than a million photos from the ImageNet collection. This network excels at categorizing images into 1000 item categories, spanning various objects, animals, keyboards and more. The network receives a 299 by 299-pixel picture as input. Multiple convolutional filters of various sizes merge with residual connections in the Inception-Resnet block. This fusion allows Inception\_ResNet\_v2 to capture intricate features and patterns at different abstraction levels using its deep layers, while mitigating the vanishing gradient problem. In addition to avoiding the degradation issue brought on by deep structures, the inclusion of residual connections shortens training time.

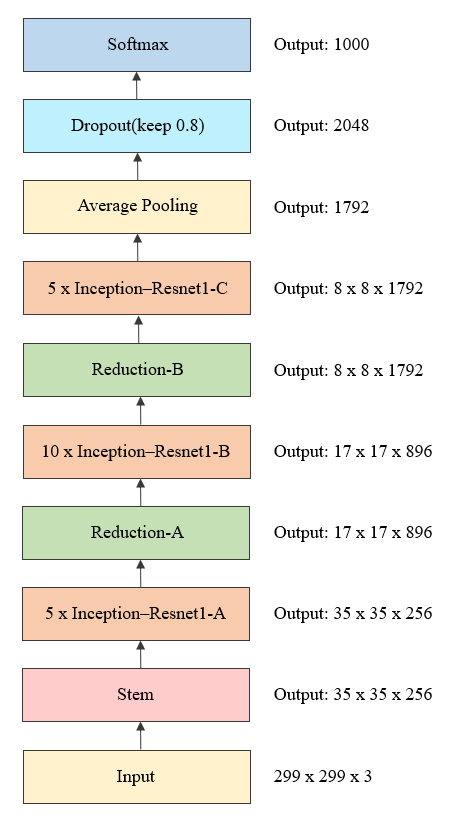
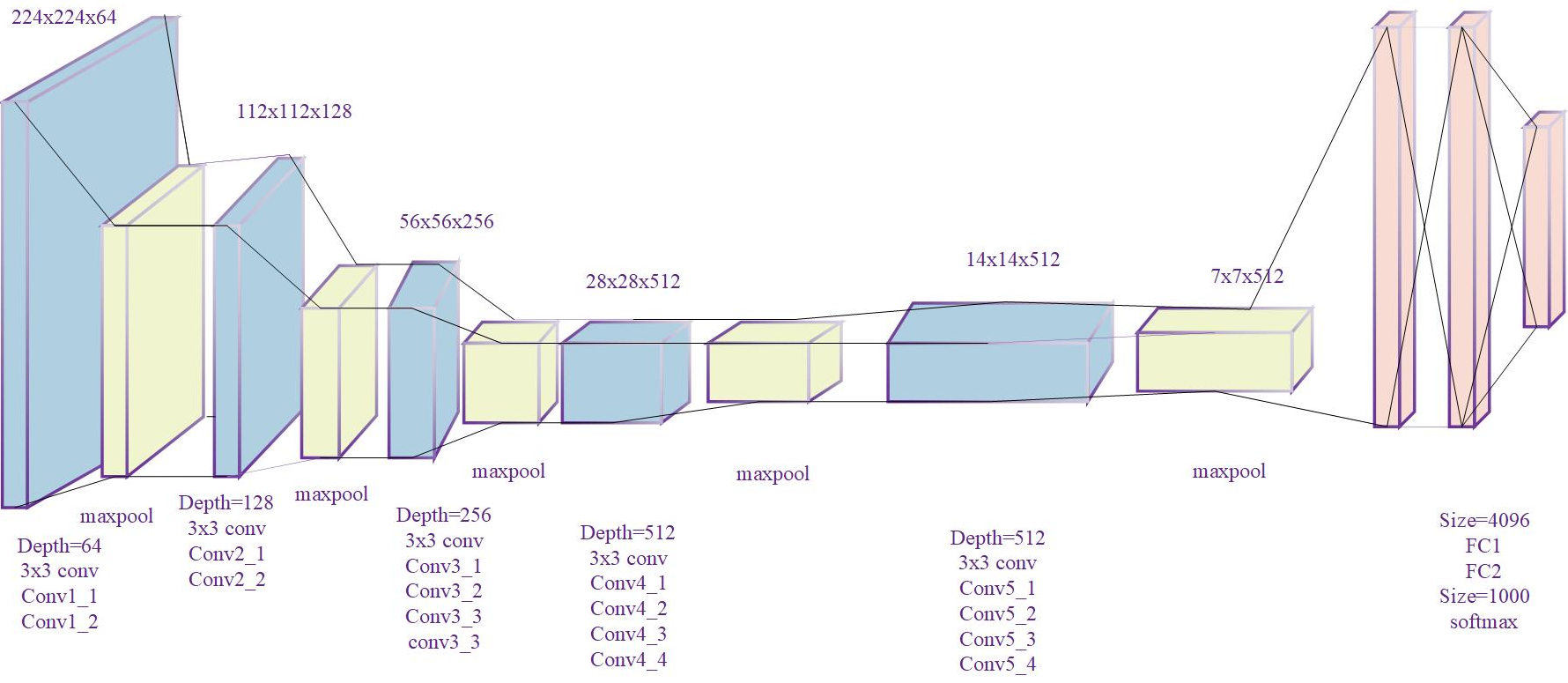


Figure 2: Architecture of InceptionResnetV2

Figure 2 architecture represents the Inception-ResNet-v2 model which serves as an advanced convolutional neural network tailored for image classification tasks. The architecture starts from the Input layer which takes images of dimensions 299×299×3 that include height, width and RGB color channels. The Stem layer follows the Input layer by performing initial convolution and pooling operations to detect low-level image features such as edges and textures with an output shape of 35×35×256. The model employs five Inception-ResNet-A modules to merge Inception layer efficiency with residual connection learning strength which allows it to capture complex patterns without degradation while producing an output size of 35×35×256. The Reduction-A module decreases spatial dimensions but expands depth to produce feature maps suitable for advanced learning with a result size of 17×17×896. The network then uses 10 Inception-ResNet-B modules to process and enhance mid-level features through residual learning while keeping the dimensions at 17×17×896. Reduction-B follows next by reducing spatial resolution and increasing depth to 8×8×1792 for enhanced semantic feature extraction. The model utilizes five Inception-ResNet-C modules at this point to capture abstract features while keeping the output dimensions unchanged. Average Pooling then transforms each feature map into a single value through global averaging which produces a 1792-dimensional vector. The model uses a Dropout layer with 80% neuron retention rate to combat overfitting by omitting neurons randomly throughout training before connecting to a Softmax layer that produces probabilities for 1000 classes. All stages function as essential components for step-by-step visualization refinement which leads to precise classification results.

1. *VGG19:*

The Visual Geometry Group at the University of Oxford developed the deep convolutional neural network model known as VGG19. Figure 3 shows the architecture of VGG19 model. VGG19 network comprises 19 layers. The network structure contains convolution layers and MaxPooling layers alongside fully connected layers and a SoftMax layer. Five convolutional blocks make up the overall architecture with each block followed by a MaxPooling2D layer. Block 1 contains two convolutional layers that use 64 filters. Block 2 has two convolutional layers utilizing 128 filters. Block 3 includes four convolutional layers with 256 filters. Block 4 features four convolutional layers, each containing 512 filters. Block 5 also contains four convolutional layers using 512 filters. Each MaxPooling2D layer decreases the spatial dimensions while capturing the critical features for classification. Through these 5 blocks, VGG19 follows a structured and step-by-step approach to feature extraction, allowing it to thoroughly understand and represent input images with increasing levels of detail.

Figure 3: Architecture of VGG19

1. *CNN:*

The Convolutional Neural Network (CNN) deep learning algorithm focuses on performing image recognition and processing tasks. The neural network consists of several layers which encompass convolutional layers followed by pooling layers and ending with fully connected layers. CNN analyzes images through pixel matrix inputs which are normalized to enable efficient processing. The convolution operation starts by moving a small n\*n kernel (filter) across the image to multiply pixel values by its weights which helps in identifying patterns to create feature maps. An activation function such as ReLU adds non-linearity to the process while keeping only essential features. MaxPooling and similar pooling layers decrease the size of feature maps but maintain essential information which results in more efficient computation. As layers deepen, CNNs become more adept at recognizing complex patterns but may also lose fine details due to pooling, and computational costs increase. To counteract this, dilated convolution introduces gaps (zero-weight holes) within filters, expanding their receptive field without adding extra parameters, enabling the model to capture broader contextual information. Extracted features are then flattened and passed through fully connected layers, where neurons determine relationships between features, leading to the final classification output. The loss function measures how well predictions align with actual labels, guiding backpropagation and gradient descent to optimize weights for better accuracy. CNNs are widely used in image classification, object detection, facial recognition, medical imaging, and various other computer vision applications due to their ability to automatically learn and extract meaningful image features. Figure 4 shows the architecture of CNN model.

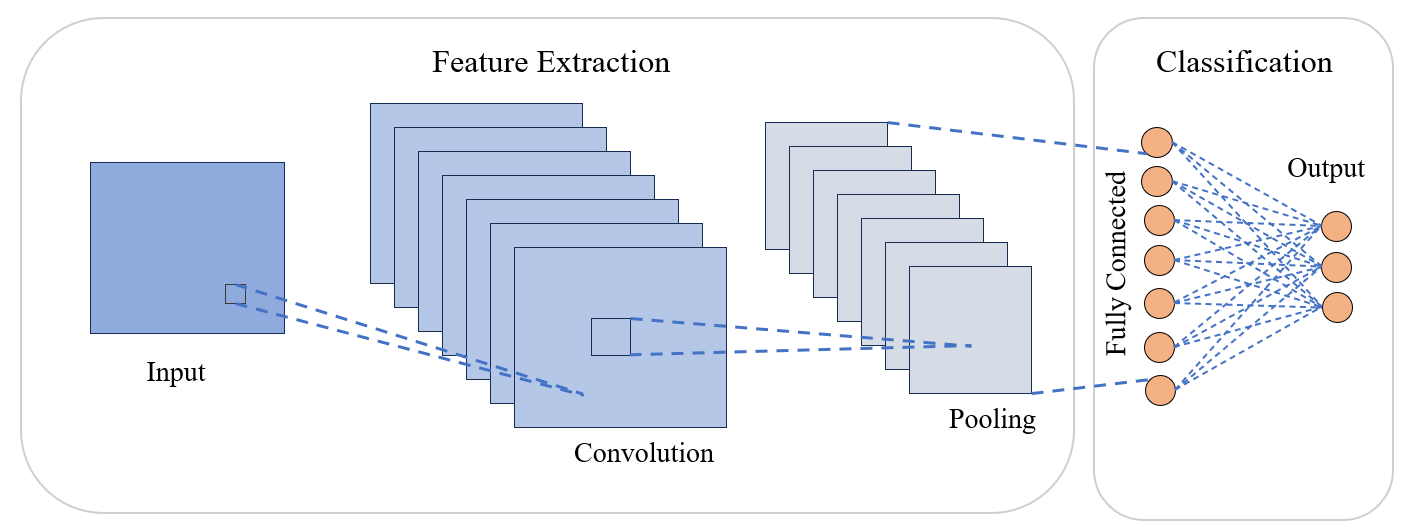


Figure 4: Architecture of CNN

1. *Xception:*

The Xception (short for Xtreme Inception) architecture, inspired by the Inception concept, is a convolution neural network with 71 layers. It employs depth-wise separable convolutions, followed by pointwise convolution, which split channel-wise and spatial convolutions into separate layers [22]. Figure 5 represents Xception model. In the first stage, the input data is convolved channel-wise with separate learnable filters. This process captures spatial features within each input channel (R, G, B) independently. This strategy reduces computation and enhances feature learning efficiency. The second stage involves pointwise convolutions, where the depth-wise convolved outputs are linearly combined using 1x1 convolutions. This step enables the network to learn cross-channel feature interactions. As such, Xception learns spatial and cross-channel correlations separately and thus learns complex dependencies in the input data. Xception obtains this level of performance with a more parameter efficient architecture that uses depth wise separable convolutions to reduce the number of parameters. Both features representation and computation efficiency have been Xception's goal in a variety of image analysis applications and it has been successful in achieving them.

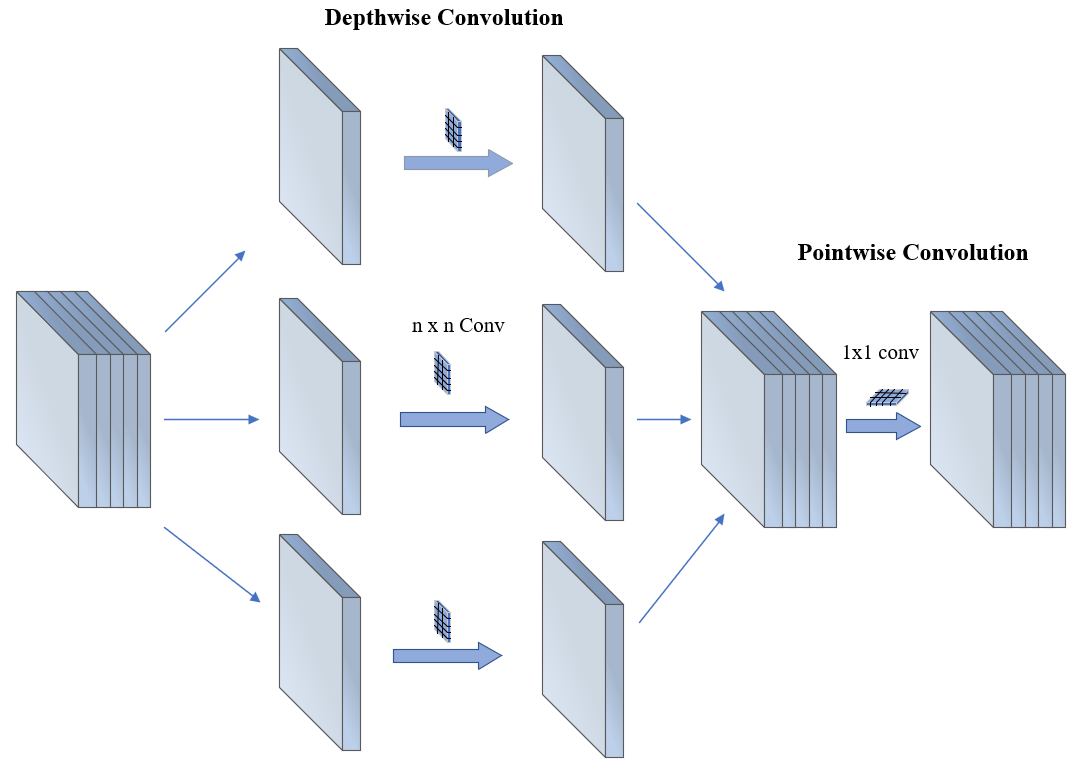


Figure 5: Architecture of Xception

## Fine Tuning

The performance and training efficiency of deep learning models heavily depend on properly chosen hyperparameters. Hyperparameters determine several training aspects such as learning rate and batch size before training begins while model parameters emerge from the training process. The correct adjustment of hyperparameters helps the model perform well on new data while preventing problems such as overfitting and underfitting. The study implemented carefully chosen hyperparameters to enhance deepfake detection model training and ensure high performance across different evaluation metrics. Table-1 gives the hyperparameters used in my research.

|  |  |
| --- | --- |
| **Hyperparameters** | **Values** |
| Optimizer | Adam |
| Loss Function | Categorial\_Cross entropy |
| Image size | 128\*128 |
| Batch size | 64 |
| Epochs | 50 |
| Patience | 3 |
| Activation | Softmax, Relu |
| Verbose | 1 |
| Metrics | Accuracy, Precision, F1 Score, Recall, Specificity |

Table-1

# Evaluation Results

In the evaluation of results, 4 deep learning models were subjected to rigorous testing to assess their performance in deep fake image detection. Accuracy, support precision, recall, F1-score and specificity were used to evaluate the models to assess their efficacy in distinguishing between authentic and manipulated images. Upon analyzing the comparative performance metrics illustrated in Figures 6 through 14, the Xception model clearly outperformed the other architectures across all key metrics. Figure 6 gives the detailed comparison of the results obtained for the above metrices. The Xception model reached top accuracy levels of 99.39%. This implies that Xception made fewer errors in identifying images as fake or real than other models. When comparing precision and recall metrics Xception achieved the top results demonstrating its effectiveness at both detecting real deepfakes and reducing false positives. The F1-score demonstrates model robustness by combining precision and recall through their harmonic mean which validates performance in practical applications. CNN and InceptionResNetV2 also delivered competitive results. However, the VGG19 model showed poor performance across all evaluation metrics when dealing with complex facial expressions and texture manipulations detection.

The study's findings demonstrate that depth-wise separable convolutions and optimized feature extraction methods in the Xception model establish its position as the most dependable deepfake detection approach.

1. Accuracy: Accuracy measures the overall correctness of the predictions= (TP + TN) / (TP + TN + FP + FN)

2. Precision: Precision measures the proportion of true positive predictions out of all positive predictions = TP / (TP + FP)

3. Recall: Recall measures the proportion of true positive predictions out of all actual positive instances =TP / (TP + FN)

4. F1-score: F1-score is the harmonic mean of precision and recall, providing a balance between the two metrics = 2 \* (Precision \* Recall) / (Precision + Recall)

5. Specificity: Specificity measures the proportion of true negative predictions out of all actual negative instances = TN / (TN + FP)

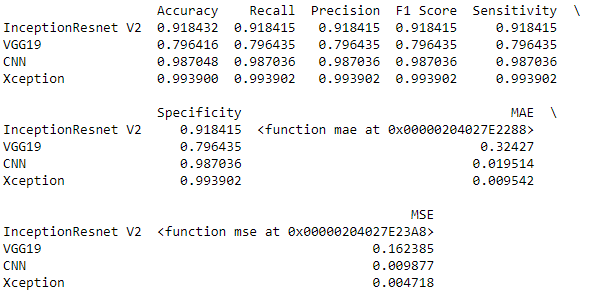


Figure 6: Comparison of results for various metrics

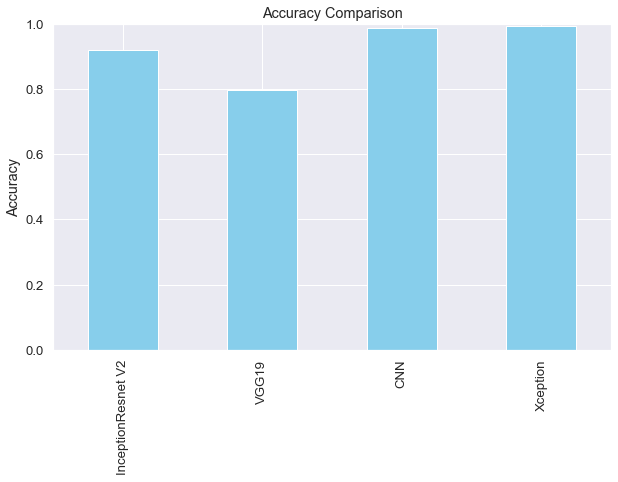


Figure 7: Accuracy comparison

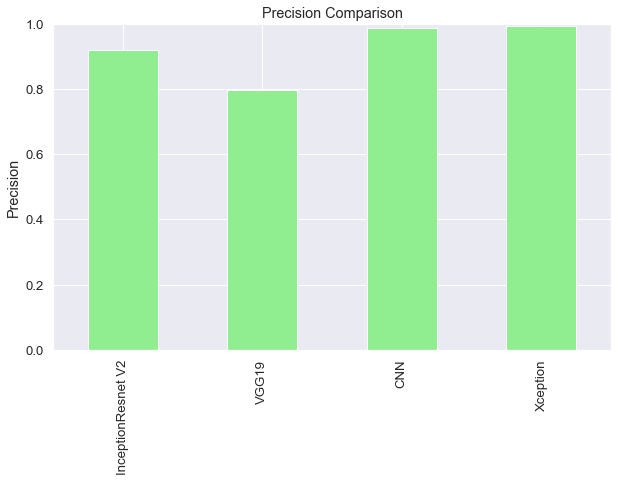


Figure 8: Precision Comparison

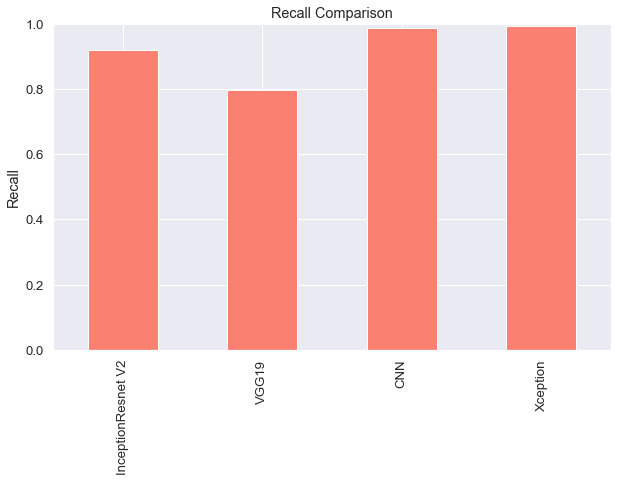


Figure 9: Recall Comparison

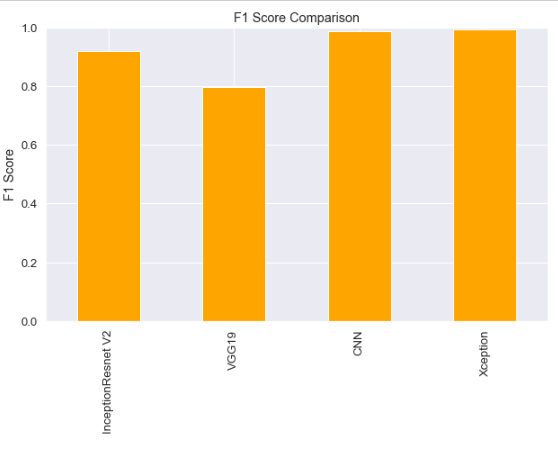


Figure 10: F1 Score comparison

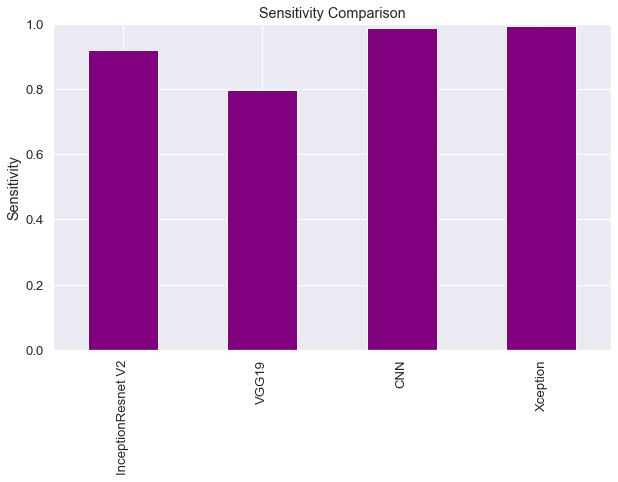


Figure 11: Sensitivity Comparison

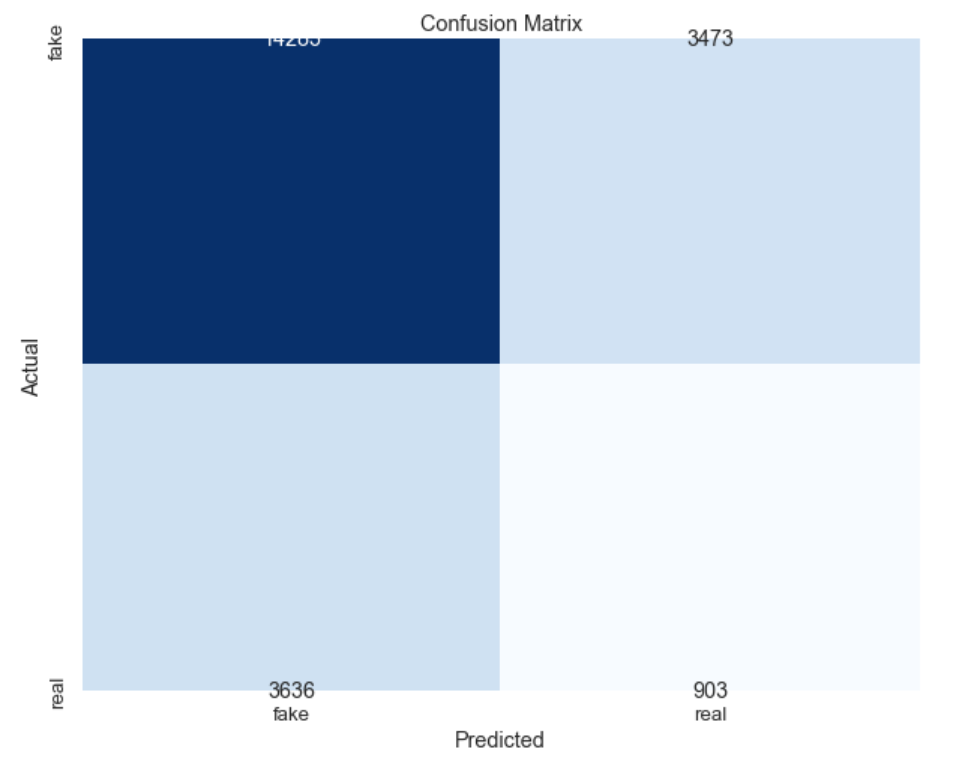


Figure 12: Confusion matrix \_Xception model

The Xception model successfully identified 14,205 fake images and 903 real images according to the confusion matrix. While this model reached 99.39% accuracy, it also incorrectly classified 3,473 fake images as real and 3,636 real images as fake. Although these errors make up a small portion of the correctly classified results they demonstrate why precision and recall optimization is essential when minor classification mistakes could lead to significant consequences.

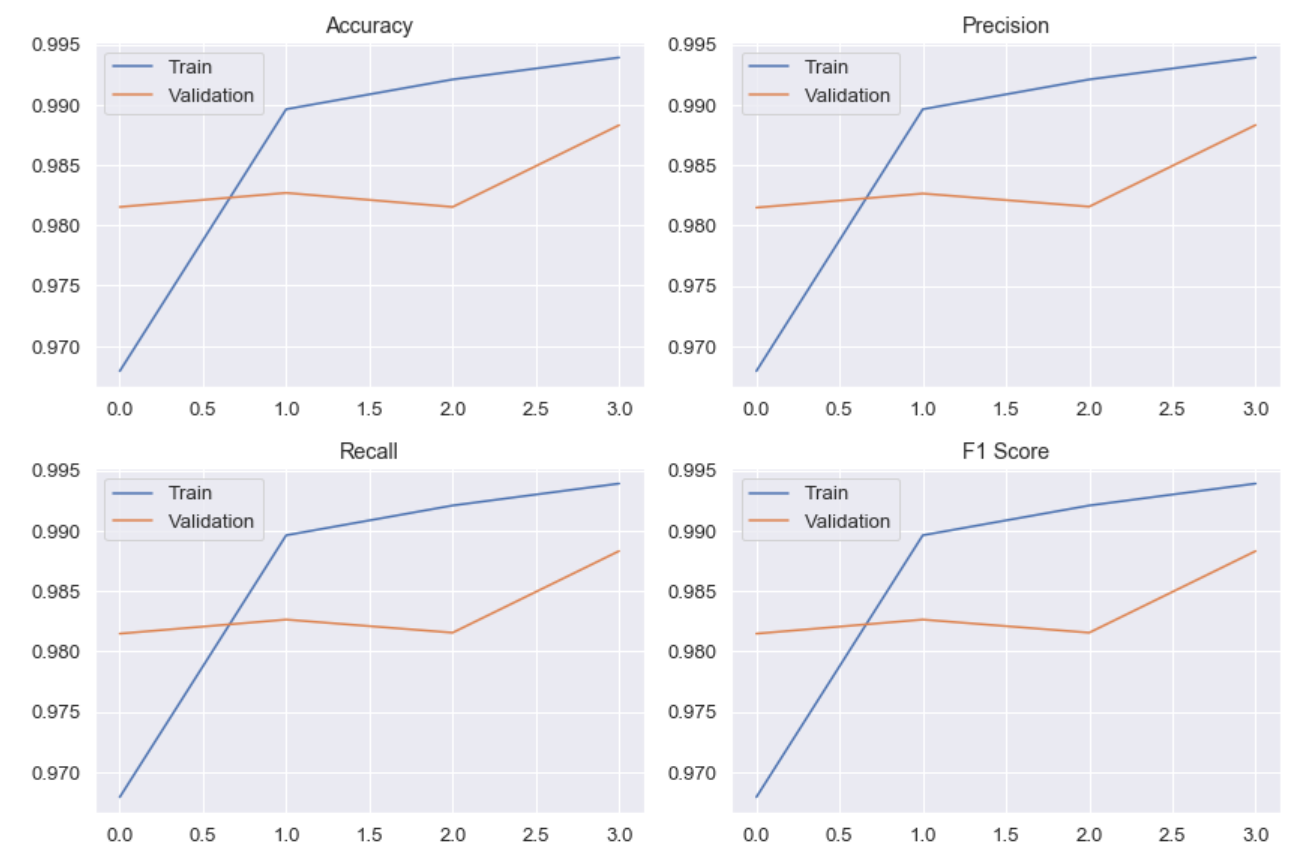


Figure 13: Training vs Validation Metrics for Xception Model

Training curves demonstrate high performance levels for accuracy, precision, recall, and F1 score which improve steadily throughout each epoch. Validation metrics show slightly lower values but maintain the same trend which demonstrates effective generalization without significant overfitting.

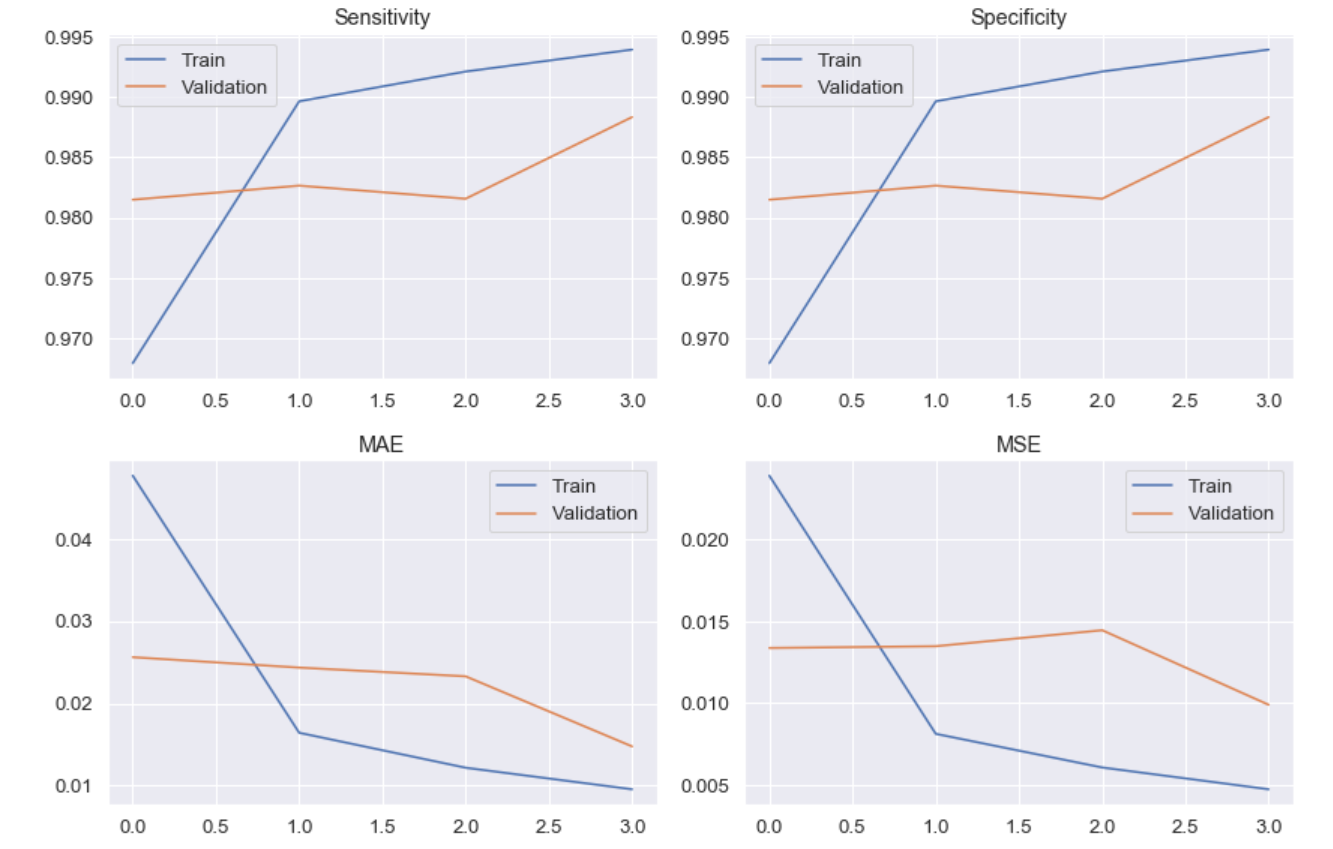


Figure 14: Sensitivity, Specificity, MAE, and MSE Curves for Xception Model

Across both training and validation phases the model maintains high sensitivity and specificity which shows its proficiency in accurately distinguishing between fake and real samples. The consistent reduction of Mean Absolute Error (MAE) and Mean Squared Error (MSE) throughout epochs shows strong convergence and better prediction accuracy.

# Conclusion and Future Work

In conclusion, this research conducted a comprehensive comparative analysis of four pre-trained convolutional neural network architectures—InceptionResNetV2, VGG19, CNN, and Xception—evaluating their varying performance. The incorporation of preprocessing techniques, feature extraction methodologies, and classification strategies contributes to the robustness and reliability of the detection models. Of the CNN architectures investigated Xception emerged as the optimal model, achieving highest accuracy on the test set. Overall, our study highlights the importance of selecting appropriate pre-trained models and underscores the potential of transfer learning for enhancing deep fake image detection capabilities.

With the growth of deepfake technology, subsequent studies should rather focus on hybrid models that combine the best features of these technologies while overcoming their respective weaknesses, including the computational complexity of QNNs and the scalability issues of blockchain systems. Much attention needs to be paid to the improvement of the speed, accuracy and scalability of the detection process. Moreover, to prevent unauthorized use of deepfake videos for political manipulation, there is a requirement to develop techniques that can not only detect such videos but also reverse them, rendering them infeasible for use in their original form. This could involve creating opposing narratives or highlighting factual errors present in the false video. One key area is real-time deepfake detection, which would allow platforms and security systems to identify manipulated content instantly before it spreads. Enhancing the efficiency of quantum-assisted AI models and vision transformers could further boost detection capabilities while reducing computational costs.

Furthermore, Interdisciplinary cooperation between computer vision experts, cybersecurity professionals and policymakers is crucial to develop effective protection measures for deepfake threats. The next generation of detection systems must combine high accuracy with interpretability by delivering clear explanations of their results to gain user and stakeholder trust. Combining audio, visual data with contextual metadata generates a comprehensive approach for synthetic content identification through multimodal analysis. The rapid advancement of AI technologies demands continuous development and thorough benchmarking with varied datasets to maintain detection methods that are both resilient and ethical while staying relevant against ever-evolving synthetic media strategies. Strong digital governance frameworks along with international regulatory cooperation and defined policy structures need to be established to promote ethical deployment of deepfake technologies while holding malicious users accountable. Legal frameworks need to advance alongside technological developments to prevent abuse while safeguarding personal and social integrity.

##### References

1. Goodfellow, Ian J., Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. "Generative adversarial nets." Advances in neural information processing systems 27 (2014).J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.
2. Ratliff, Lillian J., Samuel A. Burden, and S. Shankar Sastry. "Characterization and computation of local Nash equilibria in continuous games." In 2013 51st Annual Allerton Conference on Communication, Control, and Computing (Allerton), pp. 917-924. IEEE, 2013K. Elissa, “Title of paper if known,” unpublished.
3. Li, Yuezun, Ming-Ching Chang, and Siwei Lyu. "In ictu oculi: Exposing ai created fake videos by detecting eye blinking." In *2018 IEEE International workshop on information forensics and security (WIFS)*, pp. 1-7. Ieee, 2018.
4. Yang, Xin, Yuezun Li, and Siwei Lyu. "Exposing deep fakes using inconsistent head poses." In *ICASSP 2019-2019 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pp. 8261-8265. IEEE, 2019.
5. Matern, Falko, Christian Riess, and Marc Stamminger. "Exploiting visual artifacts to expose deepfakes and face manipulations." In *2019 IEEE Winter Applications of Computer Vision Workshops (WACVW)*, pp. 83-92. IEEE, 2019.
6. Afchar, Darius, Vincent Nozick, Junichi Yamagishi, and Isao Echizen. "Mesonet: a compact facial video forgery detection network." In *2018 IEEE international workshop on information forensics and security (WIFS)*, pp. 1-7. IEEE, 2018.
7. Nguyen, Huy H., Junichi Yamagishi, and Isao Echizen. "Capsule-forensics: Using capsule networks to detect forged images and videos." In *ICASSP 2019-2019 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pp. 2307-2311. IEEE, 2019.
8. Sandler, Mark, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. "Mobilenetv2: Inverted residuals and linear bottlenecks." In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4510-4520. 2018.
9. Szegedy, Christian, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. "Rethinking the inception architecture for computer vision." In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2818-2826. 2016.
10. Huang, Gao, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q. Weinberger. "Densely connected convolutional networks." In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4700-4708. 2017.
11. Tan, Mingxing, and Quoc Le. "Efficientnet: Rethinking model scaling for convolutional neural networks." In *International conference on machine learning*, pp. 6105-6114. PMLR, 2019.
12. Rana, Md Shohel, Mohammad Nur Nobi, Beddhu Murali, and Andrew H. Sung. "Deepfake detection: A systematic literature review." *IEEE access* 10 (2022): 25494-25513.
13. Güera, David, and Edward J. Delp. "Deepfake video detection using recurrent neural networks." In *2018 15th IEEE international conference on advanced video and signal based surveillance (AVSS)*, pp. 1-6. IEEE, 2018.
14. Heidari, Arash, Nima Jafari Navimipour, Hasan Dag, Samira Talebi, and Mehmet Unal. "A novel blockchain-based deepfake detection method using federated and deep learning models." Cognitive Computation (2024): 1-19.
15. Katı, Bekir Eray, Ecir Uğur Küçüksille, and Güncel Sarıman. "Enhancing Deepfake Detection Through Quantum Transfer Learning and Class-Attention Vision Transformer Architecture." *Applied Sciences* 15, no. 2 (2025): 525.
16. Verma, Akshay, Dipesh Gupta, and Manish Kumar Srivastava. "Deepfake detection using inception-ResnetV2." In 2021 First International Conference on Advances in Computing and Future Communication Technologies (ICACFCT), pp. 39-41. IEEE, 2021.
17. Hayathunnisa, V., P. Kuppusamy, and A. Manimaran. "Art of Detection: Custom CNN and VGG19 for Accurate Real Vs Fake Image Identification." In 2023 6th International Conference on Recent Trends in Advance Computing (ICRTAC), pp. 306-312. IEEE, 2023.
18. Lei, Xinyu, Hongguang Pan, and Xiangdong Huang. "A dilated CNN model for image classification." *Ieee access* 7 (2019): 124087-124095.
19. Joshi, Paritosh, and V. Nivethitha. "Deep Fake Image Detection using Xception Architecture." In 2024 5th International Conference on Recent Trends in Computer Science and Technology (ICRTCST), pp. 533-537. IEEE, 2024.
20. Dolhansky, Brian, Joanna Bitton, Ben Pflaum, Jikuo Lu, Russ Howes, Menglin Wang, and Cristian Canton Ferrer. "The deepfake detection challenge (dfdc) dataset." arXiv preprint arXiv:2006.07397 (2020).
21. Ravale, Ujwala, Riya Ramesh Tattu, Ashish Baban Bhoir, and Sneha Bhaskar Mahajan. "Deepfake Detection using InceptionResNetV2 Model." In 2024 IEEE 3rd World Conference on Applied Intelligence and Computing (AIC), pp. 873-879. IEEE, 2024.
22. Chollet, François. "Xception: Deep learning with depthwise separable convolutions." In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 1251-1258. 2017.