Grenze International Journal of Engineering and Technology, June Issue

Deepfake Detection using EfficientNET-B0 and Transfer Learning

Jane Rubel Angelina Jeyaraj[1], E. Sai Pradeep[2], K. Kowshik[3], K. Guru Vardhan Reddy[4] and D. Jithendra Reddy[5]
[1-5]Kalasalingam Academy of Researach and Education, Computer Science and Engineering, Krishnankoil, Tamilnadu, India
Email: j.janerubelangelina@klu.ac.in, saipradeep536@gmail.com, kowshik.koka@gmail.com, guruvardhanreddy127@gmail.com, jithendrareddydevireddy@gmail.com

***Abstract*— Deepfake technology has advanced significantly over the last few years, enabling the development of highly realistic fake facial images, which are a threat to the security of digital information, the ability to manage the spread of misinformation, and identity verification. Therefore, a deepfake detection model based on deep learning and the efficient model, EfficientNet-B0, with transfer learning has been proposed to address the challenge of deepfake images. This model was trained and tested using a public dataset containing 29,449 real and fake facial images. Additionally, the images were preprocessed based on resizing, normalization, and augmentation for the model's robustness. The proposed model was proven to be accurate, with a value of 92.44, F1-score of 92.00, and AUC-ROC of 97.83, indicating that the model was better than a few CNN-based models. Furthermore, a web interface was created through the use of Gradio, Flask, and FastAPI to carry out a real-time test of the trained model, showing that the proposed model was efficient in the detection of deepfake images through the efficient model, EfficientNet-B0, with transfer learning.**

***Keywords*— Deepfake detection, EfficientNet-B0, Transfer Learning, Image Classification, Convolutional Neural Networks, Binary Classification, SDG 4 (Quality Education), SDG 6 (Peace, Justice and Strong Institutions).**

I. INTRODUCTION

Recently, various advancements in the field of artificial intelligence and generative models have enabled the creation of realistic synthetic facial images called deepfakes with the help of Generative Adversarial Networks (GANS) and autoencoders. While these models are beneficial in the field of media and entertainment, there are various risks of misinformation, identity theft, cybercrime, and digital manipulation with the help of these models. As the models for creating deepfakes are becoming more and more sophisticated, it is becoming increasingly difficult to detect these synthetic images. The problem of deepfake detection can be considered a binary classification problem that distinguishes between real and deepfake facial images, in which traditional image processing techniques are not able to detect the minute changes in the image. Convolutional Neural Networks (CNN) have achieved remarkable results in this problem but are costly in terms of data and computational requirements. Transfer learning is an efficient workaround to achieve the task. EfficientNet is a model proposed by Google to scale the depth, width, and resolution of the network compoundly. This results in a model with high accuracy and less complexity.

Grenze ID: 01.GIJET.12.2.371_24

In the present research, the EfficientNet-B0 model is used with the help of transfer learning to design a computationally efficient deepfake detection system with the potential to attain high accuracy, generality, and real-time.

II. Related Work

The problem of deepfakes has been identified as an emerging area of research because of the swift misuse of visually produced content through AI. The first detection systems were based on hand-designed features such as irregularities in eye blinking, differences in colors, and texture differences; however, these systems were found to be ineffective against the latest deepfake technology that produces highly realistic content. The latest benchmarking tests such as Face Forensics++ proved the effectiveness of the CNN-based detection system but also proved the poor generalization ability of the system. The lightweight system such as Meso Net helped in the efficient detection of deepfakes but lacked the required accuracy. The other systems such as Face X-Ray focused on the detection of the presence of blending artifacts and were found to be ineffective in the absence of the same. Lip Forensics focused on the audio-visual consistency and were found to be ineffective in the absence of audio or improper cropping. Multi-attentional and multi-branch networks improved the detection process but introduced additional complexity. Frequency domain approaches uncovered GAN patterns in the frequency domain but were not robust in the long run as standalone solutions. Hybrid spatial-frequency methods were more robust but introduced additional complexity during training. Transformer networks provided excellent detection results but were computationally expensive and required large amounts of training data. Models with sequential and recurrent components were more stable at the video level but were sensitive to accurate face tracking. Reconstruction frameworks improved model interpretability but introduced additional latency during inference. In general, while many state of the art approaches improve detection accuracy, they may introduce complexity in terms of simplicity and deployability.

III. Proposed Methodology

The suggested deepfake detector system comes with a structured system from the collection of the data sets to the real time system using a state-of-the-art deep learning architecture that is optimized for high accuracy and generalization. The system starts with the creation of a curated list of real and deepfake images and videos, which are treated equally using resizing, normalization, and augmentation to increase robustness to variations in poses, lighting conditions, and compression artifacts. The images are then used as input to the EfficientNet-B0 model pre-trained on the ImageNet data set, and transfer learning is used to boost the rate of convergence to high-quality feature representations using its compound scaled stem, MB-Conv layers, and classification head. The training is done using the AdamW optimizer to keep the weights updated constantly and the Cosine Annealing Warm Restarts to vary the learning rate to avoid stagnation and boost the convergence rate. The loss and accuracy are evaluated on a per-epoch basis using a supervised learning loop and early stopping and checkpointing to save and select the best model. The trained model is then used to test unseen test data to validate the model and is saved as a Torch Script or ONNX model for fast inference on various platforms. The system designed can be used to develop a simple web interface using Gradio, Flask, FastAPI, and other libraries to upload an image and return a state of theart deepfake prediction of its authenticity and confidence levels.

A. Data Collection

The training and evaluation data set comprises 29,449 face images, which are divided into two classes: real and fake. The data set comprises images collected from an open-source introduction to a Roboflow deepfake data set, which comprises a variety of images synthesized through the help of various techniques of data synthesis. The data set will be split into training, validation, and testing sets to ensure that the evaluation process does not leak data from one phase to the next. This will ensure that the model generalizes well to unseen data. The data set comprises a wide range of data, including different lighting, pose, expressions, and quality of synthesis, which can be used to train a deepfake detector.

B. Data Preprocessing

All the images are resized to 224 x 224 pixels, which is the input size required by EfficientNet-B0. Following the resizing of the images, the pixel values of the images are normalized with respect to the mean and standard deviation of the ImageNet dataset to make sure the input distribution is exposed to the same environment as it was during the pretraining. To avoid overfitting and to make the network more robust, the techniques used to

augment the images to improve the training process include random flipping, rotation, cropping, and jittering. The images are then passed through the PyTorch Data-Loaders with the correct batching of the images.

C. Model Training

The proposed system will utilize a deep learning-based architecture for detecting fake images and real images. The proposed architecture will have four main phases: input data acquisition, data preprocessing, proposed architecture, proposed training strategy, followed by the prediction output. The proposed workflow will be explained in Figure 1.

Input Data

The system uses a dataset with 29,449 images for training and validation purposes. The dataset consists of two classes:

- Real Images
- Fake Images/ AI Generated Images

These images are fed into the proposed system. A large number of images in the dataset will improve the generalization ability of the model.

Data Preprocessing Pipeline

Before feeding the images into the network, a number of preprocessing techniques are used to enhance the quality of the input images and improve the robustness of the network.

- Image Resizing: The input images are resized to a size of 224 x 224 pixels. This process standardizes the dimensions of the images, which are compatible with the EfficientNet-B0 model. The resizing process reduces the complexity of the images, enabling the training of the convolutional neural network efficiently.
- Data Augmentation: To ensure the diversity of the training set and prevent overfitting, data augmentation operations are performed. These operations include: Horizontal Flipping, Rotation (±15 degrees),Color Jitter (brightness, contrast, and saturation changes). These operations enable the model to learn invariant features, hence enhancing its ability to generalize.
- Image Normalization: The images are then normalized using the ImageNet normalization parameters. Normalization of images is important to ensure the proper scaling of the pixel values for the pretrained networks.
- Weighted Random Sampling: To overcome the problem of class imbalance, a Weighted Random Sampler is used during the training process. This ensures a balance between the two classes, real and fake, by giving more probability to the underrepresented class.

Core Model Architecture

The core classification model architecture is based on EfficientNet-B0, which is a lightweight yet very effective CNN architecture. Efficient-Net uses compound scaling to achieve optimal depth, width, and resolution for the CNN, which results in high accuracy with fewer parameters compared to standard CNN architectures.

- Input Tensor: Each input image is represented as a tensor of size:(1, 3, 224, 224),1 represents the batch size,3 represents the number of color channels,224 * 224 represents the dimensions of each input image.
- MBConv Blocks: The Efficient-Net network is composed of Mobile Inverted Bottleneck Convolution (MBConv) blocks, each of which includes several important components:
  - Depthwise Convolution: In depth wise convolution, multiple convolutional operations are performed independently across each channel, greatly reducing the complexity of the operation without compromising the ability to extract features.
  - Squeeze-and-Excitation (SE) Module: The squeeze and excitation mechanism helps the network selectively enhance important features and discard unwanted information.
  - Batch Normalization: Batch normalization helps stabilize the training process by normalizing the output of each layer, thus speeding up convergence and reducing internal covariate shift. The EfficientNet-B0 model used in the system contains approximately 4.01 million trainable parameters.
- Global Average Pooling: After the convolutional layers used for feature extraction, the Global Average Pooling (GAP) layer is used. In this layer, the spatial dimensions of the feature maps are reduced based on the average value of the feature maps. The GAP layer reduces the parameters of the network and prevents overfitting.
- Fully Connected Layer: After the pooled feature vector is obtained, it is passed through the Fully Connected Layer. In this layer, the features extracted from the convolutional layers are mapped to the classes. The network is designed as a binary classifier with two classes: real and fake.

- Softmax Activation: The final layer is the Softmax activation function. In this function, the class probabilities are obtained. The class probabilities are expressed as: P(Real),P(Fake)The class with the highest probability is chosen as the final class.
- Transfer Learning: Transfer learning is used in the proposed system. In this method, the weights of the EfficientNet-B0 network are pre-trained on the ImageNet dataset. Only the final layers of the network are finetuned. Transfer learning reduces the training time and increases the accuracy of the network, especially when the dataset is moderately sized.

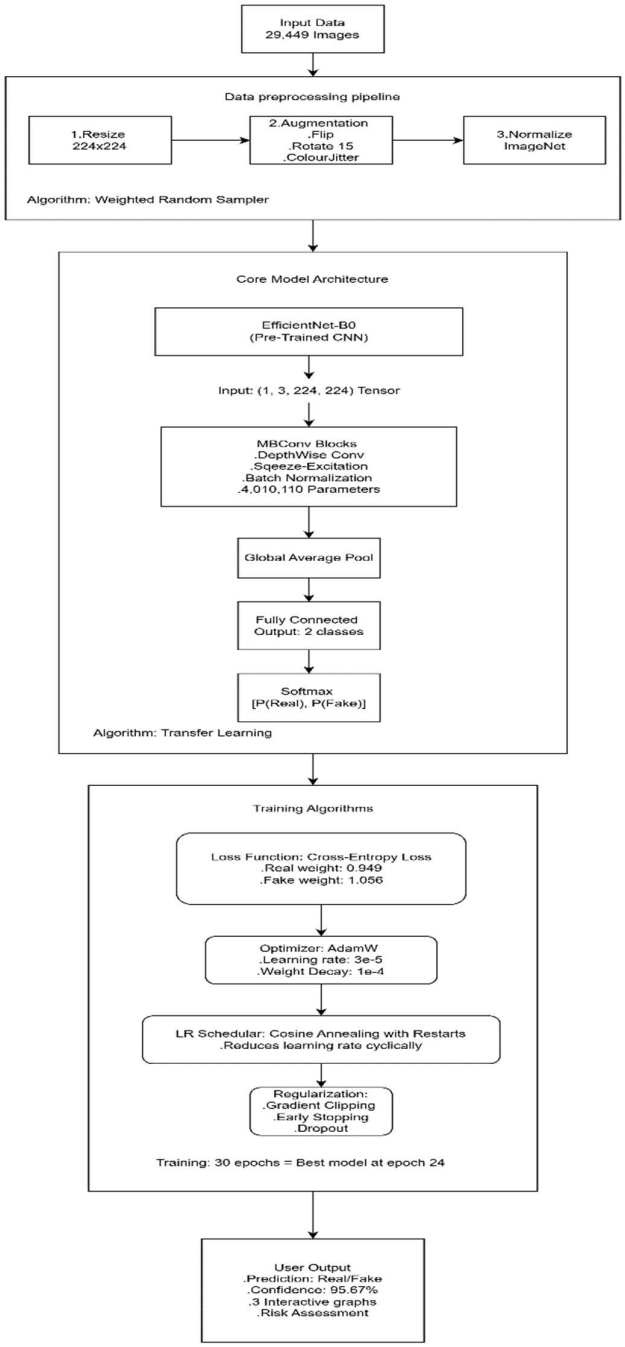

Fig. 1: Architecture Daigram of Proposed Method

Training Strategy

There are various optimization approaches that are utilized for effective training and improved performance.

- Loss Function: The model utilizes the Cross-Entropy Loss function for training, which is best for classification-based approaches. To avoid class imbalance, class weights are provided as follows: Real class weight:0.949,Fake class weight:1.056.These weights are utilized for balancing learning for both classes.
- Optimizer: The parameters are optimized by the AdamW optimizer. AdamW incorporates learning rate optimization with weight decay for improved generalization and avoidance of overfitting. The parameters are provided as follows:
  - Learning rate: 3x10 -5
  - Weight decay: 1x10-4
  - AdamW improves generalization and prevents overfitting compared to traditional optimizers.
- Learning Rate Scheduler: A Cosine Annealing Learning Rate Scheduler with Restarts has been used to optimize the learning rate. It reduces the learning rate following the cosine shape.
- Regularization Techniques: Several regularization techniques are used to optimize the network to prevent overfitting. The techniques used are:
  - Gradient Clipping
  - Early Stopping
  - Dropout
  - These techniques are used to prevent excessive update in the weights.

Model Training

The model is trained for 30 epochs. During the experiment, the best model was recorded at epoch 24, as it performed optimally. The model is then used for inference.

Output Layer (User Output)

The final system output includes the following:

- Prediction: Real or Fake Image
- Confidence Score: e.g., 95.67%
- Interactive Performance Graphs
- Risk Assessment for Authenticity Detection

IV. ALGORITHMS USED

- Data Preprocessing Augmentation: The data preprocessing augmentation is done to prepare the input data before it is fed into the EfficientNet-B0 model. In this case, all the images are resized to a size of 224x224 pixels to match the input size required by the model. In addition, the pixel values are normalized to the mean and standard deviation values of the ImageNet dataset to match the parameters required by the model. In order to add more diversity to the dataset, various forms of data augmentation are done on the images. These include random flip, rotation, color jitter, and random resized crops. These transformations mimic the real-world scenarios experienced by the objects in the images. This will enable the model to learn more generalized features from the input data. Hence, the process will enhance the performance of the model, reducing the chances of overfitting.
- Transfer learning-Based Model training Algorithm: This algorithm is based on the EfficientNet-B0 architecture, which has been modified to suit the binary classification problem of distinguishing real from fake images. In this case, the EfficientNet-B0 architecture is loaded, and the last classification layer is modified to a fully connected layer with two output neurons. In the training phase, the parameters of the network are optimized through the Adam optimizer with a cross-entropy loss function. In addition, the network is trained for a specified period, and the performance is monitored on the validation data set. To enhance the performance of the network and prevent overfitting, the network is subjected to early stopping, checkpointing, and scheduling.
- Deepfake Classification and Prediction Algorithm: The above algorithm describes the process of how the untold image is classified by the model that has been created. After the image is uploaded through the web interface, it is first passed through the same preprocessing steps that were used during the training process, so that the quality of the image is high. Then, the processed image is passed through the EfficientNet-B0 model, where it is represented by a feature vector, and two probability scores are generated, one for "Real" class and one for "Fake" class, both being probabilities (as calculated by the softmax activation function).

The class with the highest probability is selected as the last prediction made by the model. Apart from that, the model also produces a confidence score, which describes the reliability of the prediction made by the model. This algorithm is the backbone of the real-time deepfake detection system, allowing the algorithm to be used in the real world as a reliable screening tool.

V. Experimental Results

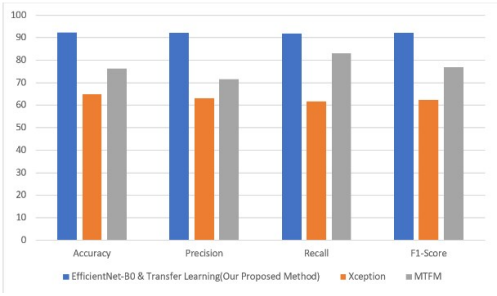

Fig. 2: Performance Comparison after Proposed Approach with Existing Deepfake Detection Techniques

The bar graph is used to show the performance of the deepfake detection techniques using percentage values. The EfficientNetB0 and Transfer Learning model is performing well with 92.4% accuracy, 92.1% precision, 91.9% recall, and 92.0% F1- score values. The percentage values are quite high for the model. For the Xception model, the percentage values are less compared to the other models. The percentage values for the Xception model are 64.8% accuracy, 63.1% precision, 61.8% recall, and 62.4% F1-score values. The MTFM model is performing moderately with 76.3% accuracy, 71.6% precision, 83.0.

The above confusion matrix in the image above indicates the performance of the deepfake detection model on the test dataset. Of the real images, 2163 were detected correctly as real, while 164 were incorrectly identified as fake. Additionally, 1921 fake images were detected correctly, while 170 were incorrectly identified as real. The model performs well in the detection of real and fake images with an accuracy of 93.0% and 91.9%, respectively.

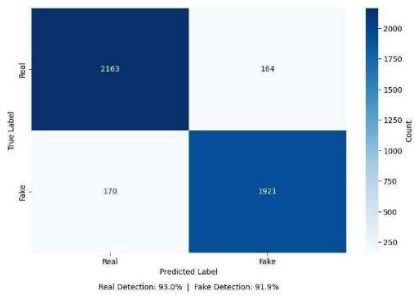

Fig. 4: Confusion Matrix

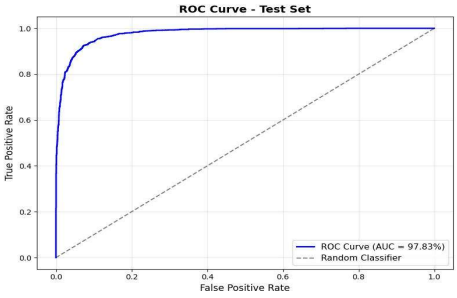

Fig. 5. ROC Curve

The image above shows the ROC curve of the deepfake detection model on the test dataset. The Curve is always close to the top-left corner, which means that the model has a high true positive rate and a low false positive rate. The model has an Area Under the Curve (AUC) of 97.83%, which is an indication of excellent discriminative ability. The model performs significantly better than the random classifier baseline.

Model	Accuracy	Precision	Recall	F1-Score	AUC
EfficientNet-B0 and Transfer Learning	92.44%	92.13%	91.87%	92.00%	97.83%
Xception	64.80%	63.07%	61.84%	62.45%	70.33%
MTFM	76.35%	71.58%	82.97%	76.65%	85.58%

Fig. 6: Performace Summary Table of all Method

The performance of the three deepfake detectors, EfficientNet-B0 and Transfer Learning, Xception, and MTFM, is compared using the standard evaluation metrics of accuracy, precision, recall, F1- score, and AUC. The results show that the proposed EfficientNetB0 model performs better than the other two models in all evaluation metrics. The performance of the Xception model is relatively poor, with lower precision and recall values, which indicates a higher misclassification rate. The MTFM model performs better than Xception but is still inferior to the proposed EfficientNet-B0 model. The AUC comparison further demonstrates the superior discriminative power of the proposed model.

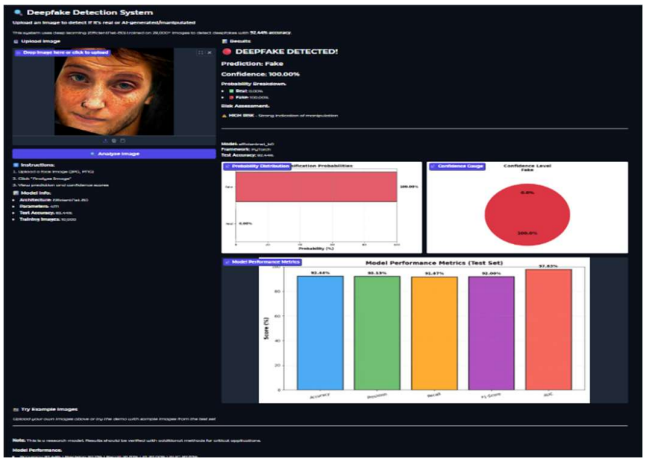

Fig. 7: System Interface and Result

The image shows a web-based deepfake detection tool designed for image-level analysis using a deep learning model. The user can upload a facial image, which is then analyzed using a trained EfficientNet-B0 model with around 4 million parameters. In the result shown in the image, the tool is able to detect that the uploaded image is fake with a test accuracy of 92.44%, which is obtained after training the model on a dataset of 10,000 images. The confidence level analysis shows that there is a 0.01% chance that the image is real and a 99.99% chance that the image is fake, which clearly indicates a confident prediction. Based on the confidence level of the prediction, the tool is able to provide a high-risk level, which clearly indicates that the image is manipulated.

VI. Conclusion

This research paper presents a deepfake detection model using EfficientNet-B0 and transfer learning. The proposed system effectively differentiates between real and fake facial images and demonstrates strong robustness and accuracy. The achieved performance indicates promising prospects for practical implementation. Future studies may extend the system to support video deepfake detection, multimodal training, or transformer-based architectures to further improve perfor- mance. In addition to its strong empirical results, the architecture is computationally efficient and stable, making it suitable for real time applications such as content moderation systems, identity verification systems, and digital forensic tools. The compound scaling properties of EfficientNet-B0 balance ac- curacy and computational cost, enabling deployment even on resource constrained devices. Furthermore, data augmentation, early stopping, and optimized learning strategies enhance robustness across different testing conditions. These advantages make the proposed system a competitive alternative to more complex architectures, such as Xception and transformer-based models, while achieving superior accuracy and generalization.

Future Work

Future research may involve extending the proposed EfficientNetB0-based system to level deepfake detection by incorporating temporal modeling across consecutive frames. The framework can be enhanced through multi-modal learning by combining facial images with audio in- puts, lip movement consistency, and physiological signals. Transformer-based models may be explored to capture long- range spatial dependencies and semantic inconsistencies, bustness against unseen manipulation techniques can be im- proved using self-supervised and contrastive learning ap- proaches. Additionally, spatial–frequency feature fusion may enhance detection in

compressed and low-quality media. Expanding the dataset with more diverse real-world deepfakes will further strengthen generalization. Optimizing the system for edge and mobile platforms is another promising direction for low resource environments.

ACKNOWLEDGEMENT

The authors were acknowledged and overwhelming sup- ported by our respected professor and staff from Department of Computer Science and Engineering, Faculty of Engineering, Kalasalingam Academy of Research and Education.

REFERENCES

[1] A. Ro¨ssler, D. Cozzolino, L. Verdoliva, C. Riess, J. Thies, and M. Nießner, "FaceForensics++: Learning to Detect Manipulated Facial Images," in Proc. IEEE Int. Conf. Computer Vision (ICCV), 2019.
[2] D. Afchar, V. Nozick, J. Yamagishi, and I. Echizen, "MesoNet: A Compact Facial Video Forgery Detection Network," arXiv preprint arXiv:1809.00888, 2018.
[3] Y. Li, J. Bao, Y. Wu, et al., "Face X-Ray for More General Face Forgery Detection," in Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), 2020.
[4] A. Haliassos, H.-Y. Hsu, M. Marrese-Taylon, et al., "LipForensics: A Generalizable and Robust Approach to Face Forgery Detection," in Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), 2021.
[5] H. Zhao, W. Zhou, D. Chen, T. Wei, W. Zhang, and N. Yu, "Multi- Attentional Deepfake Detection," in Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), 2021.
[6] J. Frank and O. Ronneberger, "Leveraging Frequency Analysis for Deep Fake Image Recognition," in Proc. ICLR Workshop, 2020.
[7] I. Masi, A. Killekar, R. Mascarenhas, et al., "Two-Branch Recurrent Network for Isolating Deepfakes in Videos," in Proc. European Conf. Computer Vision (ECCV), 2020.
[8] Y. Jeong, D. Kim, Y. Ro, and J. Choi, "FrePGAN: Robust Deepfake Detection Using Frequency-level Perturbations," 2022.
[9] Y. Xu et al., "TALL: Transformer-Based Local-to-Long Deepfake De- tection," arXiv preprint, 2023.
[10] B. Wang et al., "Spatial-Frequency Feature Fusion Based Deepfake Detection," 2024.
[11] Anonymous, "Robust Sequential DeepFake Detection," arXiv preprint, 2023.
[12] B. Yan et al., "Deepfake Detection Based on Joint Reconstruction and Classification," 2024.
[13] P. Chen et al., "Detecting Compressed Deepfake Images Using Two- Branch Networks," IEEE Trans. Circuits Syst. Video Technol., 2022.
[14] "Multi-Branch Deepfake Detection Algorithm Based on Fine-Grained Features," TechScience, 2023.
[15] "Frequency-Aware Deepfake Detection: Frequency-Domain Masking and Spatial Interaction," AAAI / MDPI, 2024.
[16] D. Gu¨era and E. J. Delp, "Deepfake Video Detection Using Recurrent Neural Networks," in Proc. IEEE AVSS, 2018, pp. 1–6.
[17] D. Afchar, V. Nozick, J. Yamagishi, and I. Echizen, "Mesonet: A Compact Facial Video Forgery Detection Network," in Proc. IEEE WIFS, 2018, pp. 1–7.
[18] H. Dang, F. Liu, J. Stehouwer, X. Liu, and A. K. Jain, "On the Detection of Digital Face Manipulation," in Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), 2020, pp. 5781–5790.
[19] S. Tariq, S. Lee, H. Zhang, J. Ho, K. Kim, and S. S. Woo, "Detecting Both Deepfakes and Face Swaps Using CNN-LSTM Architecture," IEEE Access, vol. 9, pp. 21146–21158, 2021.
[20] B. Zi, M. Chang, J. Chen, X. Ma, and H. Xue, "WildDeepfake: A Challenging Real-World Dataset for Deepfake Detection," in Proc. ACM Multimedia (MM), 2020, pp. 2382–2390.