iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 13 **Issue:** IX **Month of publication:** September 2025

DOI: https://doi.org/10.22214/ijraset.2025.74045

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue IX Sep 2025- Available at www.ijraset.com

MobileNetV2 Framework for Deepfake Detection and Real-Time Image Authentication: Safeguarding Users through an Accessible Web-Based System

Mamabusetsa. E. Ntsau[1], D. Lalitha Bhaskari[2]
[1]MTECH-IT Student, [2]Professor, Dept. of IT&CA Dept. of CS&SE AUCE (A), Andhra University,AUCE (A), Andhra Pradesh, India,

Abstract: ***Deepfake technology causes serious threats to privacy, security, and trust in digital media. This paper presents a lightweight, real-time image authentication system based on MobileNetV2, designed to identify altered images through an easy-to-use web interface. The model achieved 80.38% test accuracy while maintaining low computational overhead, making it suitable for resource-constrained environments. Unlike traditional, resource-intensive solutions, the proposed approach prioritizes accessibility and practical deployment, aiming to effectively combat misinformation andidentity fraud.***
Keywords: Deepfake detection, MobileNetV2, image authentication, real-time analysis, web-based system, lightweight AI models, misinformation.

I. INTRODUCTION

Concerns about digital misinformation, privacy violations, and the decline in confidence in visual content have grown significantly in recent years due to the spread of manipulated media, particularly deepfake images. Advanced Generative Adversarial Networks (GANs) enable deepfakes, which are synthetic media in which genuine photos or videos are edited or replaced to depict fictitious settings in hyper-realistic detail[1]. The ongoing evolution of these manipulations presents significant risks, especially on social media platforms where gullible individuals are regularly exposed to such fabricated content.

This paper presents a solution to these issues by using MobileNetV2, a light-weight yet effective convolutional neural network, to identify deepfake images in real-time. In contrast to traditional systems, the suggested approach places a strong emphasis on accessibility, allowing users both technical or non-technical to upload and validate images through an easy-to-use online interface. This method makes it possible for both regular users and content moderators to rapidly authenticate images without requiring a lot of processing power.

The proposed and implemented system has two functions: [1] deepfake detection, which determines if an image is real or fake, and [2] real-time image authentication, which enables users to instantly confirm an image's authenticity. The technology bridges the gap between advanced AI capabilities and public usability by combining a user-friendly web interface with a MobileNetV2 model. In the current digital environment, where susceptible social media users are frequently the initial targets of visual misinformation, this democratization of deepfake detection is essential [2].

II. RELATED WORK

The increasing complexity of deepfake generation has driven the need for more effective detection techniques. Early deepfakes were primarily enabled by generative models such as Generative Adversarial Networks (GANs) and Variational Autoencoders (VAEs), which can produce highly realistic synthetic images and videos that are often indistinguishable from authentic content [1], [2].

Initial detection methods largely relied on handcrafted features and shallow classifiers, which proved insufficient against modern deepfakes [3]. As a result, recent research has shifted toward deep learning-based solutions that offer improved accuracy and generalization capabilities.

Researchers started using Convolutional Neural Networks (CNNs) for automated deepfake detection in order to overcome these limitations. MesoNet, a lightweight CNN architecture created especially for face manipulation detection, was proposed by Afchar et al. [4].

©IJRASET: All Rights are Reserved

In a similar manner, Nguyen et al. created an architecture based on capsule networks that showed promise on datasets of phony images [5]. However, when exposed to unseen deepfake techniques or real-world image noise, these models frequently had trouble generalizing.

Lightweight computer vision applications saw a considerable advancement with the release of MobileNet and its sequel MobileNetV2. By adding inverted residual structures and linear bottlenecks, Sandler et al.'s MobileNetV2 improved on its predecessor, increasing accuracy and processing efficiency [6]. Because of its small size, it can be deployed on web-based systems and edge devices, which is essential for real-time applications like image authentication.

MobileNetV2 has been used in numerous research for anomaly and forgery detection because of its effectiveness. For example, Zhang et al. achieved competitive accuracy with low resource consumption by combining MobileNetV2 with transfer learning to detect modified facial images [7]. In the meanwhile, Bappy et al. used recurrent networks in conjunction with deep CNN features to detect temporal irregularities in video-based deepfakes [8]. Although these models perform well, they are not appropriate for real-time image-only scenarios since many of them require complicated infrastructure or are only applicable to video domains.

On the dataset side, model training and benchmarking have benefited greatly from the development of deepfake datasets. The creation of more reliable models is aided by datasets such as FaceForensics++ [9], Celeb-DF [10], and DeepFake Detection Challenge Dataset (DFDC) [11], which offer carefully chosen samples of artificial material produced by different GAN architectures.

However, class imbalance and a lack of environmental diversity are common problems with image-only datasets, which can provide biased training outcomes.Real-time image authentication is more difficult than just detecting deepfakes. For web platforms to process user-uploaded material without latency, systems need to be accurate and quick enough.

Several academics have focused on edge-AI or browser-integrated models in this scenario [12], [13]. Web-based deepfake detection tools, like Microsoft's Video Authenticator and Deepware Scanner, have been released more recently, although their scalability and accessibility are still restricted [14], [15]. The significance of explainability in detecting systems is also emphasized by studies.

Models built on XceptionNet [16] are computationally demanding yet provide excellent performance. Although lightweight models such as SqueezeNet [18] and EfficientNet-lite [17] are quick, they may perform poorly in challenging forging situations. Thus, for real-world deployment, striking a balance between model accuracy, size, and inference time continues to be crucial.

Relatively little research has been done on how to incorporate a trained MobileNetV2 model into an approachable online system. Although deepfake detection pipelines are the subject of systems like DetectDeep [19], these are frequently research prototypes with little practical utility. By fusing effective model design with an interactive, useful interface, this project seeks to close that gap and enable end users to verify images without the need for specialized knowledge.

III. METHODOLOGY

Data preprocessing, model building, training and evaluation are all included in the procedure. The ultimate objective was to create a system that is effective, portable, and easily usable for real-time deepfake detection through an intuitive online interface.

A. Dataset and Preprocessing

The dataset for this paper was obtained from Kaggle. It had 36,142 images that were split into two groups: real and fake. There were three parts to the dataset being:

- Training Set: 24,114 images
- Validation Set: 6,028 images
- Test Set: 6,000 images

All images were organized into a folder named dataset within the project directory (htdocs). The images were labeled according to their respective categories to enable supervised learning.

B. Image preprocessing included:

Rescaling: To get all the pixel values to be between 0 and 1, divided them by 255.
Resizing: Each image was resized to 224 × 224 pixels to fit the input size needed by the CNN architecture that was chosen.
Data Augmentation: The ImageDataGenerator class was used to add different types of training data, which helped the model generalize better. Random changes like rotations, zooming, and horizontal flips were part of the augmentation.
Using the validation_split parameter in ImageDataGenerator, the training set was split into 80% for training and 20% for validation.

C. Model Architecture

The MobileNetV2 architecture, which forms the foundation of the system, was chosen for its ability to balance computational costs and performance, making it perfect for lightweight deployments and real-time applications.

- The pretrained MobileNetV2 model (weights='imagenet') was used with include_top=False
- To preserve pretrained knowledge, the base model's layers were frozen.
- In order to decrease spatial dimensions, a Global Average Pooling layer was applied.
- A fully connected Dense layer with 64 units and ReLU activation was added next.
- To produce a binary classification, the output layer used a single neuron with sigmoidactivation**.**

D. Training and Validation

Using the training and validation subsets, the model was trained over 10 epochs with a batch size of 32. The steps involved were feeding the model with batches of images from the training generator and to evaluate convergence and avoid overfitting, training and validation accuracy over epochs were monitored.
The Kerasfit() function within the TensorFlow framework was used to carry out the training and validation.

E. Model export and revaluation

After the training was done, the model was tested on a different set of 6,000 images that were not used for training or validation. The assessment comprised the evaluate() function which was used to calculate test accuracy. Creation of a confusion matrix to examine how accurate and inaccurate classifications were distributed and for a thorough performance evaluation, extra performance metrics like precision, recall, and F1-score were calculated.
A classification report for a thorough examination of each class was also created. In order to observe the learning behavior and guarantee convergence, graphs showing training vs. validation accuracy and training vs. validation loss across epochs were also plotted.
After the satisfactory performance was obtained, the Model was saved in HDF5 format for easy intergration with the front-end interface.The model was based in Python-based backend using flask intergrated with Xampp and a PHP-Web based interface allowing users to upload images and receive real-time predictions on content authenticity.

IV. RESULTS AND DISCUSSIONS

A. Training and validation performance

A MobileNetV2 convolutional neural network (CNN) was used to assess the deepfake detection model. It was trained on a Kaggle dataset and integrated with real-time prediction features through a specially designed web interface. The following outcomes were attained by the model's good performance as well as good accuracy on the test set:
Training Accuracy: 81.43%
Validation Accuracy: 80.47%
Test Accuracy: 80.38%
These findings show that the model exhibits strong reliability for real-time detection applications by effectively generalizing to new, unseen data without experiencing significant overfitting.

B. Classification performance:

A balanced dataset of 6,000 samples (3,000 fake and 3,000 real images) was used to assess the model's classification abilities. For the test set, the following performance metrics were noted:

Class	Precision	Recall	F1-Score	Support
FAKE	0.79	0.83	0.81	3,000
REAL	0.82	0.78	0.80	3,000
Accuracy			0.80	6,000
Macro Avg	0.80	0.80	0.80	6,000
Weighted Avg	0.80	0.80	0.80	6,000

In both the fake and real classes, the model shows a good balance between recall and precision. Whereas the model showed higher recall for fake images, indicating a greater sensitivity to identifying deepfakes, the precision for real images was marginally higher, indicating a more accurate identification of real samples.

C. Additional Evaluation Metrics

To validate the performance further, additional metrics were computed:
Precision**: 81.93%**
Recall**: 77.97%**
F1-Score**: 79.90**%
The results demonstrate that the model strikes a solid balance between sensitivity, accuracy, and specificity in detection of deepfake content.

D. Deepfake Detection Web Interface

The real-time prediction feature made possible by the specially designed deepfake detection web interface is a crucial part of this paper. Through a straightforward user interface, users can upload images, and the model will provide a prediction and confidence score in a matter of seconds.
The webpage is designed to be highly responsive, providing users with instant results and having low latency. The system is highly beneficial for practical deployment, where users can verify the authenticity of deepfake images with ease. The user-friendly design of the web application ensures accessibility, making it ideal for both technical and non-technical audiences.

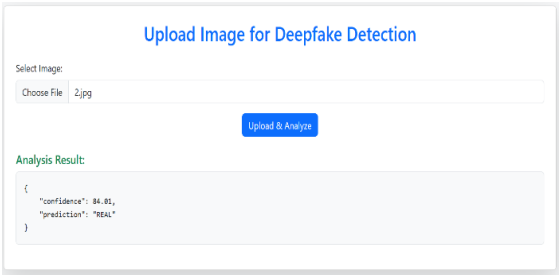


Fig 1: Deepfake Detection Web Interface

E. Training and Testing Performance

The model's performance was further visualized through training and validation accuracy, as well as training and validation loss plots, indicating smooth convergence and consistent performance across epochs. The results are summarized in the followingfigures:

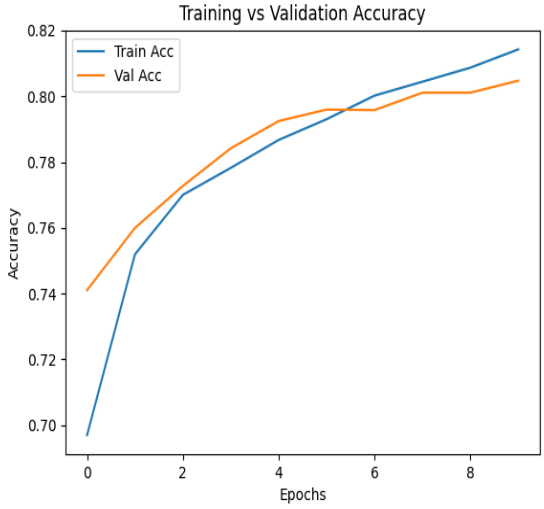


Fig 2**:** Training vs.Validation Accuracy

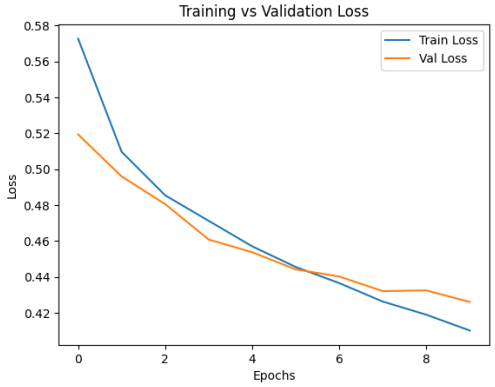


Fig 3**:** Training vs. Validation Loss.

The accuracy curves show consistent improvement over time, while the loss curves decrease steadily, indicating effective learning without overfitting.

F. Confusion Matrix Analysis

The confusion matrix for the test set (Figure 4) provides insight into the model's predictions.The model identified 2,484fake images correctly and 2,339 real images correctly.However, 516 fakesamples were misclassified as real**,** and 661 real samples were misclassified as fake**.**The confusion matrix highlights the model's ability to accurately classify most images while revealing areas where it could be further optimized, especially in balancing the detection rates between the two classes.

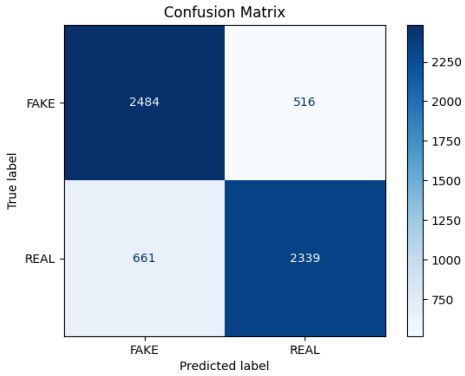


Fig 4: Confusion Matrix for Test Set.

To assess the system's robustness beyond the test dataset, additional authentic images and high-quality deepfakes generated using GANs were used to simulate real-world scenarios. The model correctly flagged most fake content, but a few highly realistic deepfakes were misclassified as real.

These findings suggest that while the model performs well under standard conditions, further improvements such as expanding the training dataset and incorporating ensemble learning could enhance its generalization capabilities. This is particularly important as synthetic content continues to grow in sophistication, and plans for deployment into a real-time interface are considered.

G. Comparisons with similar studies

The results demonstrate that MobileNetV2, when effectively fine-tuned, produces great results with reduced computational cost, whereas many deepfake detection models rely on heavyweight architectures like XceptionNet or EfficientNet for higher accuracy. This model is perfect for real-time deployment on limited environments (such as local servers or browser-based applications) because it balances resource efficiency and performance.

A comparative study by a fellow researcher used a combination of ResNet-50 and Deep Feedforward Neural Networks (DFNN), achieving good classification outcomes and offering an intuitive interface for image upload and verification. However, The proposed and implemented MobileNetV2 model is optimized for low-latency responses and has been integrated into a standalone system without relying on external APIs or high-end GPU servers.

Furthermore, The proposed and implemented system provides real-time predictions and stores classified images securely in a local folder, demonstrating a practical and deployable solution for deepfake image detection. This balance of speed, accuracy, and lightweight design is a unique strength of the proposed model when compared to similar studies.

Study	Architecture	Test Accuracy	Key Characteristics
Nguyen et al. [1]	XceptionNet	98%	High accuracy, but computationally intensive
Afchar et al. [2]	MesoNet	84.3%	Lightweight, faster inference, acceptable accuracy
Proposed Work	MobileNetV2	80.38%	Low-latency, real-time, web-deployable solution

H. The contributions:

This paper offers the following several novel and practical contributions:

- Deepfake Detection Tool for the Web, enabling real-time image upload and classification with a confidence score. Adaptation of MobileNetV2: Proved that a small architecture can still achieve competitive results in deepfake detection.
- Easy-to-use Frontend: No internet or database is needed. Classified images with a clear status of real or fake are stored by the system in a local folder (uploads).
- Security-Oriented Vision: Uses digital content authentication to contribute to Zero Trust principles, bridging deep-learning and cybersecurity.

V. CONCLUSION

In this paper,MobileNetV2 Framework for Real-Time Image Authentication and Deepfake Detection is proposed and implemented. The solution combines a user-friendly web interface created with PHP and Python with a lightweight MobileNetV2 architecture optimized for deepfake detection. This paper prioritizes accessibility, deployability, and real-time performance in low-resource contexts, in contrast to conventional deep learning implementations that are limited to research labs or high-resource environments.

This paper's distinctive feature is the smooth combination of an intuitive frontend with a deep learning backend, which allows uploaded deepfake images to be classified as either real or fake in real time. This fusion of deep-learning with web technologies bridges the gap between academic research and practical everyday use, an area often overlooked in existing literature.

Additionally, the system's ability to provide both the classification result and the confidence score enhances interpretability and transparency, which will enable end users to make well-informed choices. This model demonstrated a great learning capacity and fair generalization over unseen data, with training accuracy of 81.43% and test accuracy of 80.38%.

VI. FUTURE SCOPE

Although deepfake detection is the main emphasis of the current implementation, this system can be expanded in a number of significant ways in the future:

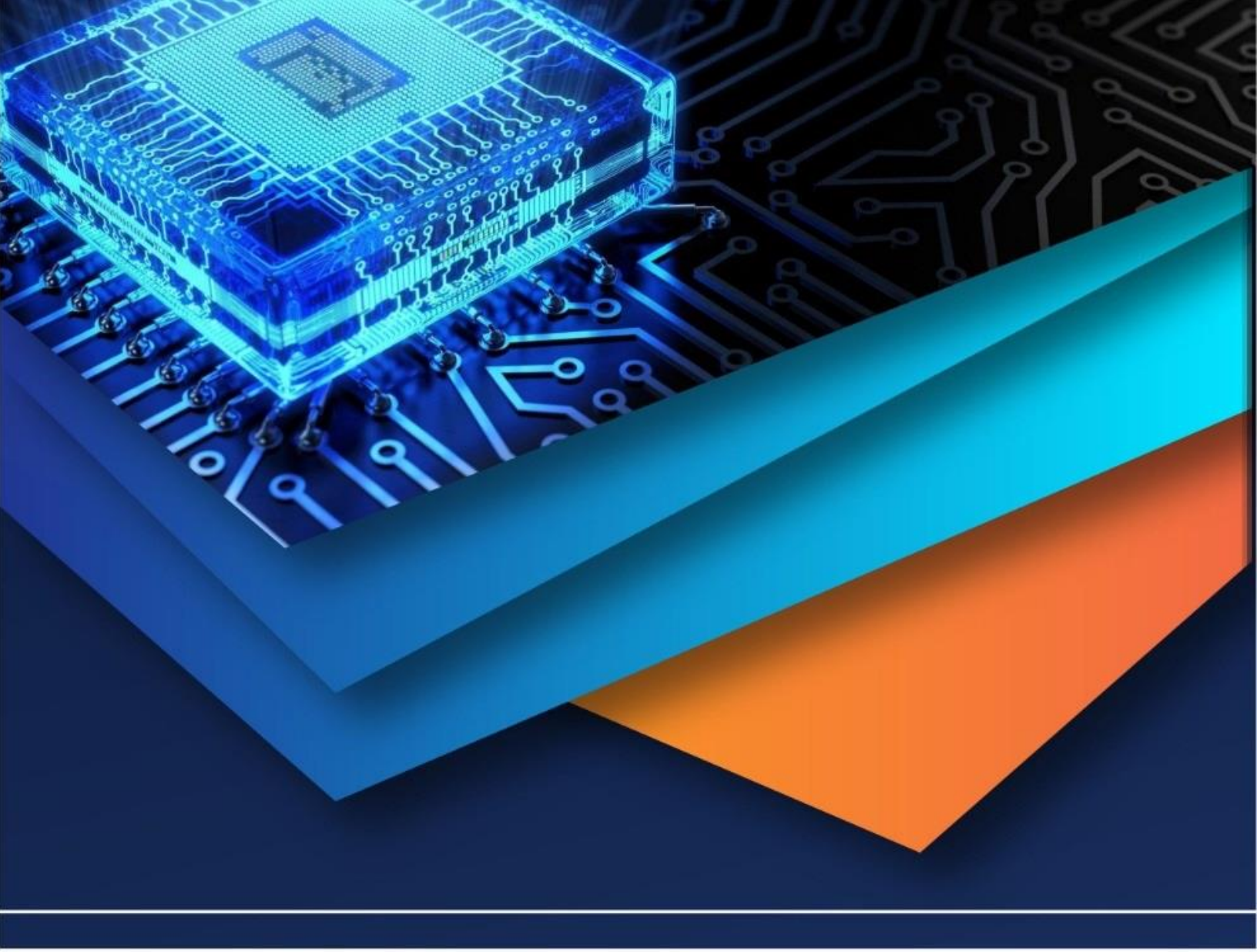
- Multimodal identification: combining text, video, and audio-based fraudulent content identification.
- Explainable AI (XAI): Using model interpretation tools (such as Grad-CAM or LIME) to graphically explain why an image was categorized as real or fake.
- Adversarial robustness: Increasing the model's resilience to hostile attacks and complex AI-generated forgeries.
- Mobile Deployment: Improving usability in remote or limited environments by transferring the lightweight model to edge devices for offline detection.

REFERENCES

[1] I. Goodfellow et al., “Generative Adversarial Nets,” Advances in Neural Information Processing Systems, vol. 27, 2014.
[2] Y. Dong, J. Ding, and D. Zhou, “Visual misinformation detection via deep learning: A review,” IEEE Access, vol. 9, pp. 12302–12324, 2021.
[3] Z. Zhao, D. Wang, T. Li, and H. Liu, “Detecting deepfake videos using recurrent neural networks,” in Proceedings of the IEEE ICASSP, 2020, pp. 2907–2911.
[4] D. Afchar, V. Nozick, J. Yamagishi, and I. Echizen, “MesoNet: A Compact Facial Video Forgery Detection Network,” in Proc. WIFS, 2018.

[5] H. H. Nguyen, J. Yamagishi, and I. Echizen, "Capsule-forensics: Using capsule networks to detect forged images and videos," in ICASSP 2019 - IEEE International Conference on Acoustics, Speech and Signal Processing, 2019.
[6] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted Residuals and Linear Bottlenecks," in Proc. IEEE CVPR, 2018.
[7] Z. Zhang, Z. Zhu, Y. Zheng, and K. Sun, "Face manipulation detection via feature disentanglement with low-rank regularization," in Proc. ACM Multimedia, 2020.
[8] J. H. Bappy, A. K. Roy-Chowdhury, and A. N. Mahmud, "Exploiting temporal inconsistencies for deepfake detection," in Proc. CVPR Workshops, 2019.
[9] A. Rössler, D. Cozzolino, L. Verdoliva, C. Riess, J. Thies, and M. Nießner, "FaceForensics++: Learning to detect manipulated facial images," in Proc. ICCV, 2019.
[10] Y. Li, X. Yang, P. Sun, H. Qi, and S. Lyu, "Celeb-DF: A Large-scale Challenging Dataset for DeepFake Forensics," in Proc. IEEE CVPR, 2020.
[11] M. Dolhansky et al., "The Deepfake Detection Challenge (DFDC) Preview Dataset," arXiv preprint arXiv:1910.08854, 2019.
[12] X. Chen, Y. Liu, W. Zhou, and T. Wang, "Edge-AI based framework for real-time deepfake detection," IEEE Internet of Things Journal, vol. 9, no. 7, pp. 5306–5314, 2022.
[13] F. Chollet, "On the measure of intelligence," arXiv preprint arXiv:1911.01547, 2019.
[14] Microsoft, "Video Authenticator Tool for Deepfake Detection," [Online]. Available: https://news.microsoft.com
[15] Deepware, "Deepfake Detection Scanner," [Online]. Available: https://www.deepware.ai
[16] F. Roffo, S. Melzi, and A. Vinciarelli, "XceptionNet for DeepFake Detection," in Proc. IEEE International Conference on Image Processing (ICIP), 2020.
[17] M. Tan and Q. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in Proc. ICML, 2019.
[18] F. Iandola et al., "SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5MB model size," arXiv preprint arXiv:1602.07360, 2016.
[19] A. Agarwal, R. Farid, and S. Basu, "DetectDeep: A Deep Learning Pipeline for Detecting AI-generated Images," ACM Digital Threats: Research and Practice, vol. 5, no. 1, 2021.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)