Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 00 (2025) 000–000

Procedia Computer Science

www.elsevier.com/locate/procedia

International Conference on Machine Learning and Data Engineering

Deepfake Detection in Ecological Monitoring: An Approach to Distinguishing AI-Generated and Camera Trap Images

Pooja Meena [a], Rakhi Kumari [a], Sadhana[a], Shweta Sharma[a], Sweeti Sah[a]

[a]*Department of Computer Science and Engineering, National Institute of Technology, Kurukshetra, Haryana, India*

Abstract

Generation Z is a data-driven generation. Everyone has the entirety of humanity's knowledge in their hands. The technological possibilities are endless. However, we use and misuse this blessing to face swap using deepfake. Deepfake is an emerging subdomain of artificial intelligence technology in which one person's face is overlaid over another person's face, which is very prominent across social media.Today AI-generated images, often referred to as deep fakes, is a growing concern, particularly for women. However, we use and misuse this blessing to face swap using deepfake. Deepfake is an emerging subdomain of artificial intelligence technology in which one person's face is overlaid over another person's face, which is very prominent across social media. Deepfake technology is increasingly misused to create explicit images and videos without consent, disproportionately targeting women. This violation of privacy leads to harassment, exploitation, reputational damage, and severe psychological impacts, including anxiety, stress, and a sense of powerlessness. This research provides object detection or classification between AI fake and Camera images using pre-trained models and for visualizing AI images. It is based on the Supervised Machine Learning Techniques. The dataset for this work is obtained from google. The MobileNet models are used for image classification. We apply this model on labelled data and also compare Model performance of MobileNet with inceptionV3 and AlexNet for image classification. Amongst all the models, MobileNet performed the best, with 95% accuracy. Besides, we obtained 89% from the AlexNet, 87% from the InceptionV3.

Keywords: Deep Learning; Supervised Machine Learning; DeepFake; Mobilenet; Real(camera) image

1. Introduction

Digital manipulation of the face images includes facial information of fake images using deep fake approaches [1] . DeepFakes are manipulated pieces of media generated to spread misinformation, hoaxes, or otherwise abusive content. With the reach of modern social media platforms, DeepFakes' inherently viral nature gives them the potential to negatively influence the opinions of millions of people, making their detection a very serious problem. Due to recent advancements in architectures like Generative Adversarial Networks (GANs) [2], DeepFake generation has become much simpler, only requiring a source image and set of intended distortions, to generate believable manipulated images. For example, one case is reported on January 2024[3] in which AI generated images of the American pop star Taylor Swift spread across social media, underscoring the damaging potential posed by artificial intelligence technology And on Sun February 4, 2024[3] A finance worker at a multinational firm was tricked into paying out $25 million to fraudsters using deepfake technology to pose as the company's chief financial officer in a video conference call. In February 2022[4] we drew attention to the scammers who were using AI-generated images to make a profit from the Turkey-Syria earthquake [5]. They would post fake images of children as a way to appeal to people's sympathy and generosity — then they'd pocket the money, and maybe the victims' credentials too. All publicly posted Deepfakes were pornographic, mainly targeting individuals in the entertainment industry [6]. The development of artificial intelligence (AI) has significantly increased the risk of Deepfakes. AI algorithms, including generative models, can now create media that are difficult to distinguish from real images. Moreover, these algorithms can be acquired at a low cost and trained on easily accessible datasets, making it easier for cybercriminals to create convincing Deepfakes for phishing attacks and scam content Deepfakes, for example, can be used to create fake identification documents, making it easier for cybercriminals to impersonate individuals or gain access to secure systems. Many software apps/tools are available through which deep fake images are created without a programming knowledge and technical side background information. Usually, profile pictures from social media are taken and fake images or videos are developed with the help of the expert. Security enhancement in the detection of face swap and the accuracy are very low. The detection of deepfake content has become one of the hot issues for individuals, businesses and governments around the world. To overcome these issues, this paper proposes a new strategy for detecting the deep fake facial images using image classification with mobilenetV3.This paper constrains the problem to binary image classification, with an image as the model input and a prediction of whether the image is real or fake as the output.

1.1 Contribution

1. Our approach employs a multi-stage classification process that encompasses image acquisition, feature extraction, and model training. This structured methodology ensures thorough analysis and processing of camera images, capturing intricate patterns indicative of various categories and objects.

2. We conduct extensive experiments to evaluate the performance of our classification system across various metrics, including accuracy, precision, recall, and F1-score. Our results demonstrate significant improvements over traditional classification methods, particularly in identifying complex and previously unseen objects and patterns.

3. The proposed system is designed to adapt to the evolving landscape of image classification by incorporating supervised learning techniques, allowing it to effectively learn from labeled data and improve its accuracy over time.

2. Motivation

With the rise of AI, new software has emerged that can create highly realistic fake images. Criminals are exploiting this technology to their advantage. These tools allow anyone to generate fake images with minimal coding, and the results often look almost indistinguishable from real camera photos. By creating these fake images, criminals frequently target women, trolling them and extorting large sums of money.

The motivation for this project stems from the increasing prevalence of AI-generated images and the challenges they pose in various applications, including media, social networks, and digital security. As AI-generated images become more realistic, distinguishing them from camera-captured images becomes critical to maintaining trust and authenticity in visual media. By harnessing the capabilities of machine learning and deep learning, we strive to create a robust, scalable, and adaptive detection system that can accurately classify AI-generated and camera-captured images. Our work aims to fill critical gaps in current detection practices and pave the way for more secure and trustworthy visual content environments.

3. Related Work

Recently, deepfake images, generated by deep learning algorithms, have attracted widespread attention. Deepfake technology can be used to perform face manipulation with high realism. The detection of deepfake content has become one of the hot issues for individuals, businesses and governments around the world. The classification of deepfake and AI-generated images is critical due to advances in AI and potential misuse. This survey reviews existing methods, emphasizing their strengths and limitations, and introduces a proposed approach. Key supervised learning models include CNNs, GANs, and RNNs, each effective in detecting patterns and anomalies in images. Challenges include the need for large labeled datasets, vulnerability to adversarial attacks, and generalization difficulties. The number of deepfake articles has grown significantly in recent years, according to data gathered.[7] A lightweight, robust fine-tuning neural network-based classifier architecture known as Fake Detection Fine-tuning Network (FDFtNet) has been developed for face detection. This network effectively identifies many new fake face image generation models [8] An unsupervised learning and CNN model approach has been used for detecting DeepFake in human face images and videos. [9] Research has delved into the creation and detection of deepfakes, providing an in-depth understanding of how these architectures function. [10]L. Minh Dang and colleagues combined Adaptive Boosting and extreme Gradient Boosting techniques to form a hybrid framework called HF-MANFA. Despite its effectiveness, this approach faced limitations in terms of high time and memory consumption during the validation of irrelevant features.[11] A multitask convolutional neural network (CNN) has been introduced to enhance the reliability of face identification and alignment.[12] Chih-Chung Hsu and collaborators introduced a deep fake detector (DeepFD) using a pairwise learning approach to improve the technique's generalization property. They employed an integrated Siamese network with Dense Net for deep fake image detection. While this approach successfully detected fake videos even in noisy environments, it required more processing time than conventional methods.

Table 1. Related works of other authors based on different supervised machine learning models and different types of datasets.

Author	**Year**	**Model**	**Dataset**	**Accuracy**	**Research Gaps**
Shraddha Suratkar & Faruk Kazi	2022 [13]	CNN,RNN	Face Forensics, Face-Forensics++, DFD	85.84%	Small dataset
Hasin Shahed Shad et .al	2021 [14]	VGG16	Dataset from Kaggle using supervised learning	92%	Overfitting the models might show high accuracy during training and validation but perform poorly on the test set or in practical applications.
Bozhi Xu1 et. al	2021 [15]	Feature Selection Method	Celeb-df and DeepFake Detection Challenge (DFDC) Preview dataset [35]	87.3%	Reduced performance on datasets generated with improved DeepFake synthesis algorithms
Haseena S et. al	2023 [16]	CNN	DFDC	94.32%	Current implementation focuses on frame-level detection
Saadaldeen Rashid Ahmed et.al	2023 [17]	RACNN	Open Source dataset Face Forensics	94.87%	The dataset size increases, the computational burden of training and updating the model grows

3.1 Research Gap

1. The research gaps for the paper given by Shraddha Suratkar & Faruk Kazi is [13] use of **small datasets**. Only 3000 images are taken in this model.
2. Gap by Hasin Shahed Shad et .al [14] is **overfitting** .Given the complexity and depth of the architectures used (DenseNet, ResNet, VGG, etc.), these models may perform exceptionally well on the training data but fail to generalize to new, unseen data. This overfitting can be exacerbated by the large number of parameters in deep networks
3. [15] especially when the dataset, despite being substantial, is limited in diversity or does not represent all variations of real and fake faces encountered in real-world scenarios.

4. Gap by Bozhi Xu1 et. al is its **reduced performance on datasets generated with improved DeepFake synthesis algorithms** while the model is effective at detecting older or simpler DeepFake methods, it struggles with more sophisticated and higher-quality fake videos.
5. Gap by Haseena S et. al [16] is needed for further research to improve the model's generalization and robustness. The current implementation focuses on frame-level detection.

4. Proposed Work

In this research work, we propose a framework "AI vs Real Image Classifier" which can effectively work on images and classify them as AI Images or Real Images with an accuracy over 95%. This framework is based on the pre-trained model of supervised deep learning model 'MobileNet'. It can effectively classify new and unique sample images without any label to them as AI generated or Real images based on the results with precision and accuracy of over 95%. In this experimental work, we give labeled datasets to the model for training and validation. We have used ImageDataGenerator for data augmentation, including rescaling, rotation, width/height shift, shear, zoom, and horizontal flip, to enhance the robustness of the model. We have improvised the accuracy of the Mobilenet model by rigorously training it on datasets from various distributions. We have used augmented dataset of a considerable size along with transfer learning and fine-tuning to produce a high performance model that can achieve higher accuracy with correct results. Also we have implemented K-Fold cross-validation to ensure a reliable and generalized performance of the model across different subsets of the data. We have trained the model using augmented training data and evaluated its performance on the validation set; and recorded and reported accuracy, precision, recall, F1-score and confusion metrics during cross-validation.

4.1 Methodology

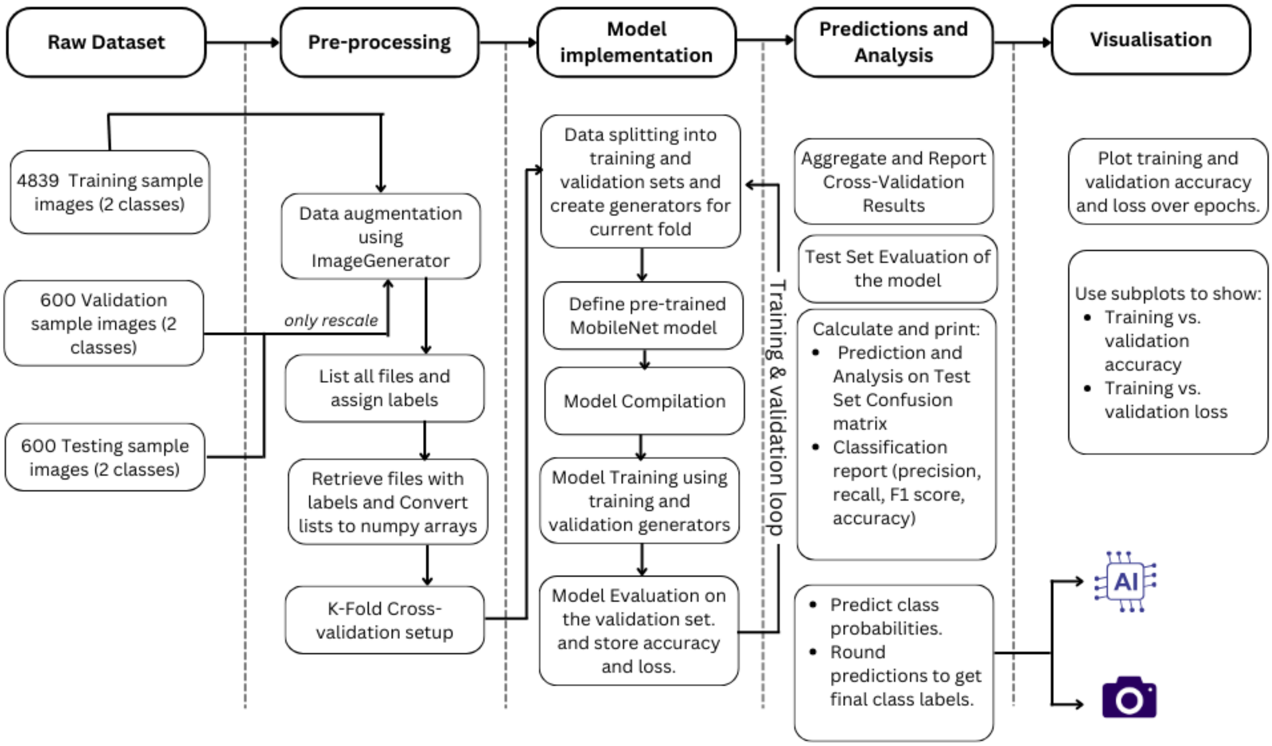

Fig.1 Conceptual Framework for Image Classification using K-Fold Cross-Validation with MobileNet

1) Raw Datasets

Datasets of images used:

a) 4839 training samples images for training of the model

b) 600 validation sample images

c) 600 sample images for the test set

Each datasets training, validation and test dataset have 2 classes as AI and Camera.

2) Data Augmentation and Preprocessing

Create data augmentation and rescaling pipelines using ImageDataGenerator for:

a) Training data: Apply augmentation techniques (rescaling, rotation, shift, shear, zoom, flip).

b) Validation data: Only rescale.

c) Test data: Only rescale.

After data augmentation, all the files of each dataset directory are listed using lists and assigned labels based on the sub-directory structure. Then all the file paths and their corresponding labels are retrieved and converted into Numpy arrays for easier manipulation. At the end of preprocessing, the K-fold cross-validation parameters (n-splits, number of folds, shuffle) are initialized.

3) Model Implementation (Training and Validation Loop)

a) After all the preprocessing steps, MobileNet Model is defined and compiled. But before implementing the model directly, the augmented datasets are split into training and validation datasets for the current fold.

b) Then data generators for the current fold are created using *flow_from_dataframe*.

c) Now for defining the model, the pre-trained MobileNet model is loaded without top layers and the custom layers are added (global average pooling, dense, dropout). Also freeze MobileNet layers to retain pre-trained weights.

d) Now compile the model with:

- Adam optimizer
- Binary cross-entropy loss
- Accuracy metric

e) Train the model using training and validation generators.

f) Evaluate the model on the validation set and store accuracy and loss for the current fold.

g) Repeat the above steps for all the folds.

4) Predictions and analysis

a) Aggregate and Report Cross-Validation Results

Calculate and print:

- Accuracy and loss for each fold.
- Average accuracy and loss across all folds.

b) Test Set Evaluation

- Create a test data generator.
- Evaluate the final model on the test set.

- Print test accuracy.

c) Prediction and Analysis on Test Set

- Reset the test generator.
- Predict class probabilities.
- Round predictions to get final class labels as AI-generated images and camera images.
- Calculate and print
  - Confusion matrix
  - Classification report (precision, recall, F1 score, accuracy)

5) Visualization

a) Plot training and validation graphs for accuracy and loss over epochs using *matplotlib*.

b) Use subplots to show:

- Training vs. validation accuracy
- Training vs. validation loss

This conceptual framework outlines the flow and interaction of the main components involved in the image classification task using K-fold cross-validation and MobileNet. Each component is responsible for specific actions contributing to the overall objective of building and evaluating a robust classification model.

4.2 Proposed Algorithm

Input: Images sample datasets

Output: Accuracy, Precision, Recall, F1- Score and Confusion Matrix for the test dataset

Data: Images divided into Training, Validation and Testing datasets consisting of 2 classes each as AI and Camera images

Step 1: *Import necessary libraries*

import tensorflow, keras, sklearn, matplotlib, numpy, os, pandas

Step 2: *Define directories*

set test_dir = 'path_to_test_directory'

set train_dir = 'path_to_train_directory'

set validation_dir = 'path_to_validation_directory'

Step 3: *Data augmentation*

initialize train_datagen with rescale, rotation_range, width_shift_range, height_shift_range, shear_range, zoom_range, horizontal_flip, fill_mode

initialize validation_datagen with rescale

initialize test_datagen with rescale

Step 4: *Set parameters*

set batch_size = 32

set epochs = 20

set num_folds = 5

Step 5: *Function to retrieve file list and labels*

define get_file_list_and_labels(directory):

set classes = list of directories in directory

initialize file_list, labels as empty lists

for each class in classes:

set class_dir = join(directory, class)

for each file in class_dir:

append file path to file_list

append class index to labels

return file_list, labels

Step 6: *Retrieve file list and labels for training*

file_list, labels = get_file_list_and_labels(train_dir)

convert file_list and labels to numpy arrays

Step 7 : *K-fold cross-validation setup*

initialize KFold with n_splits=num_folds, shuffle=True, random_state=42

Step 8: *K-fold cross-validation loop* initialize acc_per_fold, loss_per_fold as empty lists

set fold_no = 1

for each fold in KFold.split(file_list):

print 'Training fold', fold_no

Step 8a: *Data split*

set train_files, val_files = split file_list based on fold

set train_labels, val_labels = split labels based on fold

 # Convert labels to strings

convert train_labels, val_labels to string

Step 8b: *Image Data generators*

initialize train_generator with train_datagen.flow_from_dataframe(dataframe with train_files and train_labels, target_size, batch_size, class_mode)

initialize validation_generator with validation_datagen.flow_from_dataframe(dataframe with val_files and val_labels, target_size, batch_size, class_mode)

Step 8c: *Model definition*

load MobileNet with imagenet weights, include_top=False, input_shape

add GlobalAveragePooling2D, Dense, Dropout, Dense layers to base_model

create model with input=base_model.input, output=custom_layers

 # Freeze MobileNet layers

for each layer in base_model.layers:

set layer.trainable = False

Step 8d: *Compile the model*

compile model with Adam optimizer, binary_crossentropy loss, accuracy metric

Step 8e: *Train the model*

train model with train_generator, validation_generator, steps_per_epoch, validation_steps, epochs

Step 8f: *Evaluate the model*

evaluate model on validation_generator

print scores for fold_no

append accuracy and loss to acc_per_fold, loss_per_fold

increment fold_no

Step 9: *Average metrics across all folds*

print scores for each fold

calculate and print average accuracy and loss across all folds

Step 10: *Evaluate on test set*

initialize test_generator with test_datagen.flow_from_directory(test_dir, target_size, batch_size, class_mode, shuffle=False)

evaluate model on test_generator

print test accuracy

Step 11: *Predict and analyze test set*

reset test_generator

predict Y_pred with model on test_generator

round Y_pred to get y_pred

 # Calculate confusion matrix and classification report

get y_true from test_generator

calculate confusion_matrix with y_true, y_pred

calculate classification_report with y_true, y_pred

extract precision, recall, f1_score, accuracy from classification_report

print confusion matrix, precision, recall, f1_score, accuracy

Step 12: *Plot training and validation metrics*

extract acc, val_acc, loss, val_loss from history

set epochs_range = range(epochs)

plot training and validation accuracy and loss

5. EXPERIMENTAL SETUP

This section of the paper outlines the experimental setup and presents the results achieved from implementing Mobilenetv3 for deepfake.

5.1 Testing and training:

We used a dataset containing 6000 images that include 3000 images of ai from perchance [18], Microsoft Bing[19] and 3000 camera images from designer [20]. We divided this dataset into three parts- the training dataset includes 80% of the images, which equals 4800 images, are used for training the model. Testing dataset includes 10% of the images, which equals 600 images, are used for testing the model's performance. Validation dataset includes 10% of the images, which equals 600 images, are used for validating the model during training to monitor its performance and prevent overfitting.

5.2 Implementation Details:

The experimental setup involved configuring a system with Windows 11 with a 13th Gen Intel(R) Core (TM) i7-13620H 2.40 GHz ,64-bit operating system, an x64-based processor, and 16 GB of installed RAM. For implementing machine learning models, Python IDLE (Integrated Development and Learning Environment) version 3.8.80 was chosen due to its user-friendly interface and simplicity. Python IDLE is lightweight and comes bundled with Python, making it easily accessible and convenient for iterative testing and debugging through its interactive shell.

For deep learning tasks, Spyder (Scientific Python Development Environment) was utilized. Spyder offers project management features, making projects easy to organize and navigate your files. Spyder offers advanced features tailored for scientific and analytical computing, making it ideal for handling complex deep learning workflows. Its robust interface includes a powerful code editor, debugging tools, and a variable explorer, which are essential for managing large-scale projects and executing deep learning models efficiently.

5.3 Experimental Results:

The performance of machine learning models, especially identifying deepfake is critically evaluated using the confusion matrix. This matrix provides a detailed breakdown of the model's ability to distinguish between different classes (e.g., ai vs camera). The confusion matrix includes four key components: True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN).

True Positive (TP): Instances where the model correctly identifies ai image as deepfake.

True Negative (TN): Instances where the model accurately identifies camera image as non-deepfake.

False Positive (FP): Cases where the model incorrectly labels camera image as deepfake.

False Negative (FN): Cases where the model fails to identify ai image, incorrectly classifying it as a camera image.

The confusion matrix can be represented as:

Mathematically,

Table 2. Representation of the confusion matrix.

	Predicted Positive	**Predicted Negative**
Actual Positive	TP(TruePositive)	FN(FalseNegative)
Actual Negative	FP(FalsePositive)	TN(TrueNegative)

By analyzing the confusion matrix, we can derive metrics such as Precision, Recall, F-score, and Accuracy:

$$\text{Precision} = TP / (TP + FP) \quad \text{(i)}$$

Precision is the ratio of true positive predictions to the total number of positive predictions made by the model.

$$\text{Recall} = TP / (TP + FN) \quad \text{(ii)}$$

Recall (or sensitivity) is the ratio of true positive predictions to the total number of actual positives in the dataset.

$$\text{Accuracy} = (TP + TN) / (TP + TN + FP + FN) \quad \text{(iii)}$$

Accuracy the ratio of the number of correct predictions to the total number of predictions.

$$\text{F-score} = 2 * \text{Precision} * \text{Recall} / (\text{Precision} + \text{Recall}) \quad \text{(iv)}$$

The F1 score combines precision and recall into a single metric by calculating their harmonic mean. This is useful when you need a balance between precision and recall, especially in cases where the class distribution is imbalanced.

The confusion matrix is used to evaluate each technique applied in this study. The results for Mobilenet Model are as follows:

Performance Metrics:

Table 3. The confusion matrix of MobileNet Model.

	Predicted Positive	**Predicted Negative**
Actual Positive	277	13
Actual Negative	23	287

The table below presents the accuracy, precision, recall, and F-score for each model:

Table 4. Classification report of each model implemented

Model	Accuracy	Precision	Recall	F-Score
MobileNet	95.00%	95.00%	95.00%	95.00%
InceptionV3	87.00%	87.00%	87.00%	87.00%
AlexNet	89.00%	89.00%	89.00%	89.00%

The table highlights that the Mobilenet Model achieved the highest accuracy of 95.0%, showcasing its effectiveness in identifying ai image. Inception and AlexNet showed moderate performance, with 87.0% & 89.0% respectively.

6. COMPARISON WITH THE EXISTING LITERATURE

In this section, we compare the proposed "AIRIC" framework with the existing frameworks for the automated classification of AI-generated images and camera images. The existing frameworks and models implemented have various limitations and lower accuracy which makes them a little unreliable in detecting images generated by using highly advanced ai image generators. In literature, the VGG16 model (Hasin Shahed Shad et .al 2021) [14] was implemented on the dataset from Kaggle using supervised learning and overfitting was used on the models that produced high accuracy during training and validation but was performing poorly on the test set or in practical applications. In contrast, the proposed framework uses overfitting but the accuracy of the image classification remains high throughout the training and validation process as well as it performs well on test sets to produce high accuracy and good results. We used larger dataset with a range of images from Google and ai image generators for training and validation of the model with transfer learning and fine-tuning methods to ensure higher accuracy and efficient results on the test sets as well.

Due to the scarcity in the literature for the automated classification of AI-generated images and camera(real) images using supervised machine learning techniques, we compare the classification accuracy on the test sets of our proposed "AIRIC" framework with the existing frameworks as shown in Fig.2.

The existing frameworks are mostly for deepfake detection which follow different supervised machine learning techniques in which targets are already known to the algorithm and smaller datasets and high overfitting. The traditional existing frameworks are not capable of classifying unique and highly accurate images generated using Highly efficient AI-image generating techniques, confusing the masses and causing severe damages to the victims and related authorities. **Fig.2** shows that our proposed framework which works on a larger dataset for training and validation performs well on the test set and gives the best accuracy (95%) as compared to the existing frameworks.

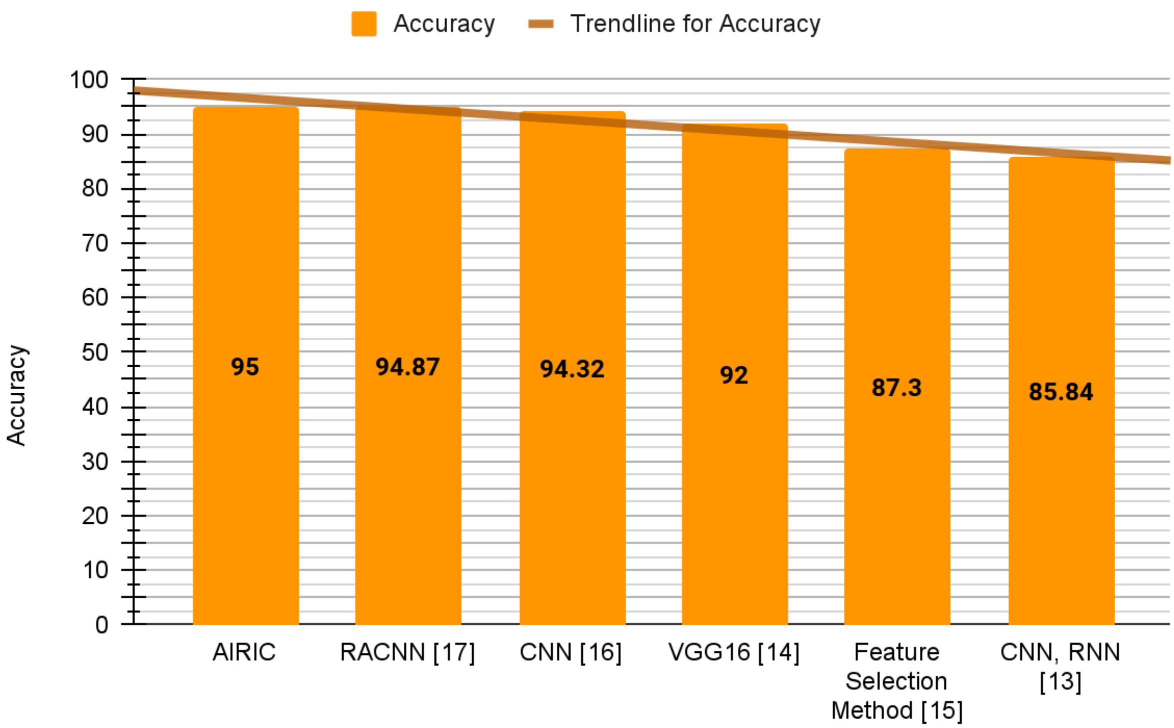

Fig.2 Comparison of accuracy of the proposed AIRIC framework for automated classification of AI-generated images and camera images with the existing traditional frameworks for deepfake detection.

7. CONCLUSIONS

Motivated by the rise in the digital manipulation of images of females all around the world which ultimately leads to financial damages or damage to dignity, we introduce a novel AIRIC framework using pre-trained model based supervised machine learning techniques with the help of transfer learning and data augmentation to give a highly accurate and good result in classifying the images as AI-generated images and camera images. The proposed AIRIC framework comprises training and validation of a large datasets consisting of labeled data as AI and camera. Along with training and validation of the datasets, cross-validation is performed as well to ensure high accuracy of the classification result. The highest accuracy achieved on the test dataset is 95% and the F-score values is 95%.

The proposed AIRIC framework shows better accuracy as compared to the existing frameworks based on supervised machine learning techniques for automated classification of AI-generated images and Camera images. The traditional existing frameworks are not capable of classifying unique and highly accurate images generated using Highly efficient AI-image generating techniques, confusing the masses and causing severe damages to the victims and related authorities. In contrast, the proposed framework uses overfitting along with larger datasets with cross-validation and gives better results.

In future work, the framework can perform better with a much higher accuracy and the much complex and highly efficient AI-generated or manipulated images can be classified with increasing the dynamic features of the complex datasets as input and perform efficiently.

REFERENCES

[1] Pavel Korshunov, Sebastien Marcel DeepFakes: A New Threat to Face Recognition?

[2] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," 2014.

[3] CNN World, Finance worker pays out $25 million after video call with deepfake 'chief financial officer' By Heather Chen and Kathleen Magramo, CNN Published 2:31 AM EST, Sun February 4, 2024

[4] Trend Micro , Fooled: Top Twelve AI Scams and Pranks by Avril Ronan | Mar 31, 2024 | For Parents, For Teachers

[5] Turkey-Syria earthquake

[6] SKYNET TODAY Putting AI News In Perspective

[7] Jeon, H., Bang, Y., Woo, S.S. (2020). FDFtNet: Facing Off Fake Images Using Fake Detection Fine-Tuning Network. In: Hölbl, M., Rannenberg, K., Welzer, T. (eds) ICT Systems Security and Privacy Protection. SEC 2020. IFIP Advances in Information and Communication Technology, vol 580. Springer, Cham. https://doi.org/10.1007/978-3-030-58201-2_28[reference]

[8] A. Malik, M. Kuribayashi, S. M. Abdullahi and A. N. Khan, "DeepFake Detection for Human Face Images and Videos: A Survey," in *IEEE Access*, vol. 10, pp. 18757-18775, 2022, doi: 10.1109/ACCESS.2022.3151186. keywords: {Information integrity;Videos;Deep learning;Media;Kernel;Forensics;Faces;Deep learning;DeepFake;CNNs;GANs},[reference]

[9] The Creation and Detection of Deepfakes: A Survey: ACM Computing Surveys: Vol 54, No 1

[10] Dang, L.M., Hassan, S.I., Im, S. and Moon, H., 2019. Face image manipulation detection based on a convolutional neural network. *Expert Systems with Applications*, *129*, pp.156-168.[reference]

[11] A. F. S. Moura, S. S. L. Pereira, M. W. L. Moreira and J. J. P. C. Rodrigues, "Video Monitoring System using Facial Recognition: A Facenet-based Approach," GLOBECOM 2020 - 2020 IEEE Global Communications Conference, Taipei, Taiwan, 2020, pp. 1-6, doi: 10.1109/GLOBECOM42002.2020.9348216. keywords: {Support vector machines;Webcams;Face recognition;Authentication;Streaming media;Video surveillance;Monitoring;Security;Trust and Privacy in e-Health;eHealth;Video Surveillance;Facial Recognition;FaceNet;Support Vector Machine} [reference]

[12] Applied Sciences | Free Full-Text | Deep Fake Image Detection Based on Pairwise Learning (mdpi.com)

[13] Suratkar, S., Kazi, F. Deep Fake Video Detection Using Transfer Learning Approach. *Arab J Sci Eng* **48**, 9727–9737 (2023). https://doi.org/10.1007/s13369-022-07321-3 [reference]

[14] Hasin Shahed Shad, Md. Mashfiq Rizvee, Nishat Tasnim Roza, S. M. Ahsanul Hoq, Mohammad Monirujjaman Khan, Arjun Singh, Atef Zaguia, Sami Bourouis First published: 16 December 2021 https://doi.org/10.1155/2021/3111676 [reference]

[15] DeepFake Videos Detection Based on Texture Features (techscience.cn); Bozhi Xu , Jiarui Liu , Jifan Liang , Wei Lu1, and Yue Zhang, Received: 09 January 2021; Accepted: 11 February 2021 [reference]

[16] S, H., S, S., D, S. D., & A., N. (2023). TVN: Detect Deep Fakes Images using Texture Variation Network. *Inteligencia Artificial*, *26*(72), 1–14. https://doi.org/10.4114/intartif.vol26iss72pp1-14 [reference]

[17] Ahmed, S.R., Sonuç, E. RETRACTED ARTICLE: Evaluating the effectiveness of rationale-augmented convolutional neural networks for deepfake detection. *Soft Comput* (2023). https://doi.org/10.1007/s00500-023-09245-y [reference]

[18] https://perchance.org/ai-text-to-image-generator [reference]

[19] https://www.deep-nude.ai/
[20] https://designer.microsoft.com/imagereatorhttps://designer.microsoft.com/image-creator