



FAKE SCOPE: AI-DRIVEN DEEPPFAKE DETECTION AND VERIFICATION

K Karthik Reddy¹, Nikhil Vishnu Malavade², Akash Ashok Banavar³, Nawaz Sharif Z N⁴

Department of Computer Science and Engineering,
GM Institute of technology, Davangere, 577004, Karnataka

ABSTRACT :

The paper discusses Fakescope, an AI-driven system for deepfake detection and verification, addressing the challenges posed by realistic, manipulated images and videos. While deepfakes have legitimate uses, their potential for misuse raises concerns about misinformation and digital impersonation. The system features a web interface built with HTML, CSS, and JavaScript, supported by a Python Flask backend and Xception-based convolutional neural network. For image classification, the model processes 224x224 pixel images and outputs a binary score. Video analysis is performed by extracting frames with OpenCV, analyzing them individually, and aggregating results for a collective prediction. The paper outlines the problem statement, objectives, related work, methodology, model architecture, implementation, and evaluation framework, aiming to establish a preliminary screening tool for suspicious media, with effectiveness to be validated through independent testing and classification metrics. Key terms include Deepfake Detection, Artificial Intelligence, and Convolutional Neural Networks.

Keywords: Deepfake Detection, Artificial Intelligence, Deep Learning, XceptionNet, Image Forensics, Video Analysis, Convolutional Neural Networks, Flask.

INTRODUCTION

The increasing prevalence of digital images and videos on platforms such as social media and journalism has escalated concerns regarding the authenticity of digital content. The rise of advanced artificial intelligence and deep learning technologies has enabled the generation and manipulation of facial appearances with remarkable realism, leading to term “deepfake detection”. While these technologies have potential benefits in areas such as creativity, visual effects, and digital content production, their misuse poses significant risks, including misinformation, identity theft, reputational harm, and the proliferation of misleading information.

The challenge of distinguishing authentic content from manipulated media is becoming more complex as the quality of synthetic media improves, making human detection increasingly difficult. Traditional methods for identifying manipulated media rely on the observation of visible artifacts, inconsistencies, or manual examination, which may not suffice when manipulations are subtle or have been altered further through compression or resizing. In contrast, deep-learning -based detection methods utilize an analysis of learned patterns from labeled datasets of genuine and manipulated media.

This research introduces Fakescope, a web-based system designed for deepfake detection that integrated image classification and video analysis within a single interface. Utilizing an Xception-based neural network, Fakescope analyzes visual inputs to generate binary predictions about authenticity. The backend is managed using Flask for handling request and model inference, while the user-friendly frontend allows without needing to engage with the underlying machine learning scripts. The focus of Fakescope is on streamlining the deepfake screening process to make it easily accessible for users seeking to verify the authenticity of visual content.

II. AIMS AND OBJECTIVE

A. AIM

The primary aim of this project is to design and implement FakeScope, an AI-driven web application that applies a deep-learning model to analyze digital images and videos. The application intends to classify visual content as either real or fake, providing users with predictions through an accessible interface.

B. Objectives

The main key goals includes: (1) Analyzing existing techniques to highlight challenges in detecting media. (2) Exploring the use of convolutional neural network, notably the Xception architecture, for binary deepfake classification. (3) Preparing and preprocessing labeled datasets of both real and fake media for model training and assessment. (4) Creating an image-classification pipeline that standardizes input images and produces real-or-fake predictions. (5) Developing a video-analysis pipeline utilizing OpenCV to select frames and prediction scores. (6) Integrating the trained model with a Python Flask backend for handling uploaded media. (7) Building user-friendly web interface using HTML, CSS, and JavaScript for image and video uploads and displaying prediction results. (8) Evaluating the system with a separate test dataset, accuracy, F1-Score and confusion matrix.

C. Expected Outcome

The expected outcome of the project is a fully operational web application that integrates various components, including media upload, preprocessing, deep-learning interface, and result presentation. This system aims to effectively demonstrate how an Xception-vbased classifier can be utilized within an image and video analysis workflow. To validate the system's performance, it is essential that implementational evaluation performance metrics will be reported only after these thorough tests have been completed, ensuring accuracy in assessing the system's capabilities.

III. LITERATURE REVIEW

A. Deepfake Detection

Deepfake detection has emerged as a critical area within digital media forensics. Researchers are studying various indicators, such as visual artifacts, inconsistencies in facial features, blending boundaries, and neural network patterns, to differentiate manipulated content from authentic media. Detection becomes particularly challenging when the altered content undergoes compression, resizing, or employs techniques not represented in the training dataset. Therefore, an effective evaluation framework must consider both performance on familiar examples and the generalization to previously unseen data.

B. Face Forensics++

Rossler et al. Introduced FaceForensics++, a benchmark designed for assessing facial manipulation detection that encompasses various manipulation techniques and compression factors. The dataset serves as a foundational resource for examining how forensic classifiers identify manipulated facial content compared to genuine materials. Its relevance to FakeScope lies in demonstrating the necessity of utilizing labeled real and manipulated examples in the development and evaluation of deepfake detectors, as well as the importance of accounting for specific artifacts arising from compression and manipulation.

C. Deepfake Detection Challenge Dataset

The Deepfake Detection Challenge dataset was established to facilitate research in detecting manipulated videos. It comprises a selection of both genuine and manipulated video material, essential for training and evaluating detection methodologies. A detector tailored for video analysis must consider frame variations, recording conditions, and manipulation techniques; however, performance on a single dataset should not be viewed as definitive evidence of reliable detection across all real-world scenarios.

D. Celeb-DF Dataset

Li et al. presented the Celeb-DF dataset, which offers a rigorous framework for studying deepfake detection under more realistic manipulation scenarios, addressing weaknesses in earlier deepfake examples that may have presented identifiable visual artifacts. This dataset is crucial for evaluating the capability of detectors to discern manipulations that appear visually convincing. For FakeScope, incorporating diverse and independent data sources aids in assessing the extent to which a model can generalize beyond its training data.

E. Xception Architecture

Developed by Challet, the Xception architecture employs depth-wise separable convolutions, which segregate spatial feature extraction from channel-wise processing, providing a novel alternative to traditional convolutional methods. A pretrained Xception network can be modified for a new classification head specific to the task and finetuning certain layers. FakeScope implements this transfer-learning approach to yield binary predictions for each input image.

F. Research Gap and Project Motivation

The existing body of research indicates that deep learning methods can effectively identify manipulated media under specific experimental settings. However, creating a practical detection application requires the integration of media preprocessing, model inference, video processing, and result presentation. FakeScope aims to address this integration by combining image classification with frame sampling video analysis within a single web application, positioning its contribution as the implementation and evaluation of a cohesive system. Any assertion of improved detection accuracy or enhanced generalization must be substantiated by comparative experimental evidence.

IV. METHODOLOGY

A. System Overview

Fake Scope operates through a sequential workflow involving media upload, input validation, preprocessing, model inference, prediction generation, and result presentation. Users select an image or video via a web interface, which is processed by a Flask backend. Images are formatted for model input, while video frames are extracted using OpenCV. Each processed image or frame is evaluated by an Xception-based classifier, yielding a score for a binary prediction. For videos, frame scores are averaged for a final video level score, which is then displayed on the frontend.

B. Dataset Preparation

The development of a supervised deepfake detector necessitates a dataset comprising labeled genuine and manipulated media, organized into training, validation, and test subsets. Careful documentation of dataset sources and counts is critical, ensuring accuracy against the actual files used. The training subset is for model parameter learning, the validation subset aids in model selection, and the test subset is reserved for final evaluation. To minimize risk of overlap, splitting should occur at the source-video or identity level.

C. Data Preprocessing

Input preprocessing is vital for consistent media representation prior to classification. For images, preprocessing involves decoding, resizing to 224*224 pixels, and converting to a three-channel format, ensuring pixel scaling matches training conditions. Video analysis employs OpenCV to select frames uniformly, processing up to 25 sampled frames per video as long as decoding is successful.

D. Tools and Technologies

TABLE 1. Tools and Technologies Employed

CATEGORY	TECHNOLOGY	PURPOSE
Frontend	HTML, CSS, JavaScript	Developing the user interface for uploading images and videos and displaying detection results
Backend	Python, Flask	Handling server-side logic, file uploads, and communication between the frontend and model
Deep Learning Framework	TensorFlow, Keras	Building, loading, and executing the deepfake detection model
Deep Learning Model	XceptionNet(Transfer Learning)	Extracting image features and classifying input media as real or fake
Image Processing	OpenCV	Reading video files, extracting frames, and preparing images for analysis
Numerical Processing	NumPy	Array operations, images data processing, and aggregation of video-frame prediction scores
Model Preprocessing	Image resizing to 224×224 pixels and normalization	Preparing input images in the format expected by the trained model
Video Analysis	OpenCV and NumPy	Sampling up to 25 frames from a video and aggregating frame-level prediction scores
Dataset	DFDC, FaceForensics++, and other dataset sources	Providing real and manipulated media for model training and evaluation
Model Storage	HDF5 (.h5) and Keras (.keras)	Saving and loading the trained deepfake detection model
Development Tools	Visual Studio Code, Git, GitHub	Code development, version control, and project collaboration

E. Xception-Based Model Architecture

Fake Scope employs an Xception-based convolutional neural network utilizing ImageNet-pretrained weights for its model architecture. It comprises an Xception feature extractor, global average pooling, batch normalization, and fully connected layers. The classification head includes a 256-unit dense layer with ReLU activation and a 0.4 dropout rate, followed by a 64-unit dense layer with ReLU and a 0.2 dropout rate, culminating in a single neuron with sigmoid activation. This sigmoid output scores between zero and one, designating class 0 for real media and class 1 for fake media, with a decision threshold are considered fake, while those below are real. The output score is model-generated and should not be misinterpreted as a calibrated probability of authenticity.

F. Model Training

Model Training outlines the training configuration using TensorFlow/Keras with binary cross-entropy loss and the Adam optimizer. The initial training phase employs a learning rate of 0.0001 for up to 10 epochs, followed by a fine-tuning phase with a rate of 0.00001 for an additional 5 epochs.

The batch size is set to 16 and early stopping monitors validation loss with a patience of four epochs. The model's final 30 layers are unfrozen during fine-tuning. Proper verification of training completion and model artifacts is necessary before reporting results.

G. Image Classification

Image Classification describes the module's processing steps for analyzing images uploaded to the system. The steps include validation, decoding, resizing to 224×224 pixels, preprocessing, passing through an Xception-based model, obtaining output scores, applying a decision threshold of 0.5, and returning the predicted label and score to the web interface. This module serves the image-level prediction functionality of FakeScope.

H. Video Classification

The video-classification module adapts the image classification pipeline for video inputs by selecting up to 25 frames across the video duration. Each frame is decoded and evaluated independently, with individual scores (S). The formula used is (Equation), where s_i is the score for each processed frame, and N is the total number of frames processed. A video is classified as fake if $S \geq 0.5$; otherwise, it is deemed real. If the video cannot be opened or no frames are processed, an error is returned instead of a misleading prediction.

V. RESULT AND DISCUSSION

A. Experimental Evaluation

It involves testing the system with a held-out dataset of labelled real and fake images and videos, using the final model artifact and preprocessing pipeline. Each image is compared with its ground truth label, while video classification produces predictions based on frame-sampling and score aggregation. Evaluation records the number of samples per class, as well as classification metrics. Verified numerical results are pending completion of the experiments.

B. Performance Metrics

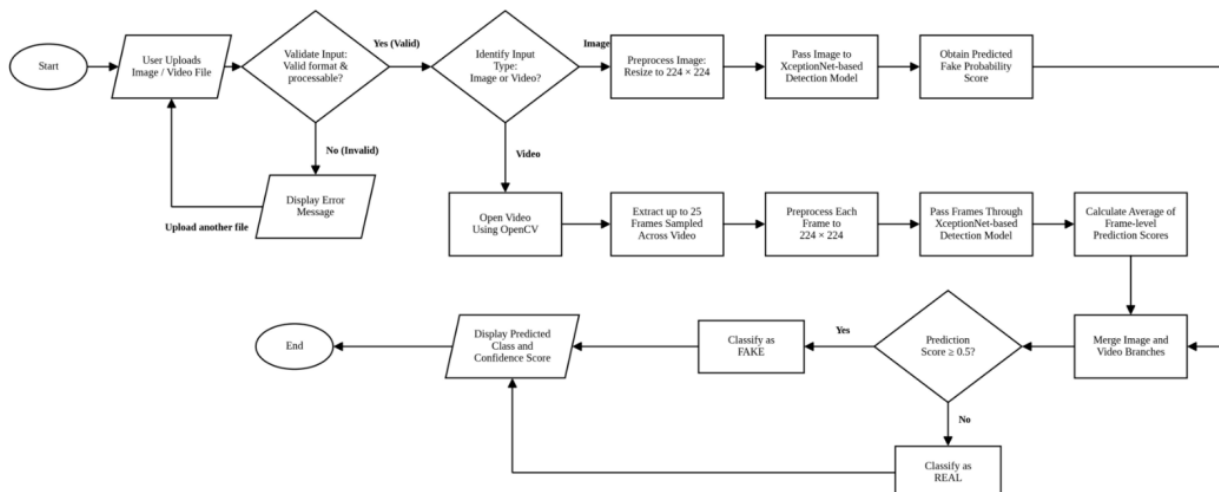
Accuracy: proportion of correctly classified samples. Formula: $\text{Accuracy} = (\text{TP} + \text{TN}) / (\text{TP} + \text{TN} + \text{FP} + \text{FN})$. [2] Precision: proportion of predicted fake samples that are actually fake Formula: $\text{Precision} = \text{TP} / (\text{TP} + \text{FP})$ [3] Recall: Proportion of actual fake samples correctly identified. Formula: $\text{TP} / (\text{TP} + \text{FN})$ [4] F1-Score: Harmonic mean of precision and recall. Formula: $\text{F1-score} = 2 \times \text{Precision} \times \text{Recall} / (\text{Precision} + \text{Recall})$.

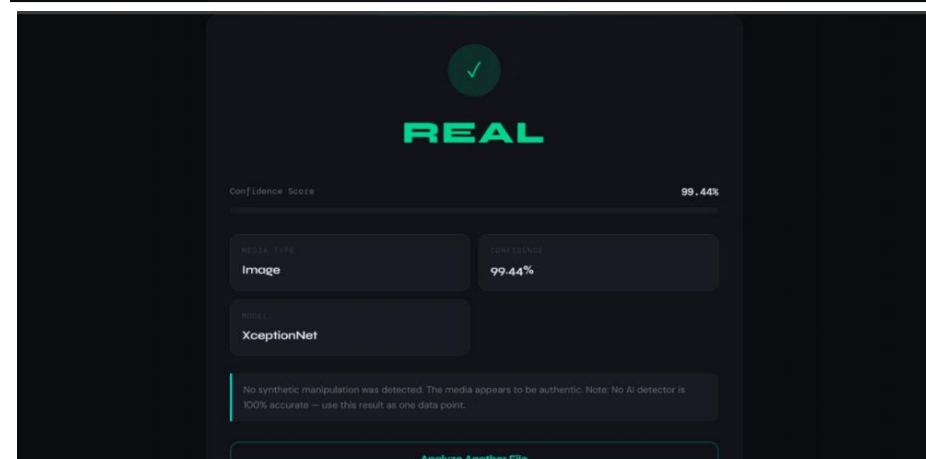
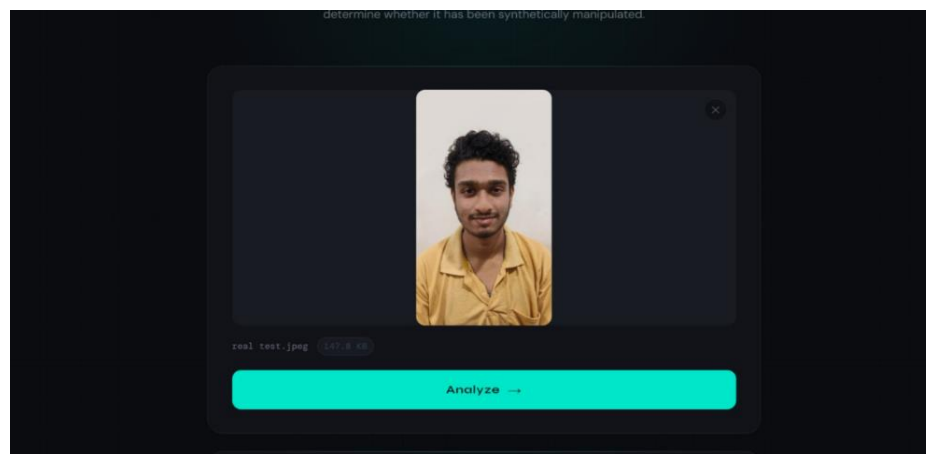
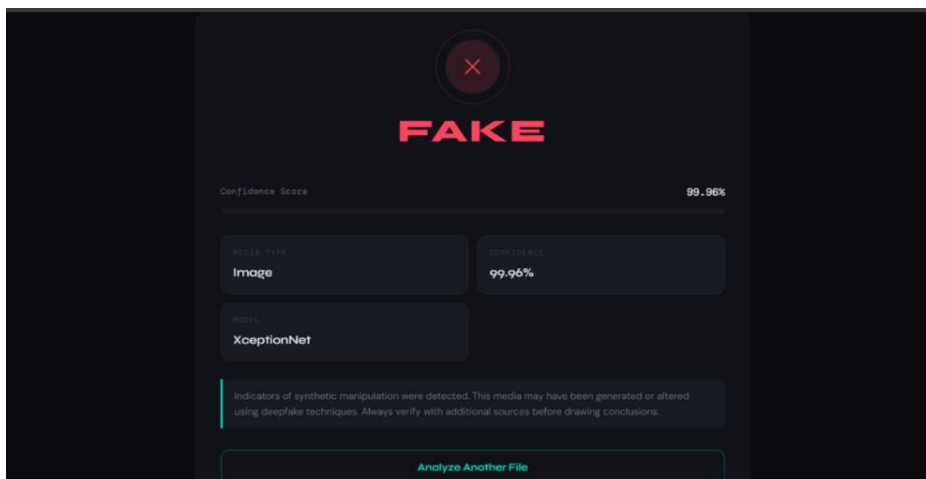
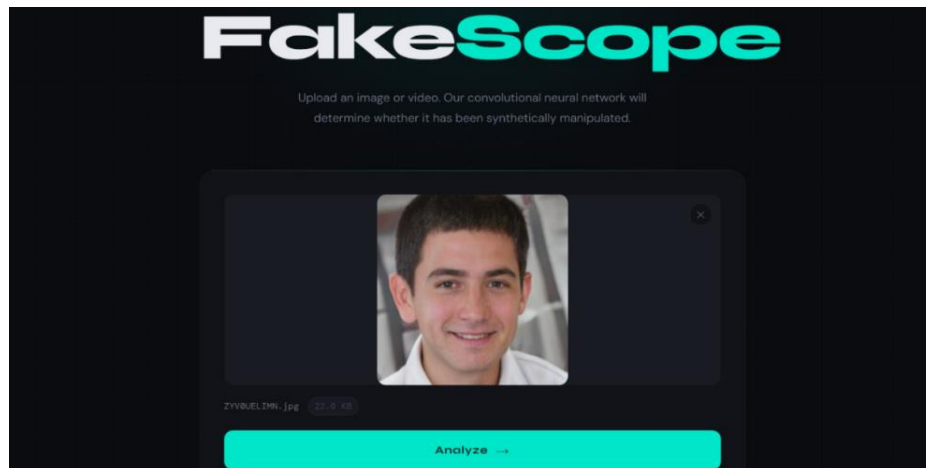
C. Test results

TABLE 2: Test Results

Evaluation metric	Measured result
Total frames evaluated	10,474
Accuracy	50.59%
Precision	0.00%
Recall (FAKE class)	0.00%
Specificity	99.98%
F1-score	0.00%
ROC-AUC	0.5392
True positives (TP)	0
True negatives (TN)	5,299
False positives (FP)	1
False neegatives (FN)	5,174

Flow Chart:



Output images:

D. Quantitative Results

It is used for classification performance on a held-out test dataset. A table outlines metrics for image and video classification including the number of test samples, accuracy, precision, recall, F1-score, all of which are yet to be measured. It notes that results should be reported separately for images and videos due to different evaluation units.

E. Discussion

Fake scope integrates a deep-learning classifier with a web interface and video screening. It employs a shared image classifier, simplifying video analysis by reusing the model across sampled frames. Uniform frame sampling reduces computational demands but relies on selected frame and aggregation methods. A fixed threshold can skew results, potentially misclassifying genuine media as fake or manipulated media as authentic. Therefore, the effectiveness of the Fake scope should be evaluated based on performance metrics across various datasets and manipulation techniques, rather than merely the presence of its interface.

VI. LIMITATIONS AND FUTURE ENHANCEMENTS

A. Limitations

First, the training data is necessary for the classification model to function. Performance may suffer on unfamiliar content if the training dataset does not accurately reflect the features of recently, created deepfakes. Second, instead of analyzing every frame, the video-analysis pipeline uses a restricted number of sampled frames. Temporal discrepancies or brief modified intervals might not be captured by this. Third, the current implementation fixes the decision threshold at 0.5. Fourth, the scope produced by the model may not be a calibrated probability. Therefore, rather than being conclusive evidence of a file's authenticity or manipulation, users should understand the given result as a model forecast.

B. Future Enhancements

Assessing the system using separate datasets with various deep-fake generations methods. [2] Examining temporal deep learning models that examine the connections between video frames. [3] Enhancing frame-selection techniques to more successfully identify potentially alerted parts. [4] Using validation data to examine model calibration and choose decision criteria. [5] Enhancing application security, processing error management, and upload validation. [6] Extending the assessment to include various recording situations, compression levels, and image resolution [7] Clearly expressing uncertainties and enhancing the way prediction scores are presented. [8] Applying consistent experimental circumstances to compare alternative topologies with Xception-based model.

VII. CONCLUSION

This paper introduces Fake scope, an AI driven web application for detecting and verifying deepfakes using an Xception-based deep-learning model. It features a web frontend and a python Flask backend, with TensorFlow/Keras for interface, image preprocessing, and OpenCV for video frame extraction. Images are resized and classified based on a sigmoid output with a threshold are examined individually, and the overall classification. Future work focuses on independent dataset validation, performance evaluation, improved video analysis, and enhanced prediction reliability. Fake Scope serves as a preliminary screening tool rather than a definitive forensic authority.

The project demonstrates an application-oriented approach to integrating a deep-learning interface with a user-facing media-analysis workflow. However, the practical effectiveness through independent testing, appropriate classification metrics, and analysis of failure cases. The system should be considered a preliminary screening tool rather than a definitive forensic authority.

REFERENCES

1. A. Rossler, D. Cozzolino, L. Verdoliva, C. Riess, J. Thies, and M. Nießner, "FaceForensics++: Learning to Detect Manipulated Facial Images," Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2019.
2. D. Afchar, V. Nozick, J. Yamagishi, and I. Echizen, "MesoNet: A Compact Facial Video Forgery Detection Network," 2018 IEEE International Workshop on Information Forensics and Security (WIFS), 2018.
3. F. Chollet, "Xception: Deep Learning with Depthwise Separable Convolutions," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017.
4. M. Tan and Q. V. Le, "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks," Proceedings of the International Conference on Machine Learning (ICML), 2019.
5. H. H. Nguyen, J. Yamagishi, and I. Echizen, "DeepFake Detection Using Deep Learning Methods," arXiv preprint arXiv:1909.11573, 2019.
6. I. Amerini, L. Galteri, R. Caldelli, and A. Del Bimbo, "Deepfake Video Detection Through Optical Flow Based CNN," Proceedings of the IEEE International Conference on Computer Vision Workshops (ICCVW), 2020.
7. H. H. Nguyen, F. Fang, J. Yamagishi, and I. Echizen, "Capsule-Forensics: Using Capsule Networks for Image Forgery Detection," ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2019.
8. Y. Mirsky and W. Lee, "Deep Learning for Deepfakes Creation and Detection: A Survey," ACM Computing Surveys, vol. 54, no. 1, pp. 1–36, 2021.