Mustansiriyah Journal of Pure and Applied Sciences, Print ISSN: 2957-9910. Online ISSN: 2957-9929, Vol. 3, No. 4, 2025, Pages 103-116
DOI: 10.47831/mjpas.v3i4.278

MJPAS

MUSTANSIRIYAH JOURNAL OF PURE AND APPLIED SCIENCES

Journal homepage:
https://mjpas.uomustansiriyah.edu.iq/index.php/mjpas

RESEARCH ARTICLE - COMPUTER SCIENCE

Enhanced Images Deepfake Detection Using YOLOv8 with MTCNN Face Extraction Method for High-Accuracy Classification

Sumaia Ali Alal [1], Sawsen Abdulhadi Mahmood [2]

1,2 Computer Science Department, College of Education, Mustansiriyah University, Baghdad, Iraq

* Corresponding author E-mail: sumaiali96@uomustansiriyah.edu.iq

Article Info.

Article history:

Received

31 July 2024

Accepted

3 September 2024

Publishing
30 September 2025

Abstract

With the increasing use of fake images and videos on social media platforms, the issue of verifying the authenticity of images has become of great importance for privacy protection .Recently, deepfake images technology has widely used for face-swapping in order to generate fake or forged data to deceive society. Detecting the rightfulness of images has become progressively critical issue due to the potential harmful effect on the human privacy and security. The main objective of this study is to develop and implement a deep fake detection system using face region extraction method and YOLOv8 deep learning model. The proposed system mainly relies on extracting the face region from the input images sample using Multi-Task Cascaded Convolutional Neural Networks (MTCNN) method to reduce the processing time of fake /real face image in the detection process. The extracted faces regions are then analysed and classified based on YOLOv8 model to obtain the fake and real information related to each input image in term of binary classification. Further, the main hyperparameters of Yolov8 model were tuned throughout model training phase to generate more robust trained model for achieving higher detection accuracy. The system performance was evaluated using multiple fake and real images datasets. The results showed that the proposed system achieved a detection accuracy = 97.5%, Precision = 97.03%, Recall = 98%, and mAp = 99%

The official journal published by the College of Education at Mustansiriya University

Keywords: Deepfake detection, Yolov8 model, Hyperparameters tuning, Face detection.

1. Introduction

Recent advances in artificial intelligence (AI), especially generative adversarial networks (GANs) and an abundance of training samples, coupled with powerful computational resources, have led to a significant increase in the production of AI-generated fake information, such as deepfakes. Deepfake technology is used to create fake visual and audio content based on a person's existing media, where their face and voice are replaced with fake media to appear realistic. Producing such fake content is an unethical act and poses a threat to society, as this technology is increasingly used in cybercrimes such as identity theft, blackmail, spreading fake news, financial fraud, and producing fake pornographic videos of celebrities for the purpose of blackmail. This technology poses many security and privacy threats, such as distortion and misinformation in politics and personal relationships[1]. Recent advancements in artificial intelligence, especially generative adversarial networks (GANs), have made deepfakes more realistic and harder to detect. These technologies enhance the quality and detail of synthetic content, blurring the line between authentic and manipulated media [1]. Object detection is one of the fundamental challenges in the field of computer vision, as it involves identifying and classifying elements within images or videos. This capability opens up vast possibilities in surveillance

and image analysis.[2]. The proposed study leverages the YOLOv8 model to address these advancements by utilizing its real-time object detection capabilities to identify subtle features and inconsistencies in deepfake content. YOLOv8's advanced architecture and high accuracy in detecting fine details make it well-suited for identifying the sophisticated manipulations introduced by modern GANs, thus improving the effectiveness of deepfake detection[3]. Deepfakes are synthetic but hyper-realistic images and videos, created by merging, overlaying, or replacing facial regions in images/videos using advanced computer vision and deep learning techniques. Deepfakes are considered one of the most dangerous forms of misinformation, posing significant security and privacy threats to government institutions and individuals worldwide. Deepfake generation algorithms are continually improving and are being exploited by malicious individuals to spread harmful content, such as ransom ware and digital kidnapping. This increasing threat has led to the development of deepfake forensics, which focuses on verifying the authenticity of digital media.[4].

This research work aims to improve the YOLOv8 model for binary classification of real and fake faces in human images. MTCNN is used for face detection, and the improved YOLOv8 model extracts facial features from images and videos. The proposed framework consists of four stages: data collection and preparation, face detection, facial landmark detection, and feature extraction using YOLOv8. A custom dataset including 8082 real and fake samples was created to enhance the model's performance. The model was trained on this data, and improvements were made by adjusting the hyperparameters of the YOLOv8 model, which showed exceptional effectiveness in detecting images deepfakes.

2. Related Works

In this section, some related works to deep fake detection methods will be presented for the convenience of the reader: Ismail et al. (2021) proposed a study that employed YOLO-CNN-XGBOOST, with YOLO functioning[5] as a face detector for extracting faces from video frames, InceptionResNetV2 CNN[6] extracting features from these faces, and XGBoost [7] acting as a classifier to determine the authenticity of the videos. Their remarkable achievement includes an accuracy of 90.73% using the merged CelebDF-FaceForensics++ dataset, comprising 2,848 training and 518 test samples selected from CelebDF [8] and FaceForensics++ [9]. Despite the dataset's size not being sufficient for training deep neural networks, their innovative deepfake detection method, YOLO-CNN-XGBoost, demonstrated exceptional performance. It obtained an AUC of 90.62% and showcased high accuracy, specificity, sensitivity, recall, and F1-measure on the combined CelebDF-FaceForencics++ (c23) dataset [7].

Yasrab et al. (2021) proposed a deep fake detection method using multi-head LSTM model trained on a dataset of original videos of four US presidents and synthetic fake videos of the same presidents. Their approach based on overhead body language analysis and achieved an accuracy of 94.39% on the test set. The dataset was created using human pose estimation and videos. They used OpenCV and OpenPose DNN applications to extract the key points of the upper body in the frames of the videos. The results showed that upper body language can effectively detect fake videos. Limitations of the method include a small data set and the use of deepfake creation techniques that may be outdated[10].

Raza et al. (2022) proposed novel deepfake predictor named (DFP) based on a hybrid of VGG16[11] and convolutional neural in model architecture. The deepfake dataset used in this research included both real and fake faces, with a total of 1081 real and 960 fake images. These images were obtained from the Department of Computer Science at Yonsei University and used to train and test neural network techniques. Their proposed approach, achieved a 95% accuracy in detecting deepfakes [12].

Awotunde et al. (2022), proposed a deep learning model consists of five-layer of convolutional neural networks (CNNs) optimized with ReLU activation function to detect and classify DeepFake videos. The model was tested using the Face2Face and first-order DeepFake motion datasets, and obained an average prediction rate of 98% for DeepFake videos dataset [13] and 95% for Face2Face videos

dataset [14] . When compared to other CNN-based systems, their proposed model had the highest accuracy rate of 86%[15].

Bansal et al. (2023), proposed DFN (Deep Fake Network), a model architecture combining mobNet, separable convolution, max-pooling with Swish activation, and XGBoost [7] classifier. This model outperforms several state-of-the-art methods such as Xception[16] and Efficient-Net [17], achieving 93.28% and 91.03% accuracy on the DFDC [13] dataset. In addition, this sturdy and lightweight model detects many facial manipulations [18].

Pinhasov et al (2024) The study proposes a new approach to detect adversarial attacks on deepfake detectors based on a modified methodology that incorporates eXplainable Artificial Intelligence (XAI). The technique includes utilizing XA [19] to create interpretability maps for a specific method and utilizing a pre-trained feature extractor while training a basic yet high performing classifier. In the study, the researchers utilized the FF++ dataset [9] and applied the method in time frame not considering it. The effectiveness of the method was evaluated according to the accuracy it achieves when dealing with specific types of adversarial attacks, which were between 61% to 84.07%[20]. Table 1. Summarized the related works mentioned in this section.

Table 1. Summarization of related works

Study	Method	Dataset	Accuracy	Additional Metrics
Ismail et al .(2021)	YOLO-CNN-XGBoost	CelebDF-FaceForensics++ dataset, comprising 2,848 training and 518 test	90.73%	AUC:90.62%
Yasrab et al. (2021)	multi-head LSTM	original videos of four US presidents and synthetic fake videos of the same presidents	94.39%	_
Raza et al. (2022)	hybrid of VGG16 and convolutional neural	1081 real and 960 fake images	95%	_
Awotunde et al. (2022)	five-layer of (CNNs) optimized with ReLU	Face2Face and DeepFake motion datasets	98% for DeepFake 95% for Face2Face	Highest CNN-Based Accuracy :86%
Bansal et al. (2023)	DFN (Deep Fake Network)	DFDC Dataset	93.28%(DFDC)	-
Pinhasov et al (2024)	XAI-base adversarial attacks detection	FF++ Dataset	61% to 84.07%	Accuracy against adversarial attacks

3. Materials

The general aim of this research work is to classify the fake and real images using the deep learning model, YOLOv8. YOLOv8 has been adopted in our framework due to its advanced object detection capabilities, including higher accuracy and faster inference speeds compared to other models. Its architecture supports real-time deepfake detection with improvements in feature extraction, object localization, and a refined backbone network that enhances performance. YOLOv8's model has the ability to process complex images efficiently aligns with the research objectives of detecting subtle deepfake manipulations quickly and accurately. Yolov8 model was launched by Ultralytics on 10th

January, 2023. The YOLOv8 model offers several specific improvements in deepfake detection compared to earlier models like YOLOv3 or YOLOv4. Here are the key enhancements:YOLOv8 incorporates more advanced and optimized network architectures compared to YOLOv3 and YOLOv4. These improvements include better backbone networks and more efficient neck components, which enhance feature extraction and integration. YOLOv8 demonstrates superior accuracy in detecting deepfakes due to its refined architecture and advanced feature extraction techniques. This allows for real-time detection of deepfakes, which is crucial for practical applications.

In this research work, we have uutilized the Yolov8 model for fake detection and retrained the Yolov8 model on a custom dataset for solving fake detection issue. MTCNN method [21] has been applied first to detect the face region in the input images and yet annotate the datasets. MTCNN method uses an edge detection procedure to quickly annotate training samples. The following sub-sections explained how to configure the necessary components of the system and prepare the necessary data sets before putting the proposed system into practice.

A. System Requirements

On the practical side, deep learning models require high specifications in terms of processor speed and amount of available storage space. In addition, implementing the yolov8 model requires the use of specialized software, such as the Windows 10 operating system and the Python programming language. Additional software such as PyCharm, Anaconda, PyTorch with Torchvision, and Cuda,ultralytics are also needed. Table 2 shows the basic aspects of the experimental conditions that used to implement the proposed deep fake detection system.

Table 2: The main requirements of the proposed deep fake detection system

Resources	Requirements
Operating System	Windows 10 Pro x64-based PC
Programming Tool	- Python programming language (python=3.10.9) - Anaconda v1.11.2 - PyCharm Community
Anacnda libraries	- torch 1.11.0+cu113 - tensorflow 2.15.0 - torchvision 0.12.0+cu113 - mtcnn 0.1.1 - numpy 1.23.5 - opencv 4.5.3 - ultralytics

B. Description of Dataset Used

The composition and detailed description of the dataset play a crucial role in enhancing the effectiveness of the YOLOv8 model across different detection scenarios. The diversity of the data ensures that the model can detect forgery more accurately, and makes it more robust against various manipulation techniques, thus enhancing its generalization ability. Careful data processing also contributes to improving the model's performance, increasing detection accuracy, and reducing errors in real-world applications. In order to build an efficient deep fake detection system, dataset samples have to gathered and prepared first. We have collected and prepared a set of fake and original images of humans' faces to use in the model training, validation and testing phases. Custom dataset composed of 8,082 real and fake images was created and utilized in the proposed framework. The custom dataset samples have been sourced from Fake-vs-Real-Faces (Hard) [22] and Real-vs-Fake [23] datasets, which contained 4041 fake images and 4041 real images. The experiments were conducted using the

original characterization of the dataset, as well as a new images characterization based on augmentation techniqyes for the training and validation phases. The entire dataset was split into 80% for training and 20% for testing.

4. The Proposed Methodology

With the development of counterfeiting and image manipulation techniques, the need arises for a powerful technology to detect fake faces to monitor counterfeiting in real time application. The aim of this research is to distinguish the forger (fake) input image/frame sample from the original one (real) using deep learning model and custom dataset. The main target of deep fake detection task requires powerful digital technology to quickly and effectively detect the forgery in the digital images. We have adopted the pre-trained YOLOv8 model for detecting fake faces due to its significant progress in object detection task with higher accuracy and speed [22]. YOLOv8 aims to strike a balance between detection accuracy and computational efficiency. The main contributions of adopting YOLOv8 model for images deepfake detection could be summarized as follows::

- Accuracy vs. Speed: YOLOv8 enhances accuracy by employing advanced architectures and training techniques while maintaining high detection speed. This is achieved through optimized network layers and efficient computational processes.
- Model Complexity: YOLOv8's architecture includes sophisticated features like improved convolutional layers and attention mechanisms, which improve accuracy but also increase computational demands. The trade-off is managed by adjusting the model size and complexity based on hardware capabilities.
- Inference Time: YOLOv8 prioritizes fast inference by using efficient algorithms and minimizing the processing time per image.

In addition, YOLOv8 helps in fast training and testing data[20[21]]. The proposed framework includes several main phases including; dataset samples gathering, preprocessing dataset, annotation of training dataset samples, region of interest extraction (ROI), Dataset collection: Real and fake images were collected from two main datasets, Fake-vs-Real-Faces (Hard)[23] and Real-vs-Fake[24]. The data samples are then processed by standardizing the image sizes to be ready for training. After that, we used MTCNN to classify the images into fake and real categories. Finally, we extracted the region of interest (ROI) to identify and analyze the important facial regions.. The retraining process of the YOLOv8 deep learning model included several significant improvements to the hyperparameters, as follows:

- Learning rate tuning: Optimizing the learning rate to ensure efficient weight updates and accelerate the convergence process during training.
- Batch size optimization: Adjusting the batch size to achieve a balance between memory usage and training efficiency, which enhanced the model’s ability to generalize.
- Increasing the number of epochs: Increasing the number of epochs to enable the model to learn more deeply from the dataset, which improves its performance. The following sub=sections illustrate the phases of the proposed framework.

A. Preprocessing Phase

In image processing, feature extraction is a fundamental step. It is done by applying various image pre-processing techniques such as downscaling, resizing, and normalization to the captured image. Then, features that may be important in image classification and recognition are extracted using feature extraction techniques[25]. In this paper, set of images was collected, including both fake and real images. After collection, the training images were carefully read. Preprocessing phase is conducted in the proposed framework to prepare the dataset samples for training, validation and testing mods, which includes two main procedures: data augmentation and dataset annotation. To enhance the model and its accuracy, the samples of dataset were enlarged using augmentation techniques. Then, all the samples of datasets used were resized to a uniform size (224 × 224) to be suitable for model training. Then the

images were divide into two sets: a training set that used in the training mode, and testing set that used for model evaluation task.

B. Annotation Phase

The combination of YOLOv8 and MTCNNs enhances deepfake detection by combining the strengths of both models. MTCNNs are highly effective at accurately detecting and extracting facial regions, which is critical to focusing analysis on relevant parts of an image. Integrating MTCNN for face region extraction enhances YOLOv8's deepfake detection by precisely locating facial areas, which improves accuracy by focusing on relevant regions. It reduces false positives by filtering out non-face content, increases efficiency by processing smaller, face-focused images, and enhances feature extraction by targeting crucial facial features. This targeted approach allows YOLOv8 to detect subtle manipulations more effectively and operate faster By feeding the accurately extracted facial regions into YOLOv8, the detection process becomes more targeted, allowing YOLOv8 to use its advanced object detection capabilities to identify subtle features and inconsistencies in deepfake content. In this phase, the pre-processed training data samples were read. Then, the MTCNN method was applied to detect the face region in each training and testing sample. After the face region was identified, each region was classified into a fake or real class based on its characteristics. Each detected region was then annotated by adding a label that identifies whether the region contains a fake or real face. Finally, the annotated regions were converted to YOLOV8 format so that the model could process them and use them in the training process. Fig.1 show the workflow of the MTCNN method.

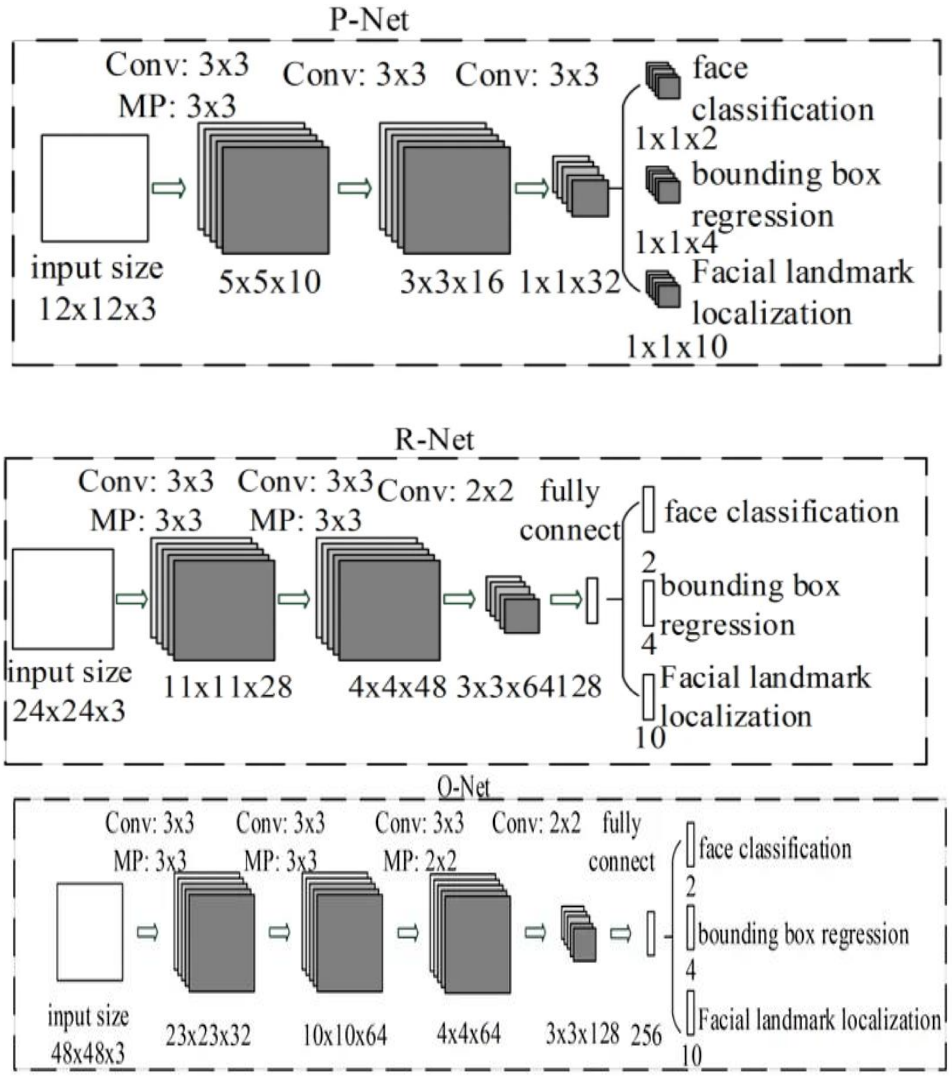


Figure 1: Workflow of the MTCNN model[21].

5. Fake and Real Features Extraction Using YOLOv8 Model

To perform the analysis of the extracted faces images and to feature the fake/real characteristics of each sample, YOLOv8 is used where the model is re-trained over the dataset employed in the present research work. This technique focuses on extracting features from the input image to detect objects and this is done in real-time hencefast and very effective in the detection of the fake and real imagess with less time as compared to other techniques. Using multiple images helps in identifying the head pose and different facial angles in each image, which contributes to the recognition of fake images. The model first takes the input image, and then splits it into a certain number of squares of a certain size called img1, where is chosen randomly. Every cell of this grid contains several boundary squares to predict such things as class probabilities, the presence of an object, and confidence scores. Non-Maximum Suppression (NMS) technique is applied in order to eliminate redundant squares and to localize objects properly. Each object is detected in NMS once and all the fake objects are deleted. For each boundary square, there is calculated a confidence score which is a measure of probability of the object belonging to one of the given classes which will only detect an object inside the square.. The boundary squares are generated by aggregating the underlying real squares from the dataset used. YOLOv8 architecture employed CNN convolutional neural network; it has a number of consecutive convolutional layers and two fully connected layers. This architecture plays an important role in making efficient feature extraction from the image, and good performance is observed in the detection processes.

6. Model Training Workflow

The proposed workflow is quite is intended to uncover fake images of people by processing images, with an emphasis on analysing the patterns and discrepancies that signal tampering. The training process involves using the YOLOv8 models and enhanced models that incorporate supervised learning to distinguish the images as genuine or fake. The model is tuned to make the least loss possible through accepting a customized and balanced data set. The Hyperparameters tuning process is conducted in the training mode, which includes; learning rate, number of epochs (12**0**), batch size (32), and optimizer method (SGD). We selected hyperparameters such as learning rate, batch size, and number of epochs based on optimizing the training process and model performance. The learning rate was adjusted to ensure efficient weight updates and convergence. The batch size was chosen to balance memory usage and training efficiency. The number of epochs was adjusted to allow sufficient training time for the model to learn from the data without overfitting. These parameters significantly affected the performance of the model by affecting the speed of convergence, generalization, and accuracy. The model training aims to improve the accuracy of the model and reduce response time to be suitable for real-time applications. The parameters were carefully tuned to select the best settings, as shown in Table 3.

Table 3. New setting of Yolov8 Hyperparameters

Hyperparameter type	Setting
sample resolution	224*****224
Batch size	32
Optimizer used	SGD
learning rate	0.0001
Activation function used	SiLU
Filter size	(3*3) for all layers
Epoch No.	120
Classes No.	2

The model has been trained for 120 epochs/iterations and the model has been used to predict new outputs on the validation set and then checked for performance on the test data set using the said

standard evaluation metrics. The optimizer is used to update the model parameters with optimal weights in order to minimize the losses measured by the loss function at a given epoch. The loss function calculates the amount of error at each epoch. The main steps of the model training workflow are described in the algorithm 1.

Hyperparameter Tuning

Hyperparameters in deep learning models include values that are set before training starts and do not change during training. These parameters are vital to the performance of the model, as incorrect settings lead to substandard results. In this research work, we have used a pre-trained YOLOv8 model and tuned hyperparameters such as learning rate (0.0001), batch size (32), momentum (SGD), and number of epochs (120). We have adopted the random search technique [26] to identify the best settings for the hyperparameters, focusing on improving accuracy and performance. This method can be more efficient than exhaustive search because it explores a broader space of possibilities with less computational cost. Compared to other methods like grid search, random search often yields better performance by avoiding overfitting and finding optimal values faster, leading to improve accuracy and generalization of the model. Performance is evaluated through metrics such as accuracy and mean average precision, and the stopping state is determined by the number of epochs selected.

Algorithm 1: Model Training Workflow
Input: Import Yolov8 model, custom dataset samples
Output: Trained model (output weights file)
Start:
Step1. Assign dataset to custom dataset
Step 2. Initialize weights
- Use original weights file as initial step to initialize weights
Step 3. Model's Hyperparameters setting according to Table 2
Step 4. Define batch number
- Batch number = number of dataset samples / batch size
Step 5. Training model (for all epochs):
- EpNo = EpNo + 1
- BaNo = 0
- For all batches:
- BaNo = BaNo + 1
- Obtain batch-sized image samples
- Pass current batch through model layers
- Record the outputs of the selected squares for the detected faces and their associated classes.
- Compute the loss between the outputs and the original results according to the annotation files of these samples.
- Apply the optimization method
- While (BaNo < batch number)
- Evaluate the model on a valid set using the tuned hyperparameters
Step 6. Return the weights file
End

7. Experimental Results

In this section, we present and analyze the performance of the Yolov8 model. The Fake-vs-Real-Faces (Hard) and Real-vs-Fake dataset have been utilized to achieve the performance evaluation process. We have selected 8082 image samples from each group, to ensure that the real and fake samples are balanced in the training and validation sets. These sets have a variety of human facial features; including fake faces generated using the StyleGAN2 method, making it more difficult to accurately distinguish. The real samples were collected to represent diverse characteristics such as age, gender, composition, and ethnicity. Standardized evaluation metrics have beem utilized as well as confusion matrix to evaluate analyze the model's performance. The confusion matrix summarizes the prediction results in a table showing:

- True Positive (TP): Cases that the model correctly classified as positive.
- False Positive (FP): Negative cases that the model incorrectly classified as positive.
- False Negative (FN): Positive cases that the model incorrectly classified as negative.
- True Negative (TN): Negative cases that the model correctly classified as negative.

Performance Evaluation Metrics

The performance of the YOLOv8 model in detecting deepfakes is evaluated using several standard metrics for object detection tasks[27], such as:

Intersection over Union (IoU): It evaluates the accuracy of predictions by determining the extent of overlap between the predicted and actual bounded squares. IoU is calculated using the eq. (1):

$$IoU = \frac{\text{Intersection area}}{\text{Union area}} \quad (1)$$

Predictions are classified as true positives (TP) or false positives (FP) if IoU is greater than or equal to 0.5.

Accuracy: Indicates the overall correctness of the model's predictions:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (2)$$

Precision: Measures the proportion of true positive detections out of all positive detections:

$$Precision = \frac{TP}{TP+TN} \quad (3)$$

Recall: Assesses the proportion of true positives detected out of all actual positives:

$$Recall = \frac{TP}{TP+FN} \quad (4)$$

F1 score: Combines precision and recall into a single metric by their harmonic mean:

$$F1 - score = \frac{2 * (\text{Precision} * \text{Recall})}{(\text{Precision} + \text{Recall})} \quad (5)$$

Average precision: Computes the precision across different recall levels, integrating the precision-recall curve:

$$\text{AP} = \sum_{k=0}^{k=n-1} [\text{Recall(k)} - \text{Recall(k} + 1)] * \text{Precision(k)} \quad (6)$$

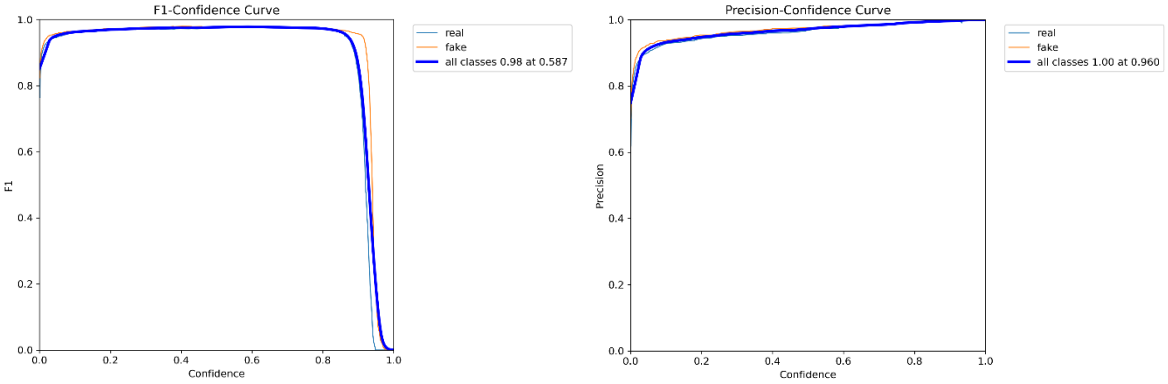
Mean average precision (mAP): Averages the AP over all classes to provide an overall measure of model performance:

$$\text{mAP} = \frac{1}{n} \sum_{k=1}^{k=n} \text{AP}k \quad (7)$$

These metrics demonstrate the effectiveness of the YOLOv8 model by providing a comprehensive assessment of its detection capabilities, accuracy, and ability to handle false positives and negatives..

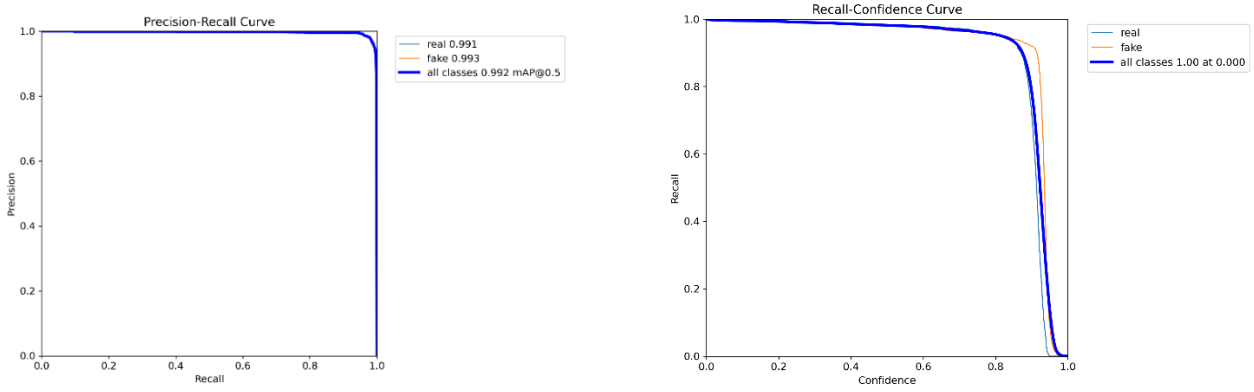
8. Experiment Implementation

The model assessment process has been conducted and implemented on the test data, which represents 20% of the total data including fake and real images. For each test sample, the inference mode is activated to detect human face (localization) first in the input image and then classify as real or fake image in term of binary classification. The performance evaluation of the model is specified through using standard evaluation metrics as precision, recall, F1 score, and mAP. In this experiment, the performance of the optimized Yolov**8** model based on hyperparameter setting is presented. The pre-processed and annotated Fake-vs-Real-Faces (Hard) and Real-vs-Fake dataset consists of 8082 images (50% fake and 50% real). The training time of this experiment was 3873 hours.The hyperparameter formation process adopted in this experiment on the sample dataset is depicted in Table (2). The visualization results obtained from these experiments are; Accuracy = 97.5%, Precision = 97.03%, Recall = 98%, mAp = 99% as shown in Figures (2-3).Through continuous training, the batch size was set to 32, depending on the image resolution and GPU memory. The training time ends after the number of epochs is completed.



(a) (b)

Figure 2: Results of F1-score, Precision metric obtained by implementing optimized yolov8 network.



(a) (b)

Figure 3: Results of (Precision -Recall), Recall metric obtained by implementing optimized yolov8 network.

The training and validation loss values are crucial measurements due to their deeper understanding of the learning performance dynamics in correlation to the number of epochs, and one can identify some of the issues with learning that result in underfitting or overfitting of the model. An illustration of The training process is monitored using tools like Tensorboard, which depict training measures such as loss function(LF) over time as shown in fig.4.

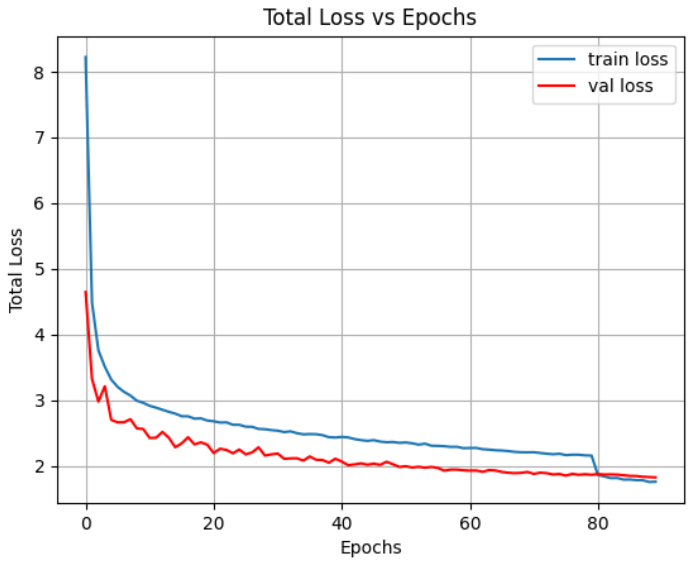


Figure 4: Simulation results in the training mode based on train/valid loss measures of the trained model of the optimized yolov8 networks.

The detection results of (real/fake) classes in the dataset samples meanwhile training and testing model are depicted in fig.5.

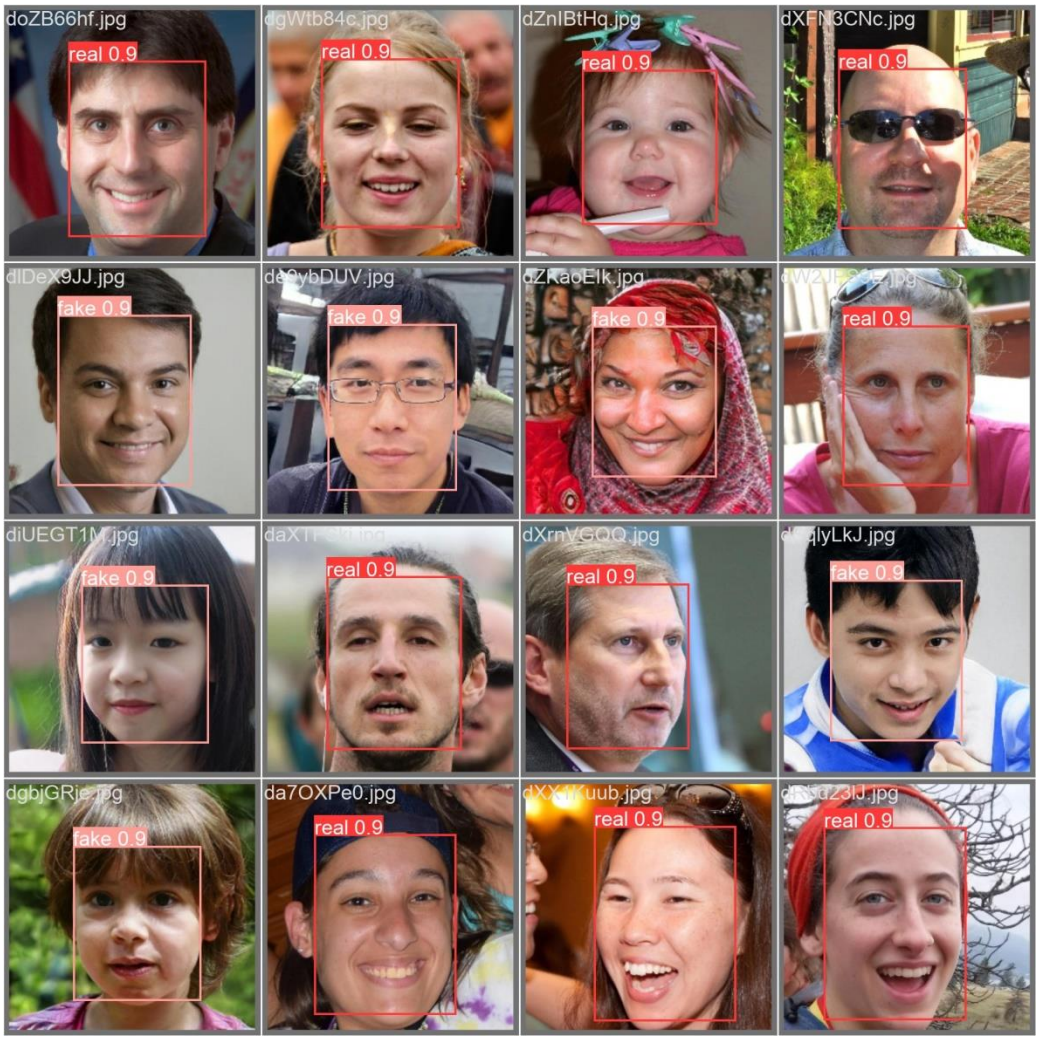


Figure 5: Detection results of yolov8 model for dataset samples in the training mode

9. Comparison Study

To evaluate the effectiveness of the proposed YOLOv8-based deepfake detection system, we compared its performance with other state-of-the-art methods. The comparison, as illustrated in Table 4, highlights the following points: **YOLOv8.** The YOLOv8 model achieved an impressive accuracy of 97.5% when trained on a customized and balanced dataset over 120 epochs. The high accuracy measurement is attributed to enhancements in hyperparameter tuning and data expansion techniques, which effectively addressed overfitting issues.

Table 4. The outcomes of comparative study

Reference	Dataset	Technique	Accuracy
[7]	CelebDF-FaceForensics++ dataset	YOLO-CNN-XGBOOST with YOLO, InceptionResNetV2 CNN, and XGBoost	90.73%
[10]	Dataset of original videos of four US presidents and synthetic fake videos	Multi-head LSTM model	94.39%
[12]	Deepfake dataset with 1081 real and 960 fake images	(VGG16 ,CNN)	95%
[15]	Face2Face and first-order DeepFake motion datasets	Five-layer CNNs optimized with ReLU	86%
[4]	FaceForensics++, Celeb-DF (V2), WildDeepfake	DFDT	FaceForensics++: 99.41%, Celeb-DF (V2): 99.31%, WildDeepfake: 81.35%
[18]	DFDC dataset	DFN with mobNet, separable convolution, max-pooling with Swish activation, and XGBoost classifier	93.28%
[20]	FaceForensics++	(XAI)	Ranging (61-84.07%)
Proposed method	Fake-Vs-Real-Faces (Hard), 140k real-vs-fake	Yolov8	Optimized Yolov8= 97.5

The main advantages of YOLOv8 represented by its outperforms compared to existing methods, especially in terms of accuracy, with notable improvements over methods like YOLO-CNN-

XGBOOST[7], Multi-head LSTM [10], and even VGG16-based models [12]. The YOLOv8 model's higher accuracy is largely due to its refined hyperparameters and data handling improvements, making it a robust solution for deepfake detection compared to other contemporary approaches. YOLOv8 often achieves higher accuracy in detecting deepfakes due to its advanced architecture and optimized hyperparameters. In additiom, YOLOv8 provides faster detection and processing times, making it suitable for real-time applications. YOLOv8 handles larger and more diverse datasets effectively, offering better generalization in term of model scalability. In contrast, models like YOLO-CNN-XGBOOST or Multi-head LSTM may have limitations in speed, scalability, or accuracy, making YOLOv8 a more robust choice for comprehensive deepfake detection.

10. Conclusions

In this paper, the YOLOv8 model is optimized based on hyperparameters tuning process to predict fake media images based on extracted face regions using the MTCNN method. The proposed framework focuses on improving the model performance through hyperparameter tuning, achieving a detection accuracy of 97.5%, an F-score of 97.51%. The results exhibited that the improved YOLOv8 outperforms in terms of speed and efficiency, and ensures excellent performance in dealing with faces of different sizes. YOLOv8-based system demonstrates superior performance in detecting deepfakes, as evidenced by its high accuracy in comparison to existing methods. Variations in dataset quality and size across different studies can significantly impact model performance. YOLOv8's superior performance can be attributed to its ability to handle high-quality, well-annotated datasets, which may lead to more accurate and reliable results compared to models trained on lower-quality data. The YOLOv8's performance improvements are also influenced by the large, balanced dataset used in the training mode, which provides a more comprehensive representation of both real and fake images., which often enhanced model's generalization and robustness. The use of data augmentation techniques in YOLOv8's training process helps to mitigate issues related to dataset size and diversity, contributing to its improved performance. However, the performance may degrade when dealing with complex backgrounds and low contrast. The future work of this research could be expanded to employ more diverse and balanced data samples to test the model in different conditions, and suggest designing and implementing a lightweight deep learning model based on compression techniques as model Pruning and quantization techniques for deepfake detection system. Further, attention layers could be adopted in model structure to minimize the impact of irrelevant background of frames.

References

[1] P. Korshunov and S. Marcel, "DeepFakes: a New Threat to Face Recognition? Assessment and Detection," pp. 1–5, 2018, [Online]. Available: http://arxiv.org/abs/1812.08685

[2] Z. H. Rasool, M. Adham, and A. Amir, "Comprehensive Image Classification using Hybrid CNN-LSTM Model with Advanced Feature Extraction on Coco Dataset," MJPAS ,vol. 2, no. 2, pp. 28–47, 2024.

[3] S. Negi, M. Jayachandran, and S. Upadhyay, "Deep fake : An Understanding of Fake Images and Videos," *Int. J. Sci. Res. Comput. Sci. Eng. Inf. Technol.*, vol. 3307, pp. 183–189, 2021, doi: 10.32628/cseit217334.

[4] A. Khormali and J. S. Yuan, "DFDT: An End-to-End DeepFake Detection Framework Using Vision Transformer," *Appl. Sci.*, vol. 12, no. 6, Mar. 2022, doi: 10.3390/app12062953.

[5] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection." [Online]. Available: http://pjreddie.com/yolo/

[6] M. Längkvist, L. Karlsson, and A. Loutfi, "Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning," *Pattern Recognit. Lett.*, vol. 42, no. 1, pp. 11–24, 2014, [Online]. Available: http://arxiv.org/abs/1512.00567

[7] A. Ismail, M. Elpeltagy, M. S. Zaki, and K. Eldahshan, "A new deep learning-based methodology for video deepfake detection using xgboost," *Sensors*, vol. 21, no. 16, Aug. 2021,

doi: 10.3390/s21165413.
[8] Y. Li, X. Yang, P. Sun, H. Qi, and S. Lyu, "Celeb-DF: A Large-Scale Challenging Dataset for DeepFake Forensics," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, pp. 3204–3213, 2020, doi: 10.1109/CVPR42600.2020.00327.
[9] A. Rössler, D. Cozzolino, L. Verdoliva, C. Riess, J. Thies, and M. Nießner, "FaceForensics: A Large-scale Video Dataset for Forgery Detection in Human Faces," 2018, [Online]. Available: http://arxiv.org/abs/1803.09179
[10] R. Yasrab, W. Jiang, and A. Riaz, "Fighting Deepfakes Using Body Language Analysis," *Forecasting*, vol. 3, no. 2, pp. 303–321, Jun. 2021, doi: 10.3390/forecast3020020.
[11] Z. P. Jiang, Y. Y. Liu, Z. E. Shao, and K. W. Huang, "An improved VGG16 model for pneumonia image classification," *Appl. Sci.*, vol. 11, no. 23, 2021, doi: 10.3390/app112311185.
[12] A. Raza, K. Munir, and M. Almutairi, "A Novel Deep Learning Approach for Deepfake Image Detection," *Appl. Sci.*, vol. 12, no. 19, Oct. 2022, doi: 10.3390/app12199820.
[13] B. Dolhansky *et al.*, "The DeepFake Detection Challenge (DFDC) Dataset," 2020, [Online]. Available: http://arxiv.org/abs/2006.07397
[14] J. Thies, M. Zollhöfer, M. Stamminger, C. Theobalt, and M. Nießner, "Face2Face," *Commun. ACM*, vol. 62, no. 1, pp. 96–104, 2018, doi: 10.1145/3292039.
[15] J. B. Awotunde, R. G. Jimoh, A. L. Imoize, A. T. Abdulrazaq, C. T. Li, and C. C. Lee, "An Enhanced Deep Learning-Based DeepFake Video Detection and Classification System," *Electron.*, vol. 12, no. 1, Jan. 2023, doi: 10.3390/electronics12010087.
[16] K. R. Avery *et al.*, "Fatigue Behavior of Stainless Steel Sheet Specimens at Extremely High Temperatures," *SAE Int. J. Mater. Manuf.*, vol. 7, no. 3, pp. 560–566, 2014, doi: 10.4271/2014-01-0975.
[17] M. Louis, "20:21," *Can. J. Emerg. Med.*, vol. 15, no. 3, p. 190, 2013, doi: 10.2310/8000.2013.131108.
[18] N. Bansal *et al.*, "Real-Time Advanced Computational Intelligence for Deep Fake Video Detection," *Appl. Sci.*, vol. 13, no. 5, Mar. 2023, doi: 10.3390/app13053095.
[19] A. Barredo Arrieta *et al.*, "Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI," *Inf. Fusion*, vol. 58, pp. 82–115, 2020, doi: 10.1016/j.inffus.2019.12.012.
[20] B. Pinhasov, R. Lapid, R. Ohayon, M. Sipper, and Y. Aperstein, "XAI-Based Detection of Adversarial Attacks on Deepfake Detectors," *arXiv*, pp. 1–20, 2024, [Online]. Available: http://arxiv.org/abs/2403.02955
[21] R. Jin, H. Li, J. Pan, W. Ma, and J. Lin, "Face Recognition Based on MTCNN and FaceNet," 2021, [Online]. Available: www.aaai.org
[22] J. Terven, D. M. Córdova-Esparza, and J. A. Romero-González, "A Comprehensive Review of YOLO Architectures in Computer Vision: From YOLOv1 to YOLOv8 and YOLO-NAS," *Mach. Learn. Knowl. Extr.*, vol. 5, no. 4, pp. 1680–1716, 2023, doi: 10.3390/make5040083.
[23] "Fake-Vs-Real-Faces (Hard)." https://www.kaggle.com/datasets/hamzaboulahia/hardfakevsrealfaces
[24] "140k Real and Fake Faces | Kaggle." https://www.kaggle.com/xhlulu/140k-real-and-fake-faces (accessed May 30, 2021).
[25] B. A. Mohammed and Z. M. Abood, "Performance Evolution Ear Biometrics Based on Features from Accelerated Segment Test," MJPAS ,vol. 1, no. 3, pp. 71–84, 2023.
[26] Francisco J. Solis and Roger J-B. Wets, "Minimization by Random Search Techniques ," vol. 6, no. 1, pp. 19–30, 2010.
[27] D. Bowes, T. Hall, and D. Gray, "Comparing the performance of fault prediction models which report multiple performance measures: Recomputing the confusion matrix," *ACM Int. Conf. Proceeding Ser.*, pp. 109–118, 2012, doi: 10.1145/2365324.2365338.