Journal of Al-Qadisiyah for Computer Science and Mathematics Vol. 15(1) 2023 , pp Comp. 137–149

JQCM University Of AL-Qadisiyah

Available online at www.qu.edu.iq/journalcm

JOURNAL OF AL-QADISIYAH FOR COMPUTER SCIENCE AND MATHEMATICS

ISSN:2521-3504(online) ISSN:2074-0204(print)

Deploying Facial Segmentation Landmarks for Deepfake Detection

Mohammed Thajeel Abdullah[a,*], Nada Hussein M. Ali[b]

[a] Informatics Institute for Postgraduate Studies, Iraqi Commission for Computer and Informatics, Baghdad, Iraq. Email: phd202010564@iips.icci.edu.iq

[b] Department of Computer Science, University of Baghdad, Baghdad, Iraq. Email: nada.husn@sc.uobaghdad.edu.iq

ARTICLE INFO

Article history:
Received: 07/01/2023
Rrevised form: 13/02/2023
Accepted : 20/02/2023
Available online: 31/03/2023

Keywords:

Deepfake detection;

Digital image forensics;

Deep learning;

CNN;

DenseNet121;

Face segmentation;

FFHQ dataset

ABSTRACT

Deepfake is a type of artificial intelligence used to create convincing images, audio, and video hoaxes and it concerns celebrities and everyone because they are easy to manufacture. Deepfake are hard to recognize by people and current approaches, especially high-quality ones. As a defense against Deepfake techniques, various methods to detect Deepfake in images have been suggested. Most of them had limitations, like only working with one face in an image. The face has to be facing forward, with both eyes and the mouth open, depending on what part of the face they worked on. Other than that, a few focus on the impact of pre-processing steps on the detection accuracy of the models. This paper introduces a framework design focused on this aspect of the Deepfake detection task and proposes pre-processing steps to improve accuracy and close the gap between training and validation results with simple operations. Additionally, it differed from others by dealing with the positions of the face in various directions within the image, distinguishing the concerned face in an image containing multiple faces, and segmentation the face using facial landmarks points. All these were done using face detection, face box attributes, facial landmarks, and key points from the MediaPipe tool with the pre-trained model (DenseNet121). Lastly, the proposed model was evaluated using Deepfake Detection Challenge datasets, and after training for a few epochs, it achieved an accuracy of 97% in detecting the Deepfake.

MSC..

https://doi.org/ 10.29304/jqcm.2023.15.1.1161

1. Introduction

Combining deep learning with computer vision has led to the development of very advanced technologies like Generative Adversarial Networks (GAN) [1], Style-Based GAN (StyleGAN) [2], etc. These techniques work to superimpose the face of the target person on the image or video of the source person, creating a so-called "Deepfake" [3]. Deepfake, a combination of the words "deep learning" and "fake," lets attackers or even people who

∗ Corresponding author

Email addresses: phd202010564@iips.icci.edu.iq

Communicated by 'sub etitor'

do not know much about machine learning manipulate a picture or video and make a new one that humans and computers cannot tell apart. Other than that, people's faith in digital media content is diminished due to Deepfake since they can no longer accept the visuals they are viewing. Research on recognizing or detecting faked media is regarded as classical research in the absence of deep learning. Hence, research into Deepfake detection is both important and urgent shall people are to keep trusting digital multimedia [4].

Deepfake detection as a technique represents the other side of Deepfake that works to discover Deepfake. It is typically considered a binary classification issue, where input video or image has to be flagged as either manipulated or not. Note that there are numerous methods proposed to detect Deepfake, some of which rely on the use of pre-trained models like Xception, Inception, EfficientNet, etc., because of their ability to extract manipulation features. Others started building their own convolutional neural network (CNN), such as [5], [6], etc., because they were interested in working on some of the specific features in the image. In this paper, the pre-trained model DenseNet121[7] is used as the input layer to extract features map from images, followed by a few layers to detect whether it is fake or not.

Other than that, deep learning algorithms need to be trained on a massive dataset to give excellent and reliable results [8]. For that, the first challenge the researchers face in Deepfake detection in images is that there are a few numerate available datasets, and some are either too small or of low quality. For this work, the Flickr-Faces-Hight Quality (FFHQ) dataset and the 1-Million Fake Faces (1MFF) dataset are used as a benchmark to show our proposed method's ability to detect tampered faces.

The system performance can improve depending on pre-process steps [9]. In the Deepfake detection, a few works highlight the functionality of pre-processing [12] and how it could contribute to improving the results of training in detecting Deepfake. Some of their pre-processing are based on segmenting the image into different parts [6] or randomly picking different parts of the face to break down into different shapes [10]. Hence, in this pre-processing, the face must be in front of the image. If you do not, parts of the image background will be added to the model as properties of the face. This will make the model noisier, hurting how well it works. Another pre-processing that is based only on the eyes [11], mouth, or teeth [5] as a feature will also have many limitations, like the face being in front and the eyes being open or the mouth being open and the teeth being visible and so on. Thus, it is necessary to intensify efforts to demonstrate the importance and effectiveness of pre-processing by avoiding these limitations to increase the network results. This research used face segmentation using facial landmark points to show this step's ability and energy to improve model performance by reducing the gap between training and validation accuracy and how it could increase testing accuracy.

According to this research experimenting with various face detector tools, such as Multitask Cascaded CNN (MTCNN) [12], Dlib [13], and Stasm [14]. The researchers noticed that they could not select a specific area of the face with high flexibility; due to the limited number of facial landmark points and their contents, which do not exceed 78 points in all except for the MediaPipe tool, having 468 points distributed all over the face. Fig. 1 shows the ability of this tool in face segmentation using facial landmarks.

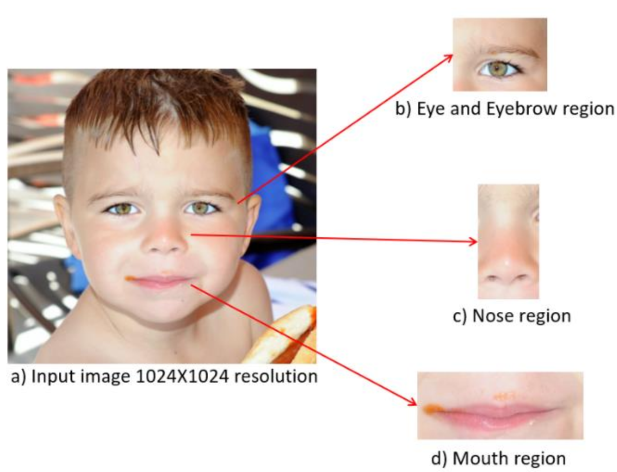


Fig. 1- Example of face segmentation process used MediaPipe Face mesh landmarks library.

The contributions of this paper are:

1) Applied the MediaPipe [15] libraries, such as Facial landmarks, Key points, and Box, to detect and crop the face from an image with single and multiple faces by identifying the posed and concerned faces.

2) Explored different feature areas (eye and eyebrow, mouth, and nose) of the face in an image to show effectiveness in Deepfake detection.

3) Showed that the model's performance could be improved with simple pre-processing by selecting a specific part of the face in high-quality images instead of the whole image. This study objective was achieved using the same proposed model in three separate cases the whole image, the face only, and the eye and eyebrow side.

The rest of the paper is organized as follows. Section 2 presents a review of the Deepfake detection method, and Section 3 displays the dataset description, which depended on this paper as evaluated benchmarks. Other than that, the proposed methodology is explained in detail in Section 4, and Section 5 is about the experimental setup. The experiment results and discussion are in Section 6, and the conclusion is in Section 7.

2. Related Works

In the last few years, the remarkable progress and development in Deepfake algorithms have produced images and videos that are difficult to distinguish by the human eye and even by software. On the other hand, artificial intelligence (AI) researchers have sought to improve and develop powerful methods that would identify and detect Deepfake in visual media. In AI Index Report 2022 by Stanford University [3], they present tracking progress over time in Deepfake detection, which shows the ability and effectiveness of AI systems to detect Deepfake and how it has evolved from a detection accuracy of 69.9% in 2012 to accuracy of 97.7% in 2021. This section presents the state-of-the-art related to this paper by proposed or developed methods in the pre-processing step of Deepfake detection.

Apart from that, pre-processing is an essential process in machine learning and can include many techniques such as face detection, augmentation, segmentation, resizing, normalization, etc. The face detection process is the first step in Deepfake detection methods [10], which enables the AI system to determine whether the image includes a human face. However, many tools serve this technique, like RetinaFace [16], OpenFace2 [17], Multitask Cascaded CNN (MTCNN), MediaPipe, etc. Note that the MediaPipe tool has achieved several successes in many studies, such as a virtual intelligent classroom system [9], which was found a more stable in improving system performance through face detection and hand gestures. Correspondingly, in human emotions detection [18], they proposed a new approach to determine the emotions of the human face in real-time to use in robot vision applications using this tool. In [19], they combined three methods: neural network, Dlib, and MediaPipe, to achieve good results in detected Hand-over-Face. This study is considered the first to deploy the MediaPipe tool as a pre-processing to detect, crop, and segment the face in the Deepfake detection technique for images.

As it is known, augmentation technology is used when training and testing data are few. Still, it can be used in learning to extract features of images in neural networks to improve their performance [20]. In [10], the researchers propose a new augmentation method called Face-Cutout. It has been used to tackle the over-fitting problem in training CNN and improve Deepfake detection by expanding data volume by eliminating one portion of the face from images based on facial landmark points or producing new pictures by injecting Gaussian Noise and Flipping to reach a good result. In another study [21], Gaussian Noise/ISO Noise, horizontal flipping, etc., were used to add new images to the data. Note that it was utilized with three different networks that were designed based on the pre-trained model to improve the accuracy of the CNN. Conflicting with the state-of-the-art, the proposed framework in this research only used a part of the high-quality image to achieve better results than some and very close to the others.

Another technique that could be implemented as a pre-processing step is image segmentation [22], which is a crucial procedure for identifying and discovering objects in images to extract relevant features that depend on the accuracy and quality of the segmentation process [23]. In their method [9], the authors divided the face after its identification into four blocks, and each block was tested alone to detect tampered using CNN. Finally, the voting process was used according to the number of fake parts to determine whether the whole image was fake or not. The main limitation of this study and most of those in this section dealt with a single face in a front direction, but our proposed model has developed solutions for these limitations.

For example, Symeon et al. [24] proposed an approach that worked on pre-process phase to address the false positive of face detection in the video by employing MTCNN to extract face detection regions and cluster these areas according to their size to recognize incorrect detections of the face. Other than that, they presented good results with the Celeb-DF and FF++ datasets utilizing the three pre-train models: MesoInception4, XcepctionNet, and EfficientNet-B4. In [25]. They proposed a face recognition and fake detection method; in pre-process step, the Kalman filter was used to remove the nose with normalization. For dimensionality reduction, they applied a fusion of Fisherface with Local Binary Pattern Histogram (FF-LBPH). In Automatic Face Weighting (AFW) [25], their system used MTCNN to find faces, EfficientNet to extract features, the AFW layer and the Gated Recurrent Unit (GRU) to decide if the video is real or not. In detecting face swapping [26], they suggested a method of seeing swapped faces by comparing faces and their context. First, they used the double shot face detector (DSFD) to find and divide the video's faces. Second, they trained four classifiers based on the architecture of Xception to obtain good accuracy.

3. Dataset Description

For the Deepfake detection research, most of the available datasets are for videos, and there are a few datasets of images that were either too small or of low quality. Therefore, the Deepfake Detection Challenge dataset was selected for this study after ensuring that it is of reasonable quality and contains a sufficient number of images.

For authentic photos, a 70k real faces dataset that was retrieved from Flickr by Nvidia for their StyleGAN paper [2] was used. Note that it includes 70,000 PNG images with 1024 X 1024 pixels with a large variety in terms of gender, age, ethnicity, and image background. It also comprises accessories such as eyeglasses, sunglasses, caps, etc. Other than that, 1-Million Fake Faces were created using Nvidia StyleGAN comprising JPG images with 1024 X 1024 resolution were used to fake photos.

After downloading the dataset, unify each image set into one folder. Lastly, 6000 images were collected randomly from it to use as a benchmark for evaluating the proposed model in three scenarios. Upon it, 3000 pictures were selected from the dataset Flickr-Faces-Hight Quality (FFHQ) and 3000 images from the dataset 1- Million Fake Faces(1-MFF).

4. Proposed Methodology

This paper proposed a Deepfake detection framework for images based on deep learning techniques and face segmentations pre-process; see Fig. 2. The main goal of this method is to show how the facial landmarks with simple pre-processing steps on high-quality images obtained good accuracy, as well as how utilizing the appropriate tool and choosing the appropriate part of the face within the image leads to improving the performance of the model and increasing the Deepfake detection of its results. The following is an explanation of the main steps in this framework.

4.1. Face detection process

In Deepfake detection methods, face detection is the first step [27], [28], for that, it is important to use an excellent technique to determine whether the image contains a face or not [29]. The face detection process in this study was performed using the deep MediaPipe framework [15]. Note that the MediaPipe tool is a super-fast solution to detect face features, and it comes with six landmarks and multi-face support. It depends on BlazeFace [30], a lightweight and well-performing face detector fitted for mobile GPU inference. Relying on the face detection library in the MediaPipe alone is not enough, as some photos contain several faces. For that, the Box score attribute was used see Fig. 3 for all images. To determine whether the face is well clear and represents the concerned face in the picture, see Fig. 3 image (a). Lastly, it was used to crop the person's face from the image. After practical experience and studying at this library, a way to pick the most prominent face in an image could be found.

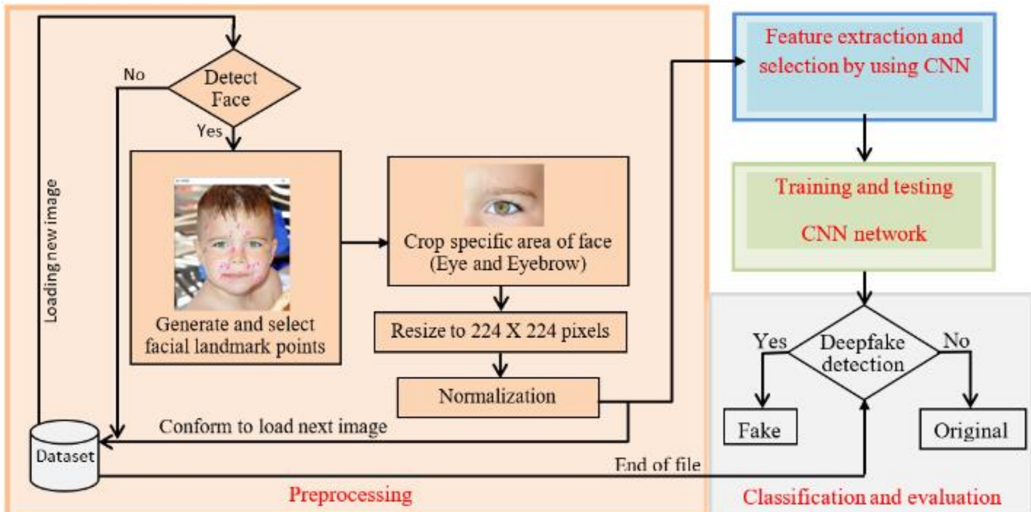
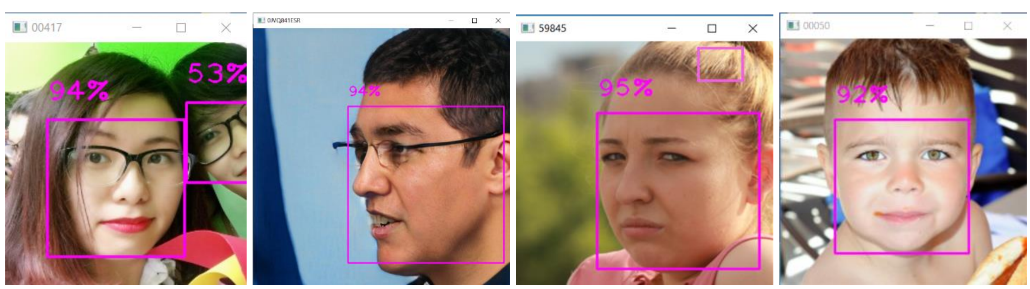


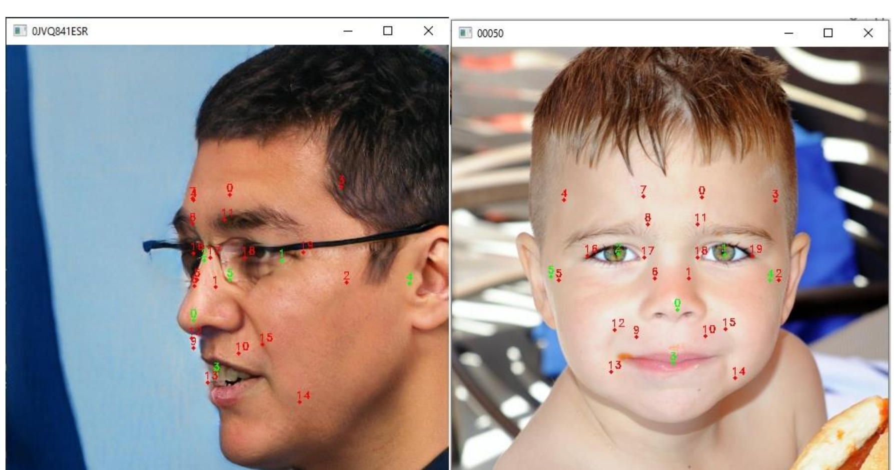
Fig. 2- The proposed method framework.

This was done utilizing the Box score attribute with a threshold of 75% and two points from the six key points (right eye, left eye, nose tip, mouth center, right ear region, and left ear region) that are all colored green in Fig. 4. By measuring the distance between the mouth center point and the nose tip point, it has reached the threshold at less than 27 as a distance by Eq. (1) to help to distinguish the prominent face from the others. Other than that, it was utilized in the face detection library, Box score attribute, and key points from the MediaPipe tool to determine the concerned face in an image containing multiple faces. This is the first point of difference of this study from the rest of the state-of-the-art.



(a) Real faces (b) Fake face (c) Real face (d) Real face

Fig. 3- Examples of MediaPipe Face detection results with Box score attribute: (a) Right multiple faces detections; (b) and (d) Right one face detection; (c) Mis detects multiple faces.



a) Fake face b) Real face

Fig. 4- Examples of MediaPipe Face Mesh and Key points generated: (a) Wrong generate for some of Face Mesh and Key points; (b) Right generate of Face Mesh

4.2. Face segmentation process

The MediaPipe face mesh is a solution that appreciates 468 3D face landmarks in real-time, even on mobile devices. It uses machine learning to infer 3D surface geometry using a single camera input without a specialized depth sensor [30]. This paper uses this library to segment the face into three areas depending on facial landmark points. After studying and experimenting with the all-facial landmark points in this library (468 points), most were founded move. Apart from that, their position changed according to the person's facial expressions, which also was confirmed by the study [18]. The determined points are the most stable, surrounding a rectangle of the eye and eyebrows, nose, and mouth. Note that there are 20 points; see Table 1 displays select key landmarks and the corresponding MediaPipe landmarks IDs; 16 of them (0-15) were used to crop these areas to deal with each one as a new image; refer to Fig. 4 for both images; the points with red color. The reset points (16-19) in this table were used to measure each eye's width. After generating and selecting facial landmark points, Claudine's law of distance, Eq. (1) shown below was employed to calculate the width of each eye according to two points for each one (left and right cornel), depending on the MediaPipe Face Mesh library. After selecting the longer eye crop, the area (eye and eyebrow) is relevant.

$$\text{Distance}= \sqrt{(x2-x1)^2+(y2-y1)^2} \qquad (1)$$

Using this suggested method in this paper enabled us to distinguish the posed face to define and crop the eye and eyebrow area for one side of the face, showing that it is sufficient to use and gave a good result. Other than that, the rest of the parts of the face that the researchers worked on, which are the areas of the nose and mouth, are easily detected depending on the facial landmark points for each of them, regardless of whether the face is a posed or front. This method represents this study's second point of difference from the rest of the state-of-the-art that only deals with the front face.

Table 1- Selected key landmarks and the corresponding MediaPipe landmarks.

Key landmark ID	MediaPipe landmark ID	Description
0	151	Left eye and eyebrow (point of the upper left corner)
1	399	Left eye and eyebrow (point of the lower left corner)
2	345	Left eye and eyebrow (point of the upper right corner)
3	284	Left eye and eyebrow (point of the lower right corner)
4	54	Right eye and eyebrow (point of the upper left corner)
5	116	Right eye and eyebrow (point of the lower left corner)
6	174	Right eye and eyebrow (point of the upper right corner)
7	151	Right eye and eyebrow (point of the lower right corner)
8	107	Nose (point of the upper left corner)
9	165	Nose (point of the lower left corner)
10	391	Nose (point of the upper right corner)
11	336	Nose (point of the lower right corner)
12	206	Mouth (point of the upper left corner)
13	202	Mouth (point of the lower left corner)
14	430	Mouth (point of the upper right corner)
15	426	Mouth (point of the lower right corner)
16	463	Left eye (point of the right corner)
17	359	Left eye (point of the left corner)
18	130	Right eye (point of the right corner)
19	243	Right eye (point of the left corner)

4.3. Three Pre-processing Scenarios

This study used more than one pre-processing since the same proposed model was trained and tested on more than one scenario for the same dataset. This illustrates the effectiveness of using the specific part of the face instead of working on the image as a whole and the effect of using the facial landmark points.

Therefore, these scenarios are:

1) The whole image was used as an input to the model, with dimensions 1024 X 1024 pixels. Subsequently, simple pre-processing was used, such as resizing images to 224 X 224 pixels and normalizing the image pixels' values.

2) The person's face was determined inside the image using the MediaPipe face detection library (sub-section 4.1). Only the face was cropped using the Box attribute to get a new image of around 500 X 700 pixels, and the same pre-processing in the first scenario (resizing and normalization) on this crop face image and inputted to the model.

3) After detecting the face, select and crop the areas of the eye and eyebrow, nose, and mouth from the face (sub-section 4.2). Consequently, deal with each of them as a new image by applying the same pre-processing in the first scenario (resizing and normalization). Lastly, inputted each part of the face to the model separately, which means training the same model three-time, each time with one of the three segments of the face.

4.4. Proposed Model

Over the years, variants of CNN architecture have been developed, leading to tremendous advances in the field. A good measure of this progress is the error rate in competitions such as the ILSVRC ImageNet challenge [31]. In addition, many deep learning models are pre-trained on the ImageNet dataset. They are made available on Keras Applications with their pre-trained weights, which can be used for prediction, feature extraction, and fine-tuning. Moreover, the DenseNet121 [10] model was utilized as an input layer to our proposed model; to input an image and get the feature map from it as output to the following layers.

A few layers followed the first layer: Flatten, Dense, Dropout, and Batch normalization. The activation functions used with the Dense layers are LeakyReLU except the last one (output layer); Sigmoid was employed because the Deepfake detection is a binary classification problem, and for kernel initializer, the he-normal value was used, and the loss function used binary cross entropy. Lastly, Adam [32] was used as an optimizer. See Table 2 presents the proposed model's architecture summary.

Table 2- Summary of the architecture of the proposed model

Model: "sequential"		
Layer (type)	**Output Shape**	**Parameter**
densenet121 (Functional)	(None, 7, 7, 1024)	7037504
flatten (Flatten)	(None, 50176)	0
batch_normalization_1	(None, 50176)	200704
dropout1 (Dropout)	(None, 50176)	0
dense1 (Dense)	(None, 256)	12845312
batch_normalization_2	(None, 256)	1024
dropout2 (Dropout)	(None, 256)	0
dense2 (Dense)	(None, 128)	32896
batch_normalization_3	(None, 128)	512
dropout3 (Dropout)	(None, 128)	0
dense3 (Dense)	(None, 32)	4128
batch_normalization_4	(None, 32)	128
dense4 (Dense)	(None, 1)	33
Total params: 20,122,241		
Trainable params: 19,937,409		
Non-trainable params: 184,832		

5. Experimental Setup

Experiments are conducted on a ZBook hp Intel (R) Core (TM) i7-7500U CPU @ 2.70GHz 2.90 GHz, VGA AMD 2 G.B., and RAM 24 G.B. with windows 10. MediaPipe libraries are used for image pre-processing; the face detection library is used for the face detection step, the Box attribute is used to crop the face, and the facial landmarks library is used for the selection and segmentation step of face parts. Furthermore, the Keras from Tensorflow was used to deploy the pre-train model (Densenet121) as an input layer to our method to extract the deep embedding futures from an image, followed by a few Flatten, Dense, Dropout, and Batch normalization layers. The precision, recall, f1-score and accuracy were used to compare the face segments results. Apart from that, the accuracy and loss results for train, validation, and testing were used to compare the three scenarios. The proposed model was evaluated using a subset (6000 images) of datasets from Deepfake Detection Challenge benchmark datasets for Deepfake detection classification, as explained in Section (3), with three different pre-process scenarios see sub-section (4.3). In addition, the same dataset fragmentation is 75% training, 25% testing, and validation 20%, with 50 epochs and 128 batch sizes used in all running and testing of the proposed model.

6. Experiment Results and Discussion

This section presents how the objectives of this study were achieved through the results obtained in two sub-sections are:

6.1. Comparative evaluation of the face segments results

This study is the first to utilize the MediaPipe tool to extract the specific areas of the face from the high-quality image based on the facial landmark points. These areas are the eye and eyebrow, nose, and mouth. One important reason this paper uses the MediaPipe tool is its many libraries and attributes. It enables us to deal with an image that includes several faces and the posed face; see sub-section (4.1). For the face segmentation method in this work, the researchers explored more than one facial landmark tool, such as Dlib, Multitask Cascaded CNN (MTCNN), Stasm, and MediaPipe. The MediaPipe tool is the best for our work because it contains many libraries and 468 face mesh points. Hence, this enables us to determine the parts of the face image with high accuracy and flexibility; the output of this pre-process is shown in Fig. 1. Subsequently, each part was inputted individually into the same model proposed in this study after performing simple operations such as resizing and normalizing.

Consequently, the proposed model was trained on all parts of the face separately and with the same hyperparameters shown in Section (5) and the same images from the same database explained in Section 3. The results later demonstrate that all parts are valid to be alone sufficient in identifying and detecting deep forgery and provided good and better results than some previous studies. Moreover, the eye and eyebrow area showed the best results for this study; see Table 3. This confirms that this area possesses features and qualities that enable the model to detect counterfeiting with high accuracy.

Table 3- Display the Precision, Recall, F1-score, and Final accuracy results of the model in three face segmentation regions; R: Real images, F: Fake images.

No	Face Segmentation	Precision	Recall	F1-score	Final accuracy
1	Mouth region	**R**: 0.96 **F**: 0.96	**R**: 0.95 **F**: 0.95	**R**: 0.96 **F**: 0.96	0.96
2	Nose region	**R**: 0.97 **F**: 0.95	**R**: 0.95 **F**: 0.97	**R**: 0.96 **F**: 0.96	0.96
3	Eye and Eyebrow region	**R**: 0.98 **F**: 0.95	**R**: 0.95 **F**: 0.98	**R**: 0.97 **F**: 0.97	**0.97**

6.2. *Comparative evaluation of the three scenarios' results*

The problem of distinguishing fake faces in high-quality photos is becoming increasingly complex for the human eye and even using software. Hence, in Deepfake detection, many studies have proposed using many complex techniques in pre-processing. The primary step in most of these studies is a pre-process to reduce the image dimension by resizing it, especially for those who's worked on the FFHQ dataset because it has high quality. In this study, with the third scenario, resizing a high-quality image is avoided. Due to the resizing process could lead to the loss of some important detail in the image, part of the image was used instead of resizing it and got a good result. During our experiment, the MediaPipe face detection tool detected more than one face as a miss detection in some pictures, see Fig. 3c. Hence, this explains these images have features outside of the person's face that are very similar to facial features. Since inputting the whole image (first scenario) into the model, missing detected areas could confuse the model or work as noise. Other than that, the same problem can be seen with the second scenario cropping box of the face see Fig. 3. All image's lower corners contain irrelevant features, with a cropped specific part of the face preventing irrelevant features, even though there are some mistakes in face mesh generation Fig. 4a.

On the practical side, when working on the complete image (first scenario), a gap was discovered between the training accuracy and the validation accuracy (about 11%), see Table 4. Furthermore, in the second one (cropping only the face), the gap was still there, with only the accuracy improved slightly. When performing the same experiments on a specific part of the face (eye and eyebrow), the results become better from the third epoch (90%), and in the end, the gap reduces to (2%). Note that the test accuracy increased to 97%; Fig. (5-7) and Table 4 show the different outcomes of the performance of the proposed model in the three cases with the same dataset and hyperparameters of the model.

Table 4 - Display the loss and accuracy results of the model in three cases study; F: Final, T: Training, A: accuracy, L: Loss, V: Validation, and Te: Testing.

No	Case study	F.T. A	F.T. L	F.V. A	F.V. L	Te. A	Te. L
1	Whole image	0.97	1.01	0.86	1.31	0.87	1.31
2	Crop only face	0.97	0.77	0.89	1.03	0.89	1.01
3	Crop only eye and eyebrow	**0.98**	**0.73**	**0.96**	**0.81**	**0.97**	**0.81**

Lastly, Table 5 illustrates how this method outperforms some state-of-the-art techniques by demonstrating the ability and effectiveness of utilizing the specified facial section based on a good tool.

Table 5 - Comparisons of the suggested framework with state-of-the-art that worked on FFHQ; Acc: Accuracy

Ref	Pre-process	Model	Acc.
[25]	1- Resized image to 256 * 256 2- Remove noise using Kalman filter 3- Normalization 4- Feature extraction (FF-LBPH)	Deep Belief Network	94%
[11]	Extracted iris pairs using DLib facial landmark and Mask R-CNN	Residual Attention Network	97%
Our	**1- Cropped specific facial region** **2- Resized facial region to 224 * 224** **3- Normalization**	**Densenet121**	**97%**
[33]	1- Extract global texture features 2- Augmentation technique	ResNet, Gram-Net, LBP-Net, and InceptionResnetV1	98%

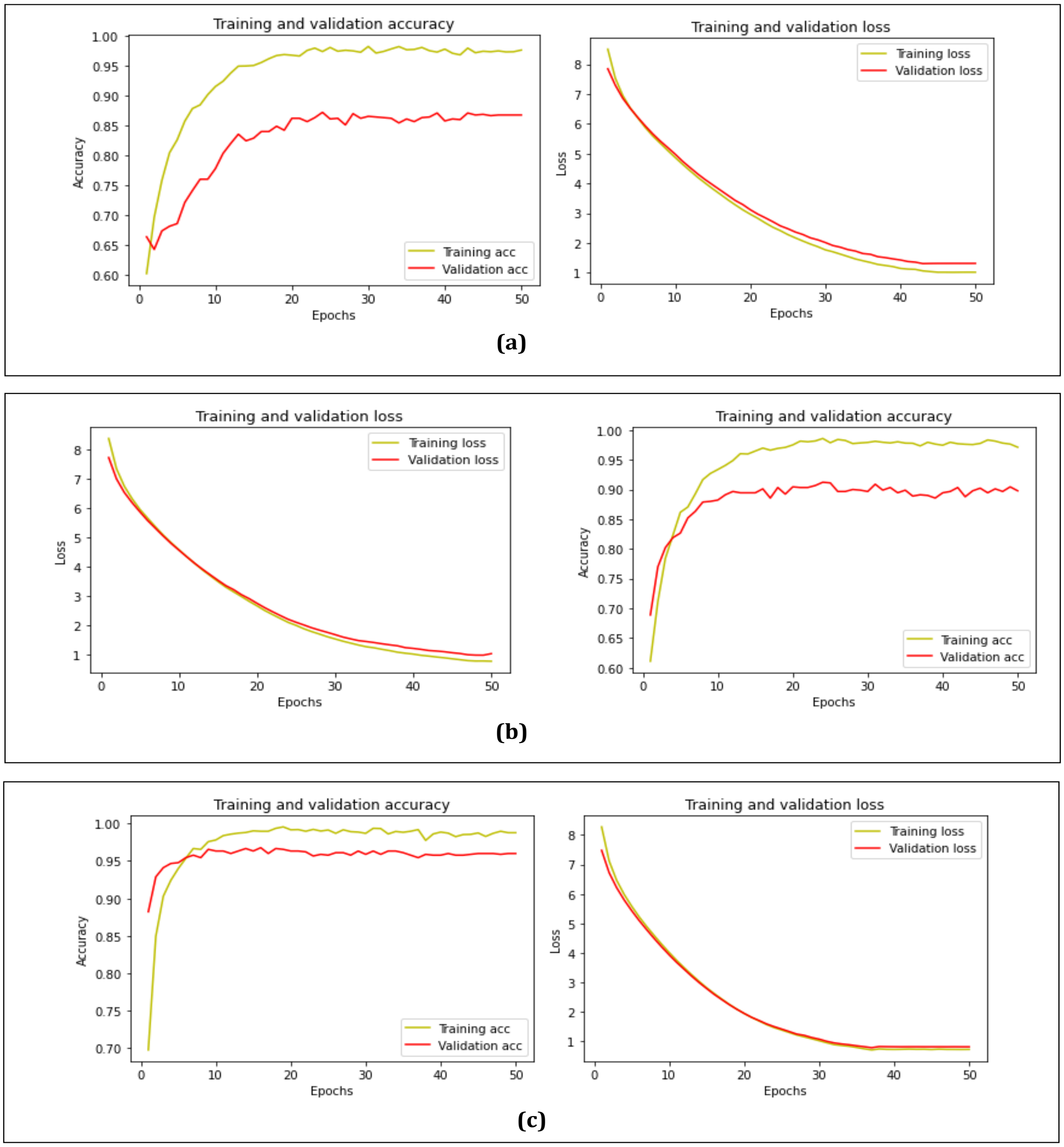


Fig. 5 - Shows the performance difference for the proposed model with the three cases in the same dataset. (a) The loss and accuracy results to the model with whole image; (b) The loss and accuracy results to the model with crop only face from image; (c) The loss and accuracy results to the model with the area of eyebrow and eye from image.

7. Conclusion

This study falls within the field of Deepfake detection in images with AI by working to identify and detect forged images within the popular databases Flickr-Faces-Hight Quality (FFHQ) and 1-Million Fake Faces. Through the results of the experiments, this study presented how far and how well facial landmark points can be used to select and crop a certain part of the face. This technique showed the possibility of training the model on the part of the face and achieving good results, better than training it on the whole image or full face when using the appropriate tool with a good feature region. Other than that, this research also demonstrated the possibility of employing more than one library and attribute for the same tool, such as the facial landmarks, key points, and box score, to avoid the problem of dealing with the posed face and multiple faces in one image.

Acknowledgements

This study was completely supported by the authors and has not received any financial support from any organization.

References

[1] I. Goodfellow *et al.*, "Generative adversarial nets," vol. 27, 2014.
[2] T. Karras, S. Laine, and T. Aila, "A Style-Based Generator Architecture for Generative Adversarial Networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PP, pp. 1-1, 01/31 2020.
[3] D. Zhang *et al.*, "The AI index 2021 annual report," 2021.
[4] P. Korshunov and S. Marcel, "Vulnerability assessment and detection of deepfake videos," in *2019 International Conference on Biometrics (ICB)*, 2019, pp. 1-6: IEEE.
[5] A. Elhassan, M. Al-Fawa'reh, M. T. Jafar, M. Ababneh, and S. T. J. S. Jafar, "DFT-MF: Enhanced deepfake detection using mouth movement and transfer learning," vol. 19, p. 101115, 2022.
[6] Yu, Chia-Mu, Ching-Tang Chang, and Yen-Wu Ti. "Detecting deepfake-forged contents with separable convolutional neural network and image segmentation." arXiv preprint arXiv:1912.12184 (2019).
[7] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4700-4708.
[8] Oraibi, Mohammed R., and Abdulkareem M. Radhi. "Enhancement Digital Forensic Approach for Inter-Frame Video Forgery Detection Using a Deep Learning Technique." Iraqi Journal of Science (2022): 2686-2701.
[9] K. Y. Lim, H. Joan, and Y. Tew, "COMPUTER PERFORMANCE EVALUATION FOR VIRTUAL CLASSROOM WITH ARTIFICIAL INTELLIGENCE FEATURES," in *International Conference on Digital Transformation and Applications (ICDXA)*, 2021, vol. 25, p. 26.
[10] S. Das, S. Seferbekov, A. Datta, M. Islam, and M. Amin, "Towards solving the deepfake problem: An analysis on improving deepfake detection using dynamic face augmentation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 3776-3785.
[11] Guo, Hui, et al. "Robust Attentive Deep Neural Network for Detecting GAN-Generated Faces." IEEE Access 10 (2022): 32574-32583.
[12] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao, "Joint Face Detection and Alignment Using Multitask Cascaded Convolutional Networks," *IEEE Signal Processing Letters*, vol. 23, no. 10, pp. 1499-1503, 2016.
[13] King, Davis E. "Dlib-ml: A machine learning toolkit." The Journal of Machine Learning Research 10 (2009): 1755-1758.
[14] S. Milborrow and F. Nicolls, "Active shape models with SIFT descriptors and MARS," in *2014 International Conference on Computer Vision Theory and Applications (VISAPP)*, 2014, vol. 2, pp. 380-387: IEEE.
[15] J. T. Camillo Lugaresi, Hadon Nash, Chris McClanahan, Esha Uboweja, Michael Hays, Fan Zhang, Chuo-Ling Chang, Ming Guang Yong, Juhyun Lee, Wan-Teh Chang, Wei Hua, Manfred Georg and Matthias Grundmann Google Research, "MediaPipe: A Framework for Building Perception Pipelines," 2019.
[16] J. Deng, J. Guo, E. Ververas, I. Kotsia, and S. Zafeiriou, "Retinaface: Single-shot multi-level face localisation in the wild," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 5203-5212.
[17] T. Baltrusaitis, A. Zadeh, Y. C. Lim, and L.-P. Morency, "Openface 2.0: Facial behavior analysis toolkit," in *2018 13th IEEE international conference on automatic face & gesture recognition (FG 2018)*, 2018, pp. 59-66: IEEE.
[18] A. I. Siam, N. F. Soliman, A. D. Algarni, F. E. Abd El-Samie, and A. Sedik, "Deploying Machine Learning Techniques for Human Emotion Detection," *Comput Intell Neurosci*, vol. 2022, p. 8032673, 2022.
[19] G. Révy, D. Hadházi, and G. Hullám, "Towards Hand-Over-Face Gesture Detection," in *29th Minisymposium of the Department of Measurement and Information Systems*, 2022, pp. 58-61: Budapest University of Technology and Economics.
[20] Shorten, Connor, and Taghi M. Khoshgoftaar. "A survey on image data augmentation for deep learning." Journal of big data 6.1 (2019): 1-48.
[21] Silva, Samuel Henrique, et al. "Deepfake forensics analysis: An explainable hierarchical ensemble of weakly supervised models." Forensic Science International: Synergy 4 (2022): 100217.
[22] N. H. Salman and S. N. Mohammed, "Image Segmentation Using PSO-Enhanced K-Means Clustering and Region Growing Algorithms," *Iraqi Journal of Science*, pp. 4988-4998, 2021.
[23] Abdulateef, Salwa Khalid, and Mohanad Dawood Salman. "A Comprehensive Review of Image Segmentation Techniques." Iraqi Journal for Electrical & Electronic Engineering 17.2 (2021).
[24] Symeon, Polychronis Charitidis Giorgos Kordopatis-Zilos, and Papadopoulos Ioannis Kompatsiaris. "AFace PREPROCESSING APPROACH FOR IMPROVED DEEPFAKE DETECTION." arXiv preprint arXiv:2006.07084 (2020).
[25] Suganthi, S. T., et al. "Deep learning model for deep fake face recognition and detection." PeerJ Computer Science 8 (2022): e881.
[26] Nirkin, Yuval, et al. "DeepFake detection based on discrepancies between faces and their context." IEEE Transactions on Pattern Analysis and Machine Intelligence (2021).
[27] S. Fang, S. Wang, and R. Ye, "DeepFake Video Detection through Facial Sparse Optical Flow based Light CNN," in *Journal of Physics: Conference Series*, 2022, vol. 2224, no. 1, p. 012014: IOP Publishing.
[28] Dhahir, Hayder Kadhim, and Nassir Hussein Salman. "A Review on Face Detection Based on Convolution Neural Network Techniques." Iraqi Journal of Science (2022): 1823-1835.
[29] A. T. Kabakus, "An experimental performance comparison of widely used face detection tools," 2019.
[30] Y. Kartynnik, A. Ablavatski, I. Grishchenko, and M. J. a. p. a. Grundmann, "Real-time facial surface geometry from monocular video on mobile GPUs," 2019.
[31] O. Russakovsky *et al.*, "Imagenet large scale visual recognition challenge," vol. 115, no. 3, pp. 211-252, 2015.
[32] Kingma, Diederik P., and Jimmy Ba. "Adam: A method for stochastic optimization." arXiv preprint arXiv:1412.6980 (2014).
[33] Y. Wang, V. Zarghami, and S. Cui, "Fake Face Detection using Local Binary Pattern and Ensemble Modeling," in *2021 IEEE International Conference on Image Processing (ICIP)*, 2021, pp. 3917-3921: IEEE.