IJARSCT

ISSN (Online) 2581-9429

International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)

International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal

Impact Factor: 7.53

Volume 5, Issue 1, January 2025

Neural Architecture Search Based Deepfake Detection Model using YOLO

Somnath Banerjee[1], Bhuman Vyas[2], Shalini Sivasamy[3], Mahaboob Subhani Shaik[4]

Data Engineer, Data, AMFAM, USA[1]

Senior Software Engineer, Enterprise Architecture, Credit Acceptance Corporation, USA[2]

AI/ML Engineer, AI Automation, Webster Bank, USA[3]

Senior Software Engineer, SAIPSIT Inc, USA[4]

Abstract*: Deepfakes are intentionally created to disseminate false information or serve malicious purposes. Detecting deepfakes has become increasingly difficult due to the advancing technology involved in their creation. This paper introduces a deep learning model based on Neural Architecture Search (NAS) that incorporates the You Only Look Once (YOLO) model for image segmentation and employs data augmentation to enhance the diversity of the dataset. The goal is to improve deepfake detection accuracy compared to current models. The study utilized the CelebDF v2 dataset, which includes 590 genuine videos and 5,639 deepfake videos. From this dataset, 100 deepfake and 100 real videos were chosen, and frames were extracted. After augmentation, the resulting dataset comprised 2,000 real and 2,000 deepfake images. The proposed model attained a testing accuracy of 99.04% and performed exceptionally well across other evaluation metrics such as F1 score, precision, and recall.*

Keywords: NAS, YOLO, GAN, augmentation

I. INTRODUCTION

In a world where advanced learning technology produces deepfakes, Generative Adversarial Networks (GANs) serve as both the creators of these illusions and the instruments required to detect them. Deepfakes are made by overlaying facial images of a target individual onto a video of a source individual, resulting in highly realistic media alterations [1]. These manipulations have generated significant debate, especially concerning public figures such as celebrities and politicians, posing risks to public trust, the integrity of information, and even individual safety [2][3][4].

While the existing deepfake detection methods, including those utilizing deep convolutional GAN (DCGAN) models, demonstrate potential, they face notable challenges. These models frequently encounter problems such as model collapse and gradient instability, making the training and convergence processes more difficult [1]. Many detection models excel in controlled settings but have difficulty generalizing real-world scenarios where the types of deepfakes are unfamiliar. This challenge primarily arises from their reliance on large, well-annotated datasets for training, which are often hard to obtain in practical situations. These limitations hinder the ability of deepfake detection systems to scale and adapt to the fast-evolving nature of deepfake technologies.

Our research builds upon these methods by presenting a deep learning model based on Neural Architecture Search (NAS) that utilizes YOLO v8n for image segmentation and data augmentation, enhancing the precision of deepfake detection. By harnessing YOLO's object detection features, we extract and isolate facial regions from video frames, which are then augmented to enrich the dataset's diversity. This approach enhances the model's capacity to accurately identify deepfakes, even with limited data, and addresses the limitations of traditional models regarding generalization and adaptability to new types of deepfakes.

II. RELATED WORK

Detecting deepfakes with Deep Convolutional GANs by [1] classifies facial manipulations and employs a DCGAN model, highlighting important features such as batch normalization and LeakyReLU activation. The adversarial training achieves 100% detection of fake images after 9 iterations.

DOI: 10.48175/IJARSCT-22938

Li Y et. al. in [2] introduced a large-scale dataset of 5,639 high-quality Deepfake videos and over 2 million frames, addressing limitations in existing datasets. By employing higher resolution synthesis (256x256 pixels) ,color transfer algorithms, they achieved a Mask-SSIM score of 0.92. Various detection methods, including Two-Stream, MesoNet, and XceptionNet , were tested, revealing Celeb-DF to be the most challenging dataset, consistently producing lower AUC scores across all methods.

A deepfake detection system that utilizes YOLO in conjunction with Local Binary Pattern Histogram (LBPH) is presented in [3] for improved accuracy. LBPH enhances detection effectiveness by emphasizing facial features. The system takes advantage of YOLO's fast processing, achieving metrics such as AUROC (88.35%) and precision (86.88%), which indicate exceptional performance across various datasets. This model is recognized for its decreased computation time and increased accuracy.

Reference [4] explores the development of a universal detector for CNN-generated images by evaluating 11 CNN-based generators. The classifier achieved high performance, with augmentation enhancing results on most datasets. The paper identifies frequency domain artifacts as key indicators of CNN-generated images but highlights challenges like false positives and robustness to social media compression.

In [5] the authors propose a deep learning-based method using YOLO models to detect manipulated medical images. YOLO models (YoloV3, V5, V8 variants) were evaluated, with YOLOV5su achieving the best recall (0.997) for lung CT scans highlighting YOLO’s efficiency, particularly for synthetic data, and its potential in medical deepfake detection.

A hybrid approach using CNN and GRU to enhance deepfake detection is seen in [6]. The core method combines a CNN using HOG descriptors and an improved XceptionNet for spatial feature extraction, merging them before applying GRUs to capture temporal features. The proposed model achieves an AUROC of 95.53%. The paper emphasizes the advantage of spatiotemporal feature learning in deepfake detection.

The work by Nela Petrželková and Jan in [7] addresses challenges in deepfake detection, including generalization to unseen generators, robustness to image degradations, and adversarial attacks. A ResNet-50 binary classifier achieved 100% accuracy while FaceApp (FA) performed best in leave-one-out testing. Additionally, the YOLOv8 model, trained on a custom dataset for partial manipulations, achieved a mAP50 score of 90%, surpassing the HiFi 2023 model.

The paper [8] focuses on a convolutional recurrent network architecture that combines YOLO v3 for face detection with EfficientNet-B5 for spatial feature extraction and Bidirectional Long Short-Term Memory (Bi-LSTM) for temporal analysis. The Bi-LSTM enhances the temporal understanding of features by processing sequences forward and backward. The proposed model is compared against other alternative combinations. The proposed model outperformed these alternatives, achieving an AUROC of 77.47% using the bagging approach.

Gihun Lee and Mihui Kim in [9] propose a deepfake video detection method that analyzes the rate of change between frames to exploit subtle inconsistencies often present in GAN- or autoencoder-generated videos. Implemented using Keras with binary cross-entropy as the loss function and the Adam optimizer, the model features five hidden layers. Their method outperforms traditional detection techniques, such as MesoNet and SVM, on the Face2Face, FaceSwap, and DFDC datasets.

Elsken, Metzen, and Hutter (2019) in [10] survey NAS, a crucial field in automating the design of neural networks. They categorize NAS methods into three dimensions: search space, search strategies and performance estimation. The NAS search space defines potential architectures, NAS search strategies, including random search, Bayesian optimization, evolutionary algorithms identify high-performing architectures. Performance estimation strategies for NAS reduce computational costs using methods like lower-fidelity training, learning curve extrapolation , balancing speed and accuracy for faster, unbiased searches. Notably, NAS has outperformed manually crafted architectures in domains like image classification and semantic segmentation.

III. METHODOLOGY

Proposed System Architecture

The suggested system architecture for deepfake detection using the YOLOv8n model consists of multiple stages, including frame extraction, face detection, augmentation, data preprocessing, and model training.

Frame Extraction

From the dataset of Celeb-DF v2 videos, 100 real and 100 deepfake videos were selected. For each video, five representative frames were extracted, resized to 256x256 pixels to standardize input dimensions, and then passed to the next stage.

Face Detection with YOLOv8n

The extracted frames underwent processing with the YOLOv8n model for face detection. YOLOv8n was selected for its rapid inference and accurate localization, producing bounding boxes around faces in an average of 6.4 milliseconds per image. This process ensured that only facial features crucial for deepfake detection were preserved. Furthermore, the system included a duplication check to eliminate redundant faces from the dataset, improving data quality.

Data Augmentation

To improve the model's robustness, a variety of data augmentation techniques were implemented. Each detected face underwent the following transformations:
- Horizontal and Vertical Flipping: Creates variations in orientation.
- Brightness and Contrast Adjustments: Mimics different lighting scenarios.
- Rotation and Scaling: Introduces geometric variability.
- Noise Injection: Reflects real-world image flaws.
- Contrast Limited Adaptive Histogram Equalization (CLAHE): Enhances contrast, particularly useful in low-light frames.

These augmentations increased the dataset to 1,048 real images and 1,060 fake images, resulting in a balanced and varied dataset.

Data Preprocessing and Storage

All face images, including their augmented versions, were resized to a uniform dimension of 256x256 pixels. The dataset was subsequently saved in HDF5 format, which allows for efficient data retrieval and storage for training needs.

Model Training with NAS

The last phase consisted of training the neural network model with a NAS framework. NAS explored various model configurations, focusing on optimizing performance metrics like accuracy, speed, and resource efficiency.

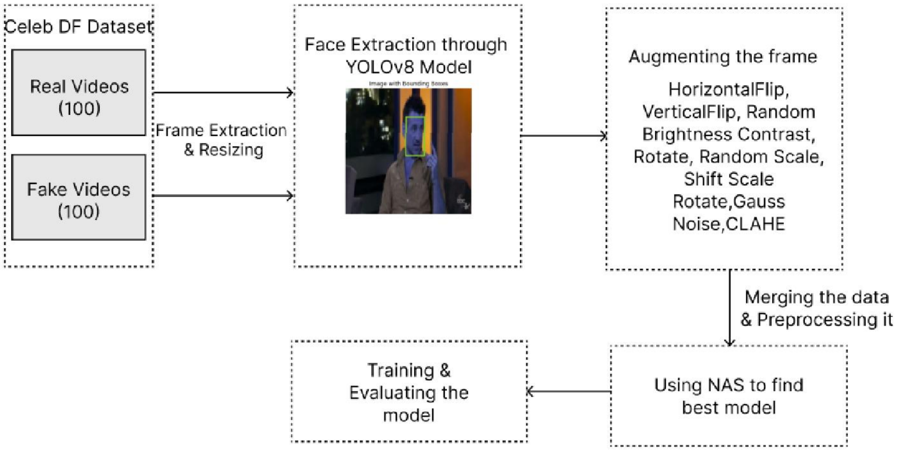

Celeb DF Dataset
Real Videos (100)
Fake Videos (100)
Frame Extraction & Resizing
Face Extraction through YOLOv8 Model
Image with Bounding Boxes
Augmenting the frame
HorizontalFlip, VerticalFlip, Random Brightness Contrast, Rotate, Random Scale, Shift Scale Rotate,Gauss Noise,CLAHE
Merging the data & Preprocessing it
Using NAS to find best model
Training & Evaluating the model

Fig. 1. Proposed model for Deepfake Detection demonstrating all the steps involved

This procedure was carried out over 10 epochs with a maximum of 5 trials, allowing for the selection of the most effective model architecture for deepfake detection. Proposed Model Figure 1 below illustrates the proposed model for

Deepfake Detection and outlines the fundamental steps involved in the entire process, starting from dataset creation and extending to the training and evaluation of the proposed model.

The suggested model for Deepfake Detection utilizes YOLOv8n for object detection along with a Convolutional Neural Network (CNN) identified via Neural Architecture Search (NAS). The system leverages YOLOv8n's rapid inference ability to identify faces in real-time and generate bounding boxes around them. The faces, taken from both authentic and fake video frames, are compiled to create the dataset for training the classification model.

Face Detection and Augmentation

Using YOLOv8n, the model conducts single inference on each video frame to identify and extract faces. The inference time for each image is roughly 6.4 milliseconds. After extracting the faces, they are augmented, with each frame undergoing two augmentations utilizing methods like flipping, brightness and contrast modifications, rotation, scaling, noise addition, and CLAHE. This approach greatly boosts the diversity of the dataset, raising the counts of real and fake faces from 524 and 530 to 1,048 and 1,060, respectively.

Neural Network Architecture

The NAS process explores various architectural configurations, testing different model parameters over 10 epochs to identify the optimal network. The best model determined by NAS is a Convolutional Neural Network (CNN), which balances accuracy, speed, and resource efficiency after assessing numerous architectures, thereby fulfilling the high-performance requirements for deepfake detection. The model begins with an input layer designed to process preprocessed 256x256 face images. These images undergo normalization and are then passed through several convolutional layers with increasing filter sizes (such as 32 and 64 filters), followed by max pooling to minimize spatial dimensions. This architecture enables the model to learn the spatial hierarchies of patterns within the images. The multiple layers facilitate the capture of both low-level features, like edges and textures, and high-level features, such as shapes and objects. The flattened output is subsequently directed to a dense layer featuring a single output neuron for binary classification (real or fake). The output employs a sigmoid activation function, making it appropriate for binary classification tasks. The final trained model successfully distinguishes between real and fake videos based on facial features extracted from the video frames. This system offers high-speed face detection and reliable classification outcomes, presenting an efficient solution for verifying video authenticity.

Dataset

The paper [1] by authors Li, Y., Yang, X., Sun, P., Qi, H., Lyu, S introduces the Celeb-DF Dataset, a high-quality, large-scale dataset designed to advance DeepFake forensics. This dataset, comprising 5,639 celebrity DeepFake videos and over 2 million frames, offers a challenging benchmark for evaluating and improving detection methods. Celeb-DF achieves high-resolution (256x256 pixels) , unlike earlier datasets—such as UADFV, DF-TIMIT etc that often suffer from visible artifacts Celeb-DF better mirrors high-quality DeepFakes commonly found online.

Celeb-DF was evaluated for its effectiveness by comparing it with various leading DeepFake detection models, including Two-Stream, MesoNet, Xception, and HeadPose. This assessment utilized the area under the ROC curve (AUC) at the frame level specifically for keyframes. The findings revealed that Celeb-DF consistently produced lower detection scores, highlighting its distinctive ability to challenge and possibly improve the resilience of existing DeepFake detection algorithms.

Celeb-DF (v2) builds upon Celeb-DF (v1) by adding 590 original videos featuring individuals of various ages, ethnic backgrounds, and genders, alongside 5,639 corresponding DeepFake videos to enhance the representation of real-world diversity. For our model, we selected 100 original videos and 100 DeepFake videos, from which we extracted 10 frames each, resulting in a total of 1,000 original images and 1,000 DeepFake images, all resized to 256x256 and stored in HDF5 format to optimize RAM usage. This dataset established the foundation for both the baseline model and the YOLO model, which was improved with data augmentation and face detection using YOLOv8.

The data augmentation process improves the diversity of the image dataset by implementing various probabilistic transformations that change image orientation, brightness, contrast, scale, and noise. This includes horizontal and vertical flips, random adjustments to brightness and contrast, as well as rotations and scaling within defined limits.

Additional modifications introduce Gaussian noise and use CLAHE for adaptive contrast adjustments, mimicking different real-world lighting scenarios and noise levels. This increased variability helps models learn more robust features and enhances their ability to generalize by exposing them to multiple augmented versions of each training image.

IV. IMPLEMENTATION OF MODEL

The paper presents two distinct approaches:

1) image segmentation without the YOLO v8 model and without data augmentation,

2) image segmentation using the YOLO v8 model along with data augmentation.

Both approaches utilize the NAS model, a machine learning method designed to automate the creation and optimization of neural network architectures. The NAS model identifies the most efficient and high-performing neural network architecture for the specific task. This NAS model was employed to determine the best-performing neural network in both approaches, with five trials conducted, each consisting of ten epochs. In the first case, where the YOLO v8 model and data augmentation were not used, a selection of 100 real and 100 fake videos was made, from which 10 frames were extracted from each video, resulting in 1,000 images per category and a total of 2,000 images. The best validation loss achieved after five trials was 0.0287, with the efficientnetb7 model being selected by the NAS method. Following normalization, the NAS model identified data augmentation techniques such as random translation and random flip, which were then applied to the efficientnetb7 model. The output from the efficientnetb7 model was passed to a global average pooling 2D layer, converting the feature vector from 2D to 1D. The model architecture proposed by the NAS method for case 1 is illustrated in Fig. 2 below, showcasing all the layers employed, the number of filters in each layer, the data augmentation layers utilized, and the sequential control flow from one layer to the next.

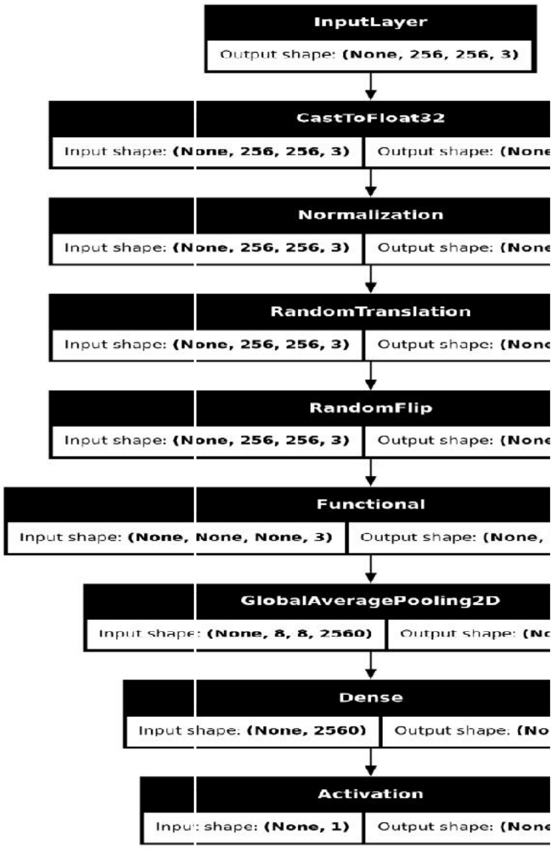


Fig. 2. : Architecture of model proposed by using NAS involving case 1 (without using YOLO v8 model for image segmentation and without doing data augmentation)

After identifying the best model, it was compiled using the Adam optimizer and binary cross entropy for the loss function. The best model was trained for 20 epochs with an 80-20 split of the data for training and testing. A batch size of 8 was selected for the training process to enhance computational efficiency. The parameters used for finding and training the best model in case 1 and case 2 are presented in Table 1 below.

Table 1. Parameters used in finding and training best model in case 1 and case 2

Parameter	**Value**	
	Case 1	**Case 2**
Max Trials	5	5
Epochs	10	10
Batch Size	8	8
Optimizer	Adam	Adam
Loss	Binary Crossentropy	Binary Crossentropy
Epoch	20	20

In case 2, which involves utilizing the YOLO v8 model for image segmentation and implementing data augmentation, 100 real and 100 fake videos were selected.

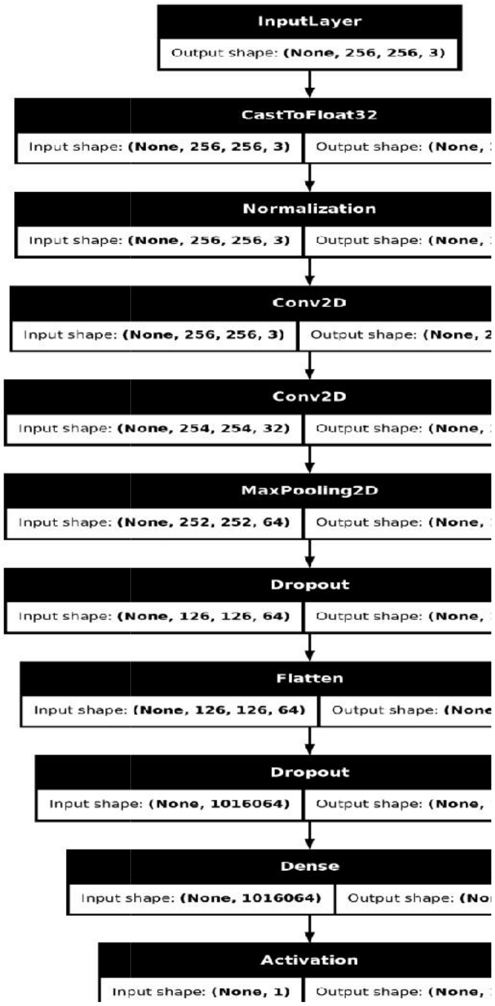
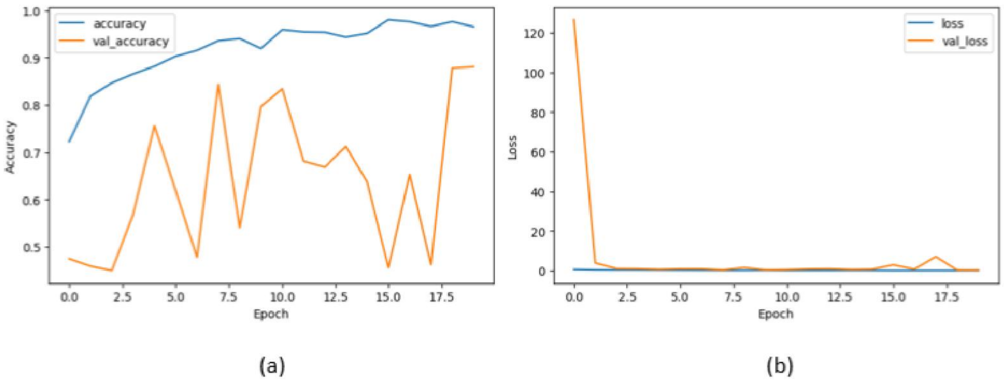

InputLayer
Output shape: (None, 256, 256, 3)
CastToFloat32
Input shape: (None, 256, 256, 3)
Output shape: (None,
Normalization
Input shape: (None, 256, 256, 3)
Output shape: (None,
Conv2D
Input shape: (None, 256, 256, 3)
Output shape: (None, 2
Conv2D
Input shape: (None, 254, 254, 32)
Output shape: (None,
MaxPooling2D
Input shape: (None, 252, 252, 64)
Output shape: (None,
Dropout
Input shape: (None, 126, 126, 64)
Output shape: (None,
Flatten
Input shape: (None, 126, 126, 64)
Output shape: (None
Dropout
Input shape: (None, 1016064)
Output shape: (None,
Dense
Input shape: (None, 1016064)
Output shape: (No
Activation
Input shape: (None, 1)
Output shape: (None,

Fig. 3.: Architecture of model proposed by using NAS involving case 2 (Using YOLO v8 model and data augmentation)

From each video, 5 frames were extracted, resulting in 500 images per category. These images were then augmented using the Augmentations library. After the augmentation process, the original set of 500 images was doubled, yielding 1000 images for each category, totaling 2000 images. The best validation loss achieved after 5 trials was 0.0427, and the basic CNN model was chosen through the NAS method. Initially, the input layer was utilized, and its output was converted to float32. Following this, normalization was applied to the float32 values. After normalization, two 2D convolution layers were employed for feature extraction from the images, succeeded by a max pooling layer to reduce the overall dimensions. A dropout layer was then introduced, followed by a flatten layer, another dropout layer, a dense layer, and finally, the classification head. The architecture of the model proposed by the NAS method for case 2 is illustrated **in Fig. 3 below**, showcasing all the employed layers, the number of filters in each layer, and the sequential control flow from one layer to the next.

After identifying the optimal model, it was compiled using the Adam optimizer and binary crossentropy as the loss function. The best model was trained with a batch size of 8 to enhance computational efficiency. Result and Analysis In case 1, which involved no YOLO v8 model and no data augmentation, all images were resized to 256x256 prior to applying the NAS method to determine the best model.

The test accuracy achieved was 89.49% with a test loss of 0.258, while the training accuracy reached 89.43% after training the best model for 20 epochs. Figure 4.(a) illustrates the accuracy versus epoch graph for training and validation accuracies obtained from the model proposed by NAS in case 1, and Figure 4.(b) displays the loss versus epoch graph for training and validation loss from the same model. 1.



(a) (b)

Fig. 4. : (a) Accuracy vs Epoch graph for proposed model using NAS for case 1 , (b) Loss vs Epoch graph for proposed model using NAS for case 1

From Fig. 4(a), it can be inferred that the validation accuracy varies throughout the 20 epochs, indicating that the model's performance is inconsistent on unseen data, which may suggest overfitting. In Fig. 4(b), the validation loss begins at a high level, decreases sharply after the first epoch, and then fluctuates at a lower level, pointing to potential issues during early training.

Table 2. shows other performance metrics(training or testing)for the model proposed in case 1.

Phase	**Accuracy**	**Precision**	**Recall**	**F1 Score**
Training	0.8943	0.8980	0.8943	0.8941
Testing	0.8950	0.9017	0.8950	0.8945

The high training and testing accuracy demonstrate the model's effective performance, with closely aligned training and testing accuracy suggesting that the model does not overfit the data, as shown in Fig. 4(a). The elevated precision during both training and testing indicates a higher number of true positives and a significantly lower number of false positives. Additionally, high recall reflects a low false negative rate in both phases, while a high F1 score signifies a balance between precision and recall for the proposed model. In case 2, which involves using the YOLO v8 model, all images were resized to 256x256 before applying the NAS method to identify the optimal model. A test accuracy of 98.81% and a test loss of 0.073 were achieved, along with a training accuracy of 99.5% after training the best model for 20 epochs. Fig. 5(a) illustrates the accuracy versus epoch graph for the training and validation accuracies obtained from the NAS-proposed model in case 2, while Fig. 5(b) presents the loss versus epoch graph for the training and validation loss for the same model.

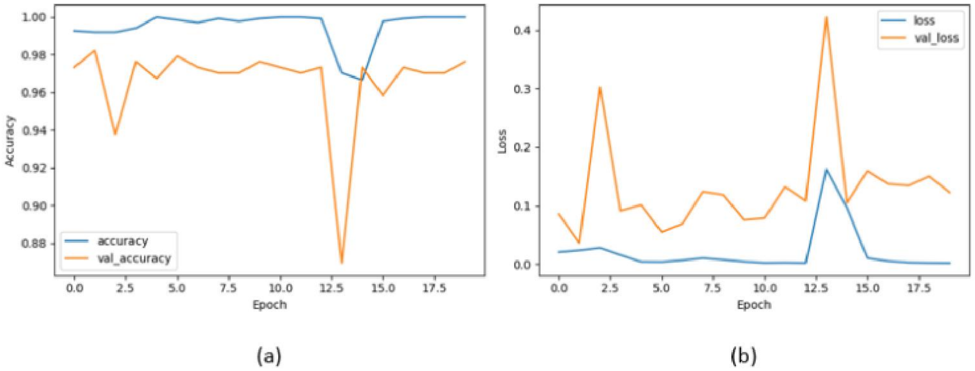


Fig. 5. : (a) Accuracy vs Epoch graph for proposed model using NAS for case 1 , (b) Loss vs Epoch graph for proposed model using NAS for case 1

From Fig. 5(a), it can be observed that the validation accuracy experiences a notable decline around the 13th epoch, indicating it may have faced some difficult data. However, post the 13th epoch, there is a swift recovery, which mirrors the trend seen in the training accuracy curve. In Fig. 5(b), both training and validation loss show a significant increase around the 13th epoch, aligning with the drop in accuracies noted in Fig. 5(a). Table 3 presents additional performance metrics for the model proposed in case 2.

Table 3. : Performance metrics for model proposed by NAS method for case 2

Phase	Accuracy	Precision	Recall	F1 Score
Training	0.9952	0.9952	0.9952	0.9952
Testing	0.9881	0.9881	0.9881	0.9881

V. CONCLUSION

Very high training and testing accuracy suggest that the model performs exceptionally well on both datasets. The close alignment of training and testing accuracy indicates that the model is not overfitting. High precision in both training and testing reflects a substantial number of true positives and minimal false positives. High recall demonstrates that the proposed model effectively captures most true positive instances. Since the F1 score is the harmonic mean of precision and recall, a high F1 score confirms that both metrics are elevated for the proposed model. Upon reviewing both scenarios, case 1 (without utilizing the YOLO v8 model for image segmentation and without data augmentation) and case 2 (utilizing the YOLO v8 model for image segmentation with data augmentation), it is clear that the model employing the NAS method outperforms case 1 significantly in case 2. This suggests that extracting faces using the YOLO v8 model and subsequently augmenting these extracted faces is advantageous for the deepfake classification task. Overall performance metrics for case 2 are superior to those in case 1.

REFERENCES

[1]. Preeti, Kumar, M., Sharma, H.K.: A GAN-Based Model of Deepfake Detection in Social Media. Procedia Computer Science 218, 2153–2162 (2023). https://doi.org/10.1016/j.procs.2023.01.191

[2]. Li, Y., Yang, X., Sun, P., Qi, H., Lyu, S.: Celeb-DF: A Large-Scale Challenging Dataset for DeepFake Forensics. In: 2020 IEEE/CVF Conference on Computer Vision and Pat-tern Recognition (CVPR), Seattle, WA, USA, pp. 3204–3213 (2020). https://doi.org/10.1109/CVPR42600.2020.00327

[3]. Hubálovský, Š., Trojovský, P., Bacanin, N., K, V.: Evaluation of Deepfake Detection Using YOLO with Local Binary Pattern Histogram. PeerJComput. Sci. 8, e1086 (2022). https://doi.org/10.7717/peerj-cs.1086

[4]. Wang, S.-Y., Wang, O., Zhang, R., Owens, A., Efros, A.A.: CNN-Generated Images are Surprisingly Easy to Spot... For Now. In: Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR), pp. X–Y (2020). https://doi.org/10.48550/arXiv.1912.11035

[5]. Karaköse, M., Yetış, H., Çeçen, M.: A New Approach for Effective Medical Deepfake Detection in Medical Images. IEEE Access 12, 52205–52214 (2024). https://doi.org/10.1109/ACCESS.2024.3386644

[6]. Ismail, A., Elpeltagy, M., Zaki, M.S., Eldahshan, K.: An integrated spatiotemporal-based methodology for deepfake detection. Neural Comput. Appl. 34(24), 21777–21791 (2022). https://doi.org/10.1007/s00521-022-07633-3

[7]. Petrzelkova, N., Cech, J.: Detection of Synthetic Face Images: Accuracy, Robustness, Generalization. arXiv:2406.17547 [cs.CV] (2024). https://doi.org/10.48550/arXiv.2406.17547

[8]. Ismail, A., Elpeltagy, M., Zaki, M., ElDahshan, K.A.: Deepfake Video Detection: YOLO-Face Convolution Recurrent Approach. PeerJComput. Sci. 7, e730 (2021). https://doi.org/10.7717/peerj-cs.730

[9]. Lee, G., Kim, M.: Deepfake Detection Using the Rate of Change Between Frames Based on Computer Vision. Sensors (Basel) 21(21), 7367 (2021). https://doi.org/10.3390/s21217367

[10]. Elsken, T., Metzen, J.H., Hutter, F.: Neural Architecture Search: A Survey. Journal of Machine Learning Research 20, 1–21 (2019). https://doi.org/10.48550/arXiv.1808.05377