A Spatio-Temporal Deep Learning Approach for Efficient Deepfake Video Detection

Raman Z. Khudhur[1] and Marwan A. Mohammed[1,2†]

[1]Department of Software Engineering, College of Engineering, Salahaddin University, Erbil, Kurdistan Region – F.R. Iraq

[2]Department of Computer Engineering, College of Engineering, Knowledge University, Erbil, Kurdistan Region – F.R. Iraq

***Abstract*—Deepfake videos have grown to be a big concern in the modern digital media landscape as they cause difficulties undermining the legitimacy of channels of information and communication. Humans often find it challenging to tell the difference between a fake and a genuine video due to the increasing realism of facial deepfakes. Identification of these misleading materials is the first step in preventing deepfakes from spreading through social media. This work introduces Spatio-temporal Intelligent Deepfake Detector (STIDD), a deep learning system including enhanced spatial and temporal modeling techniques. By means of a pre-trained EfficientNetV2-B0 model, the proposed framework efficiently extracts spatial characteristics from each frame, subsequently, and Bidirectional Long Short-Term Memory layers help to capture temporal relationships from video sequences. We evaluate STIDD on the FaceForensics++ (FF++) dataset encompassing all five manipulation techniques (DeepFakes, FaceSwap, Face2Face, FaceShifter, and NeuralTextures). The experimental results reveal that STIDD achieved precision, recall, and F1-scores all higher than 0.99 and a final test accuracy of 99.51% on the combined FF++ test set. The results demonstrate that the integration of sophisticated spatial extraction and strong temporal modeling allows STIDD to achieve high detection performance while maintaining computing efficiency at just 0.39 Giga Floating-Point Operations (GFLOPs) per inference.**

***Index Terms*—Deep learning, Deepfake detection, EfficientNet, Spatio-temporal modeling**

I. Introduction

In recent years, the digital media landscape has shifted dramatically. Advances in artificial intelligence (AI) enable the manipulation of photos and videos at an unprecedented scale, resulting in the emergence of fabricated content such as Deepfakes. Deepfake technology uses advanced methods to modify existing media or create completely synthetic content, increasingly obscuring the boundary between genuine and altered media. The dual functionality of Deepfakes, serving both advantageous and potentially detrimental purposes, poses a significant challenge for many businesses and communities (Korshunov and Marcel, 2022; Heidari *et al*., 2024).

Advances in machine learning, namely with Generative Adversarial Networks (GANs), correlate with the emergence of deepfake technology. The ability of these networks to produce artificial media that is highly realistic was first used for its artistic and entertainment usages (Kaur, et al., 2024). However, abuse is unavoidable. Nowadays, deepfakes are frequently employed in identity theft, misinformation, and other malicious activities, pushing society into a space where digital authenticity appears to be becoming more and more insecure (Shahzad, et al., 2022).

Neural networks such as GANs employ a dual□network framework in which a generator learns to create synthetic facial images and a discriminator learns to distinguish them from real images. This adversarial training paradigm yields highly realistic outputs that underpin applications such as face swapping and facial reenactment with photorealistic fidelity (Liu et al., 2021). These realistic, face-swapped deepfakes are used in a variety of contexts to fabricate fake terrorist events, inspire political fear, and enable extortion attempts.

Using a person's voice and video without permission, deepfake technology may create a variety of films, including humorous or pornographic ones (Day, 2019). The size, scope, and accessibility of deepfakes make them dangerous since they enable anybody with a single computer to produce fake movies that seem authentic. Making fake pornographic movies of celebrities, disseminating misleading information, acting as politicians, and committing financial fraud are only a few of the several uses for deepfakes (Pawelec and Pawelec Mariapawelec, 2022).

In addition to undermining individual privacy, deepfakes significantly affect the credibility of media, which in turn diminishes public trust and potentially threatens political stability (Jameel, Kadhem, and Abbas, 2022; Mubarak, et al., 2023). People have to face an unsettling truth: imagine living in a world where you cannot believe anything you come across or hear. Therefore, developing reliable and

ARO-The Scientific Journal of Koya University
Vol. XIII, No. 2 (2025), Article ID: ARO.12190. 8 pages
DOI: 10.14500/aro.12190
Received: 12 April 2025; Accepted: 19 June 2025
Regular research paper; Published: 05 August 2025
†Corresponding author's e-mail: marwan.aziz@su.edu.krd

efficient detection systems are essential. These facts make it not only an academic effort but also a critical social need to design lightweight, efficient deep learning models capable of working in various scenarios.

Although deepfake detection systems have developed rapidly, real-world applications sometimes run against problems with current methods, especially in situations limited in resources. Most current models either lack precise enough identification of deepfakes or have performance problems that cause delays in detection (Gupta, et al., 2023). This makes integrating deepfake detection methods challenging, particularly in applications where dependability and speed are critical. An effective detection method combining great accuracy and efficacy is required to ensure the successful eradication of deepfake concerns.

Given these difficulties, the goal of this research is to advance the area of deepfake detection by focusing on the development of an efficient model that is capable of precise deepfake detection to increase confidence in the information shared on different digital platforms by closing the gap between accurate deepfake detection and the limitations present in resource-constrained contexts using transfer learning approaches and optimizing for computing efficiency.

II. Related Work

One of the typical characteristics that help one to recognize someone is their face. Therefore, the rapid advancement of face synthesis technology poses a significant threat to national security. The fast development of convolutional neural networks (CNNs), GANs, and their alternatives has made it possible to generate hyper-realistic images, movies, and audio signals that are much more difficult to distinguish from genuine ones (Alom, et al., 2019; Hong, et al., 2019; Kaur, et al., 2024).

Reducing social risks are imperative. Researchers employ advanced algorithms to distinguish genuine motion movies from their manipulated counterparts. Most studies employ convolutional neural networks combined with other techniques to identify deepfakes. Alhaji, Celik, and Goel (2024) employed a deep learning architecture guided by ant colony optimization (ACO) and particle swarm optimization (PSO) to select salient attributes, yielding enhanced detection accuracy. It is worth noting that the integration of vivid temporal patterns and subtle spatial features, which were extracted from video frames through the ACO–PSO pipeline, contributed to a classifier capable of discerning genuine from manipulated content. On the other hand, Al-Adwan, et al. (2024) explore hyperparameter tuning through particle swarm optimization and reveal that adaptive parameter search often speeds convergence without degrading feature integrity. More precisely, in search of deepfake material, the proposed approach employs a hybrid EfficientNet-Gated Recurrent Unit (GRU) network.

Alanazi, Ushaw and Morgan (2023) investigate the role of specific facial regions in deepfake detection by systematically removing certain facial areas during model training and analyzing the resulting impact on performance. This approach offers valuable insights into the discriminative power of individual facial features, contributing to the refinement of face removal strategies and supporting continued progress in deepfake detection research.

Research led by Jung, Kim, and Kim (2020) focused on natural eye-blinking as a crucial indicator for identifying DeepFake videos, using an algorithm named Deep Vision. The approach was used for eight videos, effectively identifying deepfake material in seven cases, because of the predictable patterns associated with eye-blinking, a spontaneous and voluntary movement. Furthering the discussion (Luo, et al., 2024), consider different neural network designs to improve closed-eye detection and dynamic blinking patterns for signs of manipulation to identify fake facial videos generated by deep neural networks, with eye-blinking detection as the primary differentiator.

Ibnouzaher and Moumkine (2024) introduce a multi-model strategy that extracts features separately from XceptionNet, ResNet18, and EfficientNet-B5 and then refines them within a transformer-based architecture. It is crucial to highlight that this ensemble often yields more nuanced discrimination of manipulated frames. When assessed on the FaceForensics++ (FF++) and Deepfake Detection Challenge (DFDC) datasets, the ensemble method appears to leverage complementary convolutional strengths and to mitigate overfitting.

However, the complexity of the feature selection procedure and using multi-models may provide a computational efficiency challenge for the studies mentioned earlier, making them less appropriate for reliable deepfake detection in resource-constrained contexts. Mitra, et al. (2022) addressed this gap by proposing a lightweight deepfake detection approach for Internet of Things (IoT) contexts. Their method uses machine learning and texture analysis to recognize GAN-generated deepfake images at the edge, and they presented a detection API to facilitate real-world deployment. Similarly, Sridevi et al. (2022) propose an IoT-based application for detecting deepfakes, emphasizing facial motion through developing and utilizing a lightweight deep learning technique on IoT systems.

Furthermore, Xia, et al. (2022) proposed a deepfake video detection system based on MesoNet, including a preprocessing module. To enhance the differentiation among multi-color channels, the preprocessing module is first configured to process the cropped facial images. Qadir, et al. (2024) propose a hybrid model that takes input from sequential targeted frames, thereafter processing these frames through the ResNet-Swish-BiLSTM, an enhanced convolutional Bi-LSTM-based residual network for training and classification objectives. To assess the robustness of the proposed methodology, they used the DFDC and the FF++.

Despite those studies contributing to the improvement of accuracy and efficiency in deepfake detection, current approaches face significant challenges, particularly in resource-limited environments, since most current models either lack the required accuracy for deepfake detection or have performance limits that limit timely identification. Different from previous research projects, this work offers a system that maintains both accuracy and efficiency for Deepfake detection.

III. Methodology

This study proposes a hybrid approach called Spatio-temporal Intelligent Deepfake Detector (STIDD) to detect deepfakes. A thoughtful methodological approach was employed to develop and train the proposed STIDD, including the data collection and preprocessing, data augmentation, model architecture and design, training process, and model evaluation.

A. Hardware and Software Setup

The experimental hardware and software setup used for STIDD is demonstrated in Table I., This setup provided sufficient computational power and efficient storage for quick data retrieval to handle extensive video files and intensive deep learning tasks. The software stack included Python, TensorFlow, TensorFlow Lite, and Keras, offering a solid foundation for the development, training, and deployment of the deep learning model.

B. Data Acquisition and Preprocessing

The processes of data acquisition and preprocessing are essential for establishing a strong foundation in the development of an effective deepfake detection model. Thus, this phase focuses on the collection of high quality, diverse video samples and the conversion of raw video data into organized inputs that encapsulate crucial spatial and temporal details.

Data collection

The dataset has been sourced from the well-known FF++ benchmark (Rossler, et al., 2019) to improve the model's efficacy for accurate deepfake detection. The FF++ dataset was created by a collaboration of experts from the Technical University of Munich, University of Erlangen-Nuremberg, and the Federico II University of Naples to serve as a publicly available benchmark for the study of face forgery identification. The collection includes a thousand real-world videos from authentic settings such as news interviews and manipulated using five different manipulation techniques (Deepfakes, FaceShift, FaceSwap, Face2Face, and Neural Textures). FF++ was created to help with deepfake detection research and create algorithms able to precisely identify altered videos, establishing it as an acceptable benchmark in its field (Alanazi, Ushaw and Morgan, 2023).

TABLE I
Hardware and Software Environment

Component	Specification
Operating System	Windows 11 Pro
GPU	NVIDIA RTX 2080ti (11GB vRAM)
RAM	16GB
Storage	256GB SSD (for fast access) and 1TB HDD (for archive)
Software stack	Python, TensorFlow, TensorFlow Lite, Keras
Dataset	FaceForesensess++
Evaluation metrics	Accuracy, Loss, Precision (Pr), Recall (Re), F1 score, Confusion Matrix

Frame extraction and sequence formation

This approach generates individual frames by first processing the video data using the OpenCV library from a mix of real and fake MP4 videos. To reduce computational load and limit redundancy we select only key frames, specifically every tenth frame from each video. Following this, pairs of consecutive frames are formed using a sliding-window technique, as illustrated in Fig. 1. This approach ensures the preservation of temporal dynamics found in video data, apart from ensuring that every sample retains enough contextual information for the next operations.

Data splitting

Following the sequence generation, the dataset is randomly shuffled and split into training, validation, and test sets with ratios of 70% for training, 15% for validation, and 15% for test sets using a split strategy that maintains the ratio of real to fake sequences across all subsets, so guaranteeing balanced and objective training and evaluation.

C. Data Augmentation

Augmentation is done once video frames have been extracted. Zoom, brightness change, vertical flip, and horizontal flip are among the range of augmentation methods that improve the model's generalization and robustness. To accelerate the processing of massive video files, these augmentations are carried out concurrently under multi-threaded processes. As such, the system produces five variants of every extracted frame in addition to the original one. Frames must be resized suitably to convert video data into a format fit for deep learning models; hence, the extracted frames from the video data are resized to the model input size (112 × 112). Algorithm 1 shows the pseudocode for the augmentation algorithm followed in this study.

E. Model Architecture and Design

Effective hybrid model architecture is proposed for reliable and efficient deepfake detection: The STIDD, which is a lightweight yet highly accurate solution for detecting deepfake videos. The architecture consists of three primary blocks that effectively capture spatial and temporal features through a two-stage process, with the final classification component, as illustrated in Fig. 2. The network comprises roughly 7.6 million parameters, with about 7.54 million of these are trainable. These design choices enable the model to identify deepfakes with high accuracy while remaining computationally efficient.

Block 1: Spatial feature extraction

The first block focuses on extracting spatial features from each frame using an EfficientNetV2-B0 model, pre-trained on ImageNet. Wrapping this network within a TimeDistributed layer enables independent processing of every frame in the sequence, enabling the extraction of complex facial features, textures, and subtle artifacts indicating possible manipulation. Following the spatial feature map extraction, each map has been reduced into a lower-dimensional vector using a TimeDistributed GlobalAveragePooling2D layer, as

 http://dx.doi.org/10.14500/aro.12190

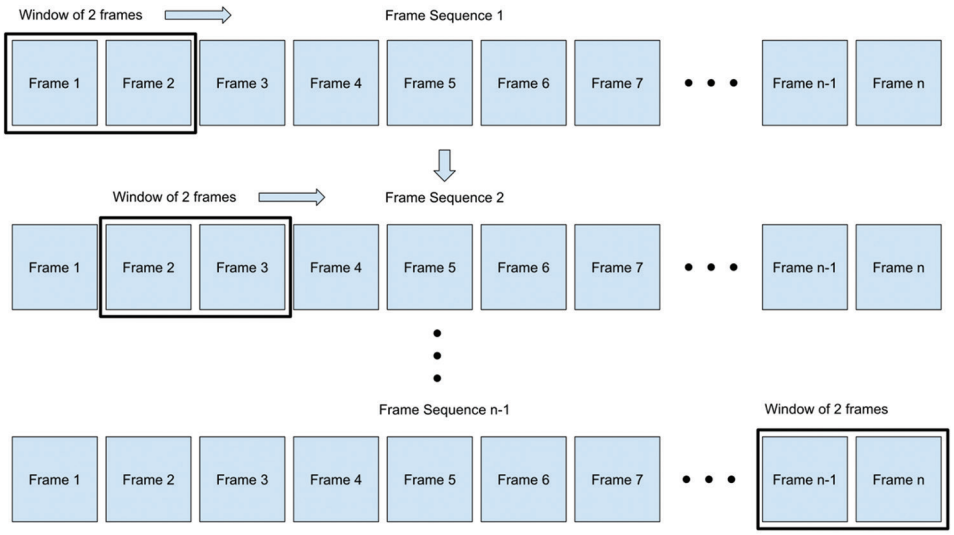

Fig. 1. Frame Sequence Generation Process using Sliding Window Approach, with n representing the number of frames in a video.

ALGORITHM 1
AUGMENTFRAME: AUGMENT A SINGLE FRAME

procedure AugmentFrame (frame, category, frameIndex, zoomFactor, brightnessFactor, targetSize)
 Concurrent Augmentation of the input frame
parallel begin
 1. Base Augmentation: Resize to target size
baseFrame←Resize (frame, targetSize)
SaveImage (baseFrame, BuildPath (outputBaseDir, "base*", category, frameIndex))*
 2. Zoom Augmentation
newWidth←frame.width×zoomFactor
newHeight←frame.height×zoomFactor
zoomedFrame←Resize (frame, (newWidth, newHeight))
startX←max (0, (zoomedFrame.width−frame.width)/2)
startY←max (0, (zoomedFrame.height−frame.height)/2)
croppedFrame←Crop (zoomedFrame, startX, startY, frame.width, frame.height)
zoomFrame←Resize (croppedFrame, targetSize)
SaveImage (zoomFrame, BuildPath (outputBaseDir, "zoom"*, category, frameIndex))*
 3. Brightness Augmentation
brightFrame←ApplyForEachPixel (frame, pixel→brightnessFactor×pixel)
brightFrame←Resize (brightFrame, targetSize)
SaveImage (brightFrame, BuildPath (outputBaseDir, "brightness"*, category, frameIndex))*
 4. Horizontal Flip Augmentation
hFlipFrame←ApplyForEachPixel (frame, (x, y) → GetPixel (frame, (frame.width−1 − x, y)))
hFlipFrame←Resize (hFlipFrame, targetSize)
SaveImage (hFlipFrame, BuildPath (outputBaseDir, "h-flip"*, category, frameIndex))*
 5. Vertical Flip Augmentation
vFlipFrame←ApplyForEachPixel (frame, (x, y) → GetPixel (frame, (x, frame.height−1 − y)))
vFlipFrame←Resize (vFlipFrame, targetSize)
SaveImage (vFlipFrame, BuildPath (outputBaseDir, "v flip"*, category, frameIndex))*
parallel end
end procedure

illustrated in Fig. 3. This pooling procedure reduces overall dimensionality while maintaining important information required for effective modeling and helps prevent overfitting by lowering the number of parameters sent to the subsequent layers.

Block 2: Temporal modeling

Temporal modeling focuses on identification of sequential patterns across frames, which is essential for detecting inconsistencies that arise over time. After the extraction of spatial features, three successive Bi-LSTM layers analyze the pooled feature vectors derived from Block 1. The first Bi-LSTM layer has 128 units and employs a dropout rate of 0.3, maintaining temporal dependencies throughout the sequence. The subsequent Bi-LSTM layer, including 64 units and same dropout rate, enhances these temporal features. Finally, a third Bi-LSTM layer containing 32 units aggregates the temporal information into a compact representation, as illustrated in Fig. 3. By modeling how features evolve across frames, this block learns dynamic signs of forgery that a single frame cannot reveal.

Block 3: Dense layers and output

The dense layers integrate spatial and temporal abstractions into higher-level representations for the final decision. After the Bi-LSTM layers, the network applies a fully connected layer with 512 neurons and a ReLU activation to learn complex interactions suggestive of deepfake artifacts. A dropout layer with rate 0.4 follows further reducing overfitting. The final output is produced by a single neuron with a sigmoid activation, yielding a probability that indicates whether the input video is authentic or manipulated, as illustrated in Fig. 2.

Equation (1) is the mathematical definition of the sigmoid activation function.

$$\sigma(x) = \frac{1}{1+e^{-x}} \tag{1}$$

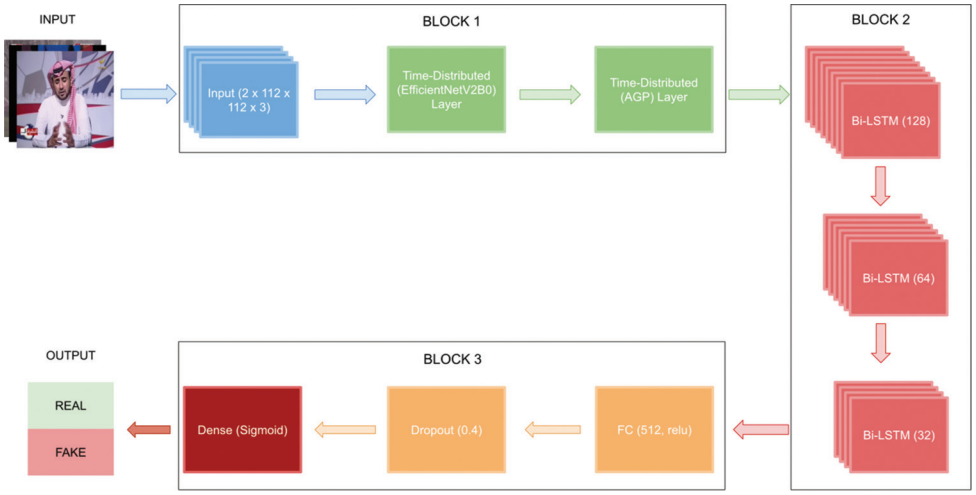

Fig. 2. Spatio-temporal intelligent deepfake detector model architecture.

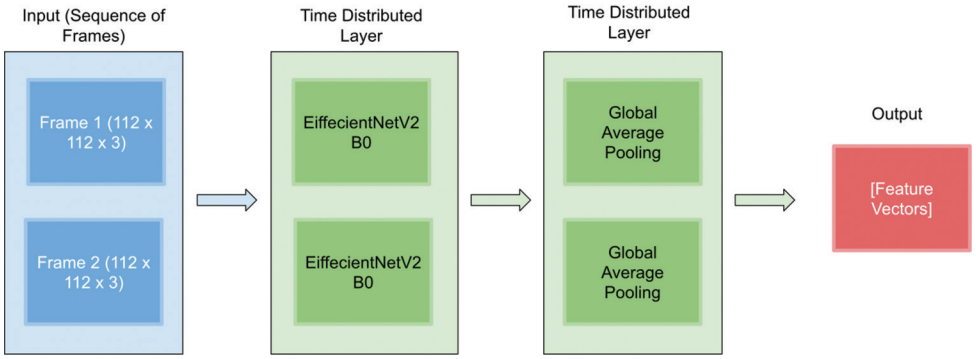

Fig. 3. STIDD block 1 layers.

F. Training method

Meticulous design of training processes is essential for a model to capture significant features and then generalize to new, unseen data. In this work, we employ a custom data pipeline alongside meticulous hyperparameter tuning, aiming to boost the model's learning capacity and improve convergence reliability. Training was carried out using a batch size of 64, selected to optimize computational efficiency while ensuring convergence stability. We used the Adam optimizer, which adapts learning rates dynamically and handles sparse gradients effectively, with an initial learning rate of 0.0001 and minimum learning rate of 1 × 10□□. Binary cross-entropy served as the loss function, which is the negative average of the log of corrected predicted probabilities, aligning with the binary classification objective of distinguishing between genuine and altered videos. The binary cross-entropy loss utilized in the training process is mathematically defined as (2):

$$L = -\frac{1}{N}\sum_{i=1}^{N}\left[y_i \log\left(\dot{y}_i\right) + \left(1-y_i\right)\log\left(1-\dot{y}_i\right)\right] \quad (2)$$

where y_i represents the actual label and $\dot{y}_i$ is the expressed probability for the *i-th* sample.

Training proceeded for up to 20 epochs, during which we monitored validation loss to prevent overfitting. Specifically, we implemented an early stopping criterion that halted training if the validation loss did not improve for five consecutive epochs.

G. Assessment Metrics

True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN) are the numbers of real video sequences correctly detected as real video sequences, fake video sequences correctly detected as fake video sequences, fake video sequences detected as real video sequences, and true video sequences detected as false video sequences, respectively. The binary cross-entropy loss function is used to quantify the error between the predicted probabilities and the actual binary labels, calculated using equation (2)

Equation (3) measures model accuracy as the percentage of accurate predictions.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (3)$$

Precision (Pr) is the ability to properly foresee an event or process result. It is usually the proportion of model or algorithm predictions that are right, calculated using (4).

http://dx.doi.org/10.14500/aro.12190

$$Precision = \frac{TP}{TP + FP} \quad (4)$$

Recall (Re) in machine learning is the capacity of a model to accurately identify all instances of a class in a dataset. It is usually calculated as a ratio of TP predictions (properly recognized class instances) to the dataset's total class instances, calculated using (5).

$$Recall = \frac{TP}{TP + FN} \quad (5)$$

The F1 score measures machine learning model performance. Take the harmonic mean of accuracy and recall. Higher F1 scores imply greater model performance, with 1 being optimum, calculated using (6).

$$F1\,Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (6)$$

IV. Experiments And Results

This section showcases the carried-out tests to assess the performance and efficiency of the STIDD model, addresses the training results and evaluation criteria, the capabilities of the model, and a comparative study with other contemporary architectures. The goal is to show that the suggested method is highly accurate while keeping minimal computational cost, thereby fitting for efficient deepfake detection.

A. Training Results and Evaluation

The model was trained on sequences of 2 frames produced by a sliding window approach with a stride of 1 frame during training. Using the Adam optimizer with a learning rate of 0.0001 and a batch size of 64 the model training executed over the 20 epochs. The training logs demonstrate quite continuous improvement throughout the course of the 20 epochs, as illustrated in Fig. 4.

For instance, as demonstrated from Table II in Epoch 1, the training loss was 0.4124 with 79.7% accuracy, while the validation loss dropped to 0.2653, yielding 88.43% validation accuracy. By Epoch 5, the training loss decreased to 0.1213 (95.15% accuracy) and validation loss fell to 0.1125 (95.57% accuracy). Progress continued steadily: at Epoch 10 the model achieved 98.97% training accuracy (loss 0.0275) and 98.82% validation accuracy (loss 0.0333). By Epoch 15, training accuracy reached 99.55% (loss 0.0118) and validation accuracy was 99.37% (loss 0.0209). The highest validation accuracy occurred at Epoch 20, with 99.52% accuracy (validation loss 0.0193) and 99.79% training accuracy (loss 0.0056). Although minor fluctuations appeared in the later epochs, the parameters corresponding to the lowest validation loss (Epoch 20) were retained for testing, ensuring optimal generalization.

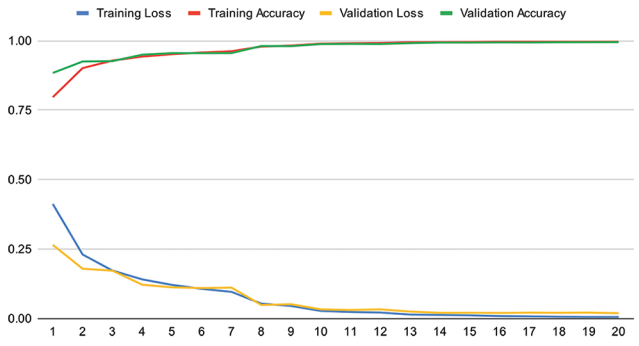

Fig. 4. Spatio-temporal intelligent deepfake detector training performance over 20 epochs.

The evaluation on the test set yielded a loss of 0.0202 and accuracy of 99.51%, demonstrating the effectiveness of our spatio-temporal design across all five FF++ subsets. The detailed classification report from Table III shows precision, recall, and F1-score all at approximately 0.99 for both real and manipulated classes, indicating balanced and robust performance. These exceptional findings validate that the integration of transfer learning, robust spatial feature extraction, excellent temporal modeling, and extensive data augmentation constitutes a high-performance detection system.

B. Ablation Experiment

To assess the contributions of temporal modeling and data augmentation, we evaluated STIDD against two variants of the EfficientNetV2-B0 backbone trained on the same dataset. The first variant omits augmentations, while the second applies our full augmentation pipeline. The backbone variants exhibited clear signs of overfitting: validation accuracy fluctuated dramatically across epochs and never approached the performance seen on the training set, as illustrated in Fig. 5. By contrast, STIDD with its Bi-LSTM layer avoids overfitting and maintains strong generalization.

As shown in Table IV, although both EfficientNetV2-B0 variants achieve only 73.45% (no augmentation) and 78.61% (with augmentation) test accuracy, whereas STIDD reaches 99.51%. These results confirm that the Bi-LSTM temporal fusion in STIDD not only enhances accuracy but also

TABLE II
Selected Training Metrics Across Epochs

Epoch	Training loss	Training accuracy (%)	Validation loss	Validation accuracy (%)
1	0.4124	79.7	0.2653	88.43
5	0.1213	95.15	0.1125	95.57
10	0.0275	98.97	0.0333	98.82
15	0.0118	99.55	0.0209	99.37
20	0.0056	99.79	0.0193	99.52

TABLE III
STIDD Classification Report (Full FF++Test Set)

Class	Precision	Recall	F1-score
Real	0.99	1.0	0.99
Fake	1.0	0.99	0.99
Accuracy			0.99
Macro average	0.99	0.99	0.99
Weighted average	0.99	0.99	0.99

STIDD: Spatio-temporal intelligent deepfake detector

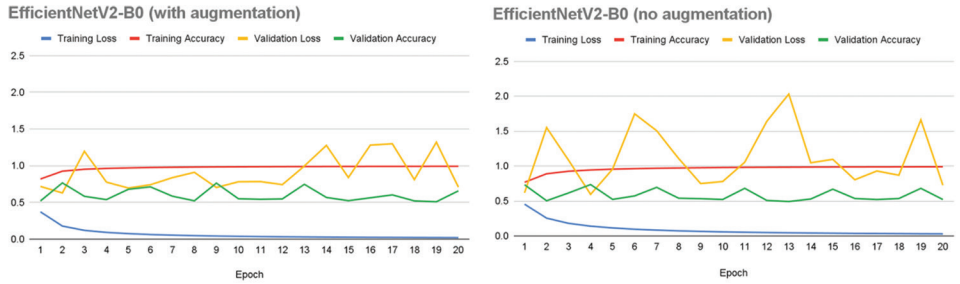

Fig. 5. Training and validation accuracy for EfficientNetV2-B0 baselines.

TABLE IV
Ablation Comparison of STIDD: Impact of Temporal Modeling and Augmentation

Model	Parameters (M)	Accuracy (%)	GFLOPs
STIDD (with Augmentation)	7.6	99.51	0.392
EfficientNetV2-B0 (no augmentation)	7.1	73.45	0.728
EfficientNetV2-B0 (with augmentation)	7.1	78.61	0.728

STIDD: Spatio-temporal intelligent deepfake detector

TABLE V
Comparison of STIDD with other SOTA Models (Same Device and Dataset)

Model	Parameters (M)	Accuracy (%)	GFLOPs	Inference time (ms)
STIDD (ours)	7.6	99.51	0.392	39
EfficientNetV2-B0	7.1	78.61	0.72	45
EfficientNetV2-S	21.5	89.22	8.40	134
MobileNetV3-Large	5.4	71.13	0.22	31

STIDD: Spatio-temporal intelligent deepfake detector

TABLE VI
Comparison of Recent Deepfake Detection Methods on the FaceForensics++

Model	Parameters (M)	Accuracy (%)	GFLOPs
STIDD (ours)	7.6	99.51	0.392
CNNs-Ensemble (Alrajeh and Al-Samawi, 2025)	25.0	95.53	5.0
ViT_CNNs-Ensemble (Alrajeh and Al-Samawi, 2025)	99.7	97.25	20.8
SCViTDW (Li, Zhou and Zhao, 2024)	89.0	99.23	5.9
TALL-Swin (Xu, et al., 2023)	86.0	98.65	47.5
ISTVT (Zhao, et al., 2023)	N/A	99.00	455.8
DepthFake (Maiano, et al., 2023)	N/A	93.00	9.218
DeepFakeNet (Gong, et al., 2021)	10.87	96.69	2.05

STIDD: Spatio-temporal intelligent deepfake detector

mitigates overfitting, and that the model's lower FLOPs stem from its smaller input resolution.

C. Comparative Analysis

The performance and efficiency of STIDD were benchmarked against three lightweight architectures trained and evaluated under identical conditions (same hardware, dataset splits, and preprocessing). Table IV summarizes each model's parameter count, test accuracy on full FF++, computational cost (GFLOPs), and inference time on a single Intel Core Ultra 3 CPU.

As shown in Table V, STIDD attains the highest accuracy (99.51%) with only 7.6 M parameters. Its 0.392 GFLOPs footprint and 39 ms inference time on the Core Ultra 3 CPU demonstrate that the spatio-temporal design is not only more accurate than EfficientNetV2-B0 (78.61% at 0.72 GFLOPs) and MobileNetV3-Large (71.13% at 0.22 GFLOPs) but also highly efficient compared to EfficientNetV2-S (89.22% at 8.40 GFLOPs, 134 ms).

To contextualize STIDD within the broader literature, we compared it to several recent deepfake detection methods that report full FF++ metrics. Table V lists each method's published parameter count, test accuracy on FF++, and GFLOPs. Where a paper did not explicitly state one of these values, "N/A" is shown.

As shown in Table VI, STIDD achieves the highest accuracy (99.51%) with only 0.392 GFLOPs, making it both the most accurate and the most efficient among the compared methods. For example, SCViTDW attains 99.23% accuracy but requires 89 M parameters and 5.9 GFLOPs, while TALL-Swin reaches 98.65% with 86 M parameters and 47.5 GFLOPs. Other models, such as DeepFakeNet (96.69%, 2.05 GFLOPs) and DepthFake (93.00%, 9.218 GFLOPs) trade efficiency for lower accuracy.

V. Challenges and Limitations

While STIDD demonstrates strong performance on the FF++ dataset, its evaluation remains limited to this single synthetic benchmark. Its generalizability across different benchmarks such as Celeb-DF or DFDC has yet to be established, and cross-dataset validation is necessary to assess robustness against unseen manipulation methods. In addition, although STIDD integrates spatial and temporal features efficiently, further reductions in parameter count could enhance adaptability for extremely constrained environments.

VI. Conclusion

This study introduces an efficient spatio-temporal deep learning model for deepfake detection, achieving 99.51% accuracy on the full FF++ dataset (all five subsets). The proposed framework combines spatial feature extraction through a pre-trained EfficientNetV2-B0 backbone with temporal modeling through Bi-LSTM layers. A detailed classification report demonstrates that precision, recall, and

F1-score are all approximately 0.99, confirming that STIDD effectively distinguishes genuine from manipulated videos. By using just 7.6 M parameters and 0.39 GFLOPs, STIDD maintains computational efficiency without compromising accuracy. Compared to recent methods that often require tens of millions of parameters or dozens of GFLOPs, our approach strikes a practical balance between performance and resource usage. Future work will explore additional architectural refinements and efficient inference strategies, as well as broader evaluations on other datasets to enhance robustness and generalizability in the face of evolving deepfake techniques.

References

Al-Adwan, A., Alazzam, H., Al-Anbaki, N., and Alduweib, E., 2024. Detection of deepfake media using a hybrid CNN-RNN model and particle swarm optimization (PSO) algorithm. *Computers*, 13(4), p.99.

Alanazi, F., Ushaw, G., and Morgan, G., 2023. Improving detection of DeepFakes through facial region analysis in images. *Electronics*, 13(1), p.126.

Alhaji, H.S., Celik, Y., and Goel, S., 2024. An approach to deepfake video detection based on ACO-PSO features and deep learning. *Electronics*, 13(12), p.2398.

Alom, M.Z., Taha, T.M., Yakopcic, C., Westberg, S., Sidike, P., Nasrin, M.S., Hasan, M., Van Essen, B.C., Awwal, A.A.S., and Asari, V.K., 2019. A state-of-the-art survey on deep learning theory and architectures. *Electronics*, 8(3), p.292.

Alrajeh, M., and Al-Samawi, A., 2025. Deepfake image classification using decision (Binary) tree deep learning. *Journal of Sensor and Actuator Networks*, 14(2), p.40.

Day, C., 2019. The future of misinformation. *Computing in Science and Engineering*, 21(1), pp.108-108.

Gong, D., Kumar, Y.J., Goh, O.S., Ye, Z., and Chi, W., 2021. DeepfakeNet, an efficient deepfake detection method. *International Journal of Advanced Computer Science and Applications*, 12(6), pp.201-207.

Gupta, G., Raja, K., Gupta, M., Jan, T., Whiteside, S.T., and Prasad, M., 2023. A comprehensive review of DeepFake detection using advanced machine learning and fusion methods, *Electronics*, 13(1), p.95.

Heidari, A. *et al.* (2024) 'Deepfake detection using deep learning methods: A systematic and comprehensive review', *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery,* 14(2), p. e1520. Available at: https://doi.org/10.1002/WIDM.1520.

Hong, Y., Hwang, U., Yoo, J., and Yoon, S., 2019. How generative adversarial networks and their variants work. *ACM Computing Surveys (CSUR)*, 52(1), pp.1-43.

Ibnouzaher, A., and Moumkine, N., 2024 Enhanced Deepfake Detection using a Multi-Model Approach. In: *International Conference on Digital Technologies and Applications*. Vol. 1101, pp.317-325.

Jameel, W.J., Kadhem, S.M., and Abbas, A.R., 2022. Detecting Deepfakes with deep learning and gabor filters. *ARO-the Scientific Journal of Koya University*, 10(1), pp.18-22.

Jung, T., Kim, S., and Kim, K., 2020. DeepVision: Deepfakes detection using human eye blinking pattern. *IEEE Access*, 8, pp.83144-83154.

Kaur, A., Hoshyar, A.N., Saikrishna, V., Firmin, S., and Xia, F., 2024. Deepfake video detection: Challenges and opportunities. *Artificial Intelligence Review*, 57(6), pp.1-47.

Korshunov, P. and Marcel, S. (2022) 'The Threat of Deepfakes to Computer and Human Visions', *Advances in Computer Vision and Pattern Recognition*, pp. 97–115. Available at: https://doi.org/10.1007/978-3-030-87664-7_5.

Li, X., Zhou, H., and Zhao, M., 2024. Transformer-based cascade networks with spatial and channel reconstruction convolution for DeepFake detection. *Mathematical Biosciences and Engineering*, 21(3), pp.4142-4164.

Liu, M.Y., Huang, X., Yu, T.C., and Mallya, A., 2021. Generative adversarial networks for image and video synthesis: Algorithms and applications. *Proceedings of the IEEE*, 109(5), pp.839-862.

Luo, A., Kong, C., Huang, J., Hu, Y., Kang, X., and Kot, A.C., 2024. Beyond the prior forgery knowledge: Mining critical clues for general face forgery detection. *IEEE Transactions on Information Forensics and Security*, 19, pp.1168-1182.

Maiano, L., Papa, L., Vocaj, K., and Amerini, I., 2023. DepthFake: A Depth-Based Strategy for Detecting Deepfake Videos. In: *Lecture Notes in Computer Science* (*Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics*). Vol. 13646. LNCS. Springer, Berlin, pp.17–31.

Mitra, A., Mohanty, S.P., Corcoran, P., and Kougianos, E., 2022. EasyDeep: An IoT friendly robust detection method for GAN Generated deepfake images in social media. In: *IFIP Advances in Information and Communication Technology*. Vol. 641. Springer, Berlin, pp.217-236.

Mubarak, R., Alsboui, T., Alshaikh, O., Inuwa-Dutse, I., Khan, S., and Parkinson, S., 2023. A survey on the detection and impacts of deepfakes in visual, audio, and textual formats. *IEEE Access*, 11, pp.144497-144529.

Pawelec, M., and Pawelec Mariapawelec, M., 2022. Deepfakes and democracy (Theory): How synthetic audio-visual media for disinformation and hate speech threaten core democratic functions. *Digital Society*, 1(2), pp.1-37.

Qadir, A., Mahum, R., El-Meligy, M.A., Ragab, A.E., AlSalman, A., and Awais, M., 2024. An efficient deepfake video detection using robust deep learning. *Heliyon*, 10(5), p.e25757.

Rossler, A., Davide, C., Luisa, V., Christian, R., Justus, T., and Matthias, N., 2019. FaceForensics++: Learning to Detect Manipulated Facial Images. In: *IEEE/CVF International Conference on Computer Vision (ICCV)*, pp.1-11.

Shahzad, H.F., Rustam, F., Flores, E.S., Luís Vidal Mazón, J., De la Torre Diez,I., and Ashraf, I., 2022. A review of image processing techniques for DeepFakes. *Sensors*, 22(12), pp.4556-4584.

Sridevi, K., Kanaprthi, S.K., Sameera, N., Garapati, Y., Krishnamadhuri, D., and Bethu, S., 2022. 'IoT Based Application Designing of Deep Fake Test for Face Animation. *ACM International Conference Proceeding Series*, pp.24-30.

Xia, Z., Qiao, T., Xu, M., Wu, X., Han, L., and Chen, Y., 2022. Deepfake video detection based on mesonet with preprocessing module. *Symmetry*, 14(5), p.939.

Xu, Y., Liang, J., Jia, G., Yang, Z., Zhang, Y., and Ran, H., 2023. TALL: Thumbnail Layout For Deepfake video detection. In: *Proceedings of the IEEE International Conference on Computer Vision*, pp.22601-22611.

Zhao, C., Wang, C., Hu,G., Chen, H., Liu, C., and Tang, J., 2023. ISTVT: Interpretable spatial-temporal video transformer for deepfake detection. *IEEE Transactions on Information Forensics and Security*, 18, pp.1335-1348.