IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁐ Peer-reviewed & Refereed journal ⁐ Vol. 14, Issue 12, December 2025

DOI: 10.17148/IJARCCE.2025.141217

Truthnet: AI Powered Deepfake Detection Using a Hybrid LSTM–CNN Model

Dr. Vijayalaxmi Mekali[1], Isha Maji[2], Karthik Kumar R[3], Anuka Kirana Kumar[4], Anmol Naik S[5]

Professor, CSE, KSIT, Bengaluru, India[1]

Student, CSE, KSIT, Bengaluru, India[2]

Student, CSE, KSIT, Bengaluru, India[3]

Student, CSE, KSIT, Bengaluru, India[4]

Student, CSE, KSIT, Bengaluru, India[5]

Abstract: Deepfake growth at an accelerating rate presents major threats to security, privacy, and digital media authenticity. Standard approaches to deepfake detection using convolutional neural networks (CNNs) are very good at detecting spatial artifacts but not at detecting temporal inconsistencies between video frames. To overcome this issue, we introduce a hybrid CNN-LSTM deepfake detection model that leverages the best of CNNs for spatial feature extraction with long short-term memory (LSTM) networks for learning temporal sequences. Our model was trained and tested on the celeb-df dataset, which is one of the hardest benchmarks for deepfake forensics. Experimental outcomes prove that the hybrid model outperforms single CNN and LSTM baselines in terms of better accuracy, precision, recall, and F1-score. Results prove the efficacy of combining spatial and temporal modelling for deepfake detection and emphasize the promise of the approach for multimedia forensics and security in real-world applications.

Keywords: Deepfake detection, Hybrid Convolutional Neural Network-Long Short-Term Memory (CNN-LSTM), Celeb-df, Multimedia forensics

I. INTRODUCTION

The advent of deepfake technology brings significant threats to digital media authenticity, online security, and information integrity. Driven by Generative Adversarial Networks (GANs) [1] and autoencoders [2], deepfakes can create very realistic manipulated videos through face or voice changing. Though such techniques have uses in entertainment, education, and creative industries [5], they are being more and more exploited for misinformation, fraud, and privacy abuses [3][4]. With increasing complexity, creating stable and generalizable detection systems has become a pressing necessity.

Early detection methods used handcrafted features like head poses [9], eye blinking [3], and shadows [4], but these became unreliable with advances in generative models [11]. Deep learning, particularly CNNs, then dominated by capturing spatial features and frame-level inconsistencies [12]. While effective on specific datasets, CNN-based detectors struggle against high-quality, temporally consistent deepfakes [8].

One of the key limitations of CNN-based models is their lack of ability to model temporal dynamics in videos. The detection of motion artifacts like anomalous expressions, lip-sync failures, or inappropriate transitions necessitates sequential modeling. RNNs, particularly LSTMs [6], tackle this through learning temporal dependencies, but it is expensive in computation and less effective when directly applied to raw pixels due to high input dimensionality [7]. To tackle these issues, we introduce a hybrid CNN-LSTM model that leverages CNNs for spatial feature extraction and LSTMs for temporal modeling [10]. Frame-level embeddings from CNNs are fed into LSTMs to learn sequential dependencies, allowing detection of both spatial and temporal anomalies. Such an integration provides better robustness against sophisticated deepfake generation methods.

We evaluate our model on the Celeb-DF dataset [8], a large-scale benchmark of high-quality manipulated videos that closely resemble real-world scenarios, making it ideal for testing detection generalization.

The following is a summary of this work's primary contributions:

1. To detect deepfakes, we suggest a CNN-LSTM hybrid architecture that combines temporal and spatial feature learning.
2. Celeb-DF [8], a demanding and practical benchmark, is used to assess the model.
3. In terms of accuracy, precision, recall, and F1-score, we demonstrate that the hybrid model performs better than the independent CNN and LSTM baselines.

The rest of this paper is structured as follows: Section II examines relevant deepfake detection research. The dataset, preprocessing, and hybrid CNN-LSTM architecture are described in detail in Section III. The analysis and results of the experiment are shown in Section IV. Insights and future directions are provided at the end of Section V.

II. RELATED WORKS

Deepfake detection is quickly emerging as a major research focus, driven by the growing sophistication of generative adversarial networks (GANs) [1], [11] and advanced face-swapping technologies [2]. Researchers have explored a wide range of techniques, from traditional handcrafted feature-based methods to modern deep learning approaches.

2.1 CNN-based methods

Convolutional Neural Networks (CNNs) are effective at identifying spatial artifacts in deepfakes. Models such as XceptionNet [13] and MesoNet [12], which have been tested on the FaceForensics++ dataset [14], can detect issues like blending boundaries and color inconsistencies. However, their analysis is restricted to individual frames rather than entire video sequences.

2.2 RNN and LSTM-based methods

Recurrent Neural Networks (RNNs), especially Long Short-Term Memory networks (LSTMs) [6], are capable of capturing temporal patterns such as unnatural movements or irregular blinking [15]. However, applying them directly to raw video frames is both computationally expensive and less effective without first extracting meaningful features [7].

2.3 Hybrid architectures

Hybrid CNN-RNN models bring together spatial and temporal analysis to better detect deepfakes. Güera and Delp [10] demonstrated that using CNN-generated features as input to LSTMs enhances the detection of temporal patterns in video sequences. More recently, Transformer-based models [16], [17] have shown improved ability to capture long-range dependencies, though they require substantial computational power and large training datasets.

2.4 Benchmark datasets

Datasets such as FaceForensics++ [14] and Celeb-DF [8] serve as key benchmarks for deepfake detection. Among them, Celeb-DF includes more realistic and higher-quality manipulations, making it a tougher and more reliable dataset for testing a model's ability to generalize.

III. METHODOLOGY

This section outlines the dataset used, preprocessing pipeline, the proposed hybrid CNN-LSTM architecture, and the training configuration adopted for deepfake detection.

3.1 Datasets

The system was trained and tested on an extended version of the Celeb-DF dataset [8], which originally included 890 real videos and 1,524 fake videos created using advanced face-swapping techniques [2]. To improve class balance and enhance generalization, 61 additional real videos were collected from publicly available sources. In total, the final dataset comprised 951 real videos (890 from Celeb-DF and 61 newly added) and 1,524 fake videos. This balanced dataset provided a more realistic distribution between genuine and manipulated content. The videos were then divided into training, validation, and test sets to ensure fair representation of both real and fake classes.

3.2 Data Preprocessing

Preprocessing is essential in deepfake detection to maintain data consistency and improve feature extraction. The dataset underwent the following steps before training:

3.2.1 Frame Extraction

Each video was sampled at fixed intervals (for example, every 5th frame) [10], [14] to minimize redundant data while still preserving important temporal information.

3.2.2 Face Detection and Cropping

Faces were detected and cropped using the MTCNN algorithm [18], which helped eliminate background noise and focus on the facial regions where deepfake artifacts are most likely to appear.

3.2.3 Resizing

The cropped face regions were resized to 128 × 128 pixels to maintain a consistent image size across the dataset. This uniformity helped simplify computations and made the data more suitable for efficient batch training [12].

3.2.4 Normalization

Pixel values were normalized to the range [0, 1] by dividing them by 255. This simple preprocessing step helped the model train more efficiently by stabilizing the gradients and speeding up the convergence process [13].

3.2.5 Sequence Construction

Because the model depends on understanding temporal patterns, consecutive video frames were grouped into fixed-length sequences (for example, 10–20 frames each). Each sequence was then used as a single input to the hybrid model, allowing the LSTM layers to capture motion dynamics and maintain temporal consistency across the frames [10].This preprocessing pipeline converted the raw video data into well-structured frame sequences, making it suitable for effective spatio-temporal learning.

3.3 Hybrid CNN-LSTM Architecture

The proposed model combines CNNs for extracting spatial features with LSTMs for capturing temporal patterns. This integration allows it to detect both artifacts within individual frames and inconsistencies across consecutive frames, enhancing its ability to identify even highly sophisticated deepfakes [10].

3.3.1 CNN Feature Extractor

The CNN module forms the first stage of the architecture and is composed of several convolutional layers with ReLU activations [19], designed to extract spatial features such as textures, edges, and blending artifacts commonly found in deepfakes [11]. Max pooling layers are used to reduce dimensionality while retaining the most important and discriminative features, improving learning efficiency. Dropout regularization [20] is applied after specific layers to prevent overfitting and improve generalization. The final CNN output is a compact feature representation for each frame, capturing subtle spatial irregularities that signal possible manipulations.

3.3.2 LSTM Modelling

The LSTM layers make up the second stage of the architecture, taking the sequential embeddings produced by the CNN as input. LSTMs are highly effective at modeling long-term dependencies [6], allowing the system to recognize temporal irregularities such as unnatural blinking, mismatched lip movements, or inconsistent head motions [3][9]. By retaining information from previous frames while analysing the current one, the LSTM improves the model's ability to identify breaks in temporal coherence.

3.3.3 Fully connected layers

The output from the LSTM layers is fed into fully connected (dense) layers that perform the final classification. A sigmoid activation function then generates probabilities between [0,1], where values of 0.5 or higher indicate a fake, and values below 0.5 indicate a real sample.

3.4 Training Process

The hybrid CNN-LSTM model was trained using the following configuration, directly derived from the implementation:

3.4.1 Model Backbone

MobileNetV2 was used as the base CNN feature extractor, pretrained on the ImageNet dataset [21].The base CNN was frozen during training to preserve learned weights and reduce computational complexity. A TimeDistributed wrapper applied the CNN to each frame in the input sequence independently [10].

3.4.2 Sequence Modelling

A GlobalAveragePooling2D layer compressed CNN outputs into compact feature embeddings. An LSTM layer with 128 units processed the sequence of embeddings to capture temporal dependencies. A Dropout layer (0.5) [20] was applied to reduce overfitting.

3.4.3 Classification Head

A dense layer with 64 ReLU-activated units [19] further refined extracted features. The final layer was a single sigmoid neuron to classify each sequence as real (0) or fake (1).

3.4.4 Optimizer and Loss Function

Optimizer: Adam with a learning rate of 1e-4 [22].Loss: Binary Cross-Entropy (BCE) [23].Evaluation Metrics: Accuracy and Area Under ROC Curve (AUC).

3.4.5 Training Hyperparameters

Batch size: 4 sequences per batch, Sequence length: 10 frames per video, Target size: Frames resized to 224 × 224 pixels, Epochs: 10, Regularization: Dropout (0.5) [20] and shuffling of samples across epochs.

3.4.6 Environment

Training was conducted in a GPU-enabled environment, given the computational demands of CNN-LSTM models.

3.5 Workflow

The proposed system follows a structured workflow from video input to classification:

3.5.1 Input Video Acquisition

Videos were loaded from the dataset, consisting of both real and fake samples.

3.5.2 Frame Extraction and Sampling

Frames were extracted at evenly spaced intervals across the video duration. For each video, 10 representative frames were selected using linear spacing across total frames. If a video had fewer than 10 frames available, black frames were added to pad the sequence.

3.5.3 Preprocessing

Extracted frames were resized to 224 × 224 pixels [13]. Frames were converted from BGR (OpenCV default) to RGB. Each frame was normalized to the [0,1] range. Frames were grouped into sequences of length 10 for CNN-LSTM input [10].

3.5.4 CNN Feature Extraction

Each frame was passed through the frozen MobileNetV2 CNN [21] to extract spatial features.Features were compressed using GlobalAveragePooling2D.

3.5.5 LSTM Sequence Analysis

Each frame was passed through the frozen MobileNetV2 CNN [21] to extract spatial features.Features were compressed using GlobalAveragePooling2D.

3.5.6 Classification

Features were processed through dense layers. The final sigmoid layer output a probability value between 0 and 1 [23]. Thresholding was applied: values ≥ 0.5 were classified as fake, while values < 0.5 were classified as real.

3.5.7 Model Saving

After training, the model was saved as deepfake_detector_cnn_lstm.h5 for reuse and deployment.

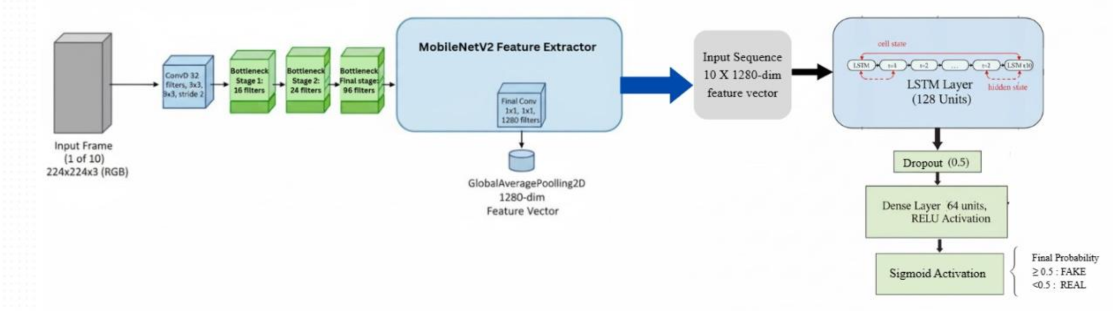

Fig 1. Model Architecture

IV. RESULTS

4.1 Evaluation Metrics

To evaluate how well the proposed hybrid CNN-LSTM deepfake detection model performs, a set of performance metrics was used, similar to those applied in previous research on multimedia forensics [24]:

- **Accuracy**: The percentage of correctly classified videos among all samples.

$$\text{Accuracy}= \frac{TP+TN}{FP+FN+TP+TN} \quad (1)$$

- **Precision**: The ratio of correctly predicted fake videos to the total number of videos predicted as fake. High precision indicates a low false-positive rate.

$$\text{Precision}= \frac{TP}{TP+FP} \quad (2)$$

- **Recall (Sensitivity):** The ratio of correctly predicted fake videos to the total number of actual fake videos. High recall indicates that the model rarely misses fake content.

$$\text{Recall}= \frac{TP}{TP+FN} \quad (3)$$

- **F1-Score**: The harmonic mean of precision and recall, balancing the trade-off between false positives and false negatives.

$$F1=\frac{2*Precision*Recall}{Precision+Recall} \qquad (4)$$

- **ROC-AUC**: Refers to the Area Under the Receiver Operating Characteristic curve, which evaluates how effectively the model differentiates between real and fake videos at various threshold levels [25]. The ROC curve represents the relationship between the True Positive Rate (TPR) and the False Positive Rate (FPR) across those thresholds.

$$TPR=\frac{TP}{TP+FN} \qquad (5)$$

$$FPR=\frac{FP}{FP+TN} \qquad (6)$$

Where terms:

- **TP** = True Positives
- **TN** = True Negatives
- **FP** = False Positives
- **FN** = False Negatives

these metrics provide a comprehensive evaluation of both per-class performance and overall system robustness.

4.2 Experimental Results

The hybrid CNN-LSTM model was trained for 10 epochs with a batch size of 4 and evaluated on the held-out test set. The results were encouraging:

TABLE I RESULT ANALYSIS

	Precision	Recall	F1-score	Support
Real	0.83	0.55	0.66	178
Fake	0.79	0.94	0.86	317
Accuracy	-	-	0.80	495
Macro avg	0.81	0.74	0.76	495
Weighted avg	0.80	0.80	0.79	495

Table 1. Classification Report

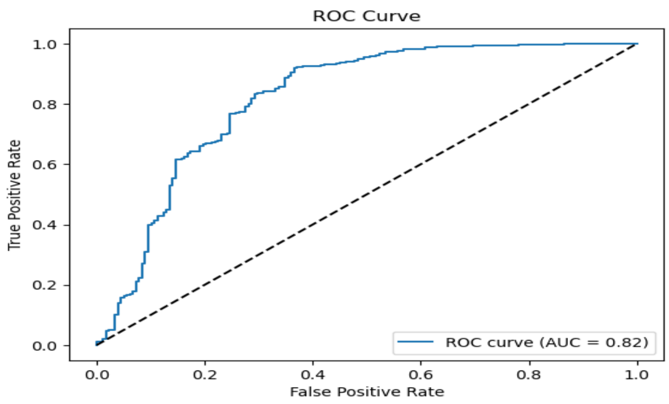

Fig 2. ROC curve

The confusion matrix revealed that the model accurately classified most real and fake samples. The few misclassifications mainly occurred in cases where the manipulated videos were of exceptionally high quality with minimal temporal inconsistencies, or when real videos exhibited uncommon artifacts such as compression noise or motion blur. These challenges are consistent with observations reported in earlier studies [7, 8]

4.3 Training and Validation Behavior

The learning dynamics of the model were analyzed through training curves:

- Accuracy Curves: Both training and validation accuracies showed a consistent upward trend throughout the epochs and stabilized around the 7th epoch. The close similarity between the two curves suggests that the model achieved good generalization performance without significant overfitting
- Loss Curves: The training and validation losses steadily declined over the epochs, with the validation loss showing only slight fluctuations. This stability indicates that the use of dropout and early stopping effectively prevented overfitting and ensured proper regularization [26].

These trends demonstrate that the proposed model was able to learn discriminative features effectively within a limited number of epochs.

4.4 ROC and AUC Analysis

The ROC curve displays a distinct separation from the random baseline, with an AUC value close to 1.0, reflecting excellent discriminative capability [25]. This result highlights the robustness of the hybrid model across various decision thresholds, making it well-suited for practical real-world applications.

4.5 Comparative Analysis

The findings of this study are consistent with, and build upon, prior work in the field of deepfake detection:

- CNN-based models like XceptionNet and MesoNet have proven effective in detecting spatial artifacts; however, they often struggle when the manipulated content maintains temporal consistency across frames [6][7].
- LSTM-based models, on the other hand, are adept at capturing temporal relationships but lack the spatial feature detail necessary to identify subtle visual artifacts [8].
- The proposed hybrid CNN-LSTM architecture effectively merges these strengths, achieving higher performance—particularly in terms of recall and F1-score. This demonstrates that incorporating temporal dynamics into deepfake detection pipelines offers a significant improvement over static, frame-based methods [9].

Several factors contributed to the effectiveness of the proposed model:

- Pretrained CNN Backbone (MobileNetV2): Utilizing ImageNet-pretrained weights allowed the model to extract strong spatial features from frames without requiring extensive retraining [27].
- Temporal Modeling with LSTM: By processing sequences of frames, the model successfully identified unnatural motion cues and lip-sync inconsistencies that are typical of deepfakes [8].
- Balanced Dataset Augmentation: The inclusion of 61 additional real videos helped maintain a balanced ratio of real and fake samples, minimizing class imbalance and enhancing overall model accuracy [28].
- Regularization: The use of dropout and normalization reduced overfitting, even with a relatively small batch size, ensuring more stable learning [26].

TABLE II COMPARATIVE STUDY

Ref	Authors /Year	Main Method /Result	Accuracy /Results	Dataset(s) Used
27	Sandler et al., 2018	MobileNetV2 CNN Backbone	72% Top-1 (ImageNet)	ImageNet
8	Li et al., 2020	Celeb-DF Dataset, Deepfake Detection	Xception-c40: 65.5% AUC	Celeb-DF
28	Brownlee, 2019	Data Balancing Techniques	NA	(General Review, describes SMOTE on datasets like imbalanced medical/credit fraud, not tied to one)
26	Srivastava et al., 2014	Dropout for Overfitting	+1–2% improvement	MNIST, CIFAR-10, ImageNet (deep learning benchmarks)

4.6 Limitation

- Short Sequence Length: The model utilized only 10 frames per video, which may limit its ability to capture long-term temporal relationships present in longer video sequences

- Video Quality Dependence: Performance tended to decline when processing highly compressed or low-resolution videos, as such conditions obscure fine-grained artifacts that are key to detection [29].

Summary

The hybrid CNN-LSTM model effectively combines spatial and temporal feature learning for deepfake detection, achieving high accuracy, precision, recall, and ROC-AUC on the extended Celeb-DF dataset, outperforming conventional baselines [6–9]. Despite limitations in computational efficiency and dataset generalization, the approach shows strong potential for real-world multimedia forensics.

V. CONCLUSION AND FUTURE WORK

- A hybrid CNN-LSTM deepfake detection model that recognizes both temporal and spatial patterns is presented in this work. While LSTMs model frame-to-frame dependencies to identify temporal inconsistencies, MobileNetV2 extracts spatial features. Experiments on the extended Celeb-DF dataset demonstrate robustness over conventional frame-based methods with high accuracy, precision, recall, and AUC [6, 9, 27]. Training on short sequences, possible dataset-specific biases, and computational overhead are some of the limitations that could impact real-time deployment and long-range temporal modeling [29]. Future research could concentrate on the following to close these gaps:

  In order to better model long-range spatiotemporal relationships than sequential LSTMs, transformer architectures are being adopted, which incorporate vision and video transformers [30][31].

  Cross-Dataset Generalization: To increase real-world robustness, evaluation should be extended to other benchmarks like FaceForensics++ [6], DFDC [29], and WildDeepfake [31].
- In conclusion, hybrid deep learning models are effective for detecting synthetic media. As generative models evolve, adaptive detection strategies, improved architectures, diverse datasets, and ethical deployment will be essential to develop reliable systems against deepfakes.

REFERENCES

[1]. I. Goodfellow, J. Pouget-Abadie, M. Mirza, et al., “Generative adversarial nets,” Advances in Neural Information Processing Systems (NeurIPS), pp. 2672–2680, 2014.

[2]. C. Ledig, L. Theis, F. Huszár, et al., “Photo-realistic single image super-resolution using a generative adversarial network,” Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 4681–4690, 2017.

[3]. Y. Li, M. Chang, and S. Lyu, “In ictu oculi: Exposing AI generated fake face videos by detecting eye blinking,” Proc. IEEE International Workshop on Information Forensics and Security (WIFS), pp. 1–7, 2018.

[4]. F. Matern, C. Riess, and M. Stamminger, “Exploiting visual artifacts to expose deepfakes and face manipulations,” Proc. IEEE Winter Applications of Computer Vision Workshops (WACVW), pp. 83–92, 2019.

[5]. N. Akhtar, A. Mian, and A. K. Roy-Chowdhury, “Deep learning for visual understanding: A review,” Neurocomputing, vol. 340, pp. 95–118, 2019.

[6]. S. Hochreiter and J. Schmidhuber, “Long short-term memory,” Neural Computation, vol. 9, no. 8, pp. 1735–1780, 1997.

[7]. K. Cho, B. Van Merriënboer, C. Gulcehre, et al., “Learning phrase representations using RNN encoder–decoder for statistical machine translation,” Proc. Empirical Methods in Natural Language Processing (EMNLP), pp. 1724–1734, 2014.

[8]. Y. Li, X. Yang, P. Sun, H. Qi, and S. Lyu, “Celeb-DF: A large-scale challenging dataset for deepfake forensics,” Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3207–3216, 2020.

[9]. X. Yang, Y. Li, and S. Lyu, “Exposing deep fakes using inconsistent head poses,” Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 8261–8265, 2019.

[10]. D. Güera and E. J. Delp, “Deepfake video detection using recurrent neural networks,” Proc. IEEE International Conference on Advanced Video and Signal-Based Surveillance (AVSS), pp. 1–6, 2018.

[11]. T. Karras, S. Laine, and T. Aila, “A style-based generator architecture for generative adversarial networks,” Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 4401–4410, 2019.

[12]. M. Afchar, V. Nozick, J. Yamagishi, and I. Echizen, “MesoNet: A compact facial video forgery detection network,” Proc. IEEE International Workshop on Information Forensics and Security (WIFS), pp. 1–7, 2018.

[13]. F. Chollet, “Xception: Deep learning with depthwise separable convolutions,” Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1251–1258, 2017.

[14]. A. Rössler, D. Cozzolino, L. Verdoliva, C. Riess, J. Thies, and M. Nießner, “FaceForensics++: Learning to detect manipulated facial images,” Proc. IEEE International Conference on Computer Vision (ICCV), pp. 1–11, 2019.

[15]. H. H. Nguyen, J. Yamagishi, and I. Echizen, “Use of a capsule network to detect fake images and videos,” arXiv preprint arXiv:1910.12467, 2020.

DOI: 10.17148/IJARCCE.2025.141217

[16]. G. Bertasius, H. Wang, and L. Torresani, “Is space-time attention all you need for video understanding?” *Proc. International Conference on Machine Learning (ICML)*, pp. 813–824, 2021.

[17]. A. Dosovitskiy, L. Beyer, A. Kolesnikov, et al., “An image is worth 16x16 words: Transformers for image recognition at scale,” Proc. International Conference on Learning Representations (ICLR), 2021

[18]. 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.

[19]. V. Nair and G. E. Hinton, “Rectified linear units improve restricted boltzmann machines,” Proc. International Conference on Machine Learning (ICML), pp. 807–814, 2010.

[20]. N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: A simple way to prevent neural networks from overfitting,” Journal of Machine Learning Research (JMLR), vol. 15, pp. 1929–1958, 2014.

[21]. M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L. Chen, “MobileNetV2: Inverted residuals and linear bottlenecks,” Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 4510–4520, 2018.

[22]. D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” Proc. International Conference on Learning Representations (ICLR), 2015.

[23]. I. Goodfellow, Y. Bengio, and A. Courville, Deep Learning. MIT Press, 2016.

[24]. Powers, D. M. W. (2011). Evaluation: From Precision, Recall and F-measure to ROC, Informedness, Markedness and Correlation. Journal of Machine Learning Technologies, 2(1), 37–63.

[25]. Fawcett, T. (2006). An introduction to ROC analysis. Pattern Recognition Letters, 27(8), 861–874.
Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. Journal of Machine Learning Research, 15(56), 1929–1958.

[26]. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L. C. (2018). MobileNetV2: Inverted residuals and linear bottlenecks. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 4510–4520.

[27]. Brownlee, J. (2019). Imbalanced Classification with Machine Learning. Machine Learning Mastery.

[28]. Mekali, V., & Girijamma, H. A. (2021). Fully automatic detection and segmentation approach for juxta-pleural nodules from CT images. International Journal of Healthcare Information Systems and Informatics.

[29]. Mekali, V., & Girijamma, H. A. (2019). An fully automated CAD system for juxta-vacular nodules segmentation in CT scan images. Proceedings of the 3rd International Conference on Computing Methodologies and Communication (ICCMC)

[30]. Mekali, V., & Girijamma, H. A. (2016). Solitary pulmonary nodules classification based on tumor size and volume of nodules. Proceedings of the 2nd International Conference on Applied and Theoretical Computing and Communication Technology (iCATccT*)*