https://doi.org/10.31436/iiumej.v26i2.3287

ENHANCING ANOMALY DETECTION PERFORMANCE: DEEP LEARNING MODELS EVALUATION

YUNUSA MOHAMMED JEDDAH[1*], AISHA HASSAN ABDALLA HASHIM[1,2*], OTHMAN OMRAN KHALIFA[1], KHMAIES OUAHADA[2]

[1]*Dept. of Electrical and Computer Engineering, International Islamic University Malaysia, Malaysia*
[2]*Dept. of Electrical and Electronic Engineering Science, University of Johannesburg, South Africa*

**Corresponding authors: yunusmj2@hotmail.com, aisha@iium.edu.my*

(Received: 8 May 2024; Accepted: 10 March 2025; Published online: 15 May 2025)

ABSTRACT: Detection of anomalies within video streams continues to be challenging, mostly due to the complexities involved in distinguishing abnormal activities from normal ones. This study aimed to enhance anomaly detection performance by evaluating different deep learning models and optimizers. Utilizing the Keras framework and Python on a Kaggle notebook, the experiment explored the effectiveness of DenseNet121, VGG19, ResNet50, and InceptionV3 models in conjunction with Adam, SGD, RMSprop, and Adagrad optimizers. A UCF Crimes dataset subset focused on Accuracy, F1 Score, and AUC evaluation metrics. The results establish that the InceptionV3 model paired with the Adam optimizer outperforms the other combinations, attaining AUC scores of 0.9918. In contrast to other state-of-the-art models such as DenseNet121 and ResNet50, InceptionV3 presents enhanced precision and adaptability in handling the variability found in video anomaly datasets. This study enhances security by providing insights into enhanced model-optimizer combinations, advancing video surveillance approaches, and providing support for developing robust anomaly detection systems.

ABSTRAK: Pengesanan anomali dalam strim video terus mencabar, kebanyakan disebabkan oleh kerumitan yang terlibat dalam membezakan aktiviti tidak normal dari biasa. Kajian ini cuba meningkatkan prestasi pengesanan anomali dengan menilai model dan pengoptimum pembelajaran mendalam yang berbeza. Menggunakan rangka kerja Keras dan Python pada komputer riba Kaggle, eksperimen ini meneroka keberkesanan model DenseNet121, VGG19, ResNet50 dan InceptionV3 bersama pengoptimum Adam, SGD, RMSprop dan Adagrad. Subset data Jenayah UCF digunakan, memfokuskan pada ketepatan, Skor F1 dan metrik penilaian AUC. Dapatan kajian menunjukkan bahawa model InceptionV3 bersama pengoptimum Adam, mengatasi kombinasi lain, mencapai skor AUC 0.9918. Berbeza dengan model canggih lain seperti DenseNet121 dan ResNet50, InceptionV3 mempunyai ketepatan dan kebolehsuaian yang tinggi dalam mengendalikan kebolehubahan yang terdapat dalam set data anomali video. Kajian ini menyumbang kepada peningkatan keselamatan dengan memberi gabungan pengoptimum bersama model yang dipertingkatkan, memajukan pendekatan pengawasan video dan menyediakan sokongan bagi pembangunan sistem pengesanan anomali yang teguh.

KEYWORDS: *Deep learning models, Video anomaly detection, Optimization Techniques, Video Surveillance, Performance Evaluation*

1. INTRODUCTION

Modern surveillance systems have become increasingly indispensable for protecting the public, especially in anomaly detection or suspicious behavior in videos. Technological

https://doi.org/10.31436/iiumej.v26i2.3287

advancements have transformed video surveillance into a component of security systems, providing critical functions in public places, transportation hubs, and other critical infrastructure. Anomalies in video streams, like unattended baggage or abnormal human behaviours [1], can hint at potential threats that demand immediate attention. Swift and precise detection of these abnormalities can prevent minor disruptions from significant safety breaches. However, real-time monitoring of video feeds is an overwhelming undertaking for human operatives, especially in dynamic and crowded environments. Automating video anomaly detection systems enhances security personnel's ability to identify potential threats in real time, where manual monitoring is challenging [2].

Despite its cybersecurity roots, the term "anomaly detection" has been widely employed in video analysis. Numerous studies have been conducted on anomaly detection in videos, highlighting its applicability in circumstances such as detecting suspicious behaviour [3], monitoring traffic violations [4], and identifying harmful objects in sensitive places [5]. However, video anomaly detection faces unique challenges, such as anomalies' rarity, appearance variability, and the complexity of describing abnormal behaviours [[6]]. These challenges necessitate employing adaptable and robust techniques to address the complexity and imbalances in video data.

This study addresses these concerns by evaluating the performance of four cutting-edge deep learning models — ResNet50, DenseNet121, VGG19, and InceptionV3 — for video anomaly detection. In addition, we evaluate the effects of four optimization algorithms — SGD, RMSprop, Adam, and Adagrad—on the models' performance. Using Python and the Keras framework on the Kaggle notebook, this study analyses key evaluation metrics, such as accuracy, F1 Score, and AUC, to reveal the most efficient model-optimizer pairings. By investigating these combinations, we aim to provide insights into enhancing the reliability and accuracy of video anomaly detection systems.

This study's primary contributions are as follows:

- A comparative assessment of four popular deep learning models for video anomaly recognition.
- An investigation of the impact of four different optimization algorithms on these models' performance.
- Optimal model-optimizer combinations identification to address variability and imbalance issues in video data.
- Providing insights to develop robust anomaly detection systems that can handle real-world scenarios.

The remaining sections of this paper are arranged as follows: The Related Works section explores current studies, focusing on advancements in research in video anomaly detection. The Method section covers the experimental setup, dataset preparation, preprocessing, and model selection. The Results section provides the findings and compares the models' performance. The Discussion section presents the interpretation of the results in context, and the Conclusion section highlights key insights, limitations, and proposes future work.

2. RELATED WORKS

Video anomaly detection has attracted much attention thanks to its significance in surveillance and security systems. Recent advances in deep learning enable the development of powerful models trained to recognize anomalies in complex video data. However, the choice

https://doi.org/10.31436/iiumej.v26i2.3287

of optimization techniques has major effects on their performance. This section reviews recent studies evaluating model-optimizer combinations for video anomaly detection.

2.1. Technology in Anomaly Detection Using Deep Learning

Deep learning models, especially CNNs, have successfully detected video anomalies. Pre-trained models such as ResNet [17], DenseNet [16], and InceptionV3 [20] have been widely used to extract hierarchical features from video frames. Wu et al. [13] assessed pre-trained CNNs on the UCSD dataset, revealing that transfer learning notably enhances anomaly detection efficiency. Wang et al. [9] also offered spatiotemporal improvements to correct the data imbalance, attaining state-of-the-art results on benchmark datasets such as UCSD Ped2 and Avenue.

Optimization methods are essential for training deep learning models. Stochastic Gradient Descent (SGD) is a primary optimization technique; however, it frequently has difficulties with noisy gradients and poor convergence. Adaptive optimizers such as Adam, RMSprop, and Adagrad mitigate these restrictions by dynamically modifying learning rates. Lydia and Francis [14] extensively assessed optimization approaches, emphasizing Adam's advantages in managing large-scale datasets with sparse gradients. Pawar and Attar [8] compared Adam and RMSprop for video anomaly detection, indicating that Adam attains rapid convergence and better accuracy.

Recent studies demonstrated how important it is to pair specific models with optimizers suited for their designs. For example, InceptionV3, with factorized convolutions and supplementary classifiers, gains substantial benefits from Adam's adaptive learning rates [20]. Conversely, ResNet50, which utilizes residual connections, demonstrates effective performance with both Adam and RMSprop, thanks to its ability to cope with vanishing gradients [17]. DenseNet121, characterized by its dense connections, demonstrates uniform performance across various optimizers, attaining optimal results with RMSprop [16]. These outcomes demonstrate reasons for the rigorous evaluation of model-optimizer pairings to optimize performance.

Numerous other recent studies have explored various methods for detecting unusual or suspicious activity in video streams, establishing a foundation for automated security solutions. [7] reviewed about 290 articles, claiming that unsupervised learning is the most frequently adopted method. Deep learning techniques, in particular, have demonstrated strong promise. For instance, in [8], they investigated deep learning applications for video-based anomaly detection, analyzing deep learning methods. The authors introduced a graphical taxonomy, addressed spatial anomalies, and compared frameworks. Other studies, such as [9] and [10], they proposed enhancements in spatiotemporal relations to tackle data imbalance, attaining significant results on popular UCSD Ped1, Avenue, and UCSD Ped2 datasets.

Traditional anomaly detection techniques, such as classical and statistical machine learning algorithms, still have a key role in certain scenarios, even though they often depend on manually crafted features. On the contrary, deep learning models can autonomously learn features from large datasets; however, they have higher computational requirements. Research on video-based anomaly detection, like [11] and [12], presented a framework for understanding the strengths and weaknesses of these methods, which we employ in this study. These studies are valuable resources for researchers and experts seeking to understand and apply deep learning techniques for anomaly detection.

Pre-trained deep learning models have proven to be useful in anomaly detection. For instance, [13] confirmed the efficiency of pre-trained convolutional neural networks (CNN) for

video-based anomaly detection on the UCSD dataset. Others, such as [14], emphasize the significance of optimizers in realizing high performance, which we further investigated in this study by evaluating four optimizers across four pre-trained deep-learning models.

2.2. Open Challenges in Video-Based Anomaly Detection

Deep learning techniques have demonstrated outstanding possibilities in detecting anomalies, showing the ability to understand complex patterns and relationships within video data. Nonetheless, notable limitations and obstacles exist in using deep learning to detect human anomalies.

Scalability presents a challenge, for example. Deep learning models typically demand significant processing power, especially with video data. The more the data grows or varies exponentially, the more deep learning systems face the challenge of efficiently handling the processing, especially regarding computing resources. The more complex the models become, the longer the training time, the greater the need for more hardware, and the higher the energy demand. Ensuring that low-latency and real-time processing capabilities are taken care of while preserving the performance and accuracy of the model is a decisive scalability issue.

Another constraint is the interpretability of deep learning models. Despite their outstanding performance, these models are often naturally considered "black boxes," making them difficult to interpret. This can impede practical deployment and regulatory compliance. Understanding the rationale behind a model's decision is essential for establishing trust and obtaining actionable insights. However, most deep learning models do not inherently provide this interpretability.

Likewise, deep learning models characteristically require large, labeled datasets for training. In the context of human anomaly detection, this involves having video data where anomalous and normal behaviors are correctly labeled. However, obtaining such labeled data can be difficult and time-consuming. Likewise, the labeling process can be subjective and susceptible to errors.

As these deficiencies persist in generalizing to real-world surveillance contexts, this study attempts to address these drawbacks by assessing model-optimizer synergies using the UCF Crime dataset, highlighting adaptation to real-world unpredictable conditions. This will culminate in figuring out which of the different ways of enhancing anomaly detection techniques proves economically and realistically viable. In this study, we investigated by pairing four pre-trained deep learning models and four optimizers to determine which combinations enhance anomaly detection.

3. METHODOLOGY

This study employs a quantitative experimental approach, investigating model-optimizer pairings via empirical metrics (Accuracy, F1 Score, AUC).

3.1. Datasets and Preprocessing

3.1.1. Data Structure and Subfolders

The UCF Crime Dataset [15] serves as the study's foundation, consisting of 1,900 real-world surveillance videos, with a total runtime of 128 hours. These videos are uncut and categorized into thirteen (13) distinct types of realistic anomalies, such as Arrest, Abuse, Arson,

https://doi.org/10.31436/iiumej.v26i2.3287

Assault, Robbery, Road Accident, Explosion, Burglary, Stealing, Fighting, Shooting, Shoplifting, and Vandalism.

The dataset used in this study is a subset of this dataset, which was divided into Train and Test subfolders, each containing seven different subfolders. The subfolders in the Train and Test folders represent distinctive classes representing individual criminal activities and normal behaviours ('Abuse', 'Arson', 'Assault', 'Fighting', 'NormalVideos', 'Robbery', and 'Shooting').

3.1.2. Preprocessing Steps

As shown in Figure 1, the preprocessing process ensures that raw video data is prepared for model evaluation. The workflow comprises the following blocks:

1. *Video Frames and Feature Extraction*: The videos were pre-processed, extracted frames, and converted to .png format. The frames were resized to 64×64 pixels for DenseNet121, VGG19, and ResNet50 models, and 75×75 pixels for the InceptionV3 model.
2. *Model Training*: This process comprised integrating pre-trained CNNs for fine-tuning, testing individual optimizers, and iteratively training the models with the selected optimizers to improve performance and minimize loss.
3. *Anomaly Detection*: The trained models were employed to predict anomaly scores for input videos, and a threshold was established to determine detected patterns as normal or abnormal.
4. *Model Evaluation*: Performed using AUC, accuracy, and F1 Score, followed by a comparison of model-optimizer combo to discover the best-performing configurations.

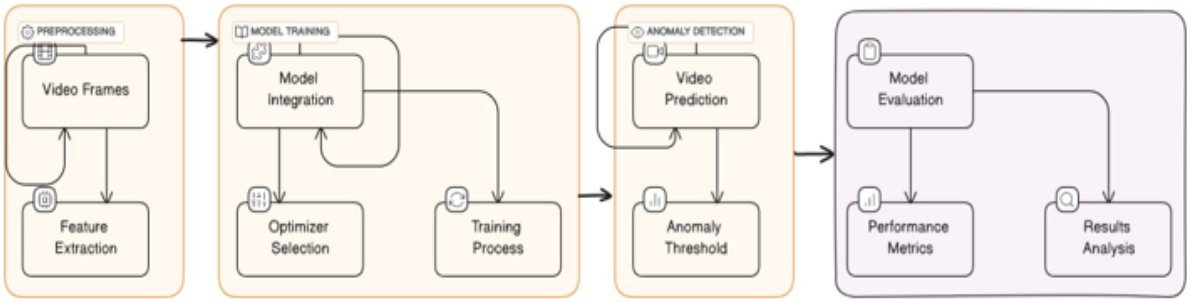


Figure 1. Anomaly Detection Process

3.2. Experimental Setup and Design

The experimental setup involves a structured workflow to evaluate the effectiveness of pre-trained deep learning models and optimization methods in video anomaly detection. The design of the experiment, as depicted in Figure 2, outlines the flowchart of the methodology:

1. *Model Selection*: This stage involves choosing the pre-trained CNN models (ResNet50, DenseNet121, VGG19, and InceptionV3) for their capabilities in image-based classification tasks.
2. *Optimizer Variation*: This stage tests four optimization algorithms—Adam, SGD, RMSProp, and Adagrad—on the selected models to analyze their impact on convergence and performance.
3. *Training on Keras API*: Leveraging the Kagle framework to streamline the training of models with diverging optimizers.

https://doi.org/10.31436/iiumej.v26i2.3287

4. *Performance Evaluation*: The evaluation metrics include Accuracy, F1 Score, and AUC to evaluate the effectiveness of the model-optimizer combinations.
5. *Best Model Found (Objectives Achieved)*: Confirm whether the study objectives, such as identifying optimal combinations, are met.
6. *Results and Insights*: Reveal which model-optimizer combinations yield the highest performance metrics and explain conclusions on model behaviors and optimization methods that enhance anomaly detection.

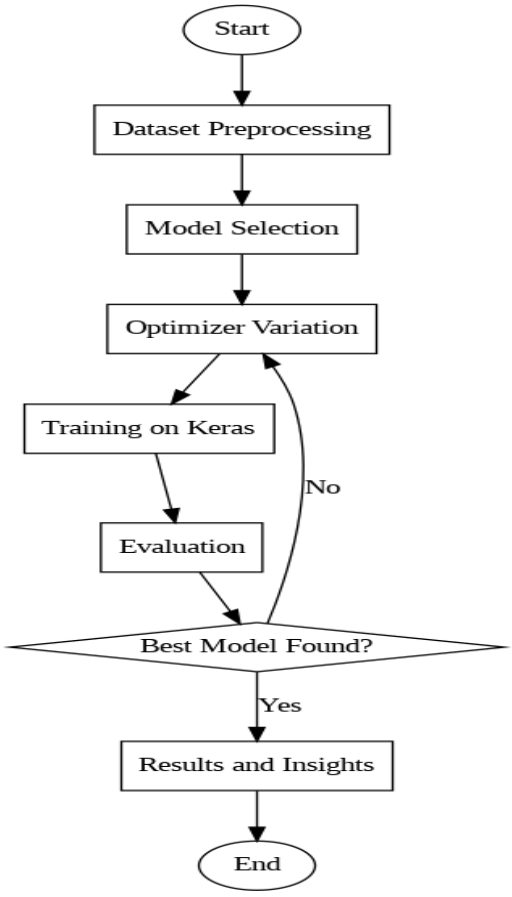


Figure 2. Flowchart illustrating the methodology: (1) Dataset preprocessing, (2) Model selection (DenseNet121, ResNet50, VGG19, InceptionV3), (3) Optimizer variation (Adam, SGD, RMSprop, Adagrad), (4) Training on Keras, (5) Evaluation using AUC, Accuracy, F1 Score.

3.3. Description of Deep Learning Models

In this study, we employed four major pre-trained deep learning models.

3.3.1. DenseNet121

DenseNet (Densely Connected Convolutional Networks) [16] is a pre-trained CNN model that achieved recognition due to its effectiveness in image-related classification. It gets its name by forward propagation by connecting every layer to another layer. There are 121 layers in this version. The bottleneck layer comprises two convolutional layers with a batch normalization layer in the middle. The initial convolutional layer is a 1x1 convolution, which decreases the

https://doi.org/10.31436/iiumej.v26i2.3287

number of input feature mappings for the bottleneck layer. The following convolutional layer is a 3x3 convolution that generates the output feature maps. The feature maps generated by both layers in a dense block are determined by the hyperparameter growth_rate, which is set to four times the inner_channel parameter. DenseNet's transition layer comprises a 1x1 convolutional layer, a 2x2 average pooling layer, and a batch normalization layer. The batch normalization layer normalizes the activations, the 1x1 convolutional layer decreases the feature maps, while the spatial dimensions are reduced by the feature maps' average pooling layer. The global average pooling layer output is passed through a fully connected (FC) layer with an activation function (ReLU), and a small number of neurons (e.g., 256). This layer decreases the feature vector dimensionality and extracts higher-level features. The FC layer's output is passed through the activation function (softmax), which gives class probabilities.

3.3.2. ResNet50

ResNet-50 (residual neural network, ResNet variant) [17] is a pre-trained CNN that is 50 layers deep (i.e., 48 convolution layers, 1 average pool layer, and 1 MaxPool layer). ResNet is an artificial neural network (ANN) that piles residual blocks on top of each other to make up a network. The model has 50 layers, comprising convolutional, batch normalization, activation, pooling, and FC layers. Residual connections are used to enable easier training of deep neural networks. This model is among the most popular variants of the ResNet architecture, having 50 layers that show impressive performance on a range of classification tasks (images). The model comprises five stages, each with convolution and identity blocks. Each convolution block has three convolutional layers, and each identity block has three. The trainable parameters of ResNet50 are more than 23 million parameters.

3.3.3. VGG19

Created by Zisserman and Simonyan of Oxford University, VGG19 has 19 layers (16 convolutions and 3 fully connected layers) [18]. It is a CNN model that uses strictly 3×3 filters with a stride of 1 and padding, alongside 2×2 max-pooling layers with a stride of 2. The model is deeper and has more layers than AlexNet. To lower the parameter count in such deep networks, it utilizes small 3×3 filters in all convolutional layers and is best used with its 7.3% error rate. This model uses (3 × 3) kernels with a 1-pixel stride size, and spatial padding is applied to preserve the image's spatial resolution. Furthermore, a 2 × 2-pixel window with a 2-pixel stride is used for max pooling. The model in question is quite complex, having undergone training on millions of photos with complicated classification tasks. With an enormous 19.6 billion FLOPs [19], it is a potent tool for picture classification and recognition.

3.3.4. InceptionV3

As a member of the Inception family, InceptionV3 [20] is an architecture of CNN introduced by Google. Compared to its predecessors, the architecture is more technologically advanced and optimized. It includes several techniques to enhance model adaptation. Among such techniques is Label Smoothing, which provides regularization and keeps the model from becoming unduly confident in its class assignments. To lower the number of parameters and computational expense, factorized 7×7 convolutions are also utilized. Similarly, an auxiliary classifier is used to convey label information to lower layers of the network. This increases the gradient signal and provides regularization. Batch normalization is also implemented in InceptionV3 at the network's side head. InceptionV3 retains excellent efficiency without sacrificing speed, even though it is deeper than its versions. Deeper networks can be created thanks to their design, which also limits parameter expansion for an increasingly effective

model. It works better than another well-known convolutional neural network, AlexNet, with fewer than 25 million parameters compared to 60 million.

3.4. Optimizers Used in the Study

Optimization is a crucial aspect of deep learning, influencing model performance. In this study, we investigated the deep learning models with the following optimizers:

1. Adam (Adaptive Moment Estimation) is an efficient optimization algorithm for large-scale problems with extensive parameters and/or data. It utilizes adaptive estimates of lower-order moments and first-order gradients to optimize stochastic objective functions. It is easy to implement and has been empirically proven effective.
2. RMSProp (Root Mean Square Propagation) was introduced by Geoff Hinton. It is an adaptive learning rate method that adjusts weight updates based on a moving average of squared gradients. It converges to a stationary point for realizable problems and a bounded region for non-realizable problems.
3. SGD (Stochastic Gradient Descent) is a fundamental set of rules that combines classical gradient descent with random subsampling to optimize the objective function. It's commonly used for neural network optimization.
4. A stochastic optimization technique called Adagrad (Adaptive Gradient Algorithm) modifies the learning rate in response to parameters. It makes smaller updates for features that occur frequently, while for features that occur infrequently, it makes more significant updates.

The optimizers were employed to train each model, ensuring consistent hyperparameters throughout trials to maintain fairness. By weighing their impact on the evaluation metrics, the study identifies the optimal model-optimizer combinations for the challenges of video anomaly detection.

3.5. Evaluation Metrics

We evaluated each of the models with each of the four optimizers using three performance evaluation metrics.

1. Accuracy: This evaluation metric measures the proportion of correctly classified samples:

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

where TP is the True Positive, TN is the True Negative, FP is the False Positive, and FN is the False Negative.

Accuracy, also referred to as top-1 accuracy, is a statistical measure that is used to demonstrate how accurately a binary classification test recognizes or rules out a condition. In other words, accuracy is described as the percentage of true positives and true negatives across every instance investigated that was predicted correctly. Although it gives a broad picture of the model's effectiveness, imbalanced datasets might not be a good fit for it. When there is a large imbalance in class in the dataset, accuracy alone may be deceiving because if the model consistently predicts the majority class, it could achieve a prominent level of accuracy.

2. F1 Score: This is simply the harmonic mean of precision and recall. It is the combination of recall and precision, delivering a single score, in which precision is the division of the number of true positive values by all sample numbers predicted as positive, comprising

those not correctly identified. The recall is calculated through the division of the total number of true positive outcomes by the total number of samples that ought to have been recognized as positive.

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

3. AUC Score: Assesses the area under the receiver operating characteristic (ROC) curve, which represents the model's ability to differentiate between classes.

$$\text{AUC} = \int_0^1 \text{TPRd(FPR)} \quad (3)$$

A receiver operating characteristic (ROC) plot is a two-dimensional representation of the classifier's performance. Multiclass AUCs are calculated by generating each class's ROC curve, determining the AUC, and then tallying the AUCs weighted by the reference class's prevalence in the video footage. The AUC performance metric is closely related to the Gini coefficient, which is occasionally employed as an alternative. The most common definition of this is double the area between the diagonal and the ROC curve. In simple geometry, $Gini + 1 = 2 \times AUC$. It is a plot of the true-positive rate (TPR) compared to the false-positive rate (FPR). A good deep-learning model will have an AUC value of nearly 1, but a random model could have a 0.5 AUC value.These metrics ensure a robust evaluation of models, especially in imbalanced datasets (such as the one used in this study) where accuracy alone is inadequate.

4. EXPERIMENTAL RESULTS

This section provides an experimental illustration of the capabilities of the chosen models and how they vary based on the optimizer. We presented the evaluation metrics used to evaluate the models' performance in tabular form.

4.1. Tabular Representation of Evaluation Metrics

Models and their performances with different optimizers are presented in tabular form.

Table 1. DenseNet121 Model

Optimizer	Accuracy	F1 Score	AUC
SGD	0.8079	0.8079	0.9307
Adam	0.8079	0.8079	0.9499
RMSprop	0.8079	0.8079	0.9537
Adagrad	0.8079	0.8079	0.9345

Table 2. ResNet50 Model

Optimizer	Accuracy	F1 Score	AUC
SGD	0.8079	0.8079	0.9307
Adam	0.8147	0.8147	0.9683
RMSprop	0.8079	0.8079	0.9701
Adagrad	0.8079	0.8079	0.9301

Table 3. VGG19 Model

Optimizer	Accuracy	F1 Score	AUC
SGD	0.8079	0.8079	0.9405
Adam	0.7583	0.7583	0.9905
RMSprop	0.7838	0.7838	0.9872
Adagrad	0.8079	0.8079	0.9391

https://doi.org/10.31436/iiumej.v26i2.3287

Table 4. InceptionV3 Model

Optimizer	Accuracy	F1 Score	AUC
SGD	0.8079	0.8079	0.9393
Adam	0.7641	0.7641	0.9918
RMSprop	0.7889	0.7889	0.9886
Adagrad	0.8079	0.8079	0.9401

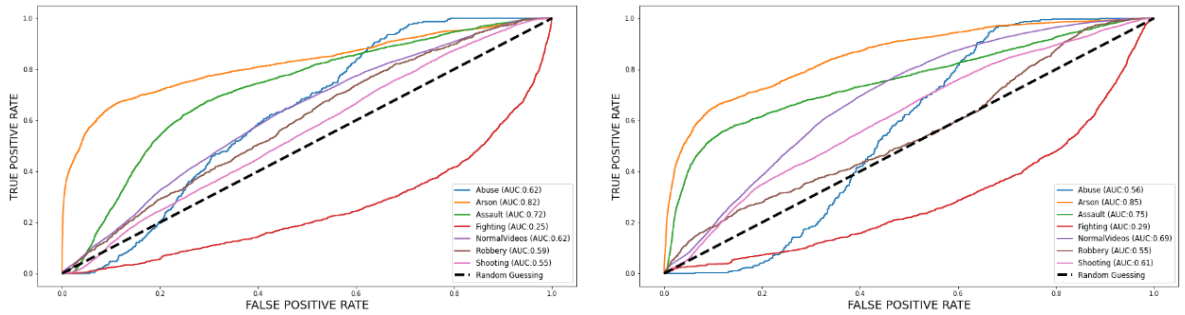
4.2. Comparative Model and Optimizer Analysis

The DenseNet121 model's performance was consistent and robust across our selected optimizers in the anomaly detection analysis. The optimizers impact the performance metrics, maintaining accuracy and F1 Score. SGD demonstrated an AUC value of 0.9307. Adam and RMSprop optimizers perform competitively, with Adam slightly outperforming RMSprop with an AUC of 0.9499.

The ResNet50 model provides consistent and competitive performance, with SGD achieving an AUC of 0.9307. Adam and RMSprop, on the other hand, outperform SGD with higher AUCs of 0.9683 and 0.9701. These optimizers improve accuracy and F1 scores, implying enhanced model correctness. Adagrad slightly lags with an AUC of 0.9301.

With regard to VGG19, SGD accomplished quite a good AUC of 0.9405. However, both Adam and RMSprop perform better than SGD, with remarkable respective AUCs of 0.9905 and 0.9872. These optimizers particularly improved the accuracy and F1 score of the model, underlining the strength of VGG19 in recognizing anomalies. While maintaining a competitive AUC value of 0.9391, Adagrad falls marginally behind Adam and RMSprop.

For the InceptionV3 model, Adam and RMSprop optimizers demonstrate considerable anomaly detection capabilities. SGD yielded an AUC of 0.9393, with constant accuracy and F1 score performances. However, Adam and RMSprop perform beyond SGD, with AUCs of 0.9918 and 0.9886, respectively. These optimizers also improve the accuracy and F1 Score of the InceptionV3 model, highlighting their ability to fine-tune the model for improved anomaly detection. Adagrad retains an AUC value of 0.9401, trailing behind the Adam and RMSprop optimizers.



(a) InceptionV3 with Adam Optimizer (b) InceptionV3 with RMSprop Optimizer

Figure 3. Visualization of the InceptionV3 Model of ROC-AUC

5. DISCUSSION

This study demonstrates that the InceptionV3 model, when optimized with Adam, surpasses other model-optimizer combinations in video anomaly detection. This conclusion is backed by its superior AUC scores of 0.9918 (Adam) and consistently high accuracy and F1 scores. InceptionV3's architecture incorporates advanced techniques like factorized convolutions and auxiliary classifiers, enabling it to capture complex spatiotemporal patterns

effectively. This capability addresses the variability and imbalance inherent in video anomaly datasets. In comparison, the other models, however robust in their own right, had lower performance due to their less optimized handling of spatial and temporal dependencies in video data. Likewise, VGG19's modest architecture and dependence on smaller filters caused slower convergence and reduced precision, further highlighting InceptionV3's advantage.

The performance observed with InceptionV3 is further enhanced by its combination with the Adam optimizer. Adam's momentum and adaptive learning rates allow for efficient parameter updates, which ensure faster convergence and improved generalization. RMSprop complements InceptionV3 by stabilizing training through adaptive learning rates tailored to sparse or noisy gradients. In comparison, Adagrad and SGD were less effective, with Adagrad exhibiting diminishing learning rates over extended training and SGD struggling to adapt to the dynamic nature of video data. This comparative study emphasizes the importance of selecting both high-performing models and appropriate optimizers to address the challenges of video anomaly detection. In this study, InceptionV3, combined with Adam, is a balanced solution that achieves high results, especially in terms of AUC, and thus emerges as the most efficient model-optimizer combination.

6. CONCLUSION

6.1. Key Insights and Observations

The distinct responses presented by each model underscore the importance of tailoring optimization methods to accommodate the distinct attributes of each CNN model. This highlights the complex relationships between models and optimizers and emphasizes the need for adaptability in optimization methods. Likewise, optimizers are important in deciding how well an anomaly detection algorithm performs. The choice of an optimizer enhances models' capabilities to detect anomalies in video footage.

As mentioned earlier, DenseNet121 demonstrated exciting strength and consistency when run in conjunction with different optimizers, suggesting that it may have flexibility in independent optimization strategies. Due to its steady performance, DenseNet121 is a reliable option for anomaly detection tasks, providing consistency even in different optimization approaches. In addition, optimizers and model architecture have a complex interaction that requires careful direction finding, as demonstrated by fine-tuning models for anomaly detection. This highlights the importance of utilizing hyperparameters to achieve the best performance results possible.

Furthermore, the synergistic impact of model-optimizer combinations can be found in the superior detection of anomaly capabilities of pairings like InceptionV3 with Adam and RMSprop optimizers. Maximizing the effectiveness of the models requires leveraging these synergies. On the other hand, choosing optimizers enforces striking a balance among many performance assessment standards, leading to variations in performance results between models and measures. Therefore, making well-informed decisions requires a thorough assessment considering variations in performance metrics.

In addition, these trends have real-world applications in anomaly detection, offering practitioners recommendations of approaches to adopt depending on their datasets. Examining these observations would greatly benefit the design and implementation of algorithms in real-world situations. Eventually, the experiment tends to spark debates on model-optimization approaches and other factors affecting performance variability. Evaluating these provides more insights into trends that have been observed and serves as a guide for further research.

6.2. Limitations and Future Directions

The experimental assessment conducted in this study encountered some limitations worth mentioning and needs insights into future studies. The most challenging constraint faced is the insufficient computing capacity, which necessitated experimenting on the Kaggle notebook, restricting the model training to a single epoch.

In the future**,** computing capacity can be addressed using stronger hardware or cloud-based platforms. This will enable extensive training sessions, allowing for many epochs and a thorough assessment of the dynamic range between the model and optimizer. Moreover, holding out the experiments with more epochs may better evaluate model convergence, stability, and long-term performance. This approach may yield valuable insights into the model's capabilities over extended training durations.

With regard to the proposed research directions, widening the scope of the investigation to involve separate deep learning models employed here would improve the overall understanding of the interactions between models and optimizers. Still, evaluating the tendency to deploy optimized models in real-world settings and their effectiveness in dynamic circumstances will be significant for their practical application.

ACKNOWLEDGEMENT

This work is supported by the Ministry of Higher Education (MOHE) Fundamental Research Grant Scheme (FRGS22-264-0873) (Grant No: FRGS/1/2022/ICT11/UIAM/01/1).

REFERENCES

[1] P. G. I. M. Chandrasekara, L. L. G. Chathuranga, K. A. A. Chathurangi, and D. M. K. N. Seneviratna, "Intelligent Video Surveillance Mechanisms for Abnormal Activity Recognition in Real- Time: A Systematic Literature Review," vol. 5, no. 1, pp. 26–40, 2023.

[2] B. Gayathri, A. Abhinav, C. H. Reddy, and G. Praneeth, "ANOMALY XPERT: A Deep Learning Approach," no. March 2024.

[3] N. Gupta, C. Science, B. B. Agarwal, and C. Science, "Recognition of Suspicious Human Activity in Video Surveillance: A Review," vol. 13, no. 2, pp. 10529–10534, 2023.

[4] A. Tonge, S. Chandak, R. Khiste, U. Khan, and L. A. Bewoor, "Traffic Rules Violation Detection using Deep Learning," in 2020 4th International Conference on Electronics, Communication and Aerospace Technology (ICECA), 2020, pp. 1250–1257. doi: 10.1109/ICECA49313.2020.9297495.

[5] U. Y. Reddy, M. S. Nikhil, P. S. S. Krishna, and S. S., "Systematic Harmful Signs Detection for Women's Safety Using Neural Networks," in 2023 International Conference on Computer Communication and Informatics (ICCCI), 2023, pp. 1–5. doi: 10.1109/ICCCI56745.2023.10128298.

[6] R. Nayak, U. C. Pati, and S. K. Das, "A comprehensive review on deep learning-based methods for video anomaly detection," Image Vis Comput, vol. 106, p. 104078, Feb. 2021, doi: 10.1016/J.IMAVIS.2020.104078.

[7] S. Bhakat and G. Ramakrishnan, "Anomaly detection in surveillance videos," ACM International Conference Proceeding Series, pp. 252–255, 2019, doi: 10.1145/3297001.3297034.

[8] K. Pawar and V. Attar, "Deep learning approaches for video-based anomalous activity detection," World Wide Web, vol.22, no.2, pp. 571–601, 2019, doi:10.1007/s11280-018-0582-1.

[9] J. Wang, D. Jia, Z. Huang, M. Zhang, and X. Ren, "Normal Spatio-Temporal Information Enhance for Unsupervised Video Anomaly Detection," Neural Processing Letters, 2023, doi: 10.1007/s11063-023-11347-5.

[10] H. Lv, Z. Cui, B. Wang, and J. Yang, "Spatio-Temporal Relation Learning for Video Anomaly Detection," Journal of Latex Class Files, vol. 14, no. 8, 2022, doi: 10.48550/arXiv.2209.13116.

[11] R. Chalapathy and S. Chawla, "Deep Learning for Anomaly Detection: A Survey," pp. 1–50, 2019.

[12] G. Pang, C. Shen, L. Cao, and A. Van Den Hengel, "Deep Learning for Anomaly Detection: A Review," ACM Computing Surveys, vol. 54, no. 2, pp. 1–36, 2020, doi: 10.1145/3439950.

[13] C. Wu, S. Shao, C. Tunc, and S. Hariri, "Video Anomaly Detection using Pre-Trained Deep Convolutional Neural Nets and Context Mining," Proceedings of IEEE/ACS International Conference on Computer Systems and Applications, AICCSA, vol. 2020-Novem, 2020, doi: 10.1109/AICCSA50499.2020.9316538.

[14] A. Lydia and F. Sagayaraj Francis, "A Survey of Optimization Techniques for Deep Learning Networks," International Journal for Research in Engineering Application & Management (IJREAM), vol. 05, no. August 2020, p. 2, 2019, doi: 10.35291/2454-9150.2019.0100.

[15] W. Sultani, C. Chen, and M. Shah, "Real-World Anomaly Detection in Surveillance Videos," Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 6479–6488, 2018, doi: 10.1109/CVPR.2018.00678.

[16] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, vol. 2017-Janua, pp. 2261–2269, 2017, doi: 10.1109/CVPR.2017.243.

[17] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 770–778. doi: 10.1109/CVPR.2016.90.

[18] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings, pp. 1–14, 2015.

[19] M. F. Amin, Z. Othman, S. S. S. Ahmad, and F. Kasmin, "Analysis on the impact of imagenet preprocessing image mode using VGG19 pre-trained model in plant disease classification," Proceedings of Mechanical Engineering Research Day 2022, no. August, pp. 154–155, 2022.

[20] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the Inception Architecture for Computer Vision," Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 2016-Decem, pp. 2818–2826, 2016, doi: 10.1109/CVPR.2016.308.