International Journal of Scientific Research in Computer Science, Engineering and Information Technology
ISSN : 2456-3307 (www.ijsrcseit.com)
doi : https://doi.org/10.32628/CSEIT2390124

Comparison of VGG-16, VGG-19, and ResNet-101 CNN Models for the purpose of Suspicious Activity Detection

Dr. Madhur Jain[1], Mayank Singh Bora[2] , Sameer Chandnani[2], Sanidhay Grover[2], Shivank Sadwal[2]

[1]Assistant Professor, Department of Information Technology, Bhagwan Parshuram Institute of Technology, Guru Gobind Singh Indraprastha University, Delhi, India

[2]UG Student, Department of Information Technology, Bhagwan Parshuram Institute of Technology, Guru Gobind Singh Indraprastha University, Delhi, India

Article Info

Publication Issue :
Volume 9, Issue 1
January-February-2023

Page Number : 121-130

Article History
Accepted: 10 Jan 2023
Published: 27 Jan 2023

ABSTRACT

This paper compares the performance of three popular convolutional neural network (CNN) models, VGG-16, VGG-19, and ResNet-101, for the task of suspicious activity detection. The VGG networks are known for their depth and the use of small convolutional filters, while ResNet is known for its residual connections that allow for deeper networks without the issue of vanishing gradients. The study utilizes a dataset of surveillance videos for training and testing the models. The results show that the VGG-19 model outperforms the other two in terms of accuracy, specifically for the detection of suspicious activities.

Overall, this study demonstrates the effectiveness of the three different models for suspicious activity detection and highlights its potential for use in real-world surveillance systems.

Keywords : Convolutional Neural Network (CNN), Image classification, Suspicious Activity Detection, VGG-16, VGG-19, ResNet-101.

I. INTRODUCTION

The importance of recognizing suspicious human activities from video surveillance is to prevent theft, vandalism, fighting, personal attacks, and fire in various highly sensitive areas such as banks, hospitals, shopping malls, parking lots, bus and train stations, airports, refineries, nuclear power plants, schools, university campuses, and borders, among others. Such attacks in public places, which are being investigated with cameras, cannot be fully prevented by current technologies. Implementation of smart surveillance using machine learning algorithms with the existing video surveillance infrastructure can prove to be highly beneficial in dealing with any mishappenings by producing an alert that can be produced through alarms, messages, or any other means necessary.

In recent years, the use of convolutional neural networks (CNNs) for video surveillance has gained significant attention due to its ability to

detect and classify objects in real-time. Among the various CNN architectures, VGG-16, VGG-19 and ResNet-101 are considered to be state-of-the-art models for image and video analysis.

The VGG models, VGG-16 and VGG-19, were proposed by the Visual Geometry Group at the University of Oxford[1], and are known for their use of small convolutional filters and deep architectures. On the other hand, ResNet-101 is a deep residual network developed by Microsoft Research Asia that uses residual connections to alleviate the problem of vanishing gradients in deep CNNs.

In this study, we evaluate the performance of these three models, VGG-16, VGG-19 and ResNet-101, for the task of suspicious activity detection in surveillance videos. This task is challenging due to the presence of various types of activities, changing lighting conditions, and occlusions. We utilize a publicly available dataset and evaluate the models based on their accuracy, speed, and number of parameters.

The results of our study indicate that the VGG-19 model outperforms VGG-16 and ResNet101 in terms of accuracy, while still maintaining a similar level of speed.

In conclusion, our study demonstrates the effectiveness of the VGG-19 model for suspicious activity detection and highlights its potential for use in real-world surveillance systems. Furthermore, the comparison of these three models provides insights into the strengths and weaknesses of these architectures for this specific task.

II. RELATED WORK

The related work suggests different approaches for detecting human behaviors from video. The objective of the works was to detect any abnormal or suspicious events in a video surveillance.

[1] The Advance Motion Detection (AMD) algorithm was proposed for detecting unauthorized entry in restricted areas. The algorithm first detects the object using background subtraction and then extracts it from the frame sequences. The second phase is the detection of suspicious activity. The system is advantageous as it works in real-time and has low computational complexity, however, it is limited in terms of storage service and can be improved by using advanced video capturing techniques in surveillance areas.

[2] A semantic-based approach was proposed. The captured video data is processed, and the foreground objects are identified using background subtraction. After subtraction, the objects are classified into living or non-living using a Haar-like algorithm. Object tracking is done using a Real-Time blob matching algorithm, and fire detection is also included in this approach.

[3] Tracking of people was proposed as a method for detecting unusual events in video footage. Humans are detected from the video using background subtraction methods, and features are extracted using CNNs. These features are fed to a Discriminative Deep Belief Network (DDBN) for comparison with labeled sample videos of classified suspicious actions. Various suspicious activities are detected by comparing the features extracted using CNNs and the labeled sample videos using a DDBN.

[4] A real-time violence detection system using deep learning was developed to prevent violence in crowds or players in sports. In a spark environment, frames were extracted from real-time videos. If the system detects any violence, it alerts the security personnel to take

preventative measures. The system detects violent actions in real-time and alerts the security forces to prevent violence.

III. TECHNOLOGIES USED

This section will discuss the technologies used in the proposed work.

A. TensorFlow

TensorFlow provides a comprehensive set of tools for building and deploying machine learning models[2], including support for deep learning. TensorFlow enables developers to create complex models using a flexible architecture, and it can run on a wide range of platforms including computers, mobile devices, and servers.

In conclusion, TensorFlow is a robust and versatile tool for building and deploying machine learning models, which is widely used in research and industry. Its ability to perform computations on a graph and its wide range of pre-built models and libraries, and its active community, makes it a valuable tool for machine learning developers.

B. Numpy

NumPy is a library for the Python programming language that offers a range of features for handling and manipulating large, multi-dimensional arrays and matrices. It provides a collection of high-level mathematical functions which can be applied to these arrays, making it an essential tool for numerical computation and data analysis tasks.Among its key features are the n array object for efficient array computations, broadcasting functions for performing operations on arrays of different shapes, mathematical and logical operations on arrays, linear algebra, Fourier transform, and random number capabilities, and interoperability with other libraries commonly used in scientific computing and data science, such as pandas and scikit-learn. Due to its efficiency in storing and manipulating large arrays of numerical data, NumPy is widely used in scientific research and data analysis.

C. Python

Python is a high-level,general purpose language that was first introduced in 1991. It is known for its readability and expressiveness, with a syntax that is easy to understand and write. One of the key features of Python is its dynamic typing system, which allows for flexibility in data types and variable assignments. Python also has an automatic memory management system, which takes care of freeing up memory for objects that are no longer in use. The language is executed by an interpreter, rather than being compiled like many other programming languages, making it more interactive and easy to use. Python supports both object-oriented and functional programming styles, giving developers the freedom to choose the most appropriate approach for their specific task.

D. Google Collab

Google Colaboratory, or Colab, is a free, cloud-based platform that enables users to conduct machine learning experimentation and research. It provides an environment for writing, executing, and sharing code in Python using Jupyter notebooks. One of the most prominent features of Colab is the availability of free GPUs and TPUs for running computationally demanding tasks.

Colab is integrated with Google Drive which facilitates easy storage and sharing of notebooks among different users. This also enables real-time collaboration on the same notebook. Colab has pre-installed libraries such as TensorFlow, Keras, and PyTorch, which eliminates the need for additional installations.

E. VGG16

VGG16 is a convolutional neural network [3] architecture that was introduced by the Visual Geometry Group (VGG) at the University of Oxford in 2014. It is a specific variant of the VGG network architecture, designed to perform image classification tasks, characterized by the use of small convolutional filters of size 3x3 and a deep architecture. VGG16 consists of 16 layers, including 13 convolutional layers and 3 fully connected layers. Additionally, it implements a technique called "very deep supervision" which adds additional supervision to intermediate layers, resulting in more accurate and stable optimization.

VGG16 was trained on the ImageNet dataset, which includes 14 million images and 1000 classes. In 2014, it achieved state-of-the-art performance on the ImageNet classification task, and since then it has been widely used as a base model for various computer vision tasks such as object detection and semantic segmentation. The pre-trained weights of VGG16 are available to the public and can be easily loaded into other models, making it a popular choice among researchers and developers as it provides a starting point for their projects without the need of re-training the model.

F. VGG19

VGG19 is a convolutional neural network architecture that is similar to VGG16, but with 19 layers. It was introduced by the Visual Geometry Group (VGG) at the University of Oxford in 2014. The main objective of this architecture is to perform image classification tasks. VGG19 utilizes small convolutional filters of size 3x3 and a deep network architecture. The architecture has five convolutional blocks, each comprising multiple convolutional layers. The first two blocks have two convolutional layers, while the remaining three blocks have four convolutional layers each. The last block is followed by three fully connected layers. VGG19 also employs the technique of "very deep supervision" which was also used in VGG16.

F. ResNet-101

ResNet101 is a deep residual neural network[4] architecture that was introduced in 2016 by Microsoft Research. It is a variant of the ResNet architecture, which stands for "Residual Network" and is characterized by the use of residual connections, which are shortcuts that bypass one or more layers in the network. The ResNet-101 architecture has 101 layers and was trained on the ImageNet dataset, which includes 14 million images and 1000 different classes. In 2016, it achieved state-of-the-art performance on the ImageNet classification task and has been widely adopted as a base model for various computer vision tasks such as object detection, semantic segmentation, and more.

IV. METHODOLOGY

A. The Testing set

In machine learning, a training set[5] is a crucial component in the model development process. It is a representative sample of the data that is used to train a model and enable it to identify the underlying patterns within the data. The model is exposed to the same training data repeatedly, over multiple iterations called epochs, as it continues to learn the features of the data. To ensure the generalization of the model and to make accurate predictions on unseen data, it is essential to have a diverse set of inputs in the training set. This way, the model can learn to identify patterns and features in a wide range of scenarios, making it robust to different situations.

It is important to note that the selection of the training set is crucial for the performance of the model, so it should be carefully chosen to be representative of the data that the model will be applied to.

B. The Validation Set

In the process of training machine learning models, the use of a validation set is a common practice to evaluate the performance of the model during the training process. The validation set, which is separate from the training set, provides valuable information that is used to adjust the model's hyper parameters and configurations accordingly. The validation set serves as a guide, indicating whether the training is progressing in the right direction or not.

The model is trained using the training set, and simultaneously, its performance is evaluated on the validation set after every iteration, known as an epoch. The main objective of using a validation set is to prevent over-fitting, which occurs when the model becomes too specialized in classifying the samples in the training set, but is unable to generalize and make accurate predictions on data it has not seen before.

By using a validation set, the model's ability to generalize can be evaluated, and thus, ensure that the model is able to make accurate predictions on unseen data. The selection of the validation set is crucial for the performance of the model, so it should be carefully chosen to be representative of the data that the model will be applied to. Overall, the use of a validation set is an important step in the model development process, and it helps to improve the robustness and generalization capabilities of the model.

C. The Test Set

In the process of training machine learning models[5], the use of a test set is a critical step in evaluating the performance of the trained model. The test set, which is separate and independent from both the training and validation sets, is used to estimate the generalization performance of the model on unseen data. The model is trained using the training set and its performance is evaluated using the validation set during the training process, however, the final evaluation of the model's performance is conducted on the test set after the model has been fully trained.

The test set provides an unbiased estimate of the model's ability to generalize to new data and serves as a means of validating the model's performance. It is important to note that the test set should only be used as the final evaluation step after the model has been fully trained and no further adjustments should be made to the model based on its performance on the test set. In conclusion, the use of a test set is an essential step in the model development process, as it helps to ensure that the model has the ability to make accurate predictions on unseen data.

D. Epoch

In the process of training machine learning models, an epoch[7] refers to a single iteration through the entire training dataset. During an epoch, the model's parameters are updated based on the training data to reduce the error of the model. The number of epochs is a crucial hyperparameter that determines the number of times the model is exposed to the training data before the training process is terminated.

As the training process progresses, the model's performance is evaluated on the validation set after each epoch. This helps to monitor the progress of the training and detect any signs of overfitting. If the model's performance on the validation set improves with each epoch, it suggests that the model is learning the features of the data correctly. However, if the performance of the model on the validation set deteriorates or remains unchanged, it may

indicate that the model is memorizing the training data rather than generalizing to new unseen data.

The number of epochs is a hyperparameter that needs to be carefully determined based on the specific problem, the size of the dataset, and the desired level of generalization. More epochs can lead to better performance of the model but also increase the risk of overfitting. Thus, it is important to find a balance and decide the optimal number of epochs that can provide the best generalization performance while avoiding overfitting. It's important to note that the selection of the number of epochs is a crucial step in the model development process, and it should be decided upon after conducting proper experimentation and analysis.

E. Accuracy

In any classification project, the assessment of accuracy is an important step in evaluating the performance of the model. This process involves comparing the classified image to a reference data source, known as ground truth data, which is considered to be accurate. The accuracy assessment allows for an estimation of the classification model's performance and identification of any potential errors or inaccuracies. This step is crucial in ensuring the quality and reliability of the classification results and identifying areas for improvement in the classification process.

It is important to note that the accuracy assessment is a fundamental step in the classification project, and it should be conducted after the model has been trained and validated. Different methodologies and metrics can be used for the assessment of accuracy, and the selection of the appropriate method should be based on the characteristics of the dataset and the classification problem. Overall, the accuracy assessment is a critical step in the classification project, and it provides valuable insights into the

performance of the model and helps to improve the classification process.

F. Validation loss

In the process of training deep learning models, the validation loss[8] is a crucial metric used to assess the performance of the model on a validation set. The validation set is a portion of the dataset that is reserved specifically for evaluating the performance of the model, separate from the training set. The validation loss is calculated by determining the sum of errors for each example in the validation set and it is similar to the training loss.

This metric is used to monitor the performance of the model during the training process, and it helps to detect overfitting or underfitting. By comparing the performance of the model on the validation set with that of the training set, the validation loss helps to identify the optimal model. It is important to note that the validation loss is a key metric in the model development process, and it should be monitored throughout the training process in order to detect and prevent overfitting. This can be achieved by comparing the performance of the model on the validation set to that of the training set, and adjusting the hyperparameters accordingly.

V. IMPLEMENTATION

In our study, we have conducted an extensive evaluation of various parameters on three different models. The aim of this evaluation is to compare and contrast the performance of the models under different parameter settings. This process helps to identify the optimal parameter settings for each model and understand the impact of the parameters on the model's performance. The results of this evaluation provide valuable insights into the behavior of the models and help to improve their performance.

We have started our evaluation by testing the impact of the first parameter.

Data Set Split = 70% Training data, 30% Testing data

Epochs=10, batch_size=4, validation_split=0.25

VGG-16 Model

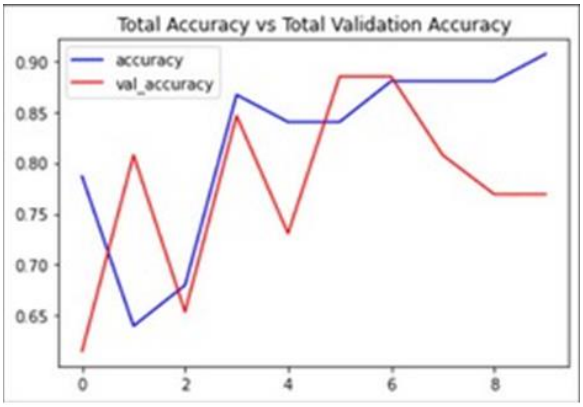

Figure 1 : Accuracy vs Validation Accuracy_1

$$Accuracy = \frac{Number\ of\ correct\ Predictions}{Total\ Number\ of\ Predictions} * 100$$

$$Accuracy = \frac{35}{41} * 100$$

$$Accuracy = 85.36\%$$

In our study, we have found that the combination of the following parameters resulted in an accuracy of 85.36% on this model under evaluation.

VGG-19 Model

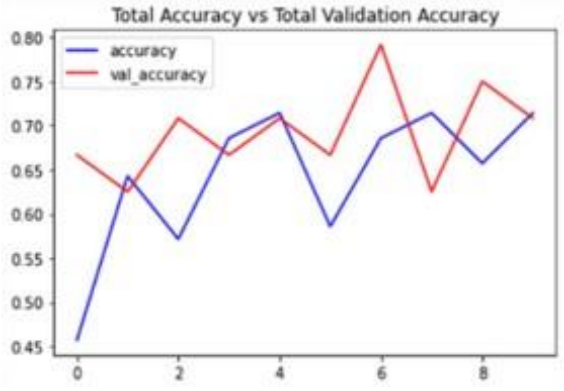

Figure 2 : Accuracy vs Validation Accuracy_2

$$Accuracy = \frac{Number\ of\ correct\ Predictions}{Total\ Number\ of\ Predictions} * 100$$

$$Accuracy = \frac{32}{41} * 100$$

$$Accuracy = 78.04\%$$

In our study, we have found that the combination of the following parameters resulted in an accuracy of 78.04% on this model under evaluation.

ResNet-101 Model

Data Set Split = 70% Training data, 30% Testing data

Epochs=10,batch_size=4, validation_split=0.25

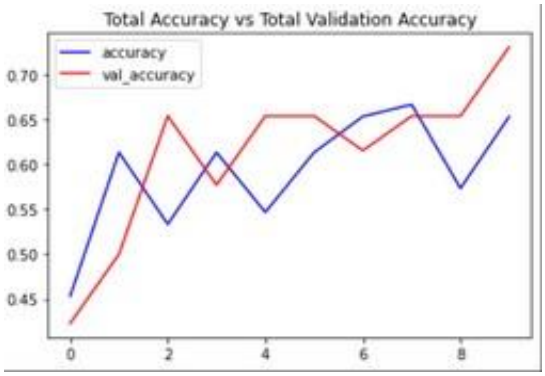

Figure 3 : Accuracy vs Validation Accuracy_3

$$Accuracy = \frac{Number\ of\ correct\ Predictions}{Total\ Number\ of\ Predictions} * 100$$

$$Accuracy = \frac{28}{41} * 100$$

$$Accuracy = 68.29\%$$

In our study, we have found that the combination of the following parameters resulted in an accuracy of 68.29% on this model under evaluation.

In our study, we have continued our evaluation process by investigating the impact of the next parameter on the performance of the three models.

Data Set Split = 70% Training data, 30% Testing data

Epochs=25,batch_size=4, validation_split=0.25

VGG-16 Model

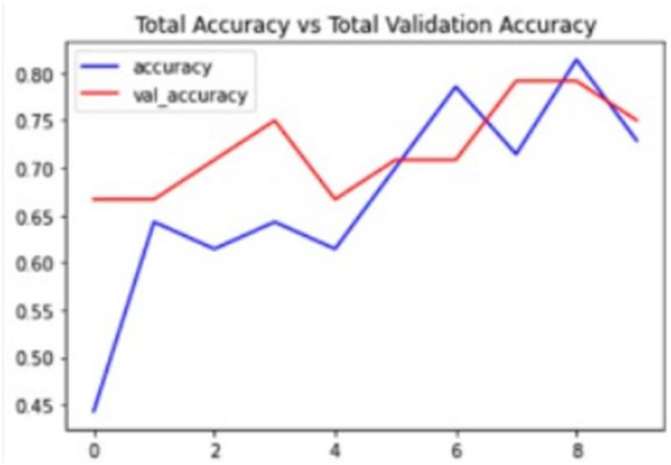

Figure 4 : Accuracy vs Validation Accuracy_4

In our study, we have found that the combination of the following parameters resulted in an accuracy of 90.24% on this model under evaluation.

VGG-19 Model

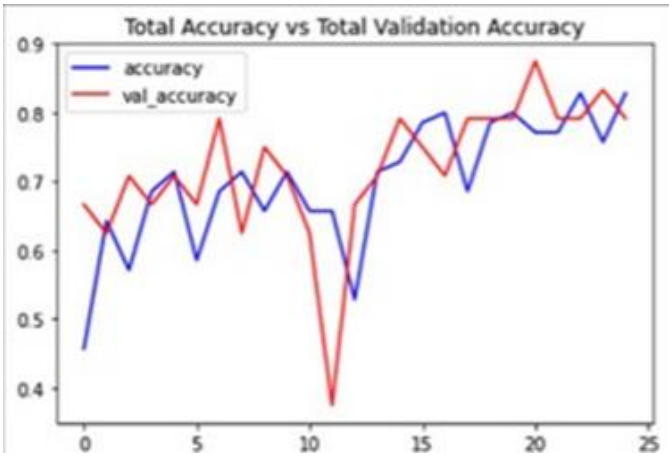

Figure 5 : Accuracy vs Validation Accuracy_5

$$Accuracy = \frac{Number\ of\ correct\ Predictions}{Total\ Number\ of\ Predictions} * 100$$

$$Accuracy = \frac{37}{41} * 100$$

$$Accuracy = 90.24\%$$

In our study, we have found that the combination of the following parameters resulted in an accuracy of 95.12% on this model under evaluation.

ResNet-101 Model

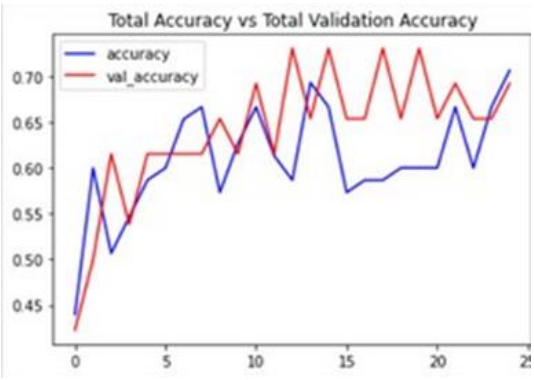

Figure 6 : Accuracy vs Validation Accuracy_6

$$Accuracy = \frac{Number\ of\ correct\ Predictions}{Total\ Number\ of\ Predictions} * 100$$

$$Accuracy = \frac{28}{41} * 100$$

$$Accuracy = 68.29\%$$

In our study, we have found that the combination of the following parameters resulted in an accuracy of 68.29% on this model under evaluation.

VI. RESULTS AND DISCUSSION

In the course of our study, we have conducted an extensive evaluation of various parameters on three different models. The results of these tests have enabled us to determine the optimal combination of parameters that resulted in the highest level of performance for each model. After a thorough analysis of the results, we have reached the conclusion that the VGG-19 Model with specific parameter settings exhibited the best performance among the models under evaluation. The accuracy metric, which measures the proportion of correctly classified instances to the total number of instances, was used as a measure of performance. The high level of accuracy achieved by the VGG-19 model with these specific parameter settings is a valuable contribution to the field and it could serve as a benchmark for future research and development in the area of deep learning.

Data Set Split = 70% Training data, 30% Testing data
Epochs=25, batch_size=4, validation_split=0.25
achieved the highest accuracy of95.12%

VII. CONCLUSION

In conclusion, this study has presented the results of evaluating various parameters on three different deep learning models for image classification tasks related to suspicious activity detection. The results have shown that the optimal configuration of parameters for achieving the highest accuracy is by setting the number of epochs at 25, using a 70% training data and 30% testing data split, with a batch size of 4 and a validation split of 0.25. The results indicate that the VGG-19 model consistently obtained the highest accuracy among the models tested. Based on these findings, it can be inferred that the VGG-19 model with the optimal configuration of parameters is the most appropriate architecture for image classification tasks related to suspicious activity detection.

It's important to note that the field of image classification using deep learning models is constantly evolving and this study is not conclusive. Nevertheless, the results presented in this study provide a valuable benchmark for future research and development in this field. Future work may include modification of the CNN models to increase the precision of the image classification. Overall, this study provides a valuable contribution to the field of image classification and suspicious activity detection, and it may serve as a foundation for future research in this area.

VIII. ACKNOWLEDGMENT

The author would like to give the deepest gratitude to the Bhagwan Parshuram Institute of Technology, Guru Gobind Singh Indraprastha University, Delhi, India for the support given to this study.

IX. REFERENCES

[1]. Understanding the VGG19 Architecture https://iq.opengenus.org/vgg19-architecture
[2]. Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, Manjunath Kudlur, Josh Levenberg, Rajat Monga, Sherry Moore, Derek G. Murray, Benoit Steiner, Paul Tucker, Vijay Vasudevan, Pete Warden, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng, Google Brain. ”TensorFlow: A system for large-scale machine learning”.12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16), USENIX Association (2016), pp. 265-283.
[3]. VGG-16 | CNN model:https://wwwgee.ksforgeeks.org/vgg-16-cnn-mod el
[4]. Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun. “Deep Residual Learning for Image Recognition”2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR).e-ISSN:1063-6919
[5]. Training and Test Sets: Splitting Data: https://developers.google.com/machine-learning/crash-course/training-and-test-sets/splitting-data
[6]. Machine Learning – Training, Validation & Test Data Set: https://vitalflux.com/machine-learning-training- validation-test-data-set/
[7]. What is Epoch in Machine Learning?: https://www.simplilearn.com/tutorials/machine-learning-tutorial/what-is-epoch-in-machine-learni ng
[8]. Training and Validation Loss in Deep Learning:https://www.baeldung.com/cs/training-validation-loss-deep-learning

Cite this article as :

Dr. Madhur Jain, Mayank Singh Bora, Sameer Chandnani, Sanidhay Grover, Shivank Sadwal, "Comparison of VGG-16, VGG-19, and ResNet-101 CNN Models for the purpose of Suspicious Activity Detection", International Journal of Scientific Research in Computer Science, Engineering and Information Technology (IJSRCSEIT), ISSN : 2456-3307, Volume 9, Issue 1, pp.121-130, January-February-2023. Available at doi : https://doi.org/10.32628/CSEIT2390124
Journal URL : https://ijsrcseit.com/CSEIT2390124