Exploring Anomaly Detection Techniques for Crime Detection

Aakanksha Singh[1,2], Ashwin Singh[2,3†], Ayush Bajaj[1,2†]

[1]Communication And Computer Engineering, The LNMIIT, Jamdoli Jaipur, 302031, Rajasthan, India.
[2]Communication And Computer Engineering, The LNMIIT, Jamdoli, Jaipur, 302031, Rajasthan, India.
[3]Communication And Computer Engineering, The LNMIIT, Jamdoli, Jaipur, 302031, Rajasthan, India.

Contributing authors: 21ucc002@lnmiit.ac.in; 21ucc128@lnmiit.ac.in; 21ucc129@lnmiit.ac.in;
[†]These authors contributed equally to this work.

Abstract

Crime anomaly detection is critical for proactive law enforcement and public safety measures. This paper introduces approaches employing deep learning techniques for anomaly detection in real-time using instances from video surveillance cameras to train the models. Leveraging current research about neural networks, the study explores multiple approaches using pre-trained neural network architectures, including VGG19, DenseNet121, ResNet50, and MobileNetV2 to aid as predictive models.

The research systematically analyzes the performance of each model using various metrics to gauge the models' ability to discern anomalies effectively. The proposed methodology provides a foundation for future research in refining crime prediction systems, contributing to advancements in law enforcement technologies

Keywords: Neural Networks, Crime, Deep Learning, Anomaly Detection, Transfer learning, CNN

1 Introduction

CCTV surveillance has been around for almost 70 years and has been a common choice among law enforcement agencies and the general population to monitor abnormal activities for public or personal safety. The global prominence of CCTV systems is evident in the exponential growth of the market, with projections soaring from a substantial $35.47 billion in 2022 to a staggering $105.20 billion by 2029, reflecting a robust CAGR of 16.8% during the forecast period, 2022-2029 [1]. However, over the years, even with the widespread deployment of CCTV cameras, the increasing population and rapid urbanization have led to an alarming surge in criminal activities[2][3]. Given the increasing abundance of data collected on surveillance feeds and the climbing crime rates, it becomes increasingly overwhelming and expensive to rely on human monitoring of surveillance cameras.

Hence, in the face of these evolving trends, the need for more sophisticated and intelligent systems for automated prediction and detection of crimes and monitoring of video surveillance arises. The use of Artificial Intelligence(AI) in video surveillance has become a hot topic for research in recent years [4]. The amount of research on the use of neural networks for real-time crime detection has also seen significant growth over the past few years [5] with the developments in machine learning practices in the 21st century. Machine learning has become a popular choice among researchers owing to how well ML techniques scale to large amounts of data [6]. Neural network(NNs) is a form of machine learning technique that attempts to model and learn based on the functionality of human brains and is considered the most powerful clustering technology available for unstructured data[6] which includes grid-like data such as images and video data. Therefore, this paper aims to propose, develop and compare deep-learning networks to identify crimes using surveillance footage. The different models are compared using metrics determining accuracy and computational costs. The problem at hand can be divided into two parts: Step 1 - Detection of a crime, Step 2 - Identifying and classifying the type of crime taking place. To accomplish this, the study focuses on using transfer learning on some well-studied and current up-to-date CNN models - Densenet121, VGG-19, ResNet50 and MobileNet V2 , and compare the different performance rates for test images. The models are re-trained on a large dataset containing millions of labelled images featuring occurrences of crimes from various CCTV footages.

Convolutional Neural Networks (CNN) or ConvNet, is one of the most widely used techniques of deep learning for the purposes of object detection, recognition and image classification problems, over the years, due to CNN achieving state-of-the-art accuracy for such tasks, CNN has been presenting an operative class of models for better understanding of content present in an image, therefore resulting in better image recognition, segmentation, detection, and retrieval [7].

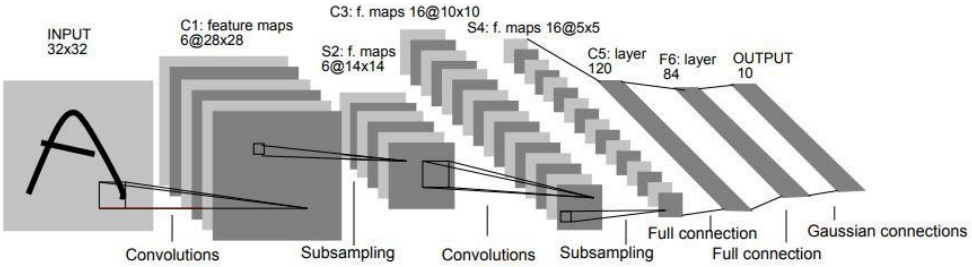


Fig. 1: Basic CNN architecture

However, their effectiveness is notably contingent on large-scale datasets and substantial computational resources. The training of CNNs demands an extensive amount of labelled data and computational power, making it a resource-intensive and time-consuming process. The paradigm of transfer learning emerges to aid with this problem. Transfer learning is an ML technique whereby a model is trained and developed for one task and then re-used for similar tasks with minimal modifications in the output or some of the hidden layers [8]. Transfer learning offers a significant benefit by alleviating the necessity for an extensive dataset when training a deep CNN, by fine-tuning a portion of the parameters from a pre-trained model in the source domain using limited labelled data from the target domain, transfer learning can yield better performance on the target dataset.[3]

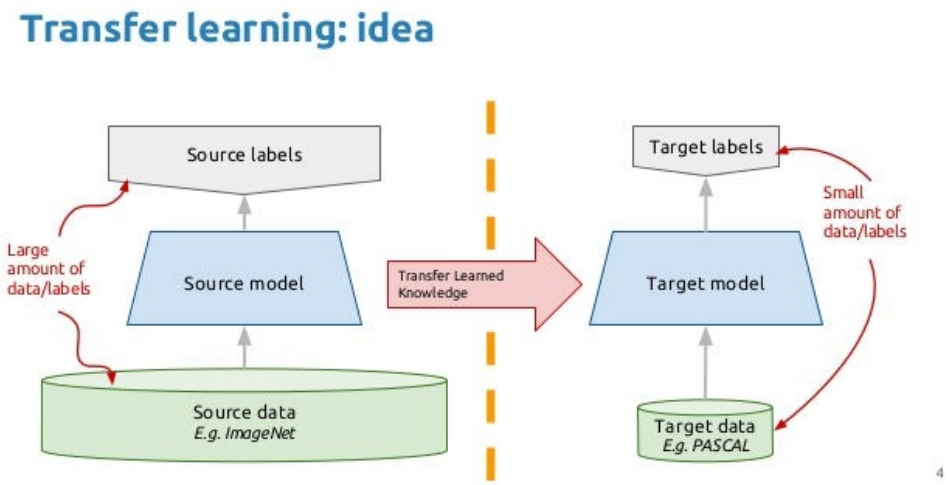


Fig. 2: Basic intuition behind transfer learning

2 Literature Review

As proposed in the Crime In India analysis paper [9], criminal cases in India have undergone an increasing trend as opposed to the world. Conventional damage control policies involve the availability of law-enforcement officers to sieve through Closed Circuit Television (CCTV) footage. Surveillance through CCTV in public spaces has been an aiding tool to solve crime, as well as prevent crime from happening [10]. Many a time, criminals will be prevented from committing the crime in areas where a CCTV is plainly in sight, thereby catching their actions. However, individually monitoring every other video snippet to observe abnormal activities gets exceedingly monotonous, elaborate, and time-consuming. It demands a workforce, and consistent attention 24*7.

Deep learning methods breakthroughs in recent years have aided the automation in menial tasks like of anomaly detection. Anomaly detection has a long history in statistics, and artificial intelligence, and is a lauded problem. Integrating Anomaly Detection with Convolutional Neural Networks has seen an increase in the recent years. Kowshik and Shoeb's paper on Real Time Crime Detection[11] proposed YOLOv5 as an effective object detection algorithm, using just a single convolutional neural network. In the research paper, YOLOv5 was compared with its predecessors on a custom real-time face recognition dataset. This lays the groundwork for the advent of deep learning techniques in tackling age-old problems like that of anomaly detection in sensitive situations pertaining to those of surveillance.

Vipin Shukla's 2015 paper on Automatic Alert of Security Threat[12] proposed the techniques of background subtraction, coupled with human outline detection using edge estimator algorithms. The result is then used to analyze human posture in subsequent frames thus classifying their activities as suspicious or benign. Although their paper proposes identifying whether an abnormal behaviour is happening, it doesn't theorize on how to gauge the nature of this behaviour.

Nandhini T J's 2023 paper [13] tackled the problem of crime objects not being visible in areas with deficit lightning. Automatic Night-time monitoring sensors are crucial to identify crime objects since most of them can be missed if inspected by the naked eye. The author compares the accuracy detecting 7 variables, namely: knife, cellphone, car, animals, gun, blood, and currency by deploying a model for object detection in IR (infrared) images. The author proposed CNN architecture and trained the model with 147 images, with the accuracy of detecting knife being the highest at 99.8%.

The following figure illustrates all the studies that were reviewed to create a comprehensive study of the existing literature on real-time crime analysis using deep learning techniques. By building upon this existing body of literature, this research paper aims to compile and compare the various methodologies used to implement the pre-existing research.

Author Reference	Origin	Purpose	Type Of Source	Major Themes
[1]	India (2023)	To curate a holistic review of the varied types of crime and their nature committed in India through the years 2000-2010	Research	Crime Trend analysis in India
[2]	India	To analyse the impact of CCTV installations with respect to property offences in a medium size district using standard GIS tools	Research	Surveillance through CCTV Cameras.
[3]	India (2023)	To provide a thorough overview of the application of deep learning in real-time crime detection. This paper digs into the various deep learning arch.	Research	Incorporate the use of deep-learning techniques in crime surveillance.
[4]	USA (2013)	To design a system that will detect threat in time under different lighting conditions using camera and sensor networks.	Research	Motor detection, Background Subtraction Algorithms to detect crime in real-time in dimlit situations.
[5]	India (2023)	To record a dark scene and identify the things at a crime scene with the help of infrared camera footage.	Research	Identifying crime objects.

Fig. 3: Review of existing related studies

3 Methodology

This section outlines the general flow of the development of the models from choosing the appropriate dataset to testing the models all of which are discussed in detail here.

3.1 Dataset

The data used for training the NN models is a modified and sized-down version of the open-source UCF-Crime Dataset, obtained from Kaggle[14]. THe UCF-Crime Dataset consists of long untrimmed surveillance videos which cover 13 real-world anomalies, including Abuse, Arrest, Arson, Assault, Road Accident, Burglary, Explosion, Fighting, Robbery, Shooting, Stealing, Shoplifting, and Vandalism. These anomalies are selected because they have a significant impact on public safety [15]. The Kaggle dataset contains images(64*64 px) extracted from every video from the UCF Crime Dataset. Every 10th frame is extracted from each full-length video and combined for every video in that class. This is done so that the size of the larger UCF dataset can be reduced without losing any of the spatial and temporal information between the images.

(a) Arson (b) Arrest

(c) Shooting (d) Explosion

(e) Road Accident (f) Normal

Fig. 4: Samples from the categories taken from UCF dataset

Fig. 3 presents a few samples of crimes from the different categories present in the UCF crime dataset. Fig. 3 (a) depicts a man pouring gasoline right outside the victim's house before igniting it, Fig. 3 (b) shows several policemen attempting to arrest someone after a car crash, Fig. 3 (c) is the footage of a person laying unconscious after getting shot, Fig 4 (d) shows a large scale explosion occurring, Fig. 3 (e) shows an incident of a road accident with a vehicle flipped over a person and Fig. 3 (f) shows an instance of normal occurrence

3.2 Data Preprocessing

After acquiring the dataset, the data was partitioned into test and training sets. The next steps in preprocessing utilize the Tensorflow Keras pipeline to resize the image data to the standard 64*64 px dimensions, apply the respective preprocessing for each of the four models and generate new training data by applying data augmentation techniques to the training data using the 'ImageDataGenerator' from the Keras library. Data augmentation improves the accuracy of the model and helps in reducing overfitting by improving the generalization ability of the model. Generalizability refers to the performance difference of a model when evaluated on previously seen data (training data) versus data it has never seen before (testing data). Models with poor generalizability have overfitted the training data [16]. The techniques used to achieve the same include horizontal flipping, random width shifts (up to 10%), and random height shifts (up to 5%). The image pixels were subsequently normalized to the range [0,1] by dividing each pixel value by 255 to reduce the computational complexity and speed up network training.

3.3 Review of training models used

Transfer learning has been used for training the following models :

3.3.1 DenseNet121

One of the key problems with these traditional CNNs is as the number of layers in the CNN increases, i.e as they get "deeper", the gradient of the loss function starts to diminish, also known as the "vanishing gradient problem", DenseNets resolve this problem by modifying the standard CNN architecture and simplifying the connectivity pattern between layers. In a DenseNet architecture, each layer is connected directly with every other layer and for 'L' layers, there are L(L+1)/2 direct connections[17]. This allows for feature reuse and requires fewer parameters than a traditional CNN, and helps in reducing overfitting[17]. DenseNet121 is a variant of DenseNet and contains 121 layers trained on large datasets such as CIFAR-100 and ImageNet. In terms of architecture, each dense block consists of a varying number of layers featuring two convolutions each; a 1x1-sized kernel as the bottleneck layer and a 3x3 kernel to perform the convolution operation followed by a transition layer containing a 1x1 convolutional layer and a 2x2 average pooling layer with a stride of 2[17]. Densenet121 has been studied for crime prediction achieving a AUC score of 82.91%[18].

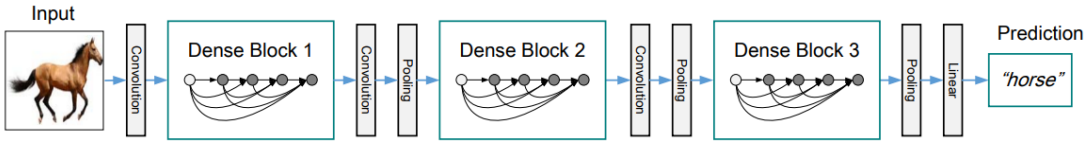


Fig. 5: DenseNet with three dense blocks. Source - G. Huang et al. 2018 from [17]

Layers	Output Size	DenseNet-121	DenseNet-169	DenseNet-201	DenseNet-264
Convolution	112 × 112	7 × 7 conv, stride 2			
Pooling	56 × 56	3 × 3 max pool, stride 2			
Dense Block (1)	56 × 56	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$
Transition Layer (1)	56 × 56	1 × 1 conv			
	28 × 28	2 × 2 average pool, stride 2			
Dense Block (2)	28 × 28	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$
Transition Layer (2)	28 × 28	1 × 1 conv			
	14 × 14	2 × 2 average pool, stride 2			
Dense Block (3)	14 × 14	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 24$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 32$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 48$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 64$
Transition Layer (3)	14 × 14	1 × 1 conv			
	7 × 7	2 × 2 average pool, stride 2			
Dense Block (4)	7 × 7	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 16$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 32$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 32$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 48$
Classification Layer	1 × 1	7 × 7 global average pool			
		1000D fully-connected, softmax			

Fig. 6: Densenet architectures. Source - G. Huang et al. 2018 from [17]

3.3.2 VGG19

VGG19 being an acronym for "Visual Geometry Group 19" is one of the most often used image recognition architecture in the present day. The term "19" connotes 19 weight layers—16 convolution layers, 3 fully connected layers, 5 maxpool layers, and 1 softmax layer. VGGNet takes an input image size of 224x224 RGB; the first two layers are convolution layers with the kernel size of 3x3 of stride 1, and these layers use 64 filters each that result in a volume of 224x224x64 of the same padding. The small size of convolution filters allow VGG to have a larger number of weight layers, leading to improved accuracy. After this a batch pooling layer with a max-pool of size 2x2 and stride 2 resulting in a reduction of height and width from 224x224x64 to 112x112x64 and so on. The VGG convolution layers are followed by a ReLu unit—it is a piecewise linear function that will output the input if positive; otherwise, the output is zero. The VGGNet has three fully connected layers, the first two having 4096 channels each, and the third has 1000 channels. VGGNet has achieved 92.7% top-5 test accuracy in ImageNet: a dataset consisting of 14 million images belonging to more than 1000 classes.

ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224 × 224 RGB image)					
conv3-64	conv3-64 **LRN**	conv3-64 **conv3-64**	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 **conv3-128**	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 **conv1-256**	conv3-256 conv3-256 **conv3-256**	conv3-256 conv3-256 conv3-256 **conv3-256**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

Fig. 7: VGG configuration. Source - Simonyan and Zisserman et al. 2014 from [**?**]

Column E showcases the internal architecture of VGG19 that we have employed as our model.

3.3.3 ResNet50

Residual Networks are another class of neural networks that solve the problem of vanishing gradient and high training error by introducing residual learning. In residual learning, instead of trying to learn some features, we try to learn some residual. Residual can be simply understood as the subtraction of feature learned from input of that layer which it achieves by introducing several shortcut or residual connections which allows the input to bypass one or two layers[19]. The skip connections perform identity mapping, and their outputs are added to the outputs of the stacked layers.ResNet-50 is a 50-layer deep convolutional neural network, trained on more than a million images from the ImageNet database and has an input image of size 224*224.The architecture is similar to the VGGNet consisting mostly of 3X3 filters. From the VGGNet, the shortcut connection as described above is inserted to form a residual network. Resnet achieved a top 5 accuracy of 92%.

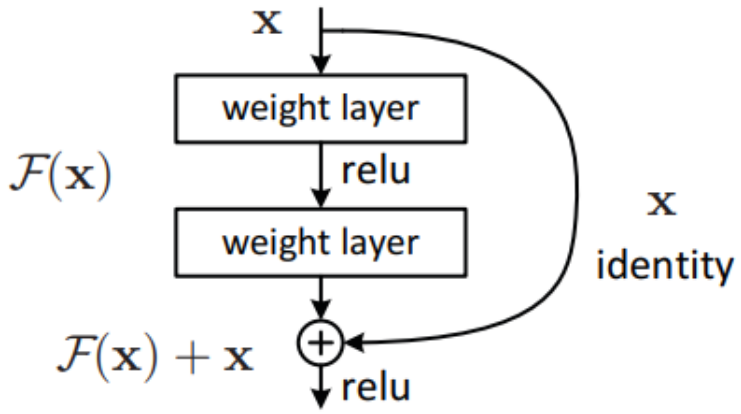


Fig. 8: Residual Connection. Source - Xiangyu Zhang et al 2015 [19]

3.3.4 MobileNetV2

As can be inferred from its name, MobileNetV2 is a CNN architecture that is aimed to perform well in mobile devices. In its previous versions, MobileNetV1 was focussed on reducing the complexity cost and model size of the network by utilizing Depthwise Separable Convolution. The basic idea is to replace a full convolution layer into two separate layers. The first layer is called a depthwise convolution, it performs lightweight filtering by applying a single convolutional filter per input channel. The second layer is a 1×1 convolution, called a pointwise convolution, which is responsible for building new features [**?**]. MobileNetV2 also employs "Inverted Residuals" building upon the intuition that the bottleneck layer, albeit being on a lower dimensionality, has all the necessary information. Using this information, MobileNetV2 establishes shortcut connections between the bottleneck layers; thereby being more memory efficient than its counterparts.

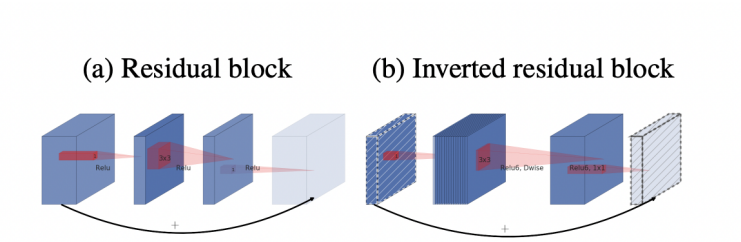


Fig. 9: Residual and Inverted Residual Connection. Source - Mark Sandler et al 2019 [**?**]

Input	Operator	Output
$h \times w \times k$	1x1 conv2d , ReLU6	$h \times w \times (tk)$
$h \times w \times tk$	3x3 dwise s=s, ReLU6	$\frac{h}{s} \times \frac{w}{s} \times (tk)$
$\frac{h}{s} \times \frac{w}{s} \times tk$	linear 1x1 conv2d	$\frac{h}{s} \times \frac{w}{s} \times k'$

Fig. 10: Bottleneck layer transformation table. Source - Mark Sandler et al 2019 [**?**]

4 Analysis and Report

4.1 Evaluation Metrics

The evaluation metrics used to compare the effectiveness of the models used are discussed here. Metrics used include precision score, F1-score and ROC-AUC score. Precision is given by equation 1.

$$\text{Precision} = \frac{TP}{TP + FP} \tag{1}$$

True Positive(TP) is the total number of occurrences where the crime/anomaly was correctly detected whereas False Positive(FP) gives the number of occurrences where the crime was falsely detected. A low precision would mean that the model predicts some false positives and labels some normal occurrences as crimes. This type of error is unwanted but allows a human analyser to review and correct the false alarm. The other useful metric is recall given by Equation 2

$$\text{Recall} = \frac{TP}{TP + FN} \tag{2}$$

Falses Negatives(FN) is the number of occurrences where the model was unable to detect criminal activity in the process.This type of error can be life-threatening since it would lead to late response time by law enforcement authorities. F1-score acts as a binding metric that unifies the precision and recall and gives a single score to judge the model's accuracy against some baseline. F1-score is given by equation 3 :

$$\text{F1} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{3}$$

The other powerful metric used is the Area under the ROC curve which has gained much popularity for multiclass classification problems. diagnostic ability of a binary classifier system as its discrimination threshold is varied. It is created by plotting the True Positive Rate (TPR) against the False Positive Rate (FPR) at various threshold settings.

The AUC score offers a quick overview of the ROC curve, shedding light on how well a classifier can distinguish between different classes. The greater the AUC score, the better the model is.

4.2 Results and discussion

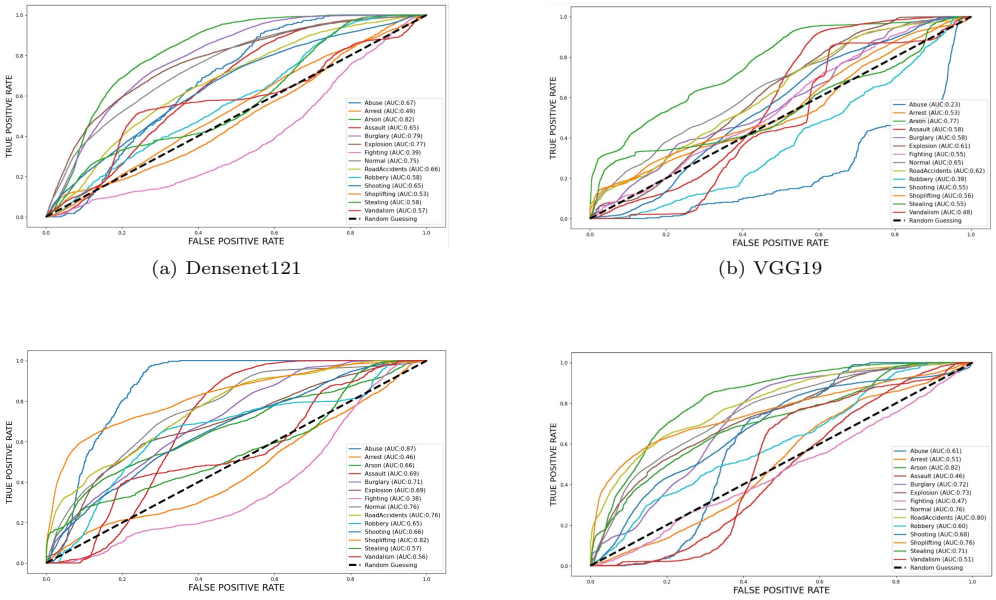
The models were implemented using the Tensorflow Keras module. The selection of hyperparameters was driven by the consideration of the dataset's relatively large size, aiming to strike an optimal balance between model convergence, stability, and computational efficiency. The models were tuned for 1 epoch (to avoid prolonged training times) with a batch size of 64 which is a common choice for deep learning models [20]. The models were compiled using SGD(for DenseNet121) and Adam(for VGG, ResNet and MobileNet) optimizers with a learning rate of 0.00003 and with the loss function of 'categorical_crossentropy'. Each model's feature extraction layers were frozen and initialized with the pre-trained weights, and the fully connected layer at the top of the network was excluded for modification to suit our classification needs. After extracting the features, Global Average Pooling (GAP) was applied to each model to reduce the feature map. Following GAP, dense layers were added to the models with ReLu activation (three dense fully connected layers with 256, 1024, and 512 units in the case of DenseNet and a single dense layer with 512 units for VGG19, ResNet50 and MobileNetv2), each dense layer being followed by a dropout layer to mitigate overfitting. Finally, a dense layer is added with a softmax activation function. This approach of freezing the extraction part of a pre-trained model and modifying the classification parts is efficient, as it saves significant computational resources and time, and effective, as it can improve model performance.

Table 1: Metrics Table

Model	Precision	F1-Score	AUC Score
DenseNet121	0.5601	0.6407	0.8361
VGG19	0.5659	0.5680	0.5464
ResNet50	0.5659	0.5660	0.6082
MobileNetV2	data 7	data 8	0.6525

Table 2: AUC scores for different classes across the models

Class	DenseNet121	VGG19	ResNet50	MobileNetV2
Abuse	0.67	0.23	0.87	0.61
Arrest	0.49	0.53	0.46	0.51
Arson	0.82	0.77	0.66	0.82
Assault	0.65	0.58	0.69	0.46
Burglary	0.79	0.58	0.71	0.72
Explosion	0.77	0.61	0.69	0.73
Fighting	0.39	0.55	0.38	0.47
Normal	0.75	0.65	0.76	0.76
Road Accident	0.66	0.62	0.76	0.80
Robbery	0.58	0.39	0.65	0.60
Shooting	0.65	0.55	0.66	0.68
Shoplifting	0.53	0.56	0.82	0.76
Stealing	0.58	0.55	0.57	0.71
Vandalism	0.57	0.48	0.56	0.51



(a) Densenet121 (b) VGG19

(c) ResNet50 (d) MobilenetV2

Fig. 11: Comparison of ROC curves for different crime classes

The AUC score of 1 indicates a perfect classifier while the score of 0.5 implies the model hasn't learnt anything but instead is making random guesses, The closer the curve is to the top-left corner, the higher the true positive rate (sensitivity) for a given false positive rate (1-specificity), which indicates a better-performing model and conversely, a point along the diagonal line from the bottom left to the top right indicates that the true positive rate equals the false positive rate, representing a classifier that performs no better than random chance.

Judging by the metrics and the ROC curves we can safely conclude that DenseNet121 performs considerably better than the other models followed by MobileNetV2. The key observation to be made is that different classes of anomalies perform better or worse on different models. 'Arson' performs better than all the other classes across all models except in ResNet50 with the top performers being DenseNet and MobileNet with an AUC score of 0.82. Whereas, the class 'Fighting' and 'Arrest' turned out to be the worst performers with below 0.5 AUC scores across the models. The paper [21], points out that the models struggle with instances of explosion and shooting due to smoke being a common entity accompanied in both instances.

5 Conclusion

The main motive of this study was to utilize and compare different frameworks for real-time anomaly detection as an aftereffect on surveillance camera footage snippets from the UCF crime dataset. We employed the fundamentals of 4 models, namely: DenseNet121, VGG19, ResNet50, and MobileNetV2. The pinnacle aim of these findings lies in the contribution to the pre-existing biosphere of literature done in the advent of Deep learning techniques in real-life situations.

The general structure of our framework follows the principle of Convolutional Neural Networks, thereby proposing a model that works upon weakly-labeled training videos. Since Densenet121 has the highest AUC score, it is deemed the best model at predicting positive and negative classes as true. Inferring from the ROC curve, we can conclude that DenseNet121 performs the best followed by MobileNetV2. ResNet50 and VGG19 do not outperform each other patently, rather they outshine others on selected anomalies for instance the category of 'Abuse'.

6 Scope For Future Work

The research done in this paper is limited to the UCF crime dataset. As noted in this paper [21], the UCF crime dataset must focus more on the crime scene frame rather than having weakly-labeled anomaly and no-anomaly clippings. This creates limitations for the accurate training of our model. UCF-Crime's testing set comprises 92.4% of normal frames and 7.6% of abnormal ones [22] , this reiterates the need for more fitting evaluation metrics to test on unbalanced datasets. Furthermore, UCF dataset only accounts for 13 anomaly classes, but doesn't train the model on how to detect normality patterns in the clippings.

The next stepping stone to further our study would be to fine-tune our dataset. This can be achieved by integrating our visual data with sensory data such as audio, and infrared to improve evaluation accuracy. Furthermore, issues pertaining to object detection in dim-light areas is another avenue to be considered. Furthermore, considering the significant variations in performance across different classes on specific models, it opens up avenues for further research in crafting more tailored models. Drawing inspiration from the discussed architectures, there is room to explore the incorporation of additional elements such as LSTM to capture and leverage temporal information present in CCTV footage. This approach aims to enhance the adaptability of models to diverse scenarios and improve overall predictive accuracy

References

[1] Business Insights, F.: CCTV camera market size, growth: Global report [2022-2029] (2023). https://www.fortunebusinessinsights.com/cctv-camera-market-107115

[2] Malik, A.A.: Urbanization and crime : A relational analysis. (2016). https://api.semanticscholar.org/CorpusID:22424407

[3] Ansari, S., Verma, A., Dadkhah, K.: Crime rates in india. International Criminal Justice Review **25** (2015) https://doi.org/10.1177/1057567715596047

[4] Nguyen, M.T., Truong, L.H., Tran, T.T., Chien, C.-F.: Artificial intelligence based data processing algorithm for video surveillance to empower industry 3.5. Computers & Industrial Engineering **148**, 106671 (2020)

[5] Mandalapu, V., Elluri, L., Vyas, P., Roy, N.: Crime prediction using machine learning and deep learning: A systematic review and future directions. IEEE Access (2023)

[6] Mena, J.: Machine learning forensics for law enforcement, security, and intelligence. (2011). https://api.semanticscholar.org/CorpusID:113740871

[7] Sharma, N., Jain, V., Mishra, A.: An analysis of convolutional neural networks for image classification. Procedia Computer Science **132**, 377–384 (2018) https://doi.org/10.1016/j.procs.2018.05.198 . International Conference on Computational Intelligence and Data Science

[8] Hussain, M., Bird, J., Faria, D.: A study on cnn transfer learning for image classification. (2018)

[9] Aakanksha Singh, A.B. Ashwin Singh: Crime in india. (2024)

[10] Rohit Malpan, M.C.: Impact of cctv surveillance on crime. (2021)

[11] Kowshik, D.Y.R.D. Shoeb: Real time crime detection using deep learning. (2023)

[12] Shukla, V., Singh, G., Shah, P.: Automatic alert of security threat through video surveillance system. (2013)

[13] J, N.T., Thinakaran, K.: Detection of crime scene objects using deep learning techniques. In: 2023 International Conference on Intelligent Data Communication Technologies and Internet of Things (IDCIoT), pp. 357–361 (2023). https://doi.org/10.1109/IDCIoT56793.2023.10053440

[14] Kaggle UCF dataset, howpublished = https://www.kaggle.com/datasets/odins0n/ucf-crime-dataset/data, note = Accessed: 2010-09-30

[15] Real-world Anomaly Detection in Surveillance Videos, howpublished = https://www.crcv.ucf.edu/projects/real-world/, note = Accessed: 2010-09-30

[16] Shorten, C., Khoshgoftaar, T.M.: A survey on image data augmentation for deep learning. J. Big Data **6**(1) (2019)

[17] Huang, G., Liu, Z., Maaten, L., Weinberger, K.: Densely connected convolutional networks. (2017). https://doi.org/10.1109/CVPR.2017.243

[18] Hasija, S., Peddaputha, A., Hemanth, M.B., Sharma, S.: Video anomaly classification using densenet feature extractor. In: Tiwari, R., Pavone, M.F., Ravindranathan Nair, R. (eds.) Proceedings of International Conference on Computational Intelligence, pp. 347–357. Springer, Singapore (2023)

[19] He, K., Zhang, X., Ren, S., Sun, J.: Deep Residual Learning for Image Recognition (2015)

[20] Bengio, Y.: Practical recommendations for gradient-based training of deep architectures (2012)

[21] Dua, A., Kalra, B., Bhatia, A., Madan, M., Dhull, A., Gigras, Y.: Crime alert through smart surveillance using deep learning techniques. In: Proceedings of the 4th International Conference on Information Management & Machine Intelligence, pp. 1–8 (2022)

[22] Caetano, F., Carvalho, P., Cardoso, J.S.: Unveiling the performance of video anomaly detection models — a benchmark-based review. Intelligent Systems with Applications **18**, 200236 (2023) https://doi.org/10.1016/j.iswa.2023.200236