Grenze International Journal of Engineering and Technology, June Issue

Anomaly Detection in Surveillance Camera using Deep Learning Techniques

Mr.B.Bizu[1], Saumya.S[2], Sowmiya.N[3] and Sowndhariya.J[4]
[1]Assistant Professor, Kongu Engineering College, Perundurai, India
Email: bizu.cse@kongu.edu
[2-4]Kongu Engineering College, Perundurai, India
Email: {saumyas,sowmiyan,sowndhariyaj}.20cse@kongu.edu

***Abstract*— The purpose of safeguarding public safety,protecting vital infrastructure, and preventing security breaches, anomaly detection in surveillance camera systems is an essential responsibility. Deep learning techniques have made significant progress in recent years in terms of increasing the precision and effectiveness of anomaly identification. The DenseNet architecture is used in this paper to provide a unique technique for anomaly identification in surveillance camera data. Convolutional neural networks (CNNs) like DenseNet are renowned for their dense connection and capacity to recognize intricate elements in images. To improve the functionality of the system, we suggest utilizing its benefits in the context of anomaly detection. Our method's primary objective is to effectively find anomalies through the utilization of the spatial correlations that already exist between pixels in images.**

***Index Terms*— Anomaly Detection, Surveillance Camera DenseNet Architecture, Deep Learning, Security,Public Safety.**

I. INTRODUCTION

The amount of video data generated by security camera proliferation in recent years has increased exponentially, making the development of intelligent systems that can identify anomalous events in real-time necessary.
Anomaly identification in surveillance footage is crucial for maintaining public safety, preventing crime, and defending important infrastructure. Conventional video surveillance systems rely mostly on human operators to monitor, which is laborious, prone to error, and oftentimes ineffective in identifying subtle irregularities or unusual actions. Therefore, utilizing cutting-edge deep learning algorithms is essential to overcoming these issues and enhancing the precision and efficacy of anomaly identification in surveillance systems.
In this research, we provide a novel approach for anomaly identification in surveillance films by using the properties of DenseNet architecture, a deep convolutional neural network well-known for its efficient feature extraction and representation learning. We intend to increase the discerning of complex spatial patterns and temporal dynamics inside video sequences, leading to enhanced anomaly detection performance compared to current approaches, by using the capability of DenseNet, which encourages feature reuse through dense connections between layers.
The primary objective of our research is to develop an intelligent framework that can automatically identify anomalous behaviors or events in real-time video frames, thereby circumventing the limitations of existing

Grenze ID: 01.GIJET.10.2.127

surveillance systems. Our proposed strategy has the potential to advance video surveillance technologies, change public safety regulations, and allow for timely action in situations where life or death is at stake.
The deployment of the DenseNet architecture for anomaly detection in security cameras is thoroughly reviewed in the article. We assess the suggested model's effectiveness using numerous surveillance datasets,accounting for various challenging conditions and contextual aspects. We then compare the outcomes with cutting-edge anomaly detection methods to show how our strategy is superior in terms of accuracy, robustness, and computing effectiveness.
The full analysis of relevant research and current approaches for anomaly detection in surveillance footage is presented . DenseNet's architecture and design ideas are described with special features and benefits in the context of anomaly detection highlighted. Our suggested methodology, which includes model training, data preparation, and assessment measures, is presented. We offer the experimental findings and a performance analysis of the suggested strategy. The fundamental motivation for this project is to overcome the issues of detecting anomalies in surveillance camera data in real time. We propose a system that can independently and accurately identify irregularities, suspicious actions, or objects of interest. A system like this would not only revolutionize video surveillance, but it would also improve public safety and security in a variety of fields.
This document is formatted as follows: An overview of pertinent research on identifying anomalies in Surveillance video is given in Section 2, along with a discussion of the drawbacks and advantages of the current approaches. In Section 3, the architecture and guiding principles of DenseNet are explored, with a focus on highlighting its distinctive features within the anomaly detection domain. Our suggested methodology,which includes model training, evaluation metrics, and data preparation, is presented in Section 4. We report the experimental findings, performance analysis, and comparisons with the most advanced anomaly detection methods in Section 5. The work is finally brought to a close in Section 6 with an overview of the main conclusions, their consequences, and suggestions for further research.
Overall, this research sets the groundwork for the creation of intelligent video analytics solutions with improved anomaly detection capabilities and supports the ongoing attempts to use deep learning techniques to increase the efficacy of surveillance systems.

II. Related Works

Anomaly detection is a difficult task, numerous approaches—such as real-time, online, detection algorithms have been developed and used in the literature. Every approach has benefits and drawbacks of its own. Nevertheless, in order to identify unusual activity, the majority of anomaly detection systems require real-time, online analysis of video frames. The literature has presented a variety of deep neural network-based methods for anomaly detection on abstraction feature learning [1]–[3] . A dictionary is created using the normal pattern as the basis for the sparse reconstruction approach [4]–[7], which models some normal patterns for anomaly detection. Patterns with a significant reconstruction error are then identified as anomalies. For anomaly detection, the researchers have also used the agents technique.
The hybrid agent is one of the method [8]. This method divides anomalous behavior into two categories: single models and groups. The two models are distinguished by static and dynamic agents. Ryotaet al. [9] coupled environment-dependent anomaly detectors with convolution neural networks to identify and explain combined aberrant occurrences. The suggested model was first trained on a large amount of environment data, and then it was taught knowledge particular to the environment derived from actual experience. A deep GMM model was created in [10], appearance and motion data were simultaneously extracted using PCANet. The anomalous events in the surveillance footage were successfully identified by the deep model. For one-class novelty detection, denoising auto-encoders and GANs were employed in [11]–[13] to adversely learn latent representations. For unsupervised anomaly identification in medical pictures, a deep convolution neural network was employed using transfer-level learning and ImageNet for feature extraction [14]. Without making any assumptions on the nature of novelties beforehand, a system was presented in [15] that employed a deep auto-encoder to learn the probability distribution of its underlying latent representations using a parametric density estimator using autoregression. Two models were presented by the authors in a recent work [16] in an effort to address the drawbacks of domain prior and model bias and enhance anomaly detection performance. Based on the log likelihood ratio of two models, the first model was created that were identical, while the second method made use of a multi-scale model that was similar to Glow.
Numerous investigators utilize traditional methods; they focus on normal patterns and label anomalies when they arise. For example, the combination of dynamic models on texturing is offered in [17], [18] demonstrates the

spatial-temporal Markov random field; [18], [19] discusses Gaussian process modeling; [20] discusses the method based on the social force model; [21, 22] discusses Models on Hidden Markov. A few writers provide unique techniques for identifying irregularities. They used the motion and appearance data from the preceding sequence to anticipate the full frame using Adversarial networks that generate new ones using U-net as a predictor [23].
Nineteen unsupervised anomaly detection techniques are evaluated on ten distinct datasets in the study " Goldstein and Uchida's " An Analysis of Unsupervised Anomaly Detection Techniques for Multivariate Data in Comparison " (2016). It attempts to rectify the deficiency of a common assessment among the scientific community and offers a fresh foundation for unsupervised anomaly identification. For the first time, the review highlights the advantages and disadvantages of the various strategies and provides guidance on choosing an algorithm for common real-world tasks.
A unique publication " A Two-Tier Classification Model and Two-Layer Dimension Reduction for Anomaly-Based Intrusion Detection in Internet of Things Backbone Networks " by Pajouh, Javidan, Khayami, Ali, and Choo (2016) presents a model for intrusion detection in IoT networks. It proposes the use of a To identify malicious activity, a two-tier classification module and two-layer dimension reduction are used, such as User to Root (U2R) and Remote to Local (R2L) attacks.. Published in the IEEE Transactions on Computing's Emerging Topics, the concept 3–4, is intended to tackle the difficulties associated with anomaly-based incursion detection in backbone networks for the Internet of Things.
A machine learning-based strategy is put out in the paper "A Machine Learning Approach for Imputing and Identifying Anomalies in Internet of Things Environment " by Vangipuram, Gunupudi, Puligadda, and Vinjamuri (2020) to solve imputation problem and anomaly behaviour in IoT environments.The technical idea is based on imputing missing data in IOT devices and using mL techniques to identify network anomalies or assaults. The work adds to the current body of research in this area by focusing on the application of cutting-edge machine learning methods to enhance IoT environments' dependability and security.
We put forth the deep learning model known as DenseNet-121 which particularly used for classification of images acurately through their dense layers.Since we need to classify labeled abnormal and normal images there is a need of models like DenseNet121 to identify image patterns , object detection which provides granular insights.

III. METHODOLOGY

A. Transfer Learning

Within the fields of machine learning and deep learning, transfer learning is a powerful concept that has fundamentally altered the development and application of models. Transfer learning is essentially the process of applying knowledge from one profession or domain to another, related but unrelated activity or domain, in order to enhance performance. It's similar to applying knowledge gained from one area of expertise to another, hence increasing the effectiveness and efficiency of learning.
Through transfer learning, a model can apply the knowledge gained from solving one problem to another. This information transfer occurs when model parameters, representations, or features are transferred from a pre-trained model to a target model. There are several tasks and domains in which transfer learning can be applied An image classification model, for example, that has been trained on a large photo dataset, can apply its feature knowledge to tasks such as object detection, segmentation, and even medical image analysis. Various models of transfer learning exist, such as:
Learning new skills by observing individuals in relevant fields or tasks is known as inductive transfer learning. Transductive Transfer Learning is the process of tailoring a model to a certain target task or topic by using sparse data. Unsupervised Transfer Learning: The process of imparting knowledge through unsupervised learning techniques. Transfer Learning's Advantages Quicker Instruction. Transfer learning starts with a model that has already been trained, which expedites the training process. Better Generalization: This often leads to better generalization on the destination work, particularly when the source task is large and important. Downsized .Lower Data Requirements: Pre-trained models can be applied to a variety of related activities, removing the need to develop new models from scratch; this allows transfer learning to succeed even in situations where there is a dearth of labeled data available for the target task.

B. DenseNet-121

Liu Gao Huang, Kilian Q, and Zhuang wrote the scholarly paper "Convolutional Networks with Dense Connections" under Laurens van der Maaten. For convolutional neural networks (CNNs), Weinberger presented

the DenseNet-121 (Densely Connected Convolutional Networks 121) deep learning architecture. The article's initial publication took place in January 2017.

The authors of the research proposed DenseNet as a state-of-the-art architecture for deep learning models, emphasizing improved network information flow. The main characteristic of DenseNet is its dense layer connections, in which every layer is directly connected to every other layer using a feed-forward method. This setting not only resolves the vanishing gradient issue, but it also promotes feature reuse. It offers advantages in terms of training efficiency, parameter efficiency, and model correctness and is frequently used in a variety of computer vision applications, particularly those involving picture categorization. Nowadays, DenseNet is a well-liked and frequently utilized architecture in the deep learning field.Unlike conventional convolutional neural networks (CNNs), DenseNet emphasizes dense connectivity across layers, expanding upon the idea of deep neural networks. Below is a detailed explanation of DenseNet:

Compact Connectivity:It is defined by the denseNet's dense connectivity pattern. In a dense block, every layer is connected to every other layer through feedforward.This implies that each layer's output and the feature maps from all earlier layers are concatenated and used as input for levels that follow. The dense connectivity promotes feature reuse and lessens the vanishing gradient problem, making it useful in terms of parameters and training stability.

Dense Blocks: Comprising multiple densely connected convolutional layers, the dense block is the fundamental component of DenseNet. The network can learn more complex representations as input flows through the layers because the packed blocks create a highly non-linear hierarchy of attributes.

Transition Layers: Transition layers in feature maps can be used to reduce the height and breadth of dense blocks while increasing the number of channels (depth). This shift allows for better parameter efficiency by controlling the model's computational complexity.

Global Average Pooling, or GAP: DenseNet typically uses global average pooling at the network edge as opposed to completely connected layers.This approach computes the average of feature maps across spatial dimensions. This helps avoid overfitting and makes the model more receptive to fine-tuning and transfer learning.

Bottleneck Layers: A few DenseNet variants have bottleneck layers, which use conventional 3x3 convolutions at first, then 1x1 convolutions to reduce the number of channels. As a result, less computing is required.

Growth Rate: DenseNet presents a hyperparameter named "growth rate," which controls the number of new feature mappings that each layer adds to the ones that already exist. This parameter has a big impact on the model's capacity and computational cost. These are performances at the cutting edge:DenseNet models have achieved state-of-the-art results in some computer vision applications, including segmentation, object detection, and image categorization. Their extensive connection, feature reuse, and effective parameter exploitation are the main reasons for their success.

Here, DenseNet's unique dense connection pattern, efficient parameter usage, and ability to create a hierarchy of feature representations make it a potent architecture for a range of computer vision tasks. It has gained popularity in the deep learning community because of its state-of-the-art performance and transfer learning success.

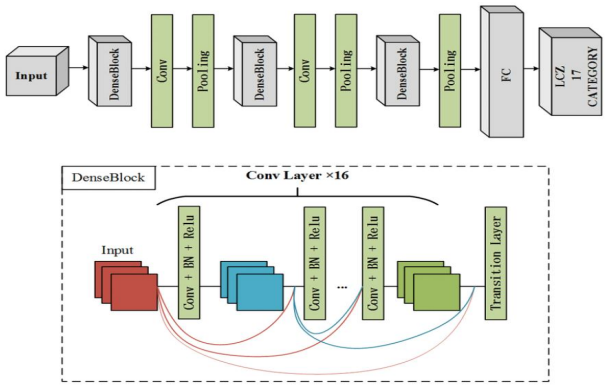

Figure 1. Dense net Architecture

IV. PROPOSED WORKS

Below flowchart shows the proposed system’s process diagram, which includes image preprocessing, training and dataset validation.

A. Dataset Description

Extracted and added to the dataset are images from the UCF crime dataset that are used for actual anomaly identification in surveillance footage.The UCF Crime collection contains images from all of the videos that make up the collection.The tenth frame is extracted and combined for every full-length video in that class.With a.png extension, every image contains 64 by 64 pixels extension. The collection has fourteen classes in all, including Road Accidents, Robberies, Shooting, Shoplifting, Stealing, Vandalism, Arson, Assault,Explosion, Fighting, and Normal Videos.The total image count for the train subset is 1,266,345.The total image count for the test subset is 111,308. All videos used for frame extraction were obtained from UCF CRIME OFFICIAL DATASET.

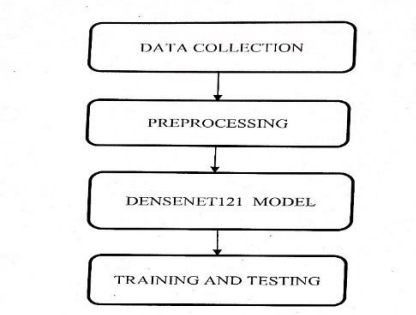

Figure 2. Modules of Proposed System

The following section provides number of images in each classes: Abuse - 297 images , Arrest - 3365 images , Arson - 2793 images , Assault - 2657 images , Burglary - 7657 images ,Explosion - 6510 images , Fighting - 1231 images,Normal - 65000 images , RoadAccidents - 2663 images , Robbery - 835 images, Shooting - 7630 images , Shoplifting - 7623 images , Stealing - 1984 images ,Vandalism - 1111 images.

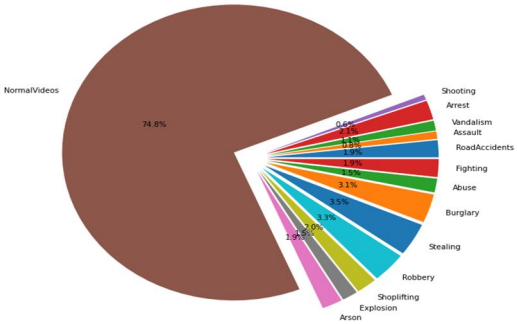

Figure 3. Share of 14 classes

B. Sample Dataset

Figure 4. 14 classes of Dataset

C. Experimental Setup

The meticulous preparation and organization of the data is the first step in setting up a technological experiment. In order to enable obvious separation of training and test sets, it is necessary to ensure that the dataset adheres to the expected directory structure. Data augmentation techniques are employed, along with a carefully considered ratio of training to validation data, to improve the model's capacity for generalization.Important preprocessing parameters for the input, including image size and normalization, are carefully set up to match the architecture of the model.

Model architecture selection, training strategy, and hyperparameter tuning are the main topics of the following phase. In order to maximize model convergence, a great deal of investigation is done to fine-tune hyperparameters, such as learning rate and batch size. To attain the intended augmentation effects, different data augmentation settings are customized. Performance gains are evaluated while making decisions on the model architecture, such as using pre-trained models and setting up the classifier's structure. Model training also heavily depends on the optimizer, loss function, and training monitoring techniques selected. Every experiment is meticulously documented, carefully analyzed, and evaluated using precise metrics to guarantee that the results are instructive for subsequent iterations. Reproducibility and scalability are hallmarks of the experimental procedure, which aims to produce the best possible outcomes in the field of crime classification.

D. Data Preprocessing

Resizing Images: A standard size is applied to all of the visuals in the dataset defined by the `IMG_HEIGHT` and `IMG_WIDTH` constants. This resizing ensures that all images have the same dimensions, making them compatible with the model's input layer. This step is crucial for ensuring uniformity in the data.

Data Augmentation: This is a technique that applies different picture changes to the training dataset in order to artificially boost its diversity. The `ImageDataGenerator` is utilized in this code to augment the data on the training set. It covers methods such as rescaling the pixel values to the range [0, 1] ({rescale{), flipping the image horizontally ({horizontal_flip}), and adjusting the image's height and width (`height_shift_range}). Data augmentation lowers the chance of overfitting and helps the model grow more adaptable to modifications in the data.

Normalization: Pixel values of an image are normalized to a common scale. This is achieved by dividing the pixel values by 255.0. Normalization helps in stabilizing the training process and can lead to faster convergence during model training.

Preprocessing Function: The code also employs a preprocessing function, which is set to preprocess_fun tf. keras. applications. densenet. preprocess_input`. This function is specific to the DenseNet pre-trained model used in the code. It applies additional preprocessing specific to the model, such as mean subtraction and scaling pixel values according to the model's pre-trained weights. This ensures that the input data is processed in a manner consistent with the model's expectations.

Data Generators: The preprocessed data is generated in batches using data generators. These generators yield batches of images and their corresponding labels during training and testing. They also handle data shuffling, which is crucial for introducing randomness and reducing potential bias during training.

These preprocessing steps collectively ensure that the input data is standardized, augmented for diversity, and prepared in a format that is suitable for training a deep learning model. In order for the model to learn and produce correct predictions based on the image that was given data, preprocessing is an essential step in the pipeline.

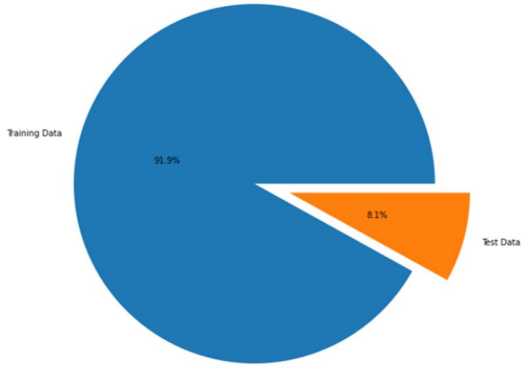

Figure 5. Share of train and test images

E. Model Architecture

A pre-trained DenseNet121 model serves as the feature extractor in the model, which establishes architecture. IMG_HEIGHT and IMG_WIDTH are the dimensions of the images that the DenseNet121 architecture is configured to capture using the `feature_extractor` function. This trained model is the foundation for feature extraction from our crime dataset.The components that make up the DenseNet architecture are the fully connected (FC), convolution (conv), Rectified Linear Unit (ReLu) layer, and pooling (pool), which is also known as the dense layer.ReLu is used to zero out missing and negative data, the pooling layer reduces the parameters, and the convolution layer extracts the feature. In order to obtain the result after pooling, the output is flattened and fed into a loop that uses the softmax activation function. For this instance, Adam is the optimizer.The Transition Layers and 1) Dense Blocks comprise DenseNet's architecture. While allowing for a range of filter types, dense blocks have the same block dimensions. The graphic DenseNet Blocks and layers represents DenseNet blocks and layers. The transition layer in CNN is an important step in the batch normalization application procedure.

E. (a)Input Layer

A 3D tensor that represents the pixel values for the height, width, and color channels (red, green, and blue in an RGB image, for example) is usually utilized to accept an image by the input layer.

E. (b) Initial Convolutional Layer

A series of filters are used to convolve the input image in the first convolutional layer in order to extract low-level information. These features might include simple patterns, edges, and color variations.

E. (c) Dense Blocks

Dense blocks are a series of densely connected units. Convolutional layers are present in many units. A crucial aspect is that, inside a single dense unit, each layer's output is concatenated with the outputs of all layers that came before it.

A dense connectivity allows the model to access and reuse features from earlier layers, promoting feature reuse and information flow throughout the network. In anomaly detection, this dense connectivity helps capture intricate patterns and abnormalities by leveraging the information from various levels of abstraction.

E. (d)Transition Layers

A 1x1 convolutional layer (to reduce channels numbers), a batch normalizing layer, and a 2x2 average pooling layer are included in the transition layer that follows each dense block. This combination helps compress information and downsample the spatial dimensions.

In anomaly detection, the transition layers assist in reducing the spatial dimensions of feature maps, which can be critical for capturing both global and local patterns. The 1x1 convolutional layer also helps in channel reduction, making the model more computationally efficient.

E.(e) Global Average Pooling Layer

The last dense block or transition layer's output is fed into the global average pooling layer, which uses it to calculate the average value for every channel across all spatial dimensions.

In anomaly detection, this layer provides a holistic representation of the entire image, condensing the information obtained from different scales and levels of abstraction. Anomalies, being deviations from the normal pattern, are expected to manifest as differences in this high-level representation.

E. (f)Fully Connected Layer

In a classification issue, the fully connected layer maps the result of global average pooling to the necessary number of classes.

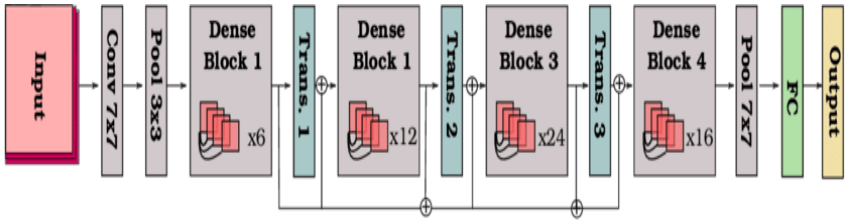

Figure 6. DenseNe Blocks and Layers

F. Model Compilation

The model's optimizer is called Stochastic Gradient Descent (SGD). a well-liked neural network optimization technique. The model's weight updates during training are influenced by the optimizer selection. The loss function mentioned in the code is categorical cross entropy, and it works well for multi-class classification applications. During training, the model will try to minimize this loss. Furthermore, the evaluation metric chosen by the code is the ROC AUC, or Area Under the Curve. This statistic offers a thorough evaluation of the model's categorization performance.

G. Training

The model then goes on to train itself. Over a predetermined number of epochs, Using the preprocessed training data, the model is trained. During training, iteratively model is adjusts its internal weights to minimize the category cross-entropy loss. This process involves forward and backward passes through the network, updating the model's parameters to improve its predictive abilities.

H. Validation

Following training on the training set, the model, its performance is assessed using the preprocessed validation data. This stage guarantees the model's generalizability to fresh, untested data. It's essential for spotting problems like overfitting, in which the model works well with training data but has trouble with fresh test data.

I. Model Evaluation

The ROC AUC score is calculated to evaluate the trained model's overall performance in classification. A single statistic that sums up the model's capacity to differentiate between various crime categories is the ROC AUC score. Additionally, the model generates ROC curves for each class, visually illustrating how well the model performs for individual crime categories. These curves help identify variations in the model's performance across different classes.

J. Visualization

The model employs various visualizations to present key information. The creation of pie charts to show the distribution of training and test data, bar charts to display the number of images per crime category, and pie charts to depict the distribution of images among different crime categories. These visualizations aid in understanding the dataset's composition and the model's performance.

V. Parameter For Evaluation

A. Accuracy

Accuracy is a popular assessment statistic in deep learning that gauges a classifier's overall effectiveness in properly predicting both positive and negative events. The classifier's accuracy is measured as the percentage of accurate predictions it produced out of all the guesses it made. The following formula can be used to calculate accuracy in binary classification tasks:

$$\text{Accuracy} = (TN + TP) / (FP + TN + TP + FN)$$

where TN stands for true negative predictions, FP for false positive predictions, FN for false negative predictions, and TP for true positive predictions.

By dividing the total number of occurrences in the dataset by the total number of occurrences that were correctly predicted, accuracy in multiclass classification problems is determined.

Being able to gauge the classifier's overall performance, accuracy is a crucial deep learning parameter. High accuracy indicates that the classifier is making accurate predictions, while low accuracy indicates that the classifier is making many incorrect predictions.

But occasionally, accuracy on its own can be deceptive, particularly if the dataset is unbalanced or the costs associated with making erroneous positive and false negative predictions are unequal. In some situations, additional metrics like F1 score, accuracy and recall could provide a more insightful evaluation of the classifier's performance.In conclusion, accuracy is a helpful indicator for assessing a classifier's overall performance; but, for a more thorough assessment of the classifier's performance, accuracy should be used in conjunction with other metrics.

B. Loss

Since every input to our model is expected to yield a black or white mask, the desired output is a binary decision for each pixel. For the purpose of classifying each pixel between the two classes of 0 and 1, the model behaves in a manner similar to a binary classifier. Binary cross-entropy, sometimes referred to as log loss, is a fairly simple and widely used loss function for binary classification. A function called binary cross-entropy determines the cross-entropy between the true and expected classes. It produces a prediction value between 0 and 1 for each pixel. This is the formula for binary cross-entropy. where y is the label (black pixels are 0) and then p(y) is represents as the predicted chance that a given pixel would be white for all N pixels.

VI. Results And Discussion

The results of trained DenseNet121 model to classify anomaly images into specific crime labelled images. In contrast with existing methods to find anomalies it detects only whether the situation is normal or abnormal whereas it do not identify the type of abnormal event. With respect to this our trained DenseNet121 model provides more efficiency and accuracy by finding types of abnormal event. It infers the importance of deep learning techniques for providing the model more advanced and efficient. Our trained DenseNet121 model classifies into 14 classes including normal events whereas others are abnormal events. We trained the DenseNet121 model on train and validation data. The DenseNet121 model's accuracy and loss function on the train and validation data are graphed against the number of epochs. The graph's ROC curve shows how the false positive rate (FPR) and true positive rate (TPR) correlate for different classification criteria. The ratio of falsely categorized negative samples as positive (FPR) to Measurement of sample proportionality is expressed in terms of correctly identified positive samples (TPR). For a given ROC, Area under the ROC curve, or AUC, is what it means.

The AUC of ROC figure indicates how well the model can distinguish between positive and negative classifications. Better performance is indicated by a higher ROC AUC score. The trained DenseNet121 model's ROC curve demonstrates that the model can obtain a high ROC AUC score across all classes. This suggests that the model can identify anomalies well.The model's ROC AUC score is 0.91, which is an excellent result. This shows that the model has a high degree of accuracy in differentiating between normal and anomalous data.

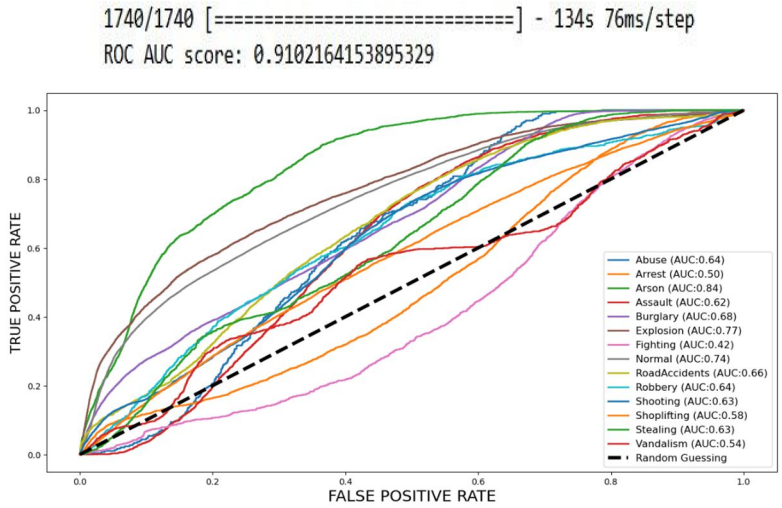

Figure 7. ROC Curve

VII. Conclusion And Future Enhancements

Deep learning methods are an effective way to find anomalies in surveillance photos. Deep learning models are scalable to handle massive volumes of visual data and can detect anomalies with extremely high accuracy. It's critical to comprehend the limitations of applying deep learning to this task, such as the need for a sizable amount of training data and the model's complexity. The system will become more beneficial in the future if the following changes are made: Once the system is trained, it can be deployed to monitor surveillance camera image in real time. The system will automatically generate alerts when it detects an anomaly. The alerts will include information about the type of anomaly and the location where it was detected.

If the system can be developed as such as then it is used to improve security and surveillance in a variety of settings, such as airports, train stations, public buildings, and private businesses and also the system can be used to develop novel uses for video monitoring, namely the prompt detection and avoidance of crimes.

REFERENCES

[1] X. Ruan , H. Lu ,Y. Zhang, L. Zhang, ‘‘When detecting anomalies, motion and appearance clues are combined.’’ Pattern Recognition, vol. 51, March 2016, pp. 443–452.

[2] L. S. Davis , A. K. Roy-Chowdhury, J. Neumann, J. Choi and, M. Hasan ‘‘Acquiring knowledge of temporal regularity in video clips”, in IEEE Conf. On Computer Vision and Pattern Recognition (CVPR), June 2016, pp. 733–742.

[3] Y. H. Tay and, Y. S. Chong ‘‘Using a spatiotemporal autoencoder to detect abnormal events in videos’’ in Neural Network Advances. Springer, Cham, Switzerland, 2017.

[4] S. Gao, W. Liu, and, W. Luo ‘‘A review of anomaly detection using sparse coding in a stacked RNN framework,” in Proc. IEEE Int. Conf. Comput. Vis. (ICCV), 341–349. October 2017.

[5] J. Liu, J. Yuan, and Y. Cong, ‘‘Disparate reconstruction expense for identifying anomalous events,’’ pp. 3449–3456 in Proc. CVPR, Jun. 2011.

[6] E. P. Xing , L. Fei-Fei, and, B. Zhao ‘‘Unusual event detection in movies online using dynamic sparse coding’’ pp. 3313–3320 in Proc. CVPR, June 2011.

[7] J. Jia, J. Shi, and, C. Lu ‘‘150 FPS abnormal event detection in MATLAB, in IEEE Int. Conf. Comput. Vis., Dec. 2013, pp. 2720–2727.

[8] H.-B. Kang and, S.-H. Cho ‘‘Detecting unusual behavior in crowded environments with hybrid agents’’ Pattern Recognition Letters, vol. 44, July 2014, pp. 64–70.

[9] S. Satoh, T. Mei, and, R. Hinami ‘‘Combined deep general knowledge learning with anomalous event detection and recounting’’ arXiv:1709.09121, 2017.[Online]. [http://arxiv.org/abs/1709.09121] is accessible.

[10] X. Lu, Y. Yuan, and, Y. Feng ‘‘Acquiring deep event models to detect anomalies in crowds,’’ Neurocomputing, Jan. 2017, vol. 219, pp. 548–556.

[11] B. Xiang ,R. Nallapati, and P. Perera,‘‘ OCGAN: GANs with limited latent representations for one-class novelty detection’’ ArXiv:1903.08550, 2019. [Online]. [http://arxiv.org/abs/1903.08550] is accessible.

[12] M. Han,R. M. Summers J. Xiao, and, Y. Tang, ‘‘Classifying normal and bad chest radiographs via deep adversarial one-class learning’’ The article 1095018 is published in the Proc. SPIE Med. Imaging, Computer-Aided Diagnosis, vol. 10950, March 2019.

[13] A. Konushin, I. Fedulova B. Bakker, and, N. Tuluptceva ‘‘Perceptual abnormality detection in images,’’ pages 164–178 of Proc. Asian Conf. Pattern Recognit., 2020.

[14] P. Krishnaswamy ,M. F. Chiang , J. Kalpathy-Cramer ,C.-S. Foo , J. P. Campbell, B. Unnikrishnan , V. Chandrasekhar H. Yang, M. Romain, H. Zenati, C. Garcin and, K. Ouardini ‘‘In order to effectively detect anomalies on retinal images without supervision,’’ at pp. 225–234 of Proc. Int. Workshop Med. Image Learn. Less Labels Imperfect Data, 2019.

[15] R. Cucchiara, S. Calderara, A. Porrello, and, D. Abati ‘‘Autoregression in latent space for detecting novelty,’’ ArXiv:1807.01653, 2018. [Online].[http://arxiv.org/abs/1807.01653] is accessible.

[16] D. Zhang , T. Ball ,Y. Zhou, and, R. Tibor Schirrmeister ‘‘Recognizing anomaly detection using deep invertible networks by using feature and distribution hierarchies,’’ arXiv:2006.10848, 2020. [Online]. [http://arxiv.org/abs/2006.10848] is accessible.

[17] N. Vasconcelos , V. Mahadevan, and, W. Li, ‘‘Identifying and locating anomalies in congested environments,’’ IEEE Trans. Pattern Anal. Mach. Intell., Jan. 2014; vol. 36, no. 1, pp. 18–32.

[18] Y.-L. Chen , Y. Ou, H. Guo, X. Wu, D. Xu, and, N. Li ‘‘Gaussian process-based anomaly detection in video surveillance,’’ International Journal of Pattern Recognition and Artif., vol. 29, no. 06, September 2015, Art. no. 1555011.

[19] W.-H. Fang , Y.-T. Chen, and, K.-W. Cheng ‘‘Gaussian process regression and hierarchical feature representation are used in the detection and localization of video anomalies.,’’ Pages. 2909–2917 in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), June 2015.

[20] M. Shah, A. Oyama, and R. Mehran ‘‘Detecting abnormal crowd behavior with the social force model,’’ in Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition, June 2009, pp. 935–942.

[21] T. Xiang, S. Gong, and, T. Hospedales ‘‘A topic model for Markov clustering to analyze video behavior’’ IEEE 12th International Conference on Computer Vision, September 2009, pp. 1165–1172.

[22] K. Nishino and, L. Kratz ‘‘Identifying anomalies in densely populated environments with spatiotemporal motion pattern modeling,’’ in Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition, June 2009, pp. 1446–1453.

[23] Y. LeCun, C. Couprie, and, M. Mathieu ‘‘Video prediction on a deep multi-scale surpassing mean square error’’ CoRR, November 2015, vol. abs/1511.05440.