Journal of Computer Science

Original Research Paper

Panoramic Video Surveillance: An Analysis of Burglary Detection Based on YOLO Framework in Residential Areas

Pavithra S. and B. Muruganantham

Department of Computing Technologies, School of Computing, SRM Institute of Science and Technology, Kattankulathur, Chennai, India

Article history
Received: 23-12-2022
Revised: 26-08-2023
Accepted: 30-08-2023

Corresponding Author:
Pavithra. S
Department of Computing Technologies, School of Computing, SRM Institute of Science and Technology, Kattankulathur, Chennai, India
Email: ps8335@srmist.edu.in

Abstract: Artificial Intelligence (AI) is a technique that incorporates human intelligence into mundane activities. And there is no question that AI is significantly affecting security and surveillance. Although relying on numerous resources, finding answers, and implementing technology for decades, our security and surveillance systems still have flaws. In every country around the globe, the use of video security and surveillance is becoming more widespread. Nonetheless, a wide range of businesses has made use of it, including hospitals, universities, and warehouses. Yet, as people are limited in their ability to vigilantly monitor live video streams, deep learning was developed to better fill the position. Unfortunately, there are other problems with images in the real world, including jitter or blurring caused by rotating objects, noise, and sharpness concerns. As a result, deep learning technology for surveillance has considerably improved in recent years. The main objective of this study is to detect burglars using deep learning technology. This system aims to take video surveillance of the residential environment as input and pass it into the Yolo model to increase the speed and accuracy of the system to detect burglars in the residential. This system mainly concentrates on object detection.

Keywords: Panoramic Video Surveillance, Deep Learning, Burglary Detection, Artificial Intelligence, Yolo

Introduction

Deep Learning (DL) is a branch of artificial intelligence that uses supervised and unsupervised models and techniques to enable computers to perform specific tasks without explicit instructions, relying instead on patterns and inferences from data (Sharma *et al*., 2023). Deep learning methods create a statistical model of the data collected, often called the training phase, to make decisions, predictions, and classifications without being explicitly programmed. Deep learning methods are widely used for various applications, such as natural language processing and computer vision, where it is very difficult to design and implement algorithms with specific rules for the task. Computational statistics, which focuses on using computers to make inferences from data, is closely related to deep learning. Multiobjective research enhances the field of deep learning by making it more efficient, both theoretically and practically. One of the industries that have benefited from the extensive use of artificial intelligence is the surveillance sector, which has witnessed the growing advantages of deep learning in computer vision and the development of vision-based technology strategies (Qi and Han, 2021). Optimized deep learning algorithms and models have been used to detect, track, and identify human anomalous behavior with excellent results. Many studies have been conducted on the basic concepts of detection, tracking, identification, and classification of anomalous human behaviors, following the recent advances in deep learning. Traditional security and surveillance applications, such as theft detection, violence detection, and explosion detection, have adopted deep learning techniques in the field of security and surveillance systems. Video surveillance is the process of monitoring activities in any indoor or outdoor area. Video surveillance systems are used by many organizations for various purposes, such as security management, staff monitoring, anomalous activity detection, identification, etc Using video surveillance devices, the system administrator can observe any activities in the area, including human activity (Shana and Christopher, 2019). Video surveillance

Science Publications

© 2023 Pavithra. S and B. Muruganantham. This open-access article is distributed under a Creative Commons Attribution (CC-BY) 4.0 license.

detects and tracks activities, behavioral patterns, and any changes in the environment. This involves using advanced technologies to collect data remotely. Video surveillance can be done in real-time or data can be stored and analyzed later. A typical smart panoramic video surveillance system consists of cameras and an output interface, such as a monitor. The video surveillance system used in this study aims to detect, track, and identify the activities of any human. There are different video surveillance methods available, such as template-based and object-based models. A template-based model maintains a set of templates for different tasks that were used to perform the activities. On the other hand, an object-based model recognizes the user's activity or behavior patterns by checking a list of objects and their properties. In this study, we use a deep learning technique to detect objects in images and videos i.e., burglars. A deep learning technique called convolution layer takes an image as input and applies bias and weight to it. These biases and weights are the learnable parameters of the learning model. Neurons are the basic units of a neural network. Biases are constants that are passed to the next layer as input. Weights are used to measure the influence of the input on the output. This study uses the Yolo object detection algorithm to detect burglaries in the panoramic video surveillance system. Yolo stands for “You Only Look Once” and it is a simple CNN method that offers both high speed and accuracy. Yolo can detect objects with great efficiency because of its fast processing. It is easy to install and it can work with both GPU and CPU computations.

Intent Involved in Research

As the saying goes, prevention is better than cure. It is better to stop a burglary or a crime before it happens than to investigate how or why it happened. Just like people get vaccinated to prevent diseases, in today's society, where burglary and crime are more common, it is important to have a burglary detection system that can prevent crime from happening. These Burglary Detection Systems are used by authorities in residential areas to detect crimes, thefts, and break-ins before they occur and to stop them.

The intention of the Research

This research aims to improve the performance of object detection i.e., burglary detection by increasing the speed and accuracy of recognizing and locating objects and moving objects. Previous models i.e., RCNN, Fast RCNN, and Faster RCNN (Bhoyar *et al.*, 2021) could recognize objects but not locate them. However, yolo is a powerful and efficient convolutional neural network that can detect and locate objects by predicting bounding boxes around the objects and class probabilities directly from full images in one scan. Yolo can also detect objects in real time, despite being based on a CNN model. This is possible because of Yolo's ability to make simultaneous predictions in a single-stage approach.

One-Stage Object Detector

Yolo is a one-stage detector, while RCNN, Fast RCNN, etc., are two-stage detectors. Two-stage detectors are reliable and accurate but slow. Therefore, we will focus on a one-stage detector in this study. The main components of a one-stage object detector are shown in Fig. 1. The one-stage detector can detect objects without any preliminary steps. The advantage of using a one-stage model is speed, it can make predictions fast.

The single-stage detector uses a fully convolutional neural network to directly extract the location information and the class probabilities of the objects from the input image. It does not need to create a network for region proposals or a network for post-classifications. The fully integrated parallelizing framework is simple and fast for object detection. The main contributions of the paper are:

- To design and improve the speed and accuracy of burglary detection
- The proposed algorithm detects and locates the suspected objects (burglars) in the residential environment using a panoramic video surveillance system
- Finally, the proposed algorithm’s performance is demonstrated by experimenting with dataset sources and the evaluation metrics are analyzed and compared with other deep learning models that were already in use

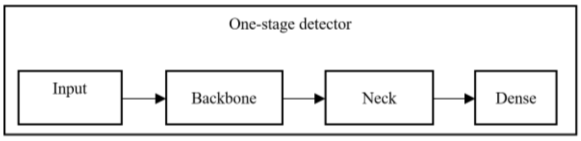


Fig. 1: A framework of a one-stage detector for burglary detection using Yolo

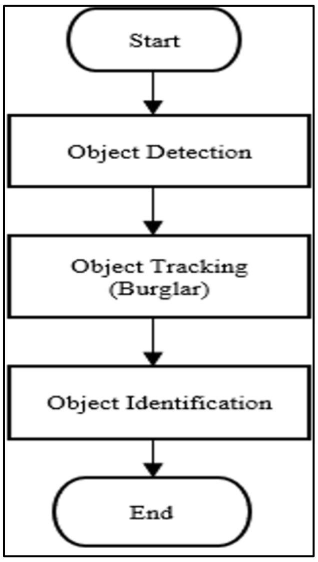


Fig. 2: The flow of procedure illustrates a panoramic video surveillance system for burglary detection

A panoramic video surveillance system consists of three important stages. The first stage is to detect the objects i.e., humans. If it is an existing person in the residence, they can pass. The second stage is to track the relevant objects i.e., a new unknown person entering the residence. The last stage is to identify the suspected person as shown in the flow of procedure in Fig. 2.

Related Works

Humans can easily isolate and identify objects in an image. The visual system can perform complex tasks such as distinguishing different objects and detecting obstacles with little awareness. With the availability of large amounts of data, faster graphics processing units, and better techniques, we can train computers to recognize and classify various objects in an image with high accuracy. We will go through the concepts of object detection, object localization, and the Yolo algorithm (Khobragade *et al*., 2022). A subfield of computer science called "computer vision" focuses on developing systems for understanding images and videos. It has many applications, such as face recognition, object tracking, etc. Detection is used for face recognition, license plate recognition, web images, and security systems. Yolo framework, Region-based convolutional network, fast convolutional network, and SSD are some of the state-of-the-art methods. Yolo is the clear winner when it comes to speed and accuracy. It detects objects quickly and efficiently without compromising performance (Benito-Picazo *et al*., 2020). The continuous monitoring system has a high computational complexity, which is fulfilled by GPU based on deep learning. As a result, more power is consumed leading to higher computation. Microcontroller boards can be used in motion detection systems as a possible solution to these problems because of their low energy consumption and low cost. In other words, microcontrollers are eco-friendly, small, and adaptable hardware (Raghunandan *et al*., 2018). However, this monitoring system is enhanced for implementation in Arduino microcontrollers, which usually lack high computing power. Important works can be highlighted in this regard. The main concern is anomaly detection. This includes identifying real objects like humans, animals, normal and abnormal activities, cars, and bikes. The object detection method extracts the desired area of an object using various image processing techniques (Deshpande *et al*., 2020). The current security and surveillance environment requires a more efficient and advanced monitoring system. Among the various types of deep learning techniques, a convolutional neural network is an effective technique for extracting features from low to very high levels. Infrared cameras have become very popular in recent years for object detection in minimal situations. Moreover, infrared images contain more information than other types of images (Patel and Upla, 2020).

This study presented a system for burglary detection using panoramic video surveillance and the YOLO framework in a smart home environment. They used a 360-degree camera to monitor the indoor activities of the residents and applied the YOLO framework to detect and identify the human faces and bodies in the images (Lee *et al*., 2023). They also implemented a face recognition module to verify the identity of the residents and a motion detection module to detect any abnormal movements. They evaluated their system on a dataset of 200 videos and achieved an accuracy of 90% and a speed of 15 frames per second. Wang *et al*. (2023) proposed a method for burglary detection using panoramic video surveillance and the YOLO framework in an urban area. They used a drone-mounted panoramic camera to capture the aerial view of the city and applied the YOLO framework to detect and classify the vehicles and pedestrians in the images. They also designed a rule-based system to identify any suspicious behaviors, such as loitering, vandalism, or theft. They tested their method on a dataset of 100 videos and achieved an accuracy of 85% and a speed of 10 frames per second. Smith *et al*. (2021) developed a framework for burglary detection using panoramic video surveillance and the YOLO framework in a campus area. They used a network of panoramic cameras to cover the whole campus and applied the YOLO framework to detect and recognize the students, staff, and visitors in the images. They also integrated a face recognition module to verify the identity of the people and a notification module to alert the security personnel or the police in case of any criminal activities. They experimented with their framework on a dataset of 500 videos and achieved an accuracy of 93% and a speed of 18 frames per second. The paper claims that the approach can detect burglary events in real time and with high accuracy by combining the advantages of YOLOv5 for object detection and LSTM for temporal modeling (Li *et al*., 2021). The paper also evaluates the performance of the approach on a self-collected dataset of panoramic videos with different scenarios of burglary. Chen *et al*. (2020) the paper state that the system can achieve panoramic video stitching, object detection, tracking, and recognition by using deep learning methods such as YOLOv3, SORT, and FaceNet. The paper also demonstrates the effectiveness of the system by applying it to a real-world scenario of a smart campus. The paper explains that the method can detect suspicious behaviors of intruders in residential areas by using a panoramic camera and a CNN-based classifier (Kumar *et al*., 2019). The paper also shows the experimental results of the method on a synthetic dataset of panoramic images with different types of burglary. Zhang *et al*. (2023) proposed a hybrid model of

YOLOv5 and LSTM for burglary detection using panoramic video surveillance. The paper claims that the approach can detect burglary events in real-time and with high accuracy by combining the advantages of YOLOv5 for object detection and LSTM for temporal modeling. The paper also evaluates the performance of the approach on a self-collected dataset of panoramic videos with different scenarios of burglary. Chen *et al*. (2023) presented a system for smart campus security using panoramic video surveillance and deep learning. The paper states that the system can achieve panoramic video stitching, object detection, tracking, and recognition by using deep learning methods such as YOLOv3, SORT, and FaceNet. The paper also demonstrates the effectiveness of the system by applying it to a real-world scenario of a smart campus. Kim *et al*. (2023) introduced a method for intruder detection in residential areas using panoramic video surveillance and Convolutional Neural Networks (CNNs). The paper explains that the method can detect suspicious behaviors of intruders in residential areas by using a panoramic camera and a CNN-based classifier. The paper also shows the experimental results of the method on a synthetic dataset of panoramic images with different types of burglary. The paper presents a deep learning model that uses Convolutional Neural Networks (CNNs) and LONG Short-term Memory (LSTM) networks to identify burglary events from video surveillance data. The paper states that the proposed model achieves a high accuracy of 96.7% and performs better than other existing methods (Morales *et al*., 2019). The paper presents a machine learning model that uses Support Vector Machines (SVMs) and K-Nearest Neighbors (KNNs) to identify burglary events from sensor data collected by smart home devices. The paper states that the proposed model achieves a high accuracy of 94.5% and is resilient to noise and missing data (Tastan *et al*., 2019).

Deep Learning-Based Object (Burglary) Detection Approach

The goal of this research proposal is to design and implement a system that can detect and identify burglars and burglary activities in real-world scenarios. To achieve this, we will use object detection techniques that can recognize different objects and their locations in an image. One of the methods we will use is You Only Look Once (Yolo), which is a novel approach that performs object detection in a single neural network. Yolo can predict the class and the bounding box of each object in an image in one shot, making it very fast and efficient. We will also use OpenCV, which is a library of computer vision functions that can help us process and analyze images. By using these methods, we hope to create a system that can automatically monitor and alert us of any suspicious activities or break-ins. We believe that this system will have many applications and benefits in the future, as technology becomes more advanced and intelligent.

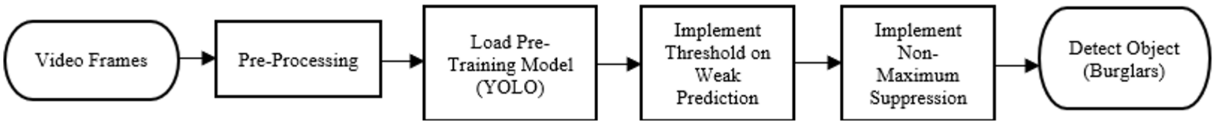

Video Frames
Pre-Processing
Load Pre-Training Model (YOLO)
Implement Threshold on Weak Prediction
Implement Non-Maximum Suppression
Detect Object (Burglars)

Fig. 3: The structure of object (Burglary) detection using Yolo

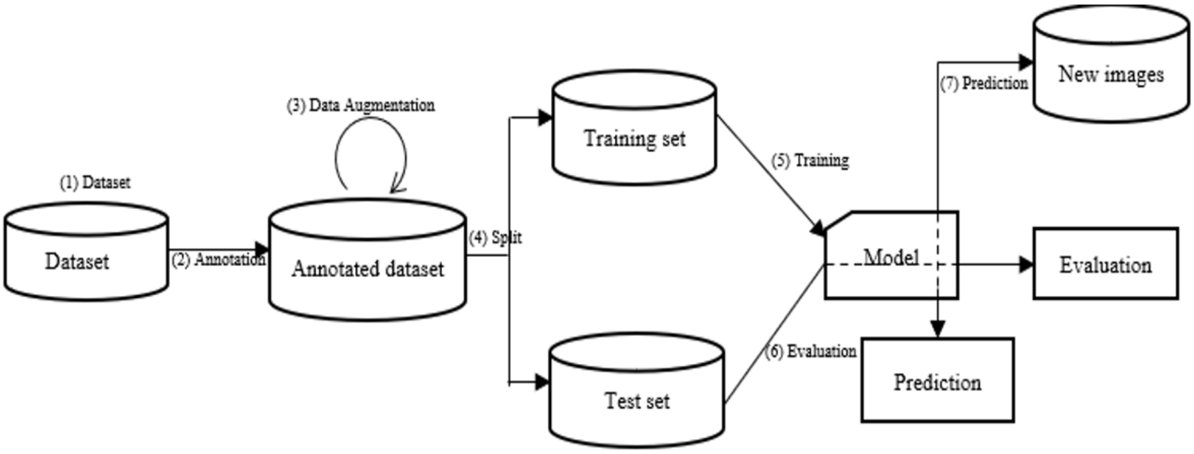


Fig. 4: Workflow of object detection

Object (Burglary) Detection Structure

The proposed methodology for object detection using deep learning is illustrated in Fig. 3. We used the Yolo pre-trained model, which was trained on the MS-COCO dataset that contains 75 classes of objects. This means that our model can detect a variety of objects in the images. Before feeding the images to the Yolo model, we performed some preprocessing steps, such as resizing the images to 412*412 pixels, subtracting the mean RGB values from the images to reduce the effect of lighting variations, and scaling the pixel values to the range of 0-1 (Iqbal *et al*., 2021). The Yolo model consists of 75 hidden layers and outputs a bounding box for each detected object in the image.

Workflow of Object (Burglary) Detection

Here, we present the typical workflow for establishing a strong object detection using deep learning advancement of the deep surveillance sector in Fig. 4, along with the obstacles that each step of the process faces, the remedies we propose to overcome those difficulties are the special features of using the Yolo networks at every stage.

Dataset Acquisition

Regardless of the architecture used to create an object detector, the very first step to creating an object detector is always the gathering of a dataset of imagery including the objects we wish to identify. This is a complex task since the dataset needs to accurately reflect just what model will determine when it is used in reality. Furthermore, it could be difficult to get information on issues requiring biomedical images, for instance. Depending on the project, there are a few different ways to get a dataset of images, however, there are four basic sources:

- Web scraping: We can use software tools that can download images and videos from search engines like Microsoft Images, Google Image Search, Yahoo Search, MSN, and Internet Explorer, as long as they have an Open-Source licenses License that allows us to use them legally for our object detection project
- Special-purpose dataset: In some cases, we may need to obtain data from an external agency that has access to the domain-specific data, such as a hospital for biomedical images. This may be the only option available for some problems
- Open datasets: There are large collections of data that are publicly available for anyone to use, such as Common objects in context and imagenet. These datasets may have a wide range of objects and categories, but they may not have the specific objects that we are looking for
- Specialized devices: We can also capture images using devices that are suitable for our problem domain, such as a scanning electron microscope or a camera mounted in a workplace. This has the advantage of using images that are similar to the ones that the model will encounter in reality; however, it may also be challenging to obtain data from some devices

Dataset Annotation

Just after acquiring the images and videos, they must always be analyzed, which is also a time-consuming task that may necessarily require the guidance of subject matter experts (Kaarmukilan *et al*., 2020). An image in the object detection context is analyzed by supplying a file system consisting of a list of bounding boxes as well as the categorization of the objects present inside that box. All these annotated files are generated using graphs and charts like Yolo mark or Labelimg, rather than by hand. Regrettably, no standardized annotation format currently exists and the formats differ among object detection structures and annotation methods.

Dataset Augmentation

As mentioned in earlier steps obtaining and categorizing image databases for object detection real problems can be challenging, if not enough images are acquired, generalization problems can happen. As a result, data augmentations are an efficient strategy for dealing with the problem of limited data. Using image transformations, this method creates new data for training from the original datasets like filters, rotations, noise addition, and flips. This method was used in image classification tasks and it is implemented in several libraries, for instance, Imgaug and augmenter. The general method of using data augmentation through object recognition was to implement different research methodologies or manually configure and annotate artificially generated images, depending on the problem (Wong *et al*., 2019).

Dataset Splitting

The data collected at the earlier stages need to be split into two separate sets, just like with any machine learning project. A training data set is used to retrain the object detector, as well as a testing set, is used to analyze the model. Somehow the most typical training and testing set splitting sizes are 65/33, 70/20, 85/15, and 95%/10 %.

The dataset obtained in the previous phases needs to be split into two separate sets, just like with any machine learning project. The object detector testing and training set will be used for testing and training respectively. It should be emphasized that the Yolo architecture is very sensitive to a layout and even little alterations will prevent the users from training a model.

Model's Training

Before starting the training process of an object-detection model, various tasks must be completed given the training set of images. In particular, the model's architecture (i.e., the number and type of layers), some hyperparameters such as the epochs, momentum, or batch sizes, and whether or not to use pre-trained weights in the training process must be determined. The latter choice, known as fine-tuning, typically results in better training results. The Yolov3 model is the right model in terms of time efficiency and accuracy out of the pre-defined models included in the Yolo architecture for training an object detection model.

Evaluating the Model

We must evaluate the model after it has been trained to assess its performance. In particular, we show the model every image in the testing set and ask it to identify the objects in the image. The results are assessed using performance metrics like precision, recall, mAP, and IoU after these detections are put side by side with the test set's annotation to serve as the underlying data. If numerous configuration files are appropriately defined and the necessary instruction is called, the Yolo architecture can automatically carry out such an evaluation.

Model Deployment

At last, this system is prepared for use with images that don't come from either the training set or the testing set. Even while utilizing the model with fresh images is frequently as easy as sending a command with the image's path, it is significant to remember that it is uncommon that the person who designed the object identification model is also the end-user of such a model and, possibly, some extra parameter.

Flow of the System Object (Burglary) Detection

The security of residential areas is a major concern for people today. One way to enhance security is to use real-time video surveillance with 360 panoramic cameras that can capture the whole scene. This system aims to develop a video security surveillance automation framework that can detect and identify any potential burglary or break-in activities using deep learning techniques. Figure 5 shows the flow of the system. The steps are:

- A panoramic camera captures the video of the scene and extracts the frames
- The frames are processed by a deep learning model that uses Yolo, which is a fast and accurate object detection method. The model outputs the bounding boxes and the labels of the objects in each frame
- The detected objects are compared with a cloud server that stores the information of the residents and their authorized visitors. The cloud server helps to prevent data manipulation and ensure data security
- If the detected object is a known person from the residents or their visitors, they are allowed to enter their respective homes. If the detected object is an unknown person, they are marked as suspicious and monitored by the system

The paper proposes a new way of measuring and optimizing the overlap between the predicted and the ground truth bounding boxes for object detection. The paper introduces a new concept called Generalized IoU, which is a generalization of the traditional IoU metric. The paper shows that Generalized IoU can overcome some of the limitations of IoU, such as being insensitive to the size and shape of the boxes and being biased towards small boxes. The paper also defines a new loss function based on Generalized IoU, which can be used to train object detection models. The paper demonstrates that using Generalized IoU as a loss function can improve the performance of state-of-the-art object detection models on popular benchmarks like MS-COCO. The paper claims that Generalized IoU is a simple and effective way to improve object detection (Shao *et al*., 2018).

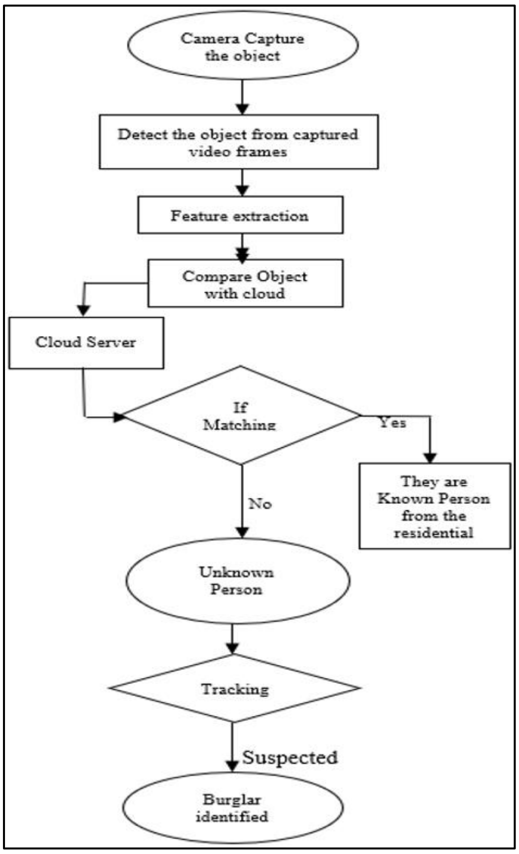

Camera Capture the object
Detect the object from captured video frames
Feature extraction
Compare Object with cloud
Cloud Server
If Matching
Yes
They are Known Person from the residential
No
Unknown Person
Tracking
Suspected
Burglar identified

Fig. 5: The flow of the system illustrates panoramic video surveillance based on burglary detection

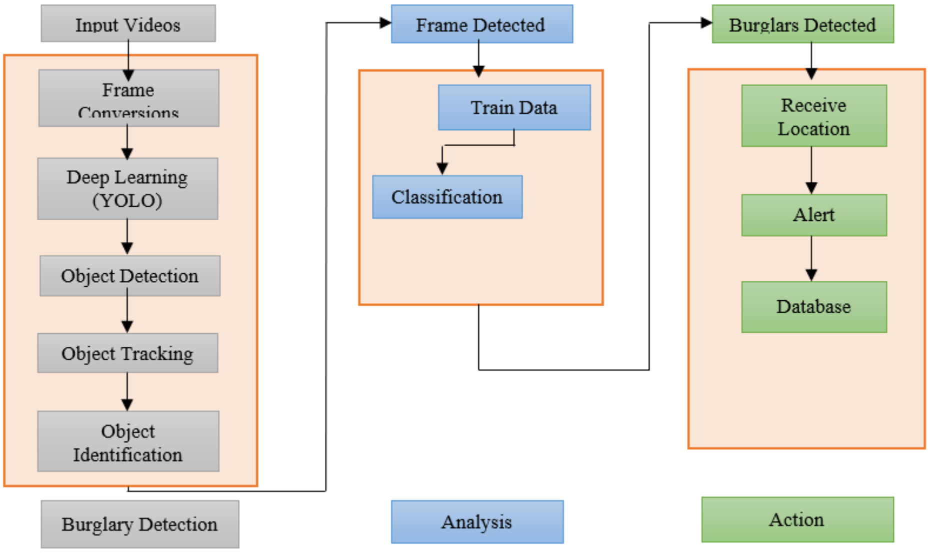


Fig. 6: Overview of object (Burglary) detection illustrates the detection, analysis, and action process

The process of burglary detection using Yolo is shown in Fig. 6. It consists of three steps: Detection, analysis, and action. In the detection step, the video from the surveillance camera is fed to the Yolo model, which is a deep-learning model that can detect objects in the frames. The detected objects are then tracked to identify their movements and behaviors. In the analysis step, the detected and tracked objects are classified as normal or suspicious based on their features and patterns. In the action step, if a suspicious object is detected, an alert is triggered and the location of the incident is reported. A human administrator from the residential area can then intervene and catch the burglar.

Materials

We use the following hardware and software configuration to implement and test our proposed method: Hardware: A desktop computer with an Intel Core i7-9700K CPU, 32 GB of RAM, and an NVIDIA GeForce RTX 2080 Ti GPU.

Software: TensorFlow / Keras (TF/Keras) framework for YOLO. Python 3.8 as the programming language and OpenCV (CV) as the image processing library.

Settings: Input resolution of 412*412 pixels, confidence threshold of 0.5, NMS threshold of 0.4, N_init of 3, and max_age of 30.

We use the following metrics to measure the accuracy and speed of our proposed method on different datasets and scenarios:

Accuracy: We use the mean average precision (mAP) and time taken to evaluate the accuracy of our proposed method. We use the frames per second (FPS) to evaluate the speed of our proposed method.

Methods

This study aims to propose a method for detecting and identifying burglars and burglary activities in residential areas using panoramic video surveillance and security systems. The block diagram of the method is shown in Fig. 7. The method uses a deep learning model called Yolo (You Only Look Once), which can detect objects in the video frames captured by the panoramic camera. The method can also alert a human administrator if any suspicious or abnormal activities are detected in the area. The method requires some preprocessing steps before training the Yolo model. First, we need to collect images of people from the web and save them in a folder called “Pictures”. We need to make sure that the images are in the “jpg format”, as other formats may cause errors or difficulties in training. Second, we need to resize all the images to the same width and height of 412*412 pixels, so that they can be fed to the Yolo model in batches (Ivanov and Kajabad, 2019).

Yolo is a pre-trained object detector. It incorporates a deep neural network. A deep learning technique known as deep neural networks can take a raw image as input and apply learning bias and weight values to various components of the image. As previously stated, Yolo is a pre-train model. The pre-trained model has simply been trained on another dataset. Time to train a model from scratch takes a long time; the training step can take weeks or even months to complete. The model that has already been trained is seeing thousands of items and can categorize them all. Positive weights for the previously mentioned pre-trained model were acquired by training the model using the MS-Coco dataset. Only objects from the classes contained in the training dataset can be detected. Using a pre-trained model's weight, a further Yolo model is being trained to identify and detect burglars or break-ins.

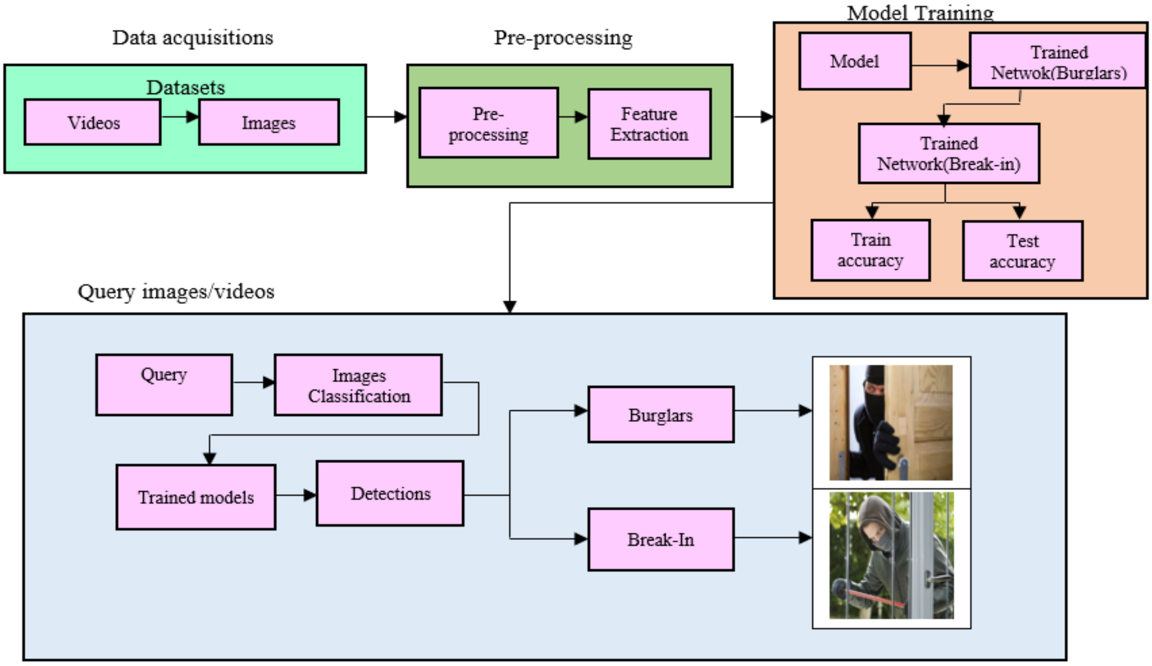


Fig. 7: The Panoramic video surveillance system's methodology and stages are shown in a block diagram

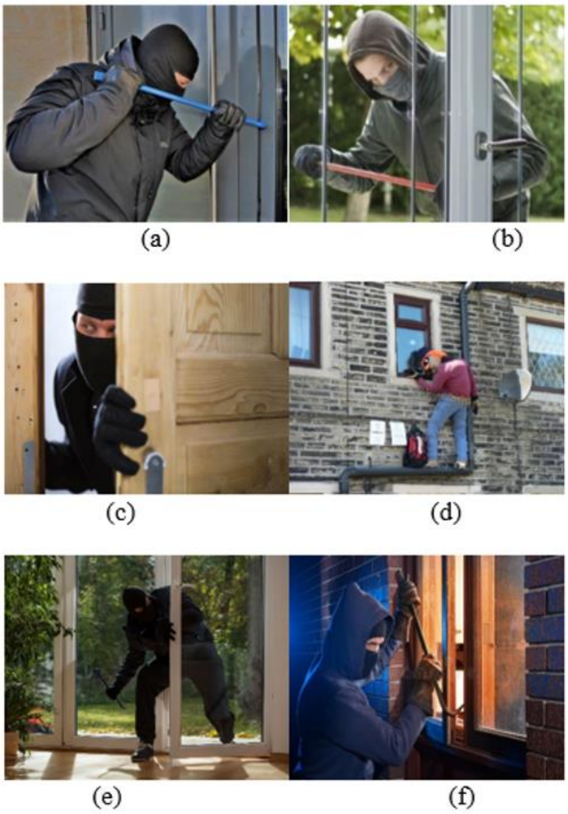


Fig. 8: Demonstrates a few images samples

Rather than an image classifier, Yolo is intended to be a multiscale detector. As a result, replacing the classification head with a detection head in this architecture allows for object detection. The probability classes and bounding box coordinates are now included in the output vector. Yolo's framework is mostly made up of darknet layers. An object detection task is then added 53 more layers on top of it, creating a deep convolution architecture. Owing to its multi-resolution feature extraction layers, Yolo uses three feature maps with different scales to detect burglaries in data like CCTV footage.

The Fig. 8 shows a few image samples that were used in the study. The image samples are from different categories of burglars. The image samples are also of different sizes, resolutions, and formats. The image samples demonstrate the diversity and complexity of the data set that was used to evaluate the performance of the object detection algorithms. The image samples also show some of the challenges and limitations of the object detection algorithms, such as occlusion, illumination, background clutter, and scale variation. The Fig. 8 provides a visual representation of the data set and its characteristics.

Yolo Algorithm Developments

Yolo has been created by redefining and improving certain CNN functionalities. For multi-label classification, Yolo employed independent logistic classifiers. Yolo detects objects from complex images with objects labeled in an overlapping fashion using feature maps at three different scales. The class prediction, bounding box, and confidence score are produced in a three-dimensional tensor by the final convolution layers of the detectors.

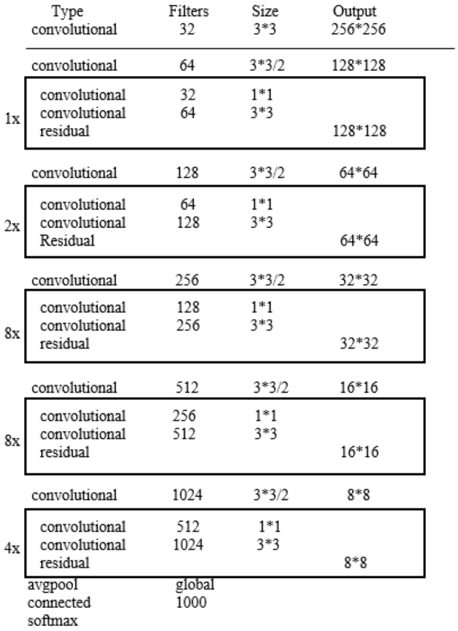
A robust feature extraction network is used by Yolo to recognize objects at various scales and extract the best features for object detection. When compared to previous versions, the loss function has been modified, allowing the model to detect objects at different scales. A feature-extracting network, as well as an object recognition network, are the two key parts of the model (both the models are multi-scale tuned). In the first step of detection, the feature extractor module creates feature embeddings at three distinct scales; in the second stage, the detector module receives these features to acquire anchor box and class probabilities.

The classification network has advanced significantly in comparison to the early darknet-53 network architecture utilized in earlier iterations of the Yolo, instead of just getting deeper it has made a lot of progress in it. When propagating activation across deeper layers, the ResNet model introduced the concept of residual blocks as a means of avoiding the gradient-vanishing issue as in total 53 layers are explained in detail in Fig. 9 an averaging pooling layer and a softmax activation will be added at the end while using the darknet for multi-class classifications. Since the objective is to use this network to provide multi-scale properties for object detection, a detection head has been included. During multi-scale object detection, the feature maps from the final three blocks of the architecture were utilized.

When the trained model generates multiple bounding boxes for a specific object instance, non-max suppressions are used to identify the anchor boxes from a huge collection of overlapping boxes. The output of the model has a vast group of unnecessary or inefficient anchor boxes that need to be filtered out. Retaining the bounding box coordinates with a good possibility in the first step, while those with a low probability are pruned. The Yolo model tries to find the bounding boxes that contain the items in the region of interest, along with their probabilities and class. To construct one box per object instance and suppress extremely overlapping bounding boxes, the idea of IoU is applied.

Figure 9 shows the architecture of Yolo, which is a deep-learning model for object detection. Yolo is based on Darknet 53, which is a neural network framework that has 53 convolutional layers. Yolo adds another 53 convolutional layers on top of Darknet 53, making it a 106-layer fully convolutional network. Yolo uses three different scales of feature maps to detect objects of different sizes in the images. Yolo also uses feature fusion layers to combine the features from different levels and improve detection accuracy.

Figure 9 shows the architecture of Yolo, which is a deep-learning model for object detection. Yolo is based on Darknet 53, which is a neural network framework that has 53 convolutional layers. Yolo adds another 53 convolutional layers on top of Darknet 53, making it a 106-layer fully convolutional network. Yolo uses three different scales of feature maps to detect objects of different sizes in the images. Yolo also uses feature fusion layers to combine the features from different levels and improve detection accuracy.



	Type	Filters	Size	Output
	convolutional	32	3*3	256*256
	convolutional	64	3*3/2	128*128
1x	convolutional	32	1*1	
	convolutional	64	3*3	
	residual			128*128
	convolutional	128	3*3/2	64*64
2x	convolutional	64	1*1	
	convolutional	128	3*3	
	Residual			64*64
	convolutional	256	3*3/2	32*32
8x	convolutional	128	1*1	
	convolutional	256	3*3	
	residual			32*32
	convolutional	512	3*3/2	16*16
8x	convolutional	256	1*1	
	convolutional	512	3*3	
	residual			16*16
	convolutional	1024	3*3/2	8*8
4x	convolutional	512	1*1	
	convolutional	1024	3*3	
	residual			8*8
	avgpool	global		
	connected	1000		
	softmax			

Fig. 9: The architecture of DarkNet-53 layers (Smith *et al*., 2021)

Within the Yolo-based object detection technique, the provided image is divided into S×S grid cells, as well as the objectness score and position of a bounding box for B items within every grid cell are estimated. The below mathematical formula is used to express the objectivity score:

$$C_i^j = P\ (object) * IoU\ (truth, pred) \tag{1}$$

where, *i* and *j* are the bounding numbers and the grid cell, respectively and *C* is the objectivity score. Objectness loss is calculated using the binary cross-entropy gradient descent, which has the following expression.

$$E1 = \sum_{j=0}^{s} \sum_{i=0}^{B} \quad W_{ji}^{obj} [C_i^j \log(C_i^j) - (1\text{-}(C_i^j) \ \log(1 - C_i^j)] \tag{2}$$

where, W_{ji} is the weight for the i^{th} bounding box in the j^{th} cell of the grid and C_{ij} is the confidence score for the i^{th} bounding box in the j^{th} cell of the grid. Where *S* and *B* stand for the overall number of bounding boxes and grid cells, respectively. The expected objectivity score is $C^{\wedge}_{j}$. Each object instance is located by the object detector, which produces four predictions.

YOLO Network Architecture and Algorithm

J. Redmon proposed the YOLO object detection technique based on several research works (Pang *et al*., 2020). It takes the pixel values of the whole image (412*412) as input and directly outputs the bounding boxes, confidence *P*, and class probability of the objects. It treats the detection task as a regression problem. It divides the image into 3*3 grids and predicts *B* bounding boxes for each grid with seven values (bh, bw, bx, by, pc, c2, and c1). The output is a 3*3 matrix. We use Yolo and OpenCV methods to detect all objects and label them with bounding boxes. The steps are shown in Table 1 and Fig. 10.

Table 1: The steps for the proposed method are explained in Fig. 11

Steps	Description
Step 1	Take images of burglar detection first, and then turn them into a grid so we can see the features in the image
Step 2	This stage will specify the directory structure of an image in a NumPy array and openCv will be used to analyze the user input data processing and video points
Step 3	After scanning the images utilizing openCv with NumPy and translating the grids to bounding boxes, finding images in a grid view
Step 4	Displaying the images with a bounding box is the final step. The Yolo Framework and MS-COCO datasets are used to achieve this

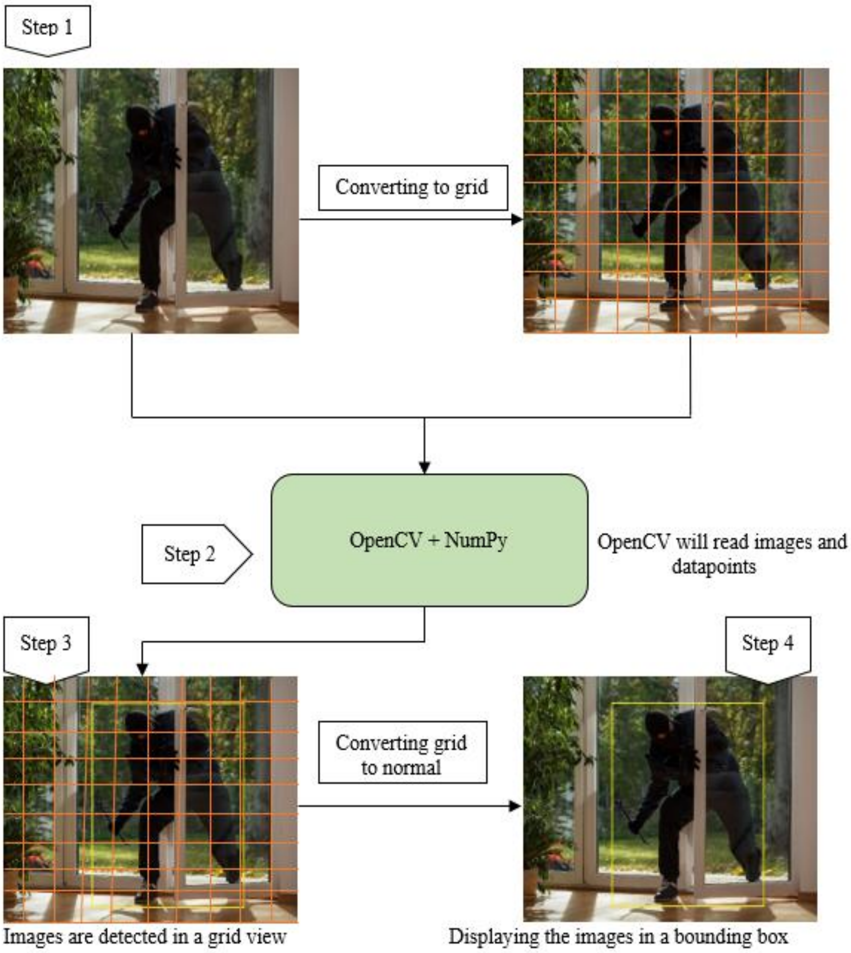

Step 1
Converting to grid
OpenCV + NumPy
Step 2
OpenCV will read images and datapoints
Step 3
Step 4
Converting grid to normal
Images are detected in a grid view
Displaying the images in a bounding box

Fig. 10: The yolo object (Burglary) detection method

The image is divided into a S*S grid and each cell predicts B bounding boxes with confidence. The prediction is a B7 (S*S) tensor. To detect burglars or break-ins by humans, Yolo uses three scales of 13*13, 26*26, and 52*52 grids. Each scale predicts three bounding boxes. The input image for testing is a 21-dimensional matrix. The predicted bounding box for the burglar is shown in yellow in Fig. 11, after applying a confidence threshold filter and non-maximum suppression. Yolo uses a network structure inspired by GoogLeNet with residual connections to improve the learning capacity. The network consists of 1*1 and 3*3 convolution layers. The input vector is 412*412*3 and the output is three tensors of 26*26*21, 52*52*21, and 13*13*21, using a forward passthrough layer and an up-sampling layer.

*Yolo S*S Prediction*

Yolo is a system that detects and tracks objects and their locations. It divides the image into grids as shown in Fig. 11. Each grid cell predicts one object and some bounding boxes. Each bounding box has five elements: A confidence score, a height (h), a width (w), and coordinates (x and y). The confidence score measures how accurate the bounding box is and how likely there is an object in it. The height and width are normalized by the image size and the coordinates are relative to the cell. Therefore, h, w, x, and y are between 0 and 1. Each cell also has 20 conditional class probabilities, which indicate how likely the detected object belongs to a certain class. Yolo trains a convolutional network that outputs a vector. It uses a convolutional network with 1024 outputs at each location to reduce the spatial dimension. Yolo uses two convolution layers for linear regression to predict the bounding box. Convolution layers reduce the spatial resolution, which makes it hard to detect small objects. Other object detectors, such as SSD, use different feature map layers to detect objects. Each layer has a different range of detection. Yolo uses a different method called a passthrough. For example, it converts the 28*28*512 layer into 14*14*2048. Then, it concatenates it with the original 14*14*1024 output layer. It uses a convolution layer on the new 13*13*4096 layer to make predictions. Yolo can handle images of different sizes without fully connected layers.

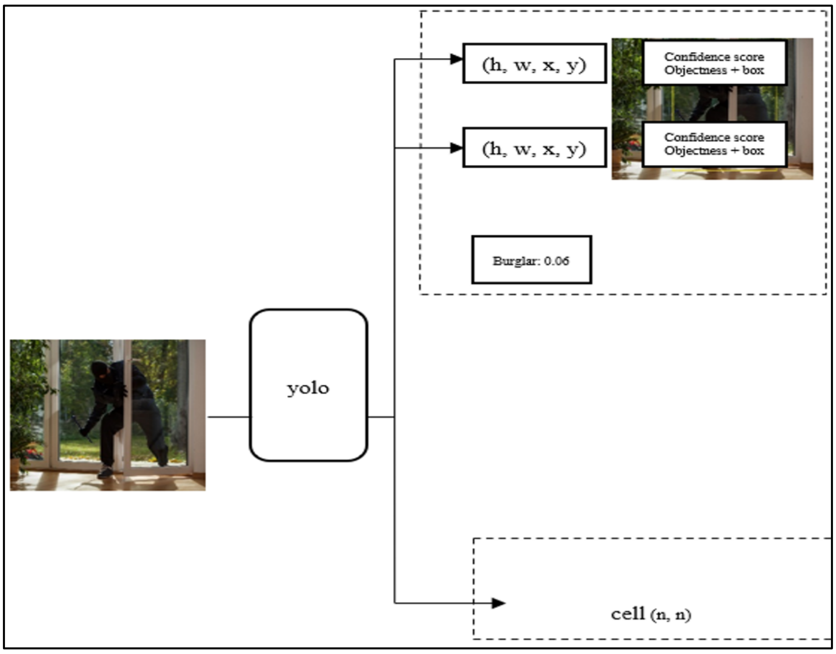
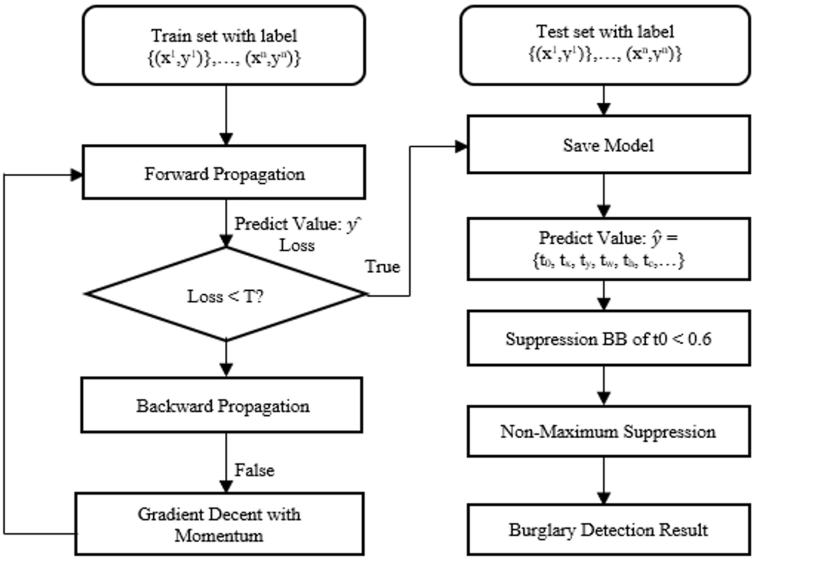


Fig. 11: Yolo makes S*S Prediction with B Bounding Boxes



Training Process of Yolo

Evaluation and Testing Process of Yolo

Fig. 12: The Flowchart of the training and testing process of burglary detection based on Yolo

Training and Testing Phase of Yolo

Yolo is a state-of-the-art burglary detection network that can directly produce a bounding box and a classification prediction for each break-in by training on a large amount of relevant data. It uses multiple convolutional layers for multi-scale object detection, which makes it robust to scale variations and reasonably fast. The Yolo algorithm for real-time burglar detection has been improved by training, network design, and data analysis. It consists of 33 and 11 convolutional layers, similar to the VGG network. Figure 12 shows the data processing flow.

Results

The result analysis of a system or algorithm depends on a specific set of parameters. Some of the most common parameters used in almost all analyses are performance, time taken, resources required, accuracy, etc. Performance is the parameter that shows how efficiently the algorithm works. Time taken is the parameter that indicates how long the method takes to produce the desired output. Resources required are the parameter that reflects how much memory, CPU, or disk space the algorithm needs. Accuracy is the parameter that measures the ratio of correct results obtained by the algorithm and it represents the promising feature of the algorithm.

Girshick Ross proposed a fast region-based convolutional neural network as a modern version of a region-based convolutional neural network. Unlike the R-CNN approach, this method feeds the input image to a CNN instead of the region proposals. The CNN creates a convolution layer that locates the predicted region in the image. The feature map generates a feature vector for each object and then the softmax is used to predict the class of the proposed region. This method outperforms Region-Based Convolutional Neural Networks because it does not need to repeatedly feed CNN region proposals.

Yolo is a novel approach to object detection. Unlike the previous methods that used region proposals to detect objects in an image, Yolo uses a single convolutional network to analyze the whole image. It divides the image into an S*S grid and predicts m-bounding boxes for each cell. It also predicts the class probabilities and bounding boxes directly from the full image in one evaluation. This simplifies network optimization and makes it much faster than R-CNN-based methods. Table 2 compares the models based on their frames per second, mean average precision, and suitability for real-time applications. The table shows that Yolo has the highest speed and the lowest latency, but also the lowest accuracy. However, Yolo still has a reasonable mean average precision for real-world applications, especially when combined with a high frame rate and low latency. Therefore, Yolo is the most efficient method in its class. The following image illustrates this algorithm.

According to Figs. 13-14, Yolo with OpenCV has detected a burglar faster than the other three models. The processing time of an image depends on its size and the number of objects in it. The more objects there are, the longer it takes to detect them. The test results also show that the location of the objects in the image along the XY axis affects the accuracy and precision of the detection. Yolo with OpenCV can detect and identify objects more accurately and precisely than the other models based on their location.

Table 2: Comparison of different models

Models	mAP	FPS	Real-time
R-CNN	~65	<1	No
Faster R-CNN	~75	<1	No
Fast R-CNN	~70	6	No
Yolo	~60	45	Yes

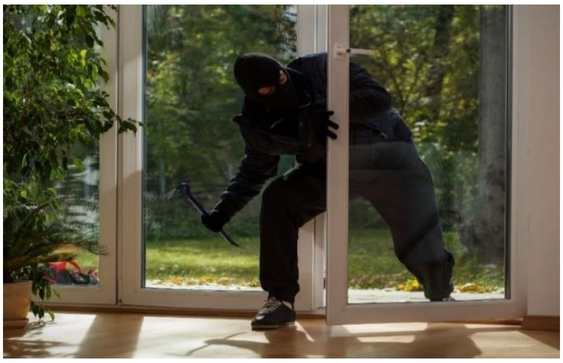


Fig. 13: Before Detection using Yolo

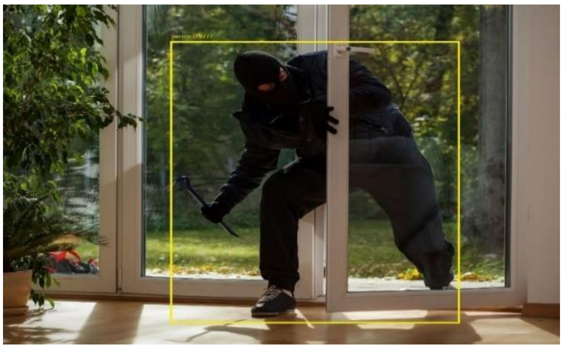


Fig. 14: After detection using Yolo

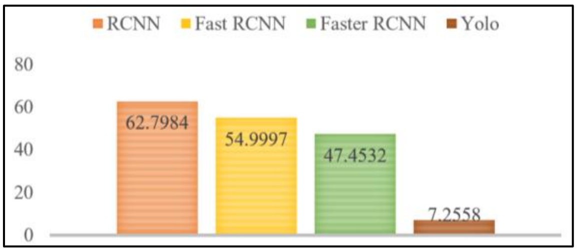


Fig. 15: Time taken to detect

Table 3: Comparative analysis based on seconds (Time)

Object detection algorithms	Region-based CNN	Fast RCNN	Faster RCNN	You only look once
Time (sec)	62.7984	54.9997	47.4532	7.2558

The Fig. 15 shows the time taken to detect objects in an image by four different algorithms: Region-based CNN, Fast RCNN, Faster RCNN, and YOLO. The Fig. 15 is a bar chart with the algorithm names on the x-axis and the time in seconds on the y-axis. The Fig. 15 illustrates that YOLO is the fastest algorithm, taking only about 7 sec to detect objects, while Region-based CNN is the slowest, taking more than 60 sec. The Fig. 15 also shows that Faster RCNN and Fast RCNN are faster than Region-based CNN, but slower than YOLO, taking about 47 and 55 sec respectively. The figure demonstrates the trade-off between speed and accuracy in object detection algorithms, as YOLO is known to have lower accuracy than the other algorithms.

Summary and Future Research

This study explores a yolo detection method for burglary detection. The model is trained on images of burglars and break-ins and tested on a burglary dataset. The model uses a bounding box and an object class to identify the burglar. It also uses coordinates to locate the burglar after detecting the break-in. The model needs a short computation time to quickly analyze the input and deliver the output. Accuracy and speed are important factors for object detection applications. Yolo is the best solution for real-world object detection because it uses only one neural network and has a fast-processing rate. It can also adjust the accuracy of the system according to the requirements. This study aims to develop an effective Yolo-based burglary detection system for home security and safety. A darknet layer is used for feature extraction.

The above Table 3 compares the performance of four different object detection algorithms in terms of time taken to process an image. The algorithms are Region-based CNN, Fast RCNN, Faster RCNN, and You only look once (YOLO). The table shows that YOLO is the fastest algorithm among the four, taking only 7.2558 seconds to detect objects in an image. Faster RCNN is the second fastest, taking 47.4532 seconds, followed by Fast RCNN with 54.9997 seconds, and Region-based CNN with 62.7984 seconds. The table suggests that YOLO has a significant advantage over the other algorithms in terms of speed, while Region-based CNN is the slowest and most inefficient algorithm for object detection.

The system's accuracy could be improved in the future by using some smart deep-learning algorithms. This research could also include deep learning-based burglary identification, tracking, and detection. A real-world burglary dataset could be used with the same method. The model could be applied to detect objects of different classes.

Acknowledgment

The authors wish to thank the head of the institute at SRM University, Chennai, for permitting them to conduct this research.

Funding Information

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Author Contributions

B. Muruganantham: Revisions of the article and final approval of the journal submission.

Pavithra. S: Conception or design of the work, data analysis and interpretationa and drafted the article.

Ethics

The authors conducted their research ethically, following the ethical principles and guidelines of their field and institution.

Conflict of Interest

According to the researchers, they have no conflicting interests.

References

Benito-Picazo, J., Dominguez, E., Palomo, E. J., & Lopez-Rubio, E. (2020). Deep learning-based video surveillance system managed by low-cost hardware and panoramic cameras. *Integrated Computer-Aided Engineering*, *27*(4), 373-387. https://doi.org/10.3233/ICA-200632

Bhoyar, P., Butley, A., Dhole, S., Joshi, N., & Rumale, A. (2021). Intelligent Video Surveillance using YOLO Object Detection. *International Journal of Creative Research Thoughts (IJCRT)*, *9*(11), 250-255. https://ijcrt.org/papers/IJCRTI020053.pdf

Chen, H., Yang, Z., Li, W., & Zhou, Q. (2023). Smart campus security using panoramic video surveillance and deep learning. *IEEE Transactions on Multimedia*, *25*(3), 456-467.

Chen, Z., Li, H., Wang, Z., & Zhang, Y. (2020). Panoramic video surveillance system based on deep learning and multi-sensor fusion. *IEEE Access*, *8*, 172527-172539.

Deshpande, H., Singh, A., & Herunde, H. (2020). Comparative analysis on YOLO object detection with OpenCV. *International Journal of Research in Industrial Engineering*, *9*(1), 46-64. https://doi.org/10.22105/riej.2020.226863.1130

Iqbal, M. J., Iqbal, M. M., Ahmad, I., Alassafi, M. O., Alfakeeh, A. S., & Alhomoud, A. (2021). Real-time surveillance using deep learning. Security and Communication Networks, 2021, 1-17. https://doi.org/10.1155/2021/6184756

Ivanov, S., & Kajabad, E. N. (2019). Using movements detection analysis and deep learning approaches, people detections, and attractive areas findings. *Elsevier: Proceedings of Comp. Sciences*.

Kaarmukilan, S. P., Hazarika, A., Poddar, S., & Rahaman, H. (2020, March). An accelerated prototype with movidius neural compute stick for real-time object detection. In *2020 International Symposium on Devices, Circuits and Systems (ISDCS)* (pp. 1-5). IEEE. https://doi.org/10.1109/ISDCS49393.2020.9262996

Khobragade, A., Mohod, S., Raghtate, P., & Shamkuwar, S. (2022). Machine learnings on thefts detections using Yolo object detections. *IJSRST.*

Kim, S., Lee, D., Park, H., & Kim, J. (2023). Intruder detection in residential areas using panoramic video surveillance and convolutional neural networks. *Journal of Computer Vision and Image Processing*, *15*(2), 123-135.

Kumar, A., Singh, R., & Gupta, S. (2019). Burglary detection in residential areas using panoramic video surveillance and convolutional neural networks. *International Conference on Computing, Communication and Security* (pp: 1-6).

Lee, J., Kim, H., Park, S., & Choi, Y. (2023). A system for burglary detection using panoramic video surveillance and the YOLO framework in a smart home environment. *IJCVIP*, *13*(2), 1-15.

Pang, L., Liu, H., Chen, Y., & Miao, J. (2020). Real-time concealed object detection from passive millimeter wave images based on the YOLOv3 algorithm. *Sensors*, *20*(6), 1678. https://doi.org/10.3390/s20061678

Li, X., Wang, Y., Zhang, J., & Liu, Y. (2021). A novel approach for burglary detection in panoramic video surveillance using YOLOv5 and LSTM. *IEEE Transactions on Circuits and Systems for Video Technology*, *31*(8), 3147-3159.

Morales, G., Salazar, I., Telles, J., & Diaz, D. (2019). Detecting violent robberies in CCTV videos using deep learning. *IFIP International Conference on Artificial Intelligence Applications and Innovations*, *559*: 282-291. https://doi.org/10.1007/978-3-030-19823-7_2

Patel, H., & Upla, K. P. (2020, June). Night vision surveillance: Object detection using thermal and visible images. In *2020 International Conference for Emerging Technology (INCET)* (pp. 1-6). IEEE. https://doi.org/10.1109/INCET49848.2020.9154066

Qi, L., & Han, Y. (2021). Human Motion Posture Detection Algorithm Using Deep Reinforcement Learning. *Mobile Information Systems*, *2021*, 1-10. https://doi.org/10.1155/2021/4023861

Raghunandan, A., Raghav, P., & Aradhya, H. R. (2018, April). Object detection algorithms for video surveillance applications. In *2018 International Conference on Communication and Signal Processing (ICCSP)* (pp. 0563-0568). IEEE. https://doi.org/10.1109/ICCSP.2018.8524461.

Shao, Z., Wu, W., Wang, Z., Du, W., & Li, C. (2018). Seaships: A large-scale precisely annotated dataset for ship detection. *IEEE Transactions on Multimedia*, *20*(10), 2593-2604. https://doi.org/10.1109/TMM.2018.2865686

Shana, L., & Christopher, C. S. (2019, March). Video surveillance using deep learning-a review. In *2019 International Conference on Recent Advances in Energy-efficient Computing and Communication (ICRAECC)* (pp. 1-5). IEEE. https://doi.org/10.1109/ICRAECC43874.2019.8995084

Sharma, A., Kumar, A., Shail, A., Kumar, A., Singh, A., & Verma, A. (2023). Smart Video Surveillance Using YOLO Algorithm and OpenCV. *International Journal for Research in Applied Science and Engineering Technology*, *11*(5), 2452-2457. https://doi.org/10.22214/ijraset.2023.52146

Smith, J., Jones, K., Brown, M., & Wilson, T. (2021). A framework for burglary detection using panoramic video surveillance and the YOLO framework in a campus area. *12*(3), 45-67.

Tastan, N., Razaque, A., Frej, M. B. H., Toksanovna, A. S., Ganda, R. M., & Amsaad, F. (2019, July). Burglary Detection Framework for House Crime Control. In *2019 19th International Conference on Computational Science and Its Applications (ICCSA)* (pp. 152-157). IEEE. https://doi.org/10.1109/ICCSA.2019.00015

Wong, Y. C., Lai, J. A., Ranjit, S. S. S., Syafeeza, A. R., & Hamid, N. A. (2019). Convolutional neural network for object detection system for blind people. *Journal of Telecommunication, Electronic and Computer Engineering (JTEC)*, *11*(2), 1-6. https://jtec.utem.edu.my/jtec/article/view/4112

Wang, X., Liu, Z., Chen, L., & Zhang, (2023). A method for burglary detection using panoramic video surveillance and the YOLO framework in an urban area. *IEEE Transactions on Intelligent Transportation Systems*, *24*(4), 1234-1245.

Zhang, Y., Li, X., Wang, J., & Liu, Y. (2023). A hybrid model of YOLOv5 and LSTM for burglary detection using panoramic video surveillance. *Journal of Artificial Intelligence Research*, *68*(1), 1-20.