1.INTRODUCTION

The meaning of surveillance is close watch kept over someone something with recent breakthroughs in AI and IoT capabilities, surveillance systems that can automatically detect items and threats to the public in real time are now more viable than ever. Consider a security camera system that can detect a wide range of on-body weapons and suspicious things. This technique has the potential to turn surveillance cameras into active observers, avoiding mass shootings in schools, stadiums, and malls. We show a prototype application of such systems in our projects, a surveillance truck with an AI-powered threat detector for smart surveillance. The captured images to take place on-site, which reduces the communication overheads and enables quick security measures. We constructed a YOLOV3 model that can consume a stream of photos directly from an off-site camera, classify them, and report the results to the user via a GUI-friendly interface on the camera side.

1.1 SURVEILLANCE

The topic of mobile vision has seen a lot of development in recent years, with aircraft [1],[2], autonomous underwater vehicles [3]-[5], and ground-based guiding vehicles (GVs) [6],[7]. Certainly, ground vehicle guidance research is an important aspect of mobile navigation research. In this topic, two research streams can be identified based on the context in which the devices are used: outdoor and indoor navigation.

The work on vision-guided road following for "Autobans" [8],[9], the Prometheus system [10], and navigation in unstructured areas are all examples of this.

Indoor navigation has gained popularity, and great progress has been made in the last few years. Many systems have been developed since the original systems, such as those proposed by Giralt et al. in 1979 and Moravec in [11].

Some academics recently looked on the issue of using GV for surveillance in both outdoor and indoor settings. Lipton and colleagues developed a multicamera system combining an aerial platform and ground vehicles to monitor activities in crowded outdoor situations via a distribution network of stationary and mobile sensors in [12].

Video surveillance has undoubtedly played a key part in research over the previous decade, with numerous systems proposed [13]-[19]. Video surveillance offers a wide range of applications, from traffic monitoring [20],[21] to human behaviour analysis. Because video surveillance applications frequently cover a large area, several types of cameras, such as fixed cameras, omnidirectional cameras, and pan and tilt cameras, are commonly employed.

To guarantee proper monitoring coverage of the area of interest when using these types of cameras, the number and positioning of the sensors must be determined in advance. There are several circumstances in which selecting on sensor location a priori hinders system performance or greatly increases costs due to heavy sensor utilisation in visual-based surveillance systems.

We're talking about instances where an alarm condition can happen with the same likelihood anywhere in the monitored environment, or situations where movable items must be tracked over large areas. The use of mobile robots outfitted with unique vision sensors for surveillance purposes in certain settings, particularly in the context of interior spaces, can become a significant concern. The integration of a mobile robot into a visual-based surveillance system can provide coverage of all types of surroundings, increase the system's perceiving capabilities, and provide an augmented reality of the observed scene to a remote operator.

We have focused our attention on the subject of indoor surveillance and security in this project, and an autonomous surveillance vehicle (SV) has been created and built to meet these goals. In addition to traditional robotic activities, the SV can perform. When a suspicious occurrence is recognised, the upper level of modules of the system can choose the target to be tracked by a remote operator. Because the current work does not focus on target selection, the reader can assume that the target is manually selected without losing generality. The tracking technique enables the system to keep the interesting objects in the image's centre and, in some situations, to move them about.

The use of video cameras mounted on a moving vehicle enhances the difficulty of detecting and tracking moving items in the scene. In order to employ change detection techniques, the development of appropriate algorithms that can take into account the camera's ego-motion is required. While stereo systems can provide certain advantages, this study focuses on monocular systems, in which frame-by-frame approaches are typically used to estimate the displacement of two consecutive frames due to camera motion [22],[23],[24],[25],[26],[27].

An image compensation is then applied in such a way that static objects exactly overlap the same objects in the previous frame. A frame-by-frame image subtraction is then applied, and the classical object detection techniques can be used as in the case of fixed cameras [28],[29]. However, these techniques demonstrate some limits in presence of outliers (i.e., object moving in the scene) [30] or when few features are extracted in the image sequence and cannot be applied in the case of a camera mounted on a mobile vehicle. The proposed ASV is able to detect moving objects by means of a direct method [31],[32],[33],[34] by computing he affine transformation for the alignment of the two consecutive frames.

The SV must move in such a way that the tracked object appears in the centre of the current image while monitoring mobile objects in the monitored scene. This constraint is required to make it easier for higher-level modules to classify the detected item, comprehend its behaviour, or locate and recognise certain portions.

As a result, surveillance logics with a variety of features supply clients with security alert strategies such as Remote Monitoring and Rpi consumer cameras. Face detection and other low-cost intelligent analytic functions are being integrated. These approaches may be employed for surveillance applications as video analysis becomes more common in massive media servers. Video surveillance can help to keep individuals secure at home while also facilitating and simplifying control of house-entrance and equipment usage functions. Understanding human behavior is a crucial component of such a home monitoring system.

The tracking of mobile objects in the monitored scene requires the SV to move in such a way that the tracked object appears in the center of the current image. This constraint is necessary in order to simplify the work of the higher- level modules in charge to classify the detected object, understand its behavior, or localize and recognize specific parts.

As a result, surveillance logics with a variety of features supply clients with security alert strategies such as Remote Monitoring and Raspberry-pi consumer cameras. Face detection and other low-cost intelligent analytic functions are being integrated. These approaches may be employed for surveillance applications as video analysis becomes more common in massive media servers. Video surveillance can help to keep individuals secure at home while also facilitating and simplifying control of house-entrance and equipment usage functions. Understanding human behaviour is a crucial component of such a home monitoring system.

1.2 DEEP LEARNING

Deep learning is a subset of machine learning that entails using a three-layer neural network. These neural networks try to mimic the behaviour of the human brain, but they fall well short of its ability to learn from enormous volumes of data. While a single layer neural network may produce approximate predictions, additional hidden layers can help to improve and tune for accuracy.

Deep learning drives many artificial intelligence (AI) applications and services that improve automation, performing analytical and physical tasks without human intervention. Deep learning technology lies behind every day products and services (such as digital assistants, voice-enabled TV remotes, and credit card fraud detection) as well as emerging technologies (such as self-driving cars).

Deep learning neural networks, or artificial neural networks, attempts to mimic the human brain through a combination of data inputs, weights, and bias. These elements work together to accurately recognize, classify, and describe objects within the data.

Deep neural networks are made up of numerous layers of interconnected nodes, each of which improves and refines the prediction or categorization. Forward propagation is the process of computations moving through a network. The visible layers of a deep neural network are the input and output layers. The deep learning model ingests the data for processing in the input layer, and the final prediction or classification is performed in the output layer.

Another process called backpropagation uses algorithms, like gradient descent, to calculate errors in predictions and then adjusts the weights and biases of the function by moving backwards through the layers in an effort to train the model. Together, forward propagation and backpropagation allow a neural network to make predictions and correct for any errors accordingly. Over time, the algorithm becomes gradually more accurate.

The above describes the simplest type of deep neural network in the simplest terms. However, deep learning algorithms are incredibly complex, and there are different types of neural networks to address specific problems or datasets.

Convolutional neural networks (CNNs), used primarily in computer vision and image classification applications, can detect features and patterns within an image, enabling tasks, like object detection or recognition. In 2015, a CNN bested a human in an

object recognition challenge for the first time. Recurrent neural network (RNNs) is typically used in natural language and speech recognition applications as it leverages sequential or times series data.

1.3 NEURAL NETWORKS

A neural network is a type of processing device, which can be either an algorithm or actual hardware that was inspired by the design and operation of animal brains and components. Artificial neural networks, often known as neural networks, have a lot to offer the computing industry.

Let us first look at how the human brain functions before moving on to artificial neural networks. The brain is an incredible processor. Its actual workings remain unknown. The most fundamental component of the human brain is a type of cell called a neuron, which does not renew. Because neurons cannot gradually replace themselves, it is considered that they are responsible for our ability to remember, think, and apply prior experiences to our every activity. About 100 billion neurons make up the human brain. Each neuron can communicate with up to 200,000 other neurons, though most connections are between 1,000 and 10,000.

The enormous number of neurons and their complex interconnections give the human mind its ability. Genetic programming and learning are also factors. There are more than a hundred different types of neurons. Individual neurons are difficult to understand. There are several pieces, subsystems, and control mechanisms in them. They transmit data through a variety of electrochemical paths. These neurons and their connections combine to generate a process that is not binary, steady, or synchronous. In short, it is unlike any electronic computer or artificial neural network now accessible.

A neural network is a processing device, which can be either an algorithm or physical hardware that was inspired by the design and function of animal brains and its components. Neural networks, often known as neural networks or neural nets, have a lot to offer the computing industry. The ability of neural networks to learn makes them incredibly versatile and powerful. There is no need to create an algorithm to execute a specific task, and there is no need to comprehend the internal mechanism of that task for neural networks.

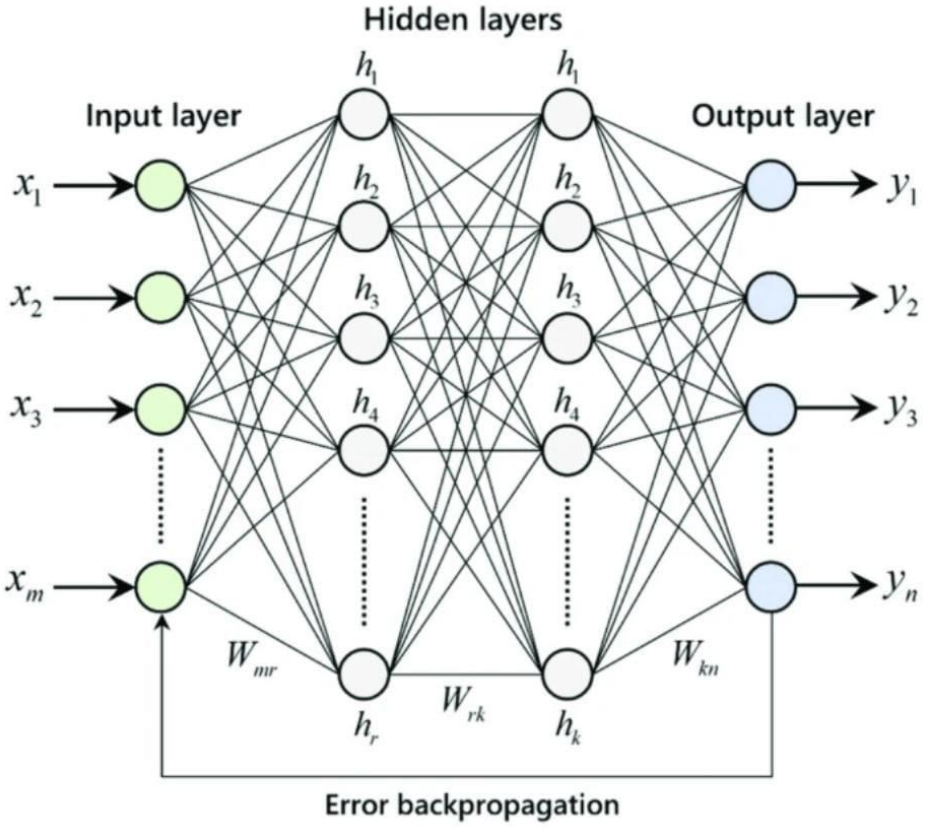


Figure 1.3.1: Artificial neural network.

1.4 TRANSFER LEARNING

Transfer learning is a machine learning method in which a model created for one job is utilised as the foundation for another task's model. Due to the enormous compute and time resources required to develop neural network models on these problems and the enormous jumps in skill that they provide on related problems, it is a popular deep learning approach in which pre-trained models are used as the starting point on computer vision and natural language processing tasks.

As the model has already been pre-trained, a fairly good model can be generated with little amount of training of data using transfer learning. Natural language processing is one transfer learning's application where giant labelled datasets require lot of experts knowledge.

There are 2 mostly used methodologies for adapting to transfer modelling:

1. Develop Model Approach
2. Pre-trained Model Approach

Develop Model Approach:

1. **Selection of Source function -** You must select a fairly predictive modelling problem with a big amount of data and some relationship between the input data, output data, and/or concepts learned throughout the mapping from input to output data.
2. **Developing of Source Model -** For the first challenge, you must then develop a skill-full model. The model must be better than a naive model to ensure that some feature learning has happened.
3. **Model Reuse - -** After that, the model fit on the source job can be utilised to create a model for the second task of interest. Depending on the modelling technique, this could mean using the entire model or just parts of it.
4. **Model Tuning -** Based on the input-output pair data available for the task at hand, the model may need to be updated or refined.

Pre-trained Model Approach:

1. **Selection of Model as Source -** From the available models, a pre-trained source model is chosen. Many research organisations publish models based on large, complicated datasets, which might be included in the pool of potential models to choose from.
2. **Model Reuse -** After then, the pre-trained model can be utilised to create a model for the second task. Depending on the modelling technique, this could mean using the entire model or just parts of it.
3. **Model Tuning -** Depending on the input-output pair data available for the task at hand, the model may need to be changed or rarefied.

Transfer learning, on the other side, is only effective if the features learned in the first task are general enough to be applied to another activity. Furthermore, the input to the model must be the same size as when it was first trained. If not, you have to add a step to resize it.

2. LITERATURE SURVEY

To the best of our knowledge, existing video surveillance solutions from academia and industry dump collected footage to the cloud for additional processing. This makes it impossible to do first video analysis on-site, resulting in higher communication costs and a slowdown in security activities. [35] This section examines existing video surveillance research from many angles, including human-weapon activity recognition [36],[37], criminal detection [38], traffic monitoring [39], monitoring indoor surveillance video [40], moving object tracking [41], and face identification [42],[43].

Lim et al. [36] compiled a dataset of 5500 guns images in various scenarios taken from 250 recorded closed circuit television systems (CCTV) recordings. The authors used the acquired data to train a multi-level object detector with a single stage. When compared to previous datasets, the collected data set improves precision accuracy by 18%, according to pyramid network experimental results. Both the training and validation phases are completed offline on a cloud-hosted centralised server with two NVIDIA Quadro P5000 GPUs and a total video memory of 32GB.

Another crime detection system based on CCTV images was proposed in [38] using deep learning models when the system detects a potential crime, it sends an SMS message to the human supervisor to take the necessary actions. The system can only detect two types of weapons, namely manual gun and knife, using the pre trained VGNET19 model however the model suffers from the poor prediction accuracy due to the limited dataset used to train the model.

GREGA et al [37] proposed an algorithm for automatic firearms detection using recorded CCTV image analysis and situation recognition the algorithm is designed to raise a flag when a firearm is detected the authors used 1000 positive examples and 3500 negative examples to train a CNN model the classification output of the CNN model the classification output of the CNN model is analyzed by a MPEG-7 classifier to determine whether the target object is a firearm. However, the collected imagery dataset suffered from poor quality and low resolution which degraded the firearms detection accuracy.

In the field of traffic monitoring, Zhang et al [39] proposed a vehicle detection and annotation algorithm based on a fine-tuned CNN model. The algorithm can identify vehicle positionsandextractvehiclepropertiesonhighwaysfromrecordedtrafficvideosthedetection algorithm can predicts.

G. K. Verma and A. Dhillon's [45] model for handheld gun detection employs Faster R-CNN. Deep learning technique for detecting different handguns held by people at various orientations and angles. This faster R-CNN employs a region proposal network, the outputs of which are fully connected neural networks subjected to regression and classification.

Even the most hyped and popular self-driving car technology uses object detection techniques to guide the vehicle's movements and motion in order to avoid accidents and collisions. [44], 3D object detection for autonomous driving ensures collision and damage-free efficient transportation.

The introduction of new object detection systems that outperform previously existing models with significantly more variations in dynamics and features has resulted in a new intervention in the deep learning domain. [48] "You Only Look Once: Unified, Real-Time Object Detection," by J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, employs a single neural network for object detection. When compared to previous models such as Faster R-CNN, SSD, Fast R-CNN, and so on, it exceeded expectations in terms of interference and mean average precision.

A newer version of YOLO with impressive performance parameters in terms of accuracy, precision, F1 score, and recall when tested on the COCO dataset. [47] In their paper "Yolov3: An incremental improvement," J. Redmon and A. Farhadi used Convolutional Neural Networks to detect multiple objects on a single image.

3. METHODOLGY

Object detection is basically the pinnacle feature of Artificial Intelligence. Object detection is a technique that uses Computer Vision and Image Processing to classify and locate specific objects in images or videos. In order to detect objects, machine learning or deep learning algorithms are used. [44] For an instance Self Driving Cars use object detection technology where Computer Vision and Image Processing are used to determine the distance or area between the car and mobile object to alert and guide the car. The fundamental principle in object detection is artificial intelligence. Data is collected using computer vision and fed into models that use Convolution Neural Networks (CNN), VGG or Residual Networks, and models are trained to detect objects in images and videos using Machine Learning or Deep Learning algorithms.

Now, the results are presented to the system in the form of alerts or instructions based on the model used and how the data was trained.

- How far is the object?
- Is the object in motion or not?
- Is the object stopped or not?

3.1 OBJECT DETECTION-YOLO V3 (You Only Look Once Version 3)

YOLO stands as an abbreviation for “You Only Look Once” was designed by Joseph Redmon and Ali Farhadi in the year 2016. Currently there are 6 versions of YOLO available for object detection. Versions (1-3) are released in years 2016,2017,2018 respectively. Other versions were released in 2020. [46] YOLO uses CNN for detection of objects in images or videos (both live and recorded). YOLO is designed for the detection of multiple objects from a single image. In order to that it can also predict the exact location of objects in the images. It has the most unique ability in detection of multiple classes (i.e., person, dogs, cats, watches etc) based on the datasets used during training. YOLO uses single and distinctive neural network, the neural network divides images into grids and generates probabilities for each grid. Following that, YOLO predicts the required number of bounding boxes and selects the best one based on the generated probabilities. Using the COCO dataset, it can detect 80 classes.

3.2 ARCHITECTURE OF YOLO V3

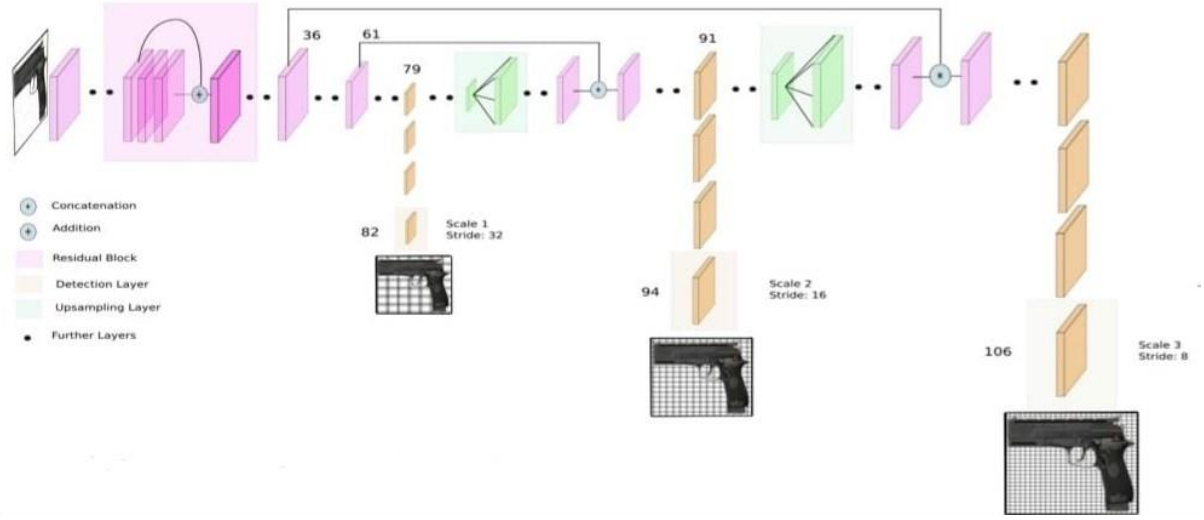


Figure 3.2.1: Architecture of YOLO v3 network

In order to completely understand YOLO V3 architecture from figure 3.2.1 the following terminologies play a crucial role. They are as follows

1. CNNs (Convolutional Neural Networks)
2. Residual Blocks
3. Skip Connections
4. Activation Function
5. IoU (Intersection Over Union)
6. Non-maximum suppression

3.2.1 Convolution Neural Networks (CNNs)

A convolutional neural network (ConvNet/CNN) is a Deep Learning system that can take an input image and assign relevance (learnable weights and biases) to different aspects/objects in the image, as well as distinguish between them. ConvNet requires significantly less pre-processing than other classification algorithms. While traditional approaches necessitate the hand-engineering of filters, ConvNets can learn these filters/characteristics with sufficient training.

A convolutional neural network (ConvNet/CNN) is a Deep Learning system that can take an input image and assign relevance (learnable weights and biases) to different aspects/objects in the image, as well as distinguish between them. ConvNet requires significantly less pre-processing than other classification algorithms. While traditional

approaches necessitate the hand-engineering of filters, ConvNets can learn these filters/characteristics with sufficient training.

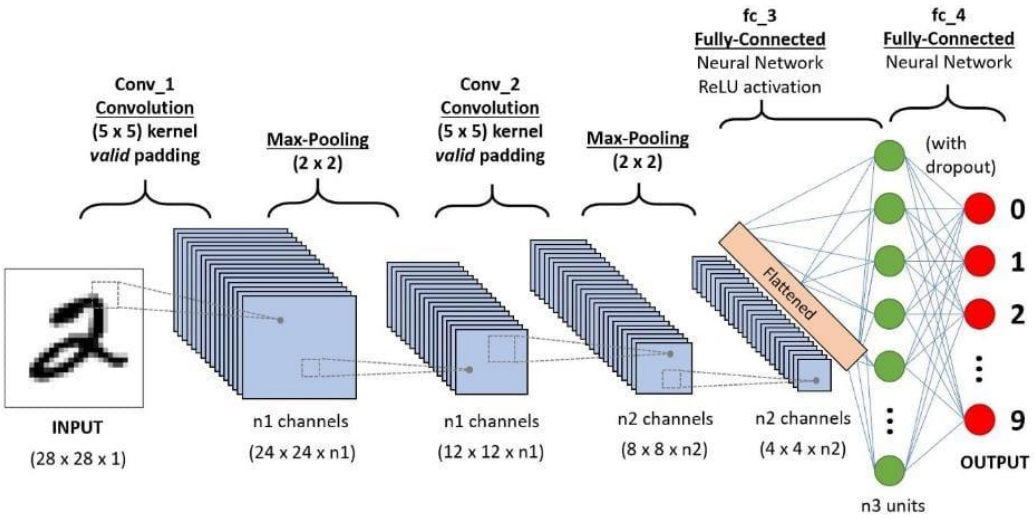


Figure 3.2.2: CNN sequence for classifying handwritten digits

When there is an exceedingly basic binary image, there can be an average precision score while predicting classes but would have only very minute to zero accuracy when it comes to complex images having pixel dependencies throughout.

A ConvNet is capable of successfully capturing the spatial as well as temporal dependencies in an image through application of relevant filters. Since there is a good reduction in the parameters and reusability of weights this architecture is apt for images or image datasets.

3.2.2 Residual Blocks and Skip Connections

Following AlexNet, the first CNN-based architecture to win the ImageNet 2012 competition, each subsequent winning architecture employs more layers in a deep neural network to reduce error rates. This works with a small number of layers, but as the number of layers increases, a common problem in deep learning called Vanishing/Exploding gradient occurs. As a result, the gradient becomes zero or excessively large. The training and testing error rate increases as the number of layers increases.

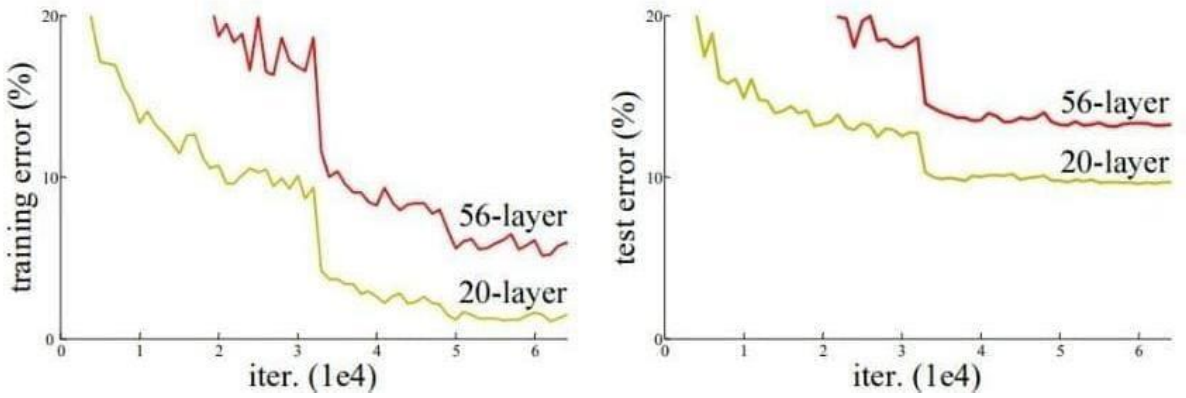


Figure 3.2.3: Comparison of 20-layer vs 56-layer architecture

Figure 3.2.3 shows that a fifty-six-layer CNN model has a higher error rate on both training and testing of the dataset than a twenty-layer CNN model. The training error in fifty-six-layer CNN should be reduced now, but it still has a high training error due to vanishing/exploding gradients.

Residual Network architecture was introduced to resolve the conflict of vanishing/exploding gradient. The skip connection technique is used to skip or omit training from a few layers and interact with the output directly. Instead of layers, this network learns the underneath mapping and fits into the residual mapping. The residual network for a simple function is shown in Figure 3.2.4.

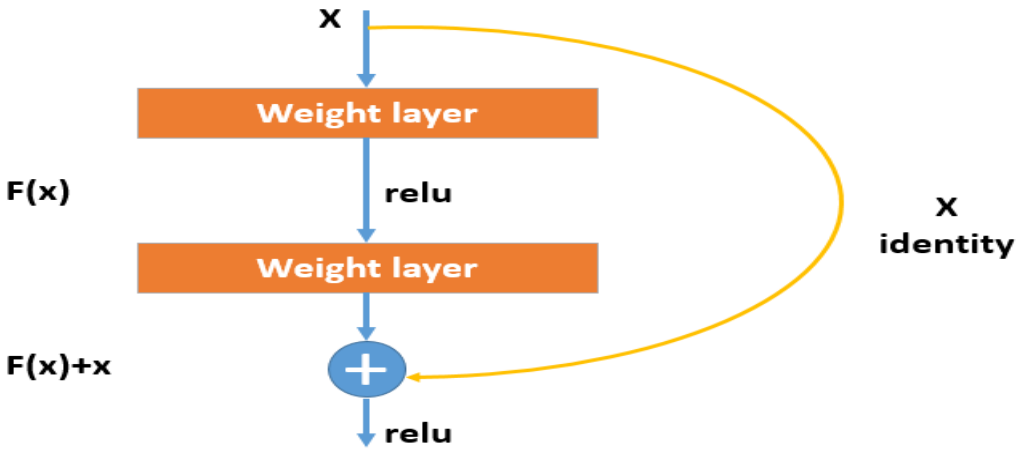


Figure 3.2.4: ResNet using skip connections

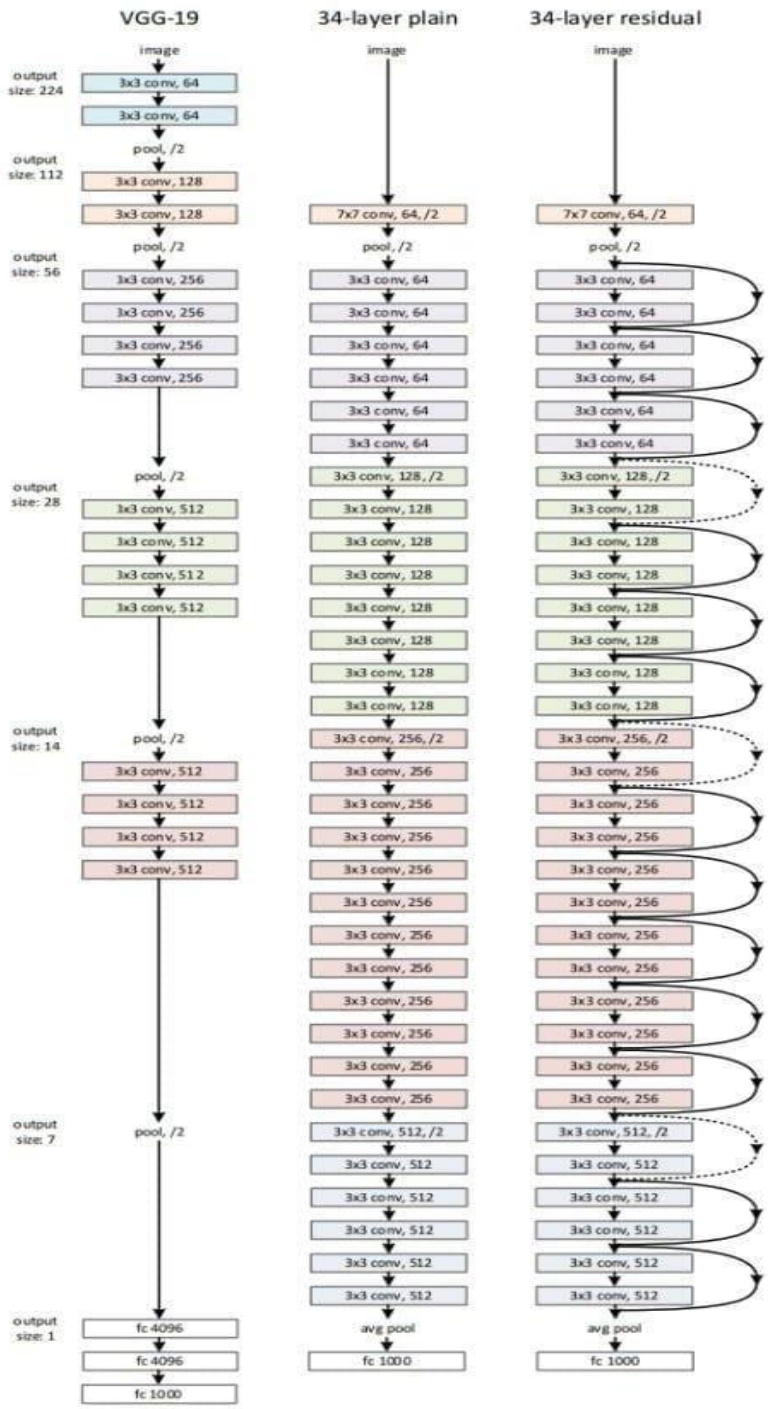


Figure 3.2.5: ResNet Architecture

According to figure 3.2.5, the ResNet employs a thirty-four layer simple network architecture based on VGG-19, with skip connections added. The architecture is then converted into a residual network as a result of these skip connections.

3.2.3 Activation Function

In a neural network, an activation function specifies how the weighted sum of the input is converted into an output from a node or nodes in a layer. Many activation functions are nonlinear, and this phenomenon is known as "non-Linearity" in layer or network architecture.

There are three types of layers in a network: input layers that accept raw data, hidden layers that take input from one layer and transfer the output to another, and output layers that produce predictions. In general, all hidden layers have the same activation function. Depending on the type of prediction required by the model, the output layer will frequently use a different activation function than the hidden layers. CNN's hidden layers employ a differentiable nonlinear activation function. As a result, the model may be able to learn more complex functions than a linear activation function-trained network. The rectified linear activation function, or ReLU activation function, shown in figure 3.2.6, is one of the most commonly used functions for hidden layers.

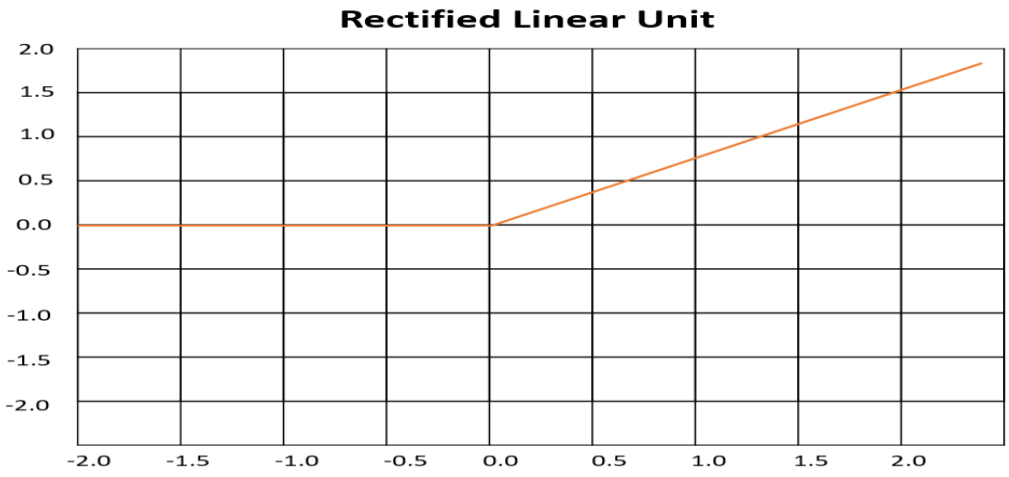


Figure 3.2.6: Rectified Linear Unit

The mathematical representation of the ReLU function is as follows

$$f(x) = \begin{cases} xi \ \ if \ xi > 0 \\ 0 \ if \ xi < 0 \end{cases}$$

where f(x) represents function on Y-axis and x represents values on X-axis

3.2.4 Intersection Over Union (IoU) and Non-Max Suppression

Non-Maximal Suppression (NMS) is a method for determining the best bounding box by rejecting predicted bounding boxes with detection probabilities less than a given NMS threshold and then eliminating all bounding boxes with IoU values greater than a given IoU threshold. Intersection over Union is a metric used to evaluate an object detector's accuracy over a given dataset. It is common to see IoU used as a metric on datasets such as PASCAL VOC. Any algorithm that generates predicted bounding boxes as output can be evaluated using IoU.

To evaluate a (arbitrary) object detector using Intersection over Union, we need the following:

1. The original bounding boxes (i.e., the labelled bounding boxes from the testing set that specify where in the image our object is).
2. The predicted bounding box from our base model.

Mathematically IoU is given by:

$$IoU = \frac{A1 \cap A2}{A1 \cup A2} = \frac{Area\ of\ Overlap(intersection)}{Area\ of\ Union}$$

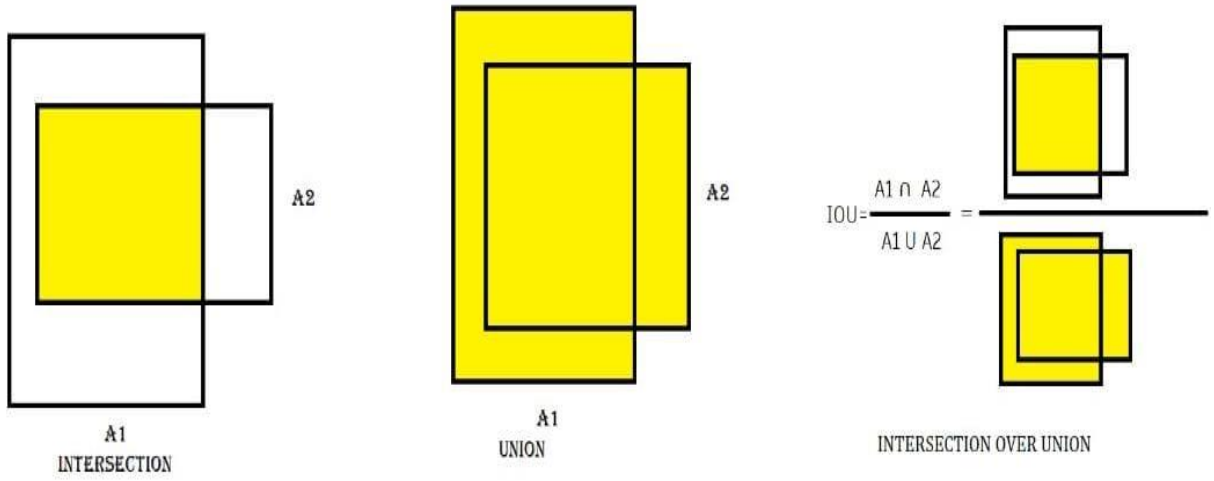


Figure 3.2.7: Intersection Over Union

From the figure 3.3.1 YOLO V3 architecture is clearly evident that it uses CNNs. YOLO V3 is uses Darknet Framework called “DarkNet-53”. YOLO V3 uses 53 convolution neural network layers for detection tasks stacked with additional 53 more layers. So, a total of 106 layers for YOLO V3.

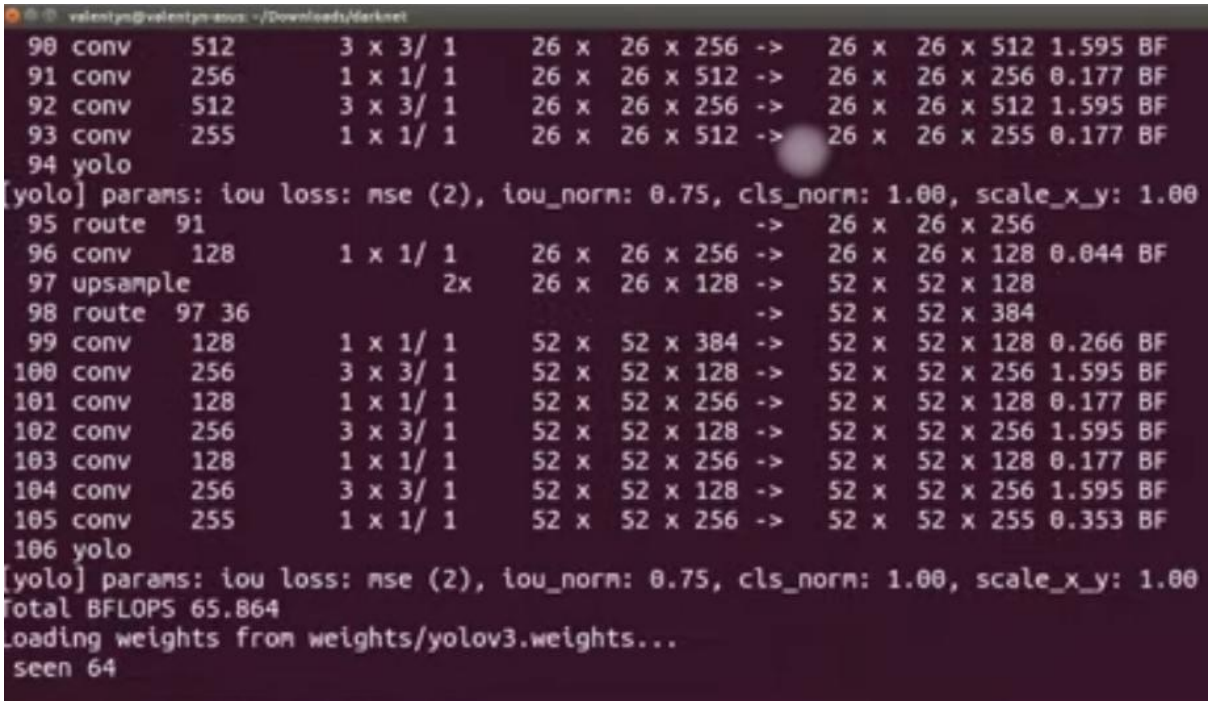


Figure 3.2.8: 106 CNN Layers in YOLO V3

The Darknet framework loads all these 106 layers during the time of object detection. Even though there are 106 CNN layers only 3 layers primarily involve in object detection those layers include CNN layer-82, CNN layer-94, CNN layer-106. During the functioning of these layers the most important elements include

- Residual Blocks
- Skip Connections
- Up-sampling

Convolution Neural Network layers are mostly followed by

- Batch Normalization
- Leaky ReLU Activation Function

YOLO V3 doesn't use pooling layers such as max pooling or average pooling it only uses convolution layers.

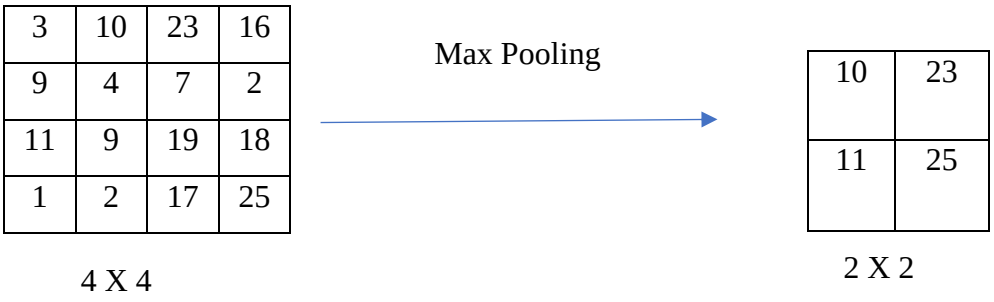


Figure 3.2.9: Max Pooling for the feature maps

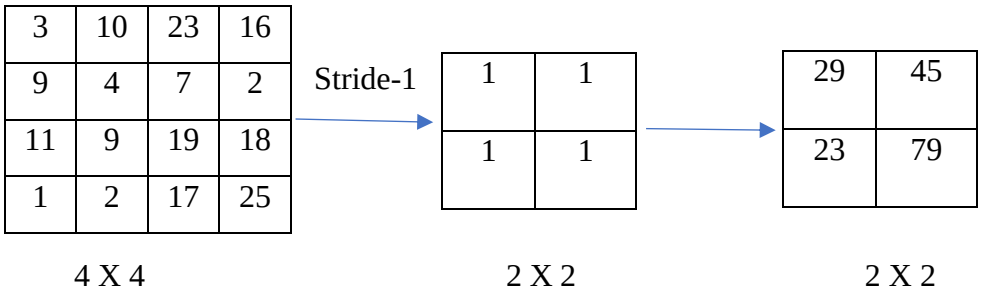


Figure 3.2.10: Convolution for the feature maps

From figures 3.2.9 and figure 3.2.10 it is clear that there is loss of information due to max pooling for the feature map, but there is no or a little loss of information when convolution is applied onto the feature map. So additional CNN layers are used to downsample the feature maps (Stride-1) this is because CNN layers downsample feature maps prevent loss of low-level features that pooling layers exclude. Hence YOLO has the ability to detect smaller objects.

3.3 INPUT TO THE NETWORK

YOLO V3 detects objects on images, video (both live and recorded), so the input to the YOLO V3 network is batches of images. These batch of images are generally of the form (n , Wt , Ht , 3) where n represents the number of images, Wt represents width of image, Ht represents the height of image, 3 represents for number of channels (RGB -Red, Green, Yellow).

Height and Width can be changed but must be divisible by 32. The very Height and Width are sometimes considered to be input to the network. A few combinations for height and

width are (416 X 416), (608 X 608), (832 X 832), (1024 X 1024). With the increment of resolution of input images, model's accuracy after training can be increased. Images can be of any size when fed to the network, there is no need to resize them as they will automatically get resized to the network.

3.4 DETECTION AT 3 SCALES

YOLO V3 detects the network at three different scales and in three different locations. As mentioned earlier CNN layer-82, CNN layer-94, CNN layer-106 are used mainly for object detection these three layers makes object detections at 3 different scales. The YOLO V3 network down-samples the input images by a reduction/stride factors of (32, 16, 8) with respect to convolution layers (82, 94,106). Figure 3.5.1 shows about how the output is present at 3 separate layers in the network.

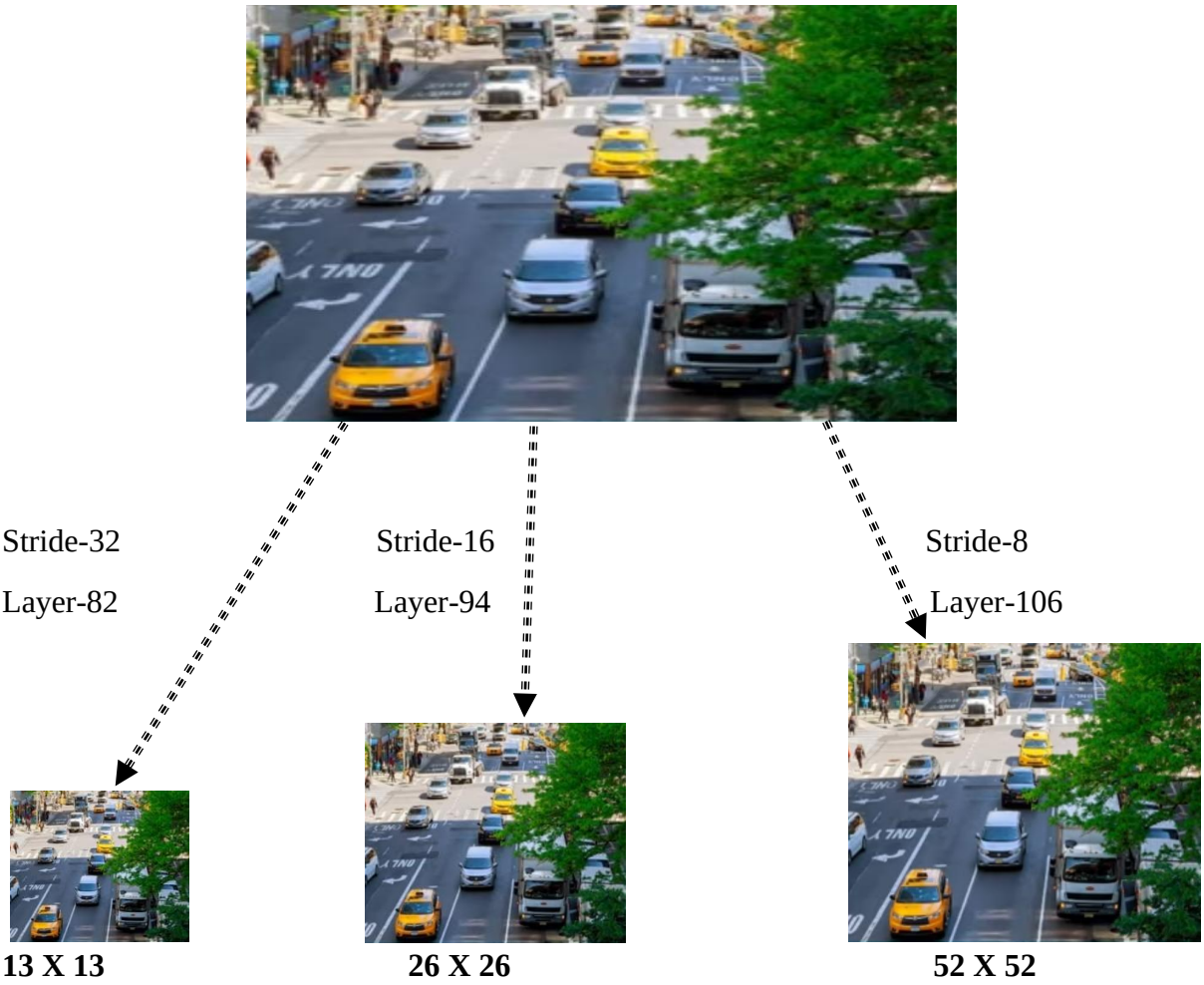


Figure 3.4.1: Downsampling of network image at 3 scales

3.5 DETECTION KERNELS

To generate output, YOLO V3 employs a (1 X 1) detection kernel at each of the network's three distinct layers. (13 X 13), (26 X 26), and apply one-to-one convolution to the downsampled input images (52 X 52). As a result, the resulting network output feature maps will have all of the special dimensions. The shape of detection kernel also has its depth that is calculated by following equation

Depth of detection Kernel $= b * (5 + c)$ pixels

Where b is the number of bounding boxes, c is the number of classes in the dataset.

For an instance let us take COCO dataset which contains 80 classes which uses 3 bounding boxes, so a total 255 attributes are detected. Hence the depth of the kernel is about 255 pixels. Each feature map produced by detection kernels at three different points in the network has one more dimensional depth and includes 255 attributes of COCO dataset bounding boxes. The feature maps have the following shapes: (13,13,255), (26,26,255), and (52,52,255). YOLO V3 has three bounding boxes for all cell of these feature maps that is why the value of b is 3.

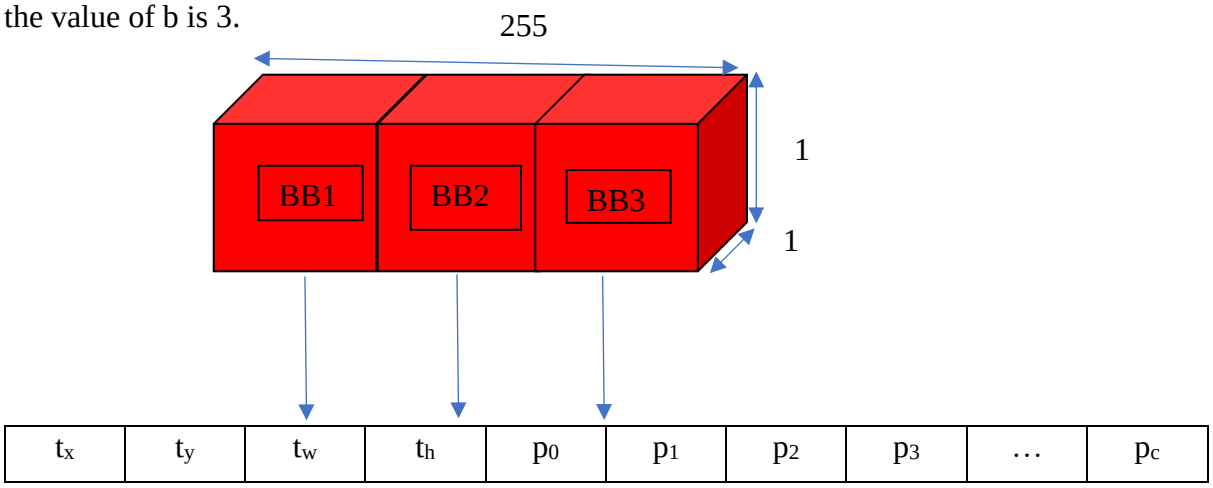


Figure 3.5.1: Kernel and its attributes

here t_x, t_y are the central coordinates of the bounding boxes.

t_w, t_h are the width and height of the predicted bounding boxes.

p_1, p_2, p_3...p_c are class confidences.

p_0, is the objectness score.

3.6 GRID CELLS

YOLO V3 predicts three bounding boxes for each cell in the feature map; each cell predicts an object through one of its bounding boxes if the centre of the object belongs to one of this cell's receptive fields, which identify the cell that falls into the centre of the object. This is one of the detection kernel-generated feature map cells. When training, YOLO V3 has a single ground truth bounding box that is responsible for detecting a single object. As a result, it is critical to determine which cell this bounding box belongs to, and to do so, consider the detection scale, where 32 is the network's stride. The input image (416 X 416) is then downsampled into a grid of cells (13 X 13). When all of the cells to which the ground truth bounding box belongs are identified, YOLO V3 assigns the centre cell to be responsible for predicting this object, and the objectness score for this cell is equal to 1. This is one of the corresponding feature map's cells that is now in charge of detecting objects, but during training, each cell predicts three bounding boxes.

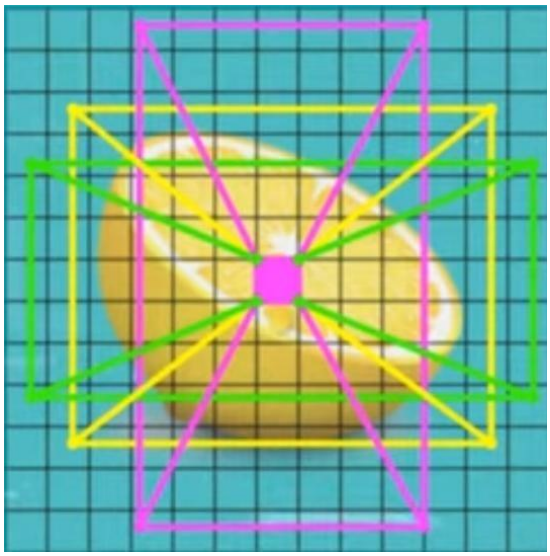


Figure 3.6.1: Centre cell prediction

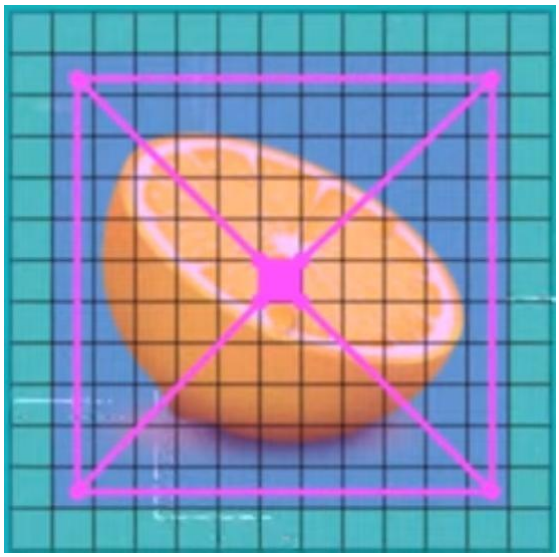


Figure 3.6.2: Centre cell prediction for ground truth bounding box

3.7 ANCHOR BOXES

Which of the three bounding boxes should be used for detection? Which one should be the best predicted object's bounding box? The answer to these questions is that YOLO V3 predicts the bounding boxes using "Anchor boxes." Anchor boxes are predefined bounding boxes, also known as "Priors." Anchor boxes have predefined width and height, which they use to calculate the width and height of required bounding boxes. There are a total of nine

anchor boxes. Each scale has three anchor boxes (82,94,106). It means that at each scale, each grid cell in the feature map can predict three bounding boxes using three anchors. In YOLO V3, the K-mean clustering technique is used to calculate these anchor boxes. The width and height of the 9 anchors in the COCO dataset are as follows.

Scale-1 (CNN Layer-82)	Scale-2 (CNN Layer-94)	Scale-3 (CNN Layer-106)
(116 X 90)	(30 X 61)	(10 X 13)
(156 X 198)	(62 X 45)	(16 X 30)
(373 X 326)	(59 X 119)	(33 X 23)

Tabular Form 3.7.1: Anchor dimensions at 3 different scales

All of these anchors are scaled in three different locations across the network. Assume we have an input image of shape (416 X 416 X 3), and the image passes through the YOLO V3 deep CNN architecture until it reaches the first step, which has the stride-32. This input image is stride down-sampled to the dimension (13 X 13 X 255) feature map produced by detection kernels.

From the figure 3.7.1 below it is clearly observed that the input image of assumed size 416 X 416 (i.e., lemon) is sent to a deep convolution neural network with an estimated stride of value 32 after performing all the operations that are described above the output image has been resized to 13 X 13 and with a kernel depth of 255, portraying 255 attributes of the image.

The three bounding boxes are helpful in detecting the probability of object that is present at each and every cell of the original input image.

It is evident that the attributes generated are helpful in detection of the confidence levels, objectness score, location coordinates of the bounding boxes, height and width of the bounding boxes.

These attributes are present for all the three bounding boxes which are in turn available across each and every cell of the grid.

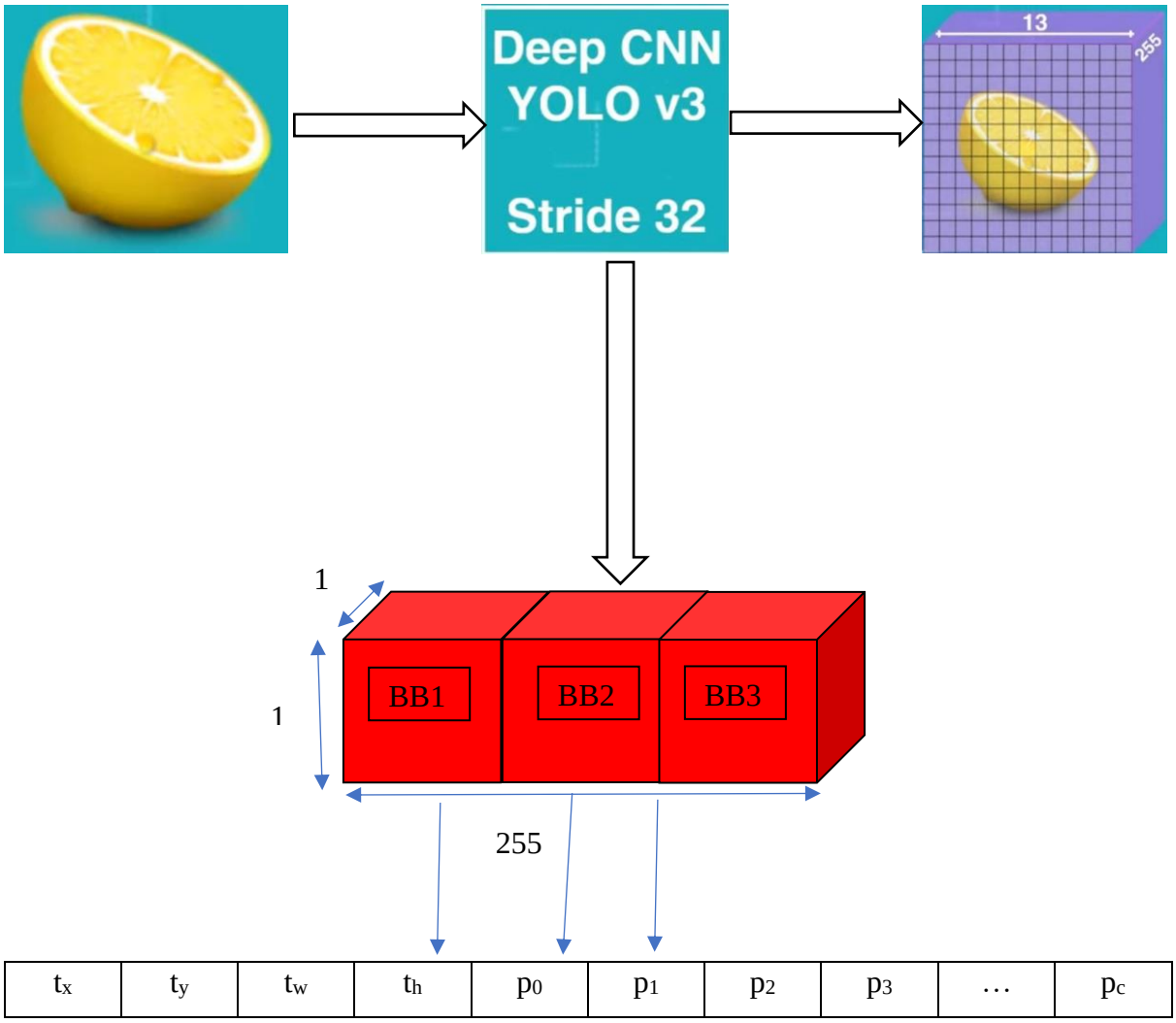


Figure 3.7.1: Attributes detection using anchors

here t_x, t_y are the central coordinates of the bounding boxes.

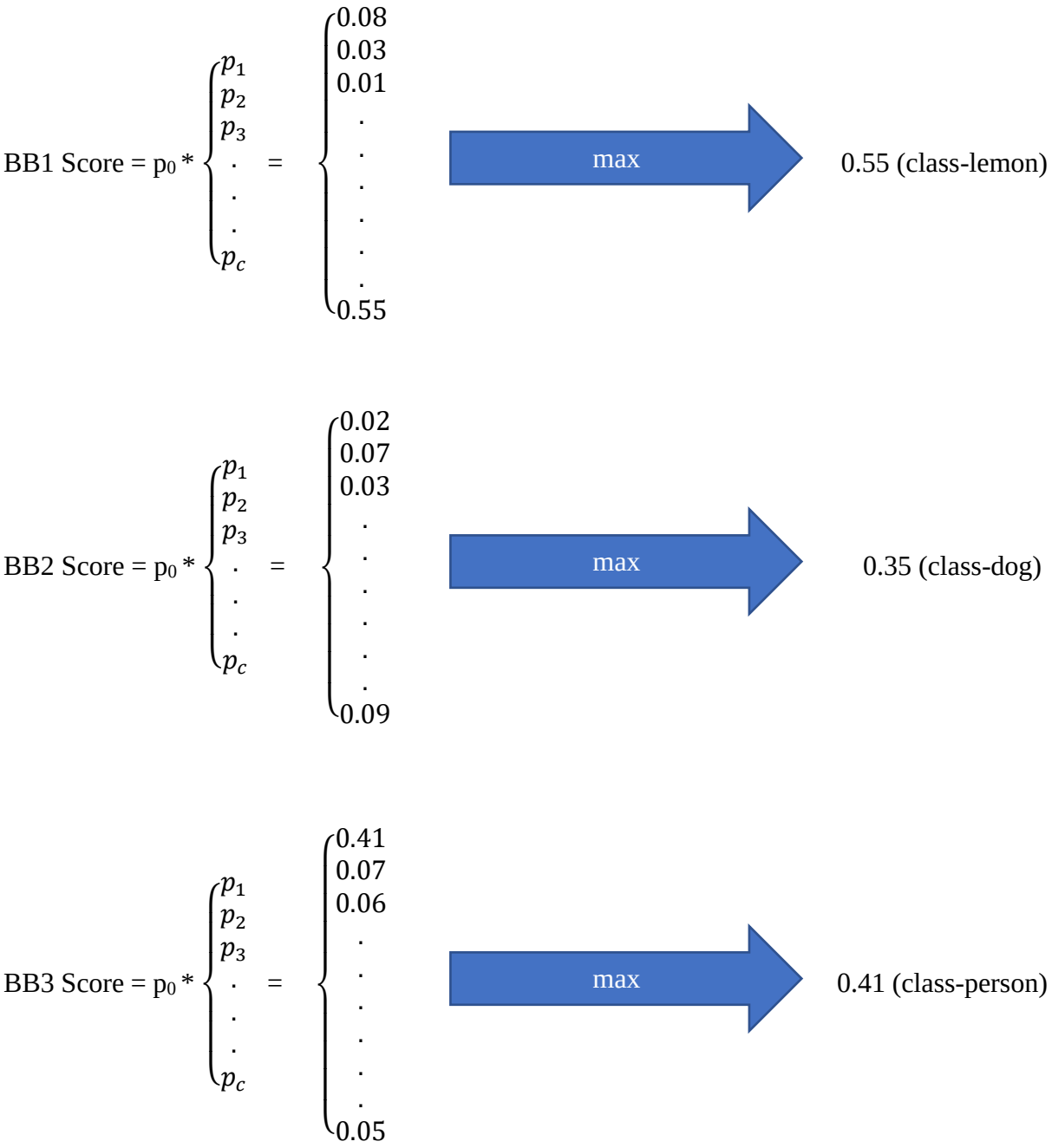
t_w, t_h are the width and height of the predicted bounding boxes.

p_1, p_2, p_3...p_c are class confidences.

p_0, is the objectness score

Since we have used COCO dataset there are 80 classes and 255 attributes. Now we must extract the probabilities among the three predicted bounding boxes of the cell in order to determine whether or not this box contains a specific class.

To do so, we compute the element-wise product of the objectness score and the list of confidences, then find the maximum probabilities and can say with certainty that this box detected a specific class.



These calculations are performed on all (13 X 13) cells in three predicted bounding boxes and 80 classes. As a result, the number of predicted boxes in the network at the first, second, and third scales is 507, 2028, and 8112. YOLO V3 predicts 10647 bounding boxes, of which only the required boxes are filtered using a non-maximum suppression technique.

3.8 CALCULATION OF PREDICTED BOUNDING BOX

Anchors (priors) are clearly defined as bounding boxes with predefined width and height that were calculated using K-means clustering. YOLO V3 calculates offsets to predefined anchors to predict real width and real height. This offset is also referred to as a "log-

space" transform. Sigmoid functions are used to predict the centre coordinates of the bounding boxes.

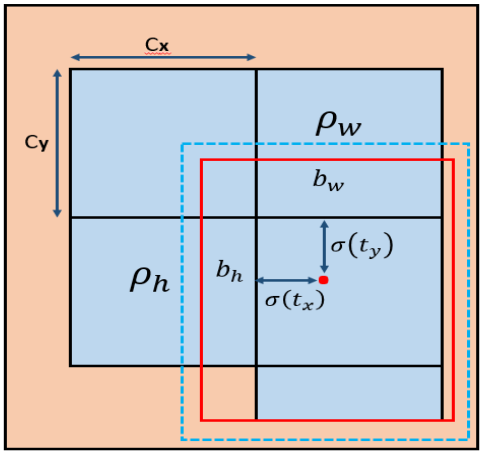


Figure 3.8.1Bounding boxes with dimension priors and prediction of location. The height and width of the box are predicted as offsets from cluster centroids.

Sigmoid function is given by

$\sigma(x) = \frac{1}{1+e^{-x}}$ where e is the Euler's value e = 2.7182818

The equations to predict the bounding boxes are as follows:

$$b_x = \sigma(t_x) + c_x$$

$$b_y = \sigma(t_y) + c_y$$

$$b_w = p_w * e^{-t_w}$$

$$b_h = p_h * e^{-t_h}$$

b_x, b_y, b_w, b_h – Represent centre coordinates, height, width of bounding box

t_x, t_y, t_w, t_h – Represent output of the network after training

c_x, c_y – Represent cells top left corner of the anchor box

p_w, p_h – Represent width and height of anchor boxes

Generally there will be some losses occurring while performing object detection. So, in order to estimate these losses which occur during of dataset we use standard loss estimation functions. The loss estimation functions is given by:

$$\lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{i,j}^{obj} \begin{bmatrix} (t_x - \widehat{t_x})^2 + (t_y - \widehat{t_y} \\ +(t_w - \widehat{t_w})^2 + (t_h - \widehat{t_h})^2 \end{bmatrix}$$

$$+\sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{i,j}^{obj} \left[-\log(\sigma(t_o)) + \sum_{k=1}^{C} BCE\left(\widehat{y_k}, \sigma(s_k)\right) \right]$$

$$+\lambda_{\text{noobj}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{i,j}^{noobj} [-\log(1 - \sigma(t_o))]$$

Assuming Prediction vector: $t_x, t_y, t_w, t_h, t_o, s_1 \ldots s_c$ and corresponding ground truth label: $\hat{t}_x, \hat{t}_y, \hat{t}_w, \hat{t}_h, \hat{t}_o, \hat{y}_0, \hat{y}_{1,}, \hat{y}_c$ where $c =$ total classes. $y \in \{0,1\}$

To improve model stability, Lambda constants are used to differentially weight the loss function. The highest penalty is for predicting coordinates with λ_{coord} =5 and when there is no object present, the penalty is the lowest. i.e. λ_{noobj} = 0.5

Binary Cross Entropy is defined as:

$$BCE(y, \hat{y}) = -\frac{1}{N} \sum_{i=1}^{N} y \cdot \log(\hat{y}_i) + (1 - y) \cdot \log(1 - \hat{y}_i)$$

y is the original value and ŷ is the predicted value. BCE measures the error in the original and predicted values.

We set the following mask to ensure focus training on boxes containing an object:

$$1_{ij}^{obj} = \begin{cases} 1 \text{ if the object exists in the i} - \text{ th cell and} \\ \text{j} - \text{ th box is responsible for detecting it} \\ 0 \text{ otherwise} \end{cases}$$

$$1_{ij}^{noobj} = \begin{cases} 1 \text{ if there is no object in i} - \text{ th cell} \\ 0 \text{ otherwise} \end{cases}$$

3.9 FINAL OUTPUT OF OBJECT DETECTION USING YOLO V3

After going through all process as mentioned earlier finally the objects get detected along with their location and object score by applying suitable bounding boxes. COCO dataset has been used to develop this object detection model (using YOLO V3).

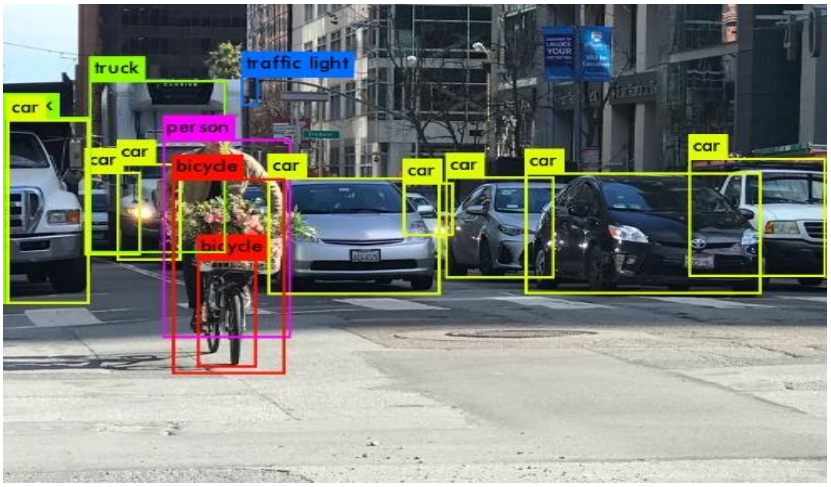


Figure 3.9.1: Object Detection System

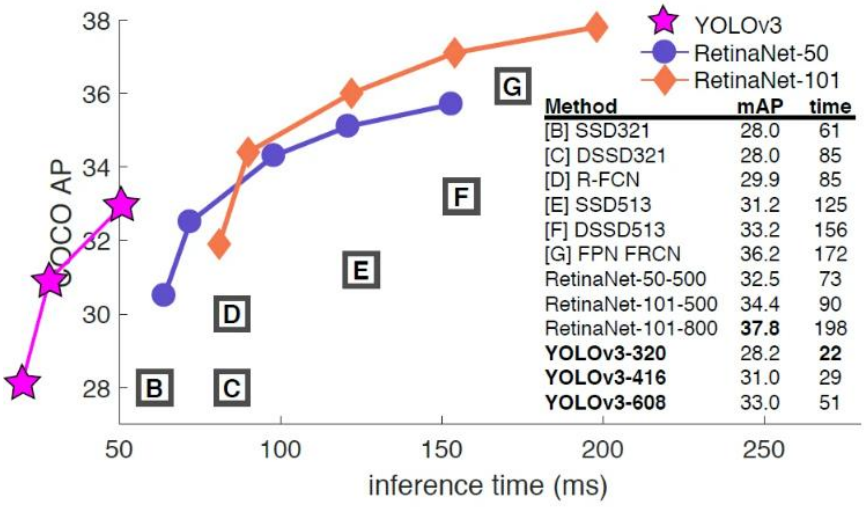


Figure 3.9.2: mAP and time interference of YOLO V3

From figure 3.9.2 we can see mean average precision in YOLO V3 is faster when campared to other object detection models. It takes less time in processing the images with grreater amount of precision.

From figure 3.9.3 and figure 3.9.4 it is clear regarding the dataset and the class that are present in it. COCO dataset provides a vast range of different class which are of 80 in number.

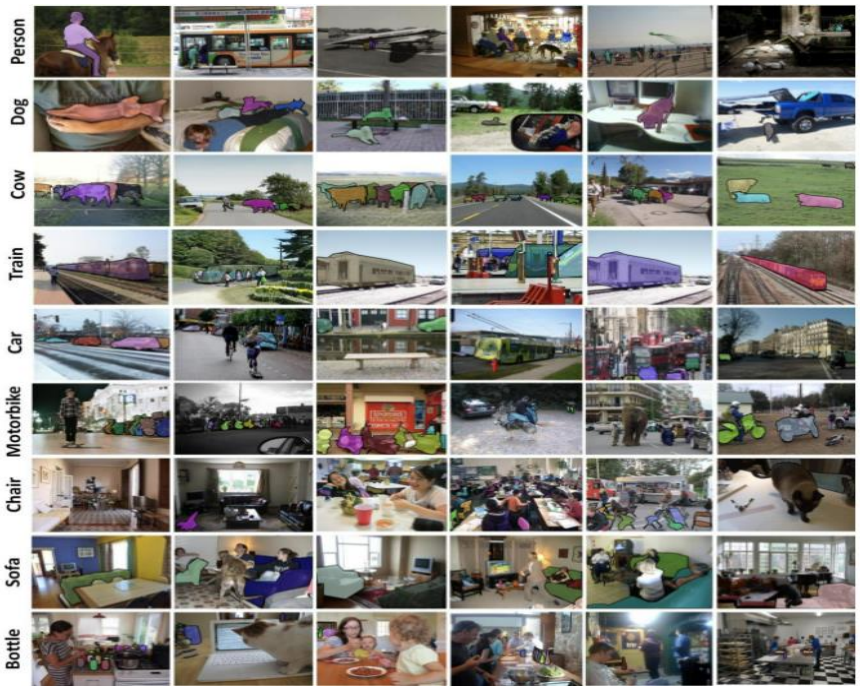


Figure 3.9.3: Few images of different classes from COCO dataset

person	fire hydrant	elephant	skis	wine glass	broccoli	dining table	toaster
bicycle	stop sign	bear	snowboard	cup	carrot	toilet	sink
car	parking meter	zebra	sports ball	fork	hot dog	tv	refrigerator
motorcycle	bench	giraffe	kite	knife	pizza	laptop	book
airplane	bird	backpack	baseball bat	spoon	donut	mouse	clock
bus	cat	umbrella	baseball glove	bowl	cake	remote	vase
train	dog	handbag	skateboard	banana	chair	keyboard	scissors
truck	horse	tie	surfboard	apple	couch	cell phone	teddy bear
boat	sheep	suitcase	tennis racket	sandwich	potted plant	microwave	hair drier
traffic light	cow	frisbee	bottle	orange	bed	oven	toothbrush

Figure 3.9.4: 80 classes of COCO dataset

4. THREAT DETECTION USING YOLO V3

Every year, many people are killed by gun violence. Children who witness high levels of violence in their communities or through the media are more likely to suffer from psychological trauma. Children who are victims, offenders, or spectators of gun violence might experience harmful psychological repercussions in the short and long term. Numerous studies have found that the portable gun is the most commonly used weapon in crimes such as break-ins, bank robberies, shoplifting, and rape. These crimes can be reduced by identifying disruptive behaviour early on and closely monitoring suspicious behaviour so that law enforcement officials can respond quickly.

The amount of gun violence varies dramatically between countries and geographical areas. The worldwide death toll from gun violence could be as high as 1,000 per day. According to statistics, every year in Pakistan, 4.2 out of every 100,000 individuals are killed in mass shootings. Many valuable lives have been lost as a result of street crimes and individual institution attacks. This also suggests that a manual monitoring system still requires the use of a human eye to notice anomalous actions, and that reporting to security personnel takes a long time.

Figure 4.1: Different types of Ammunitions.

Although the human visual framework is quick and precise, and it can also perform complex tasks like distinguishing different items and recognising snags with minimal cognizant thought, it is a common truth that if an individual watches something very similar for a long time, there is a chance of sluggishness and lack of regard.

With the availability of large datasets, faster GPUs, advanced machine learning algorithms, and more accurate calculations, we can now effectively prepare PCs and develop automated computer-based systems to distinguish and identify numerous items on a website with high accuracy. According to recent developments, machine learning and advanced image processing algorithms are playing a dominant role in smart surveillance and security systems. Aside from that, the popularity of smart devices and networked cameras has been used.

Every year, a massive population worldwide reconcile with gun-related violence. A fully automated system for identifying basic armaments, specifically guns and ammunition. Deep learning as well as transfer learning advancements have displayed important progress in object detection and recognition. The trained custom object detection model using YOLO V3 "You Only Look Once" and our dataset.

Figure 4.2: BA 338TP precision-guided bolt action.

Violence due to ammunition has a deep importance on the people's safety, pysical well-being, and psychological well-being. Every year, many people are killed by gun violence.

Children who are uncovered of gun violence, whether as victims, perpetrators, or bystanders, can suffer short and long-term psychological consequences Gun-related violence varies widely among countries and geographical areas. The worldwide death toll from gun violence could reach 1,000 per day.

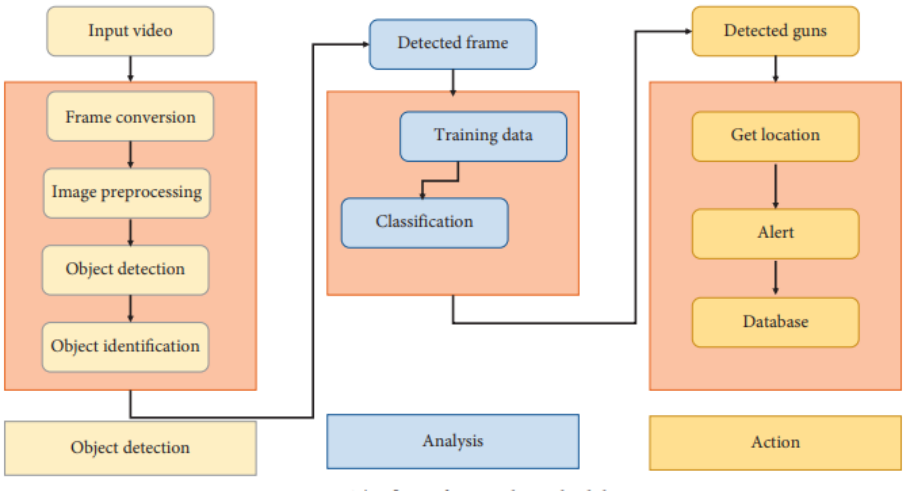


Figure 4.3: Block diagram for threat detection.

In this methodology we set out to create a consolidated structure for scrutiny and security that gradually isolates the ammunition and, if the identification is confirmed, warns/briefs security professionals on how to handle the issue by arriving at the site via IP cameras. A paradigm that gives a computer the ability to recognize dangerous weapons and can also alert a human administrator when a pistol or weapon is seen nearby.

Furthermore, we have a system in place to lock doors when the shooter appears to be armed with a lethal weapon. If at all possible, we may also share the live image with security personnel to allow them to make the move in the meanwhile using IP webcams. We've also built an information system to track all of the drills and transmit effect actions in the metropolitan areas in case of a future disaster.

This leads to the development of a database to record all of the activities in order to respond quickly in the event of a future emergency. The most significant and critical aspect of any enactment is the availability of a wanted and appropriate dataset for training machine learning models.

As a result, we interactively gathered a large number of photographs from Google. Alternatively, because the photos are processed in batches, the sizes of all the images are known before training.

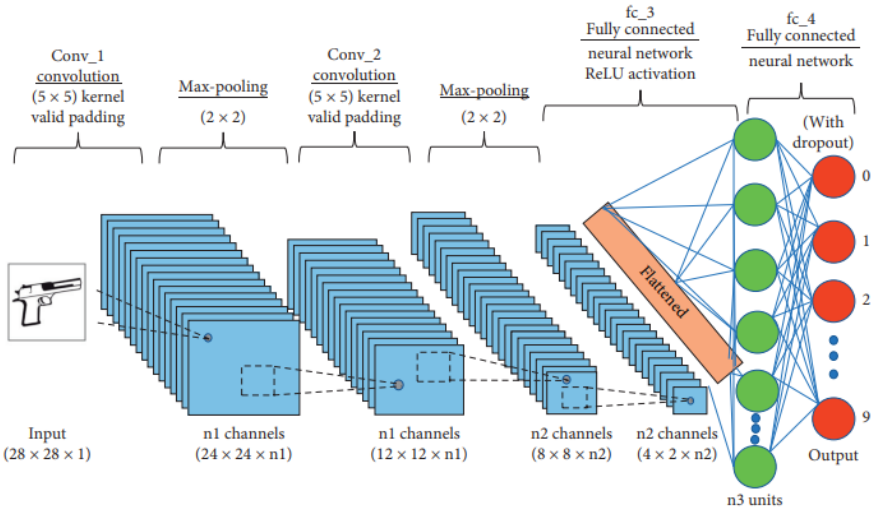


Figure 4.4: Images of gun being detected by using CNN's.

The pixels (416 X 416) are scaled to the original dimensions. Object detection is closely related to computer vision, which involves recognizing objects in digital images. Object recognition is one of the domains that has benefited considerably from recent deep learning developments. YOLO is simply a pre-trained object detector. It's based on the CNN model.

A convolution neural network is a technique capable of taking a raw input image and assigning learnable weights and biases to the image's various aspects/objects. A convolutional layer in a CNN model is responsible for taking out greater-level characteristics such as edges from the input image. This works by repeatedly applying the (K X K) kernel filter on the raw image. This produces activation maps or feature maps as a consequence. The existence of captured features from any provided input is represented by these feature maps.

We require much less pre-processing than other categorization techniques. In the old method, filters are hand-engineered, but in CNN, they are learned through iterations and training. The next layer is Max-Pooling or Subsampling, which reduces the spatial size of the convolved features. Dimensionality reduction aims to lower the amount of computer processing power required to process the data. The non-saturating activation quality is linked to ReLU, which stands for rectified linear unit activation. By setting unwanted values to zero, it

effectively removes them from an activation map. Finally, completely connected layers are used to convert the data into a one-dimensional array. The flattened output is fed into a feedforward neural network, which uses backpropagation to construct a specified long feature vector for each training iteration.

As shown by the convolutional layer's output, these layers are prone to learning nonlinear combinations of greater-level information. f(x) = maximum (0, x).

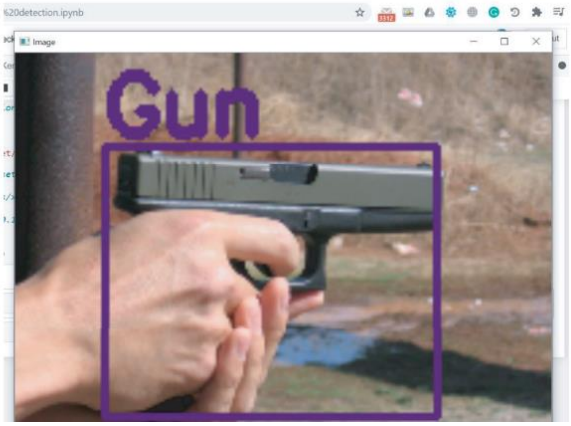


Figure 4.5: Images of gun being detected.

Training a model from start takes a long period; it can take weeks or months to complete. A trained model has seen a large number of objects and understands how each one should be categorised. Training the network on the COCO and Image net datasets yielded the weights in the above-mentioned pretrained model.

As a result, it can only detect things that are members of the classes defined in the dataset used to train the network. It employs three scale forecasts and Darknet-53 as the feature extraction backbone network. DarkNet-53 is a convolutional neural network with 53 layers. The neural network DarkNet-53 is entirely convolutional. Stride 2 uses a convolutional technique to replace the pooling layer. Furthermore, residual units are employed to avoid gradient dispersion. Initially, CNN designs were linear.

YOLO V3 is designed to be a multi-scaled detector rather than an image classifier. As a result, the head that is classified is replaced with a detected head for object detection in this design. The bounding box coordinates and probability classes will now be included in the output vector.

Darknet-53, a framework for training neural networks with 53 layers, is inherited as the backbone of YOLO V3. A total of 106 layers of fully convolutional architecture are layered on top of it for the object detection task. Because of its multiscale feature fusion layers, YOLO V3 uses three feature maps of various scales for target detection.

Figure 4.6: Images of guns being detected with different orientations.

Furthermore, dense characteristics are collected and shared with the rest of the detectors, allowing them to operate at a faster rate, which must be assessed further during the testing phase. As a result, for object subcategorization with competitive performance on many datasets, intraclass variation of the objects is presented.

Descriptors have been developed across long-range motion projections called tracklets, in contrast to standard approaches using optical flow, which only analyze edge features from two consecutive frames. On the tracklets that go through them, spatial-temporal cuboid film sequences are statistically captured.

In order to establish peace from all the violence being committed due to weapons and ammunition during wars or civil attacks or communal attacks it is not just enough to propagate ideas to counteract them, but we need to establish a system to identify and counter attack the threats that are persisting in the world.

For passive millimeter wave photography on a small-scale dataset, metallic cannons mounted on a human skeleton were employed. The Single Multi-Box Detector algorithms YOLO v3 13 and YOLO v3 53 are then compared on the Passive Millimeter Wave dataset. Furthermore, the weapon detection accuracy was calculated at 36 frames per second detection speed and 95% mean average precision. By combining the YOLO V3 method with a quicker

region-based CNN and separating the number of false negatives and false positives, real-time photos can be captured and trained using the YOLO V3 algorithm. They used four separate videos to compare faster RCNN to YOLO V3, and found that YOLO V3 imparted faster speed in real time.

Figure 4.7: Armed-forest commandos.

5. COMPONENTS DISCRIPTION

The surveillance vehicle mainly consists of

- Raspberry pi 4
- L298N motor driver
- Pi camera
- Power bank
- Robot chassis
- DC motors
- Jumper wires
- C type USB cable
- HDMI Cable

5.1. RASPBERRY PI 4

Raspberry Pi is a line of single-board computers. It was deployed in June 2019 and features a 1.5 GHz 64-bit Quad core ARM Cortex-A72 processor, on-board 802.11 ac WI-FI, Bluetooth 5, two USB 2.O ports, two USB 3.O ports, 8 GB of RAM, and dual monitor support with a pair of micro-HDMI ports for up to 4K resolution. The circuit board in the 8 GB version is different. As a result, the Pi can only be powered by 5 volts.

The Raspberry Pi uses a forty-pin or twenty-six-pin connector, depending on the model, and it is critical to understand organisation and labelling. The general- purpose input/output header provides power and interface options:

- 3.3V (on 2 pins)
- 5V (on 2 pins)
- Ground (on 8 pins)
- General purpose input and output
- PWM (pulse width modulation)
- Inter Integrated Circuit
- Integrated Inter-IC Sound Bus
- Serial Peripheral Interface

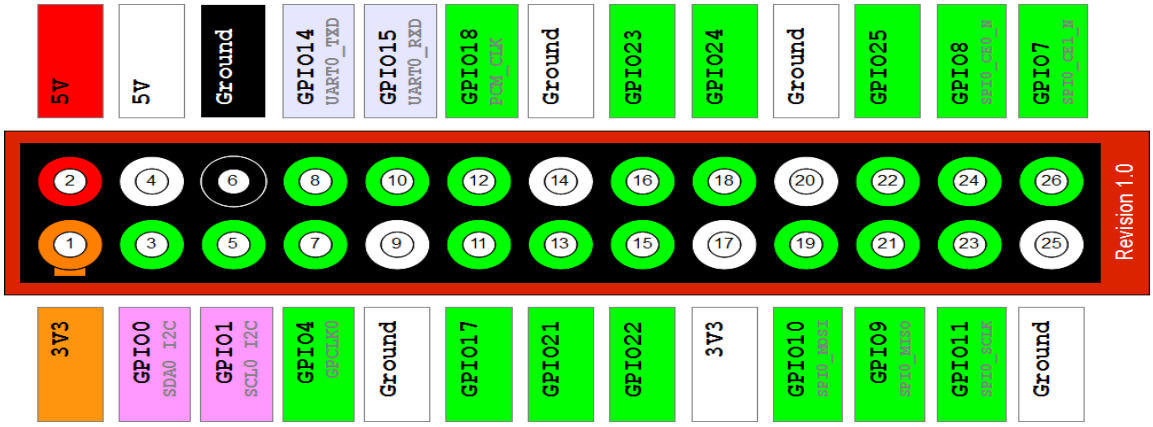


Figure 5.1.1: GPIO Pin Configuration

The Pi 4 is designed for continuous operation, with enough thermal headroom for a short performance sprint, just like the previous generations. A heat sink and a casing with a fan would be useful if you want to push your Pi 4 a little harder than the default settings.

The Broadcom Video Core VI GPU on the Pi 4 is significantly quicker than the GPU on the Pi 3. The Pi 3 managed 27.8 frames per second in the built-in time demo when running Open Arena at 1280 x 720 resolutions, whereas the Pi 4 managed 41.4 frames per second.

All of these enhancements mean that the Pi 4 is significantly more competent than its predecessor, especially when used on a desktop. The Pi 4 is smooth and responsive, unlike previous generations of devices that felt a little slow and required some waiting.

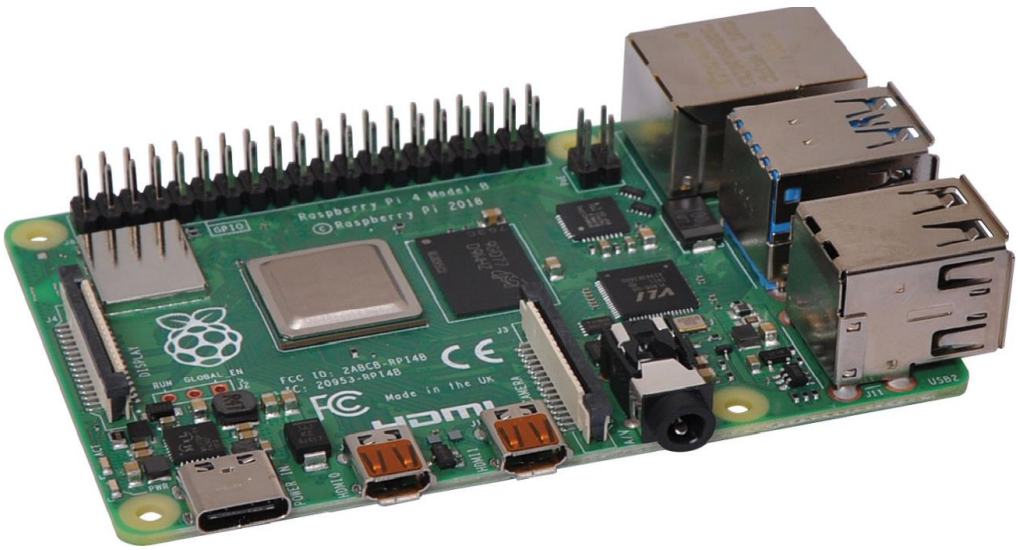


Figure 5.1.2: Raspberry Pi 4

5.2. MOTOR DRIVER IC L298N

The L298N H-bridge module is compatible with motors with a voltage range of five to thirty-five volts DC. It is simple to control one or two DC motors using the L298N H-bridge module. First, connect each motor to the L298N module's A and B connections.

If you're using two motors for a robot or anything else, make sure the polarity of the motors is the same on both inputs. If both motors are set to forward and one is set to reverse, you may need to swap them.

Connect the power supply to pin four of the L298N module and the negative/GND to pin five of the L298N module. Because we have two DC motors in this project, digital pins D9, D8, D7, and D6 will be connected to pins IN1, IN2, IN3, and IN4, respectively.

Then connect D10 to module pin number 7 (after removing the jumper) and D5 to module pin number 5 (again, remove the jumper). The direction of each DC motor is controlled by sending a HIGH or LOW signal to the drive. For example, for motor one, a HIGH signal to IN1 and a LOW signal to IN2 will cause the motor to turn in one direction, while a LOW signal to IN1 and a HIGH signal to IN2 will cause the motor to turn in the opposite direction.

The motors, however, will not move until a HIGH signal is applied to the enable pin (seven for motor one, twelve for motor two). When the LOW signal is applied to the same pin, they can be turned off. If you need to control the motor speed, use the pulse width modulation signal from the digital pin connected to the enable pin.

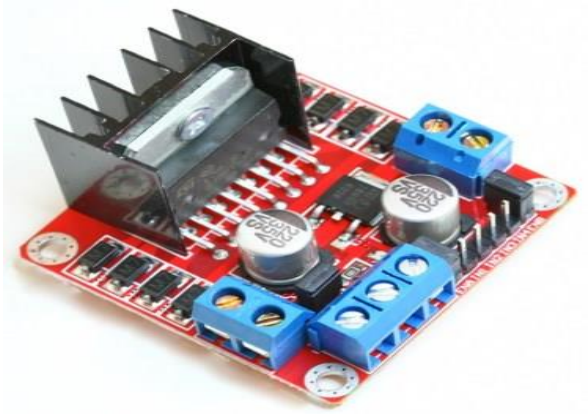


Figure 5.2.1: L298N Motor Driver

5.3. PI CAMERA MODULE

A Raspberry Pi is used to control a surveillance camera. It feeds live video and saves motion-detected segments to the cloud or a Windows shared folder for later viewing. When motion is detected, the cameras begin recording, and the Raspberry Pi device saves the footage in a protected folder.

The proposed security system gathers data and sends it over Wi-Fi to a static IP address that can be accessed via a web browser from any smart device. A Raspberry Pi is used to control a surveillance camera.

This surveillance system detects movement of any object inside its camera's range, takes the image, and uploads it to the Raspberry Pi.

Figure 5.3.1: Pi camera

5.4. PI CAMERA FLEX CABLE

The Raspberry Pi camera module uses a CSI bus devoted to carrying pixels and a flexible wire to link your camera to your board to provide high-quality images to your Raspberry Pi. The module's cord measures 150 mm in length.

When designing Raspberry Pi projects, a longer wire gives you more flexibility. And you may rest assured that the data transfer results are same.

Nothing is easier to use than a Raspberry Pi Flex Cable - unless it's another Raspberry Pi cable that's a little longer! Carefully open the CSI connector on your Pi, insert your cable, and you're good to go.

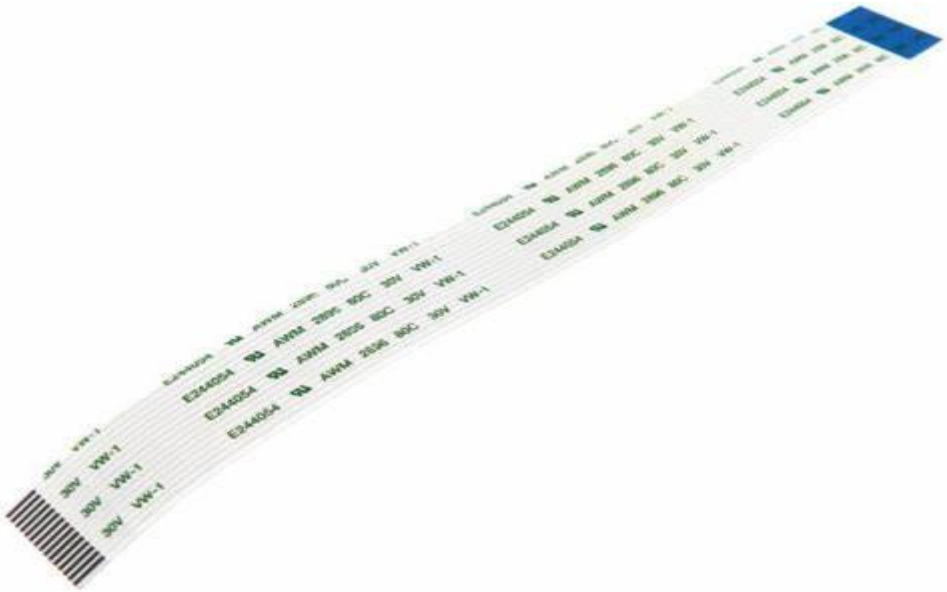


Figure 5.4.1: Pi Camera Flex cable

5.5. POWER BANK

In this project we use a 1000mAh capacity power bank for power supply to both Raspberry pi GPIO pins to motor driver

For the tourist and explorer, power banks are essential new-generation portable energy options. A power bank consists of a casing and a rechargeable battery that can be used to store energy is a physical power source

This means you'll always have power. Power banks are particularly popular among campers and hikers. They will let you to bring your favourite devices, such as your cell phone or laptop, and charge them while on the trail, allowing you to be mobile for longer.

The list of advantages and things you can recharge rapidly with one is infinite. For the road, you can charge your gadget fast and easily, with up to two or three full charges on items like your phone and other tiny devices.

Figure 5.5.1: 10000mAh power bank

5.6. DC MOTORS

Using DC current, the DC motor converts electrical energy into mechanical energy. The main advantage is that we have more control over the speed and it takes up less space. DC motors were the first widely used motors because they could be powered by existing direct-current lighting power distribution networks.

The speed of a direct current motor can be varied over a wide range by adjusting the supply voltage or the current intensity in the field windings. Small DC motors can be found in a wide range of tools, toys, and appliances.

The universal motor is a small direct current brushed motor found in portable power tools and appliances. Electric vehicle propulsion, elevator and hoist drives, and steel rolling mill drives all use larger direct current (DC) motors.

Figure 5.6.1: DC motors

5.7. JUMPER WIRES

In short, it is a kind of connecting cables. It is very useful to connect Raspberry pi 4 and L298N motor driver. There are three types of jumper wires according to the presence of male and female inputs at the ends

1. Male–Male
2. Male-Female
3. Female-Female

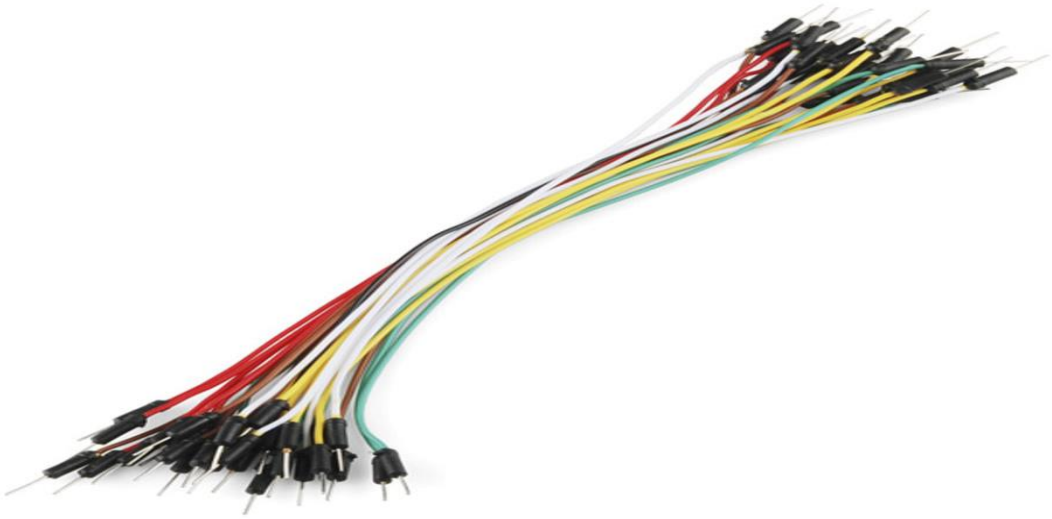


Figure 5.7.1: Jumper wires

5.8. USB TYPE C CABLE

In this project we used USB Type C cable for power supply to raspberry pi board from power bank. USB Type C cables provide extraordinarily high-speed data transfer and power. Type C cables are without a doubt becoming the most used connections in the digital world.

Reverse insertion is possible with Type-C ports, so you can plug in either end of the cable. Type-C ports have a faster data transfer rate. A USB 3.1 Type-C port can be used to transfer 4K videos. With Type-C ports, charging currents ranging from 3 to 5 A, reverse charging is possible.

Figure 5.8.1: USB Type C cable

5.9. HDMI CABLE

An HDMI cable is an item that you should always have at home. High Definition Multimedia Interface (HDMI) is a standard and a connector for sending high definition digital video and audio data between devices.

To use HDMI, you must first have devices that accept the standard. They should have an HDMI interface or port that can be connected to them using an HDMI cable. HDMI cables come in a numerous shapes and colours, but they all perform the same thing: they carry video and audio data from one device to another, such as a laptop to a television**.**

Figure 5.9.1: HDMI cable

5.10. ROBOT CHASSIS

All of the project's components are mounted on the robot chassis, including a power bank, a Raspberry Pi 4 model B, L298N motor driver and a pi camera. Robot chassis is a device it can be consist of DC motors, wheels, Upper and bottom glass sheets contain holes for wiring connection purpose.

A surveillance system is one that is utilised for security purposes. This system is intended to provide a video surveillance system that captures images and stores them for further verification

The robot is operated from a safe distance and devises a plan to deal with their activities. It acquires images from cameras via a web browser, Wi-Fi, or a mobile application.

It receives images from cameras in real time. This robot is capable of detecting weapons such as guns and knives.

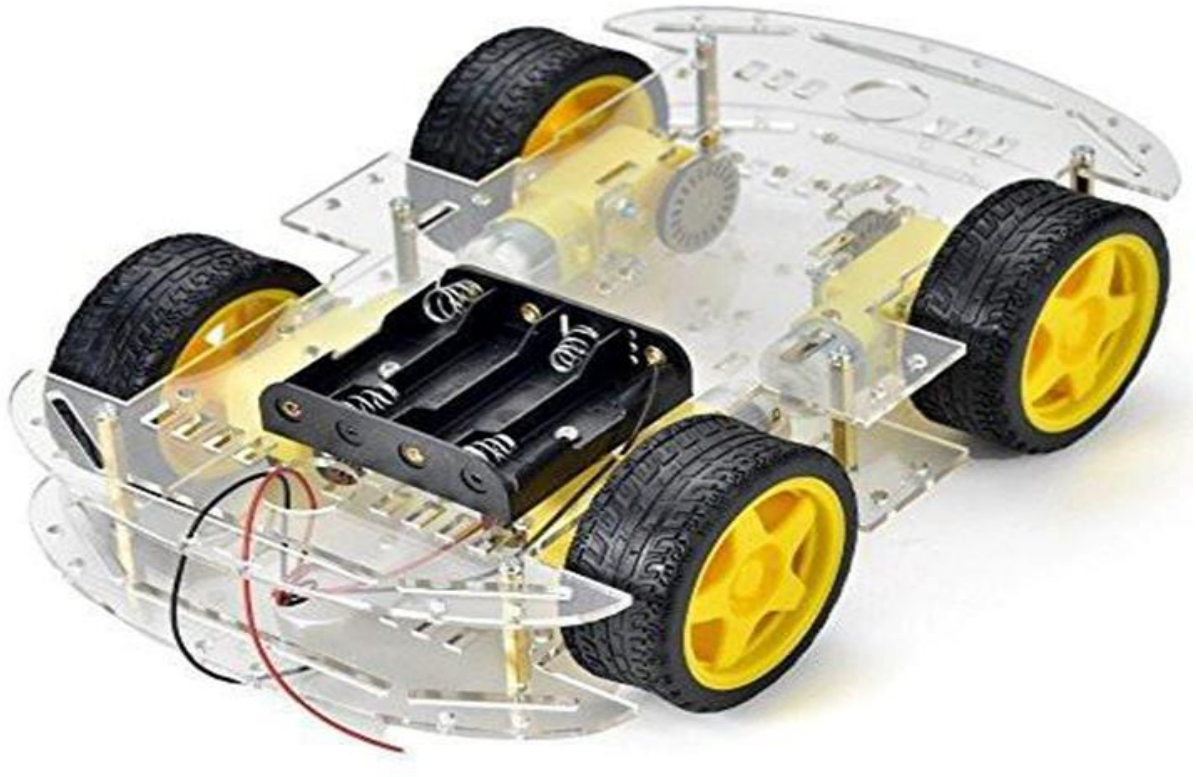


Figure 5.10: Robot chassis

6. BLOCK DIAGRAM AND FUNCTIONING OF SURVEILLANCE ROBOT

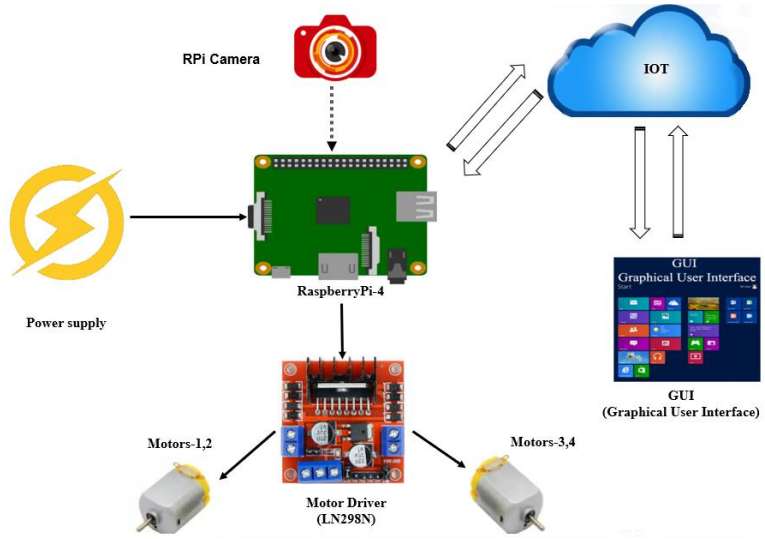


Figure 6.1: Operational Diagram for surveillance vehicle.

The robot used in this project is a mobile robot with four actuator wheels. The block diagram of a mobile robot for heavy loads, which includes a Raspberry Pi, a camera motor driver (L298N), two motors, and a power supply (power bank). For more than a half-century, robots have been a mainstay of advanced manufacturing.

Robots and their peripheral equipment are increasingly being used for entertainment, military, and surveillance purposes as they become more sophisticated, reliable, and manufactured. A remote-controlled surveillance robot is any robot that can be remotely controlled to capture images or video for specific purposes.

Remotely controlled mobile robots that capture images/videos for specific purposes. Remotely controlled mobile robots play an important role in rescue and military operations. A rescue robot is a type of surveillance robot that has been designed to rescue people.

There are numerous microcontrollers available on the market, ranging from basic input output to high-end microcontrollers. These various types of microcontrollers are designed for general use. In this project, we propose an architecture for a controllable Raspberry Pi-based

robot. Typically, surveillance robots help with law enforcement and security our project try to help personnel detect trouble while lawful citizens remain far from harm's way. The aim of this project is to aid the police in town, military forces at the borders and cities, like in parking lots or shopping malls. It is one of the finest projects of how the efforts of human security personnel are augmented by robotics helper.

The surveillance vehicle is one of the most advanced models, using a very fast processing raspberry pi, greatly defined cameras and artificial intelligence. All of these abilities shows where the technology of the day is moving. Our surveillance vehicle offers the surveillance of area to keep the officials knowledgeable about threat.

Our surveillance vehicle also provides live streaming for having better view for the operator, after reaching the enemy location safely it can also provide the objects what the enemy carrying, this helps the official to know and predict the danger in advance.

The surveillance vehicle enables remote video surveillance over medium distances using an intelligent Rpi camera, either from a stationary position or while moving and using the camera. After determining the best location for stationary video surveillance, the Rpi camera can detect people, objects, and weapons.

The operation mode involving intelligent group patrols that use artificial intelligence enables the provision of a high level of security that is not possible using technical approaches that combine CCTV cameras and patrolling security officers.

The on-board supercomputer in the video content analytics system in surveillance vehicle use deep learning algorithms to detect objects. Their neural networks are capable of distinguishing different objects.

Deep learning algorithms are used by the on-board supercomputer in the video content analytics system in the surveillance vehicle to detect objects. Their neural networks can distinguish between different objects.

7. SOFTWARE REQUIREMENTS

7.1. VNC VIEWER

Virtual Network Computing (VNC) is a graphical desktop sharing system in computing that uses the Remote Frame Buffer protocol (RFB) to control another computer remotely. It relays graphical screen updates back and forth across a network by transmitting keyboard and mouse events from one computer to another.

VNC is platform-independent, with clients and servers available for many GUI-based operating systems as well as Java. A VNC server can be accessed by multiple clients at the same time. This technology is commonly used for remote technical support and accessing files on one's work computer from one's home computer, or vice versa. Under the GNU General Public License, the original VNC source code and many modern derivatives are open source.

VNC comes in a variety of flavours, each with its own set of features. For example, some are optimised for Microsoft Windows, while others offer file transfer (which is not part of VNC itself). Many are compatible (without their extra features) with standard VNC in the sense that a viewer of one flavour can connect to a server of another; others are based on VNC code but are incompatible with standard VNC.

7.2. PUTTY

Putty is a free secure shell (and telnet) client for Windows PCs (it also includes an x-term terminal emulator). Putty is useful if you want to connect from a PC to a UNIX or other multi-user system (for example your own or one in an internet cafe). Putty is a terminal emulator, serial console, and network file transfer tool that is free and open source. SCP, SSH, Telnet, rlogin, and raw socket connections are among the network protocols supported. It is also possible to connect it to a serial port. The name "PUTTY" has no obvious meaning.

Putty was originally designed for Microsoft Windows, but it has since been ported to a variety of other platforms. Official ports for certain Unix-like platforms are available, as are work-in-progress versions for Classic Mac OS and macOS, as well as unofficial ports for Symbian, Windows Mobile, and Windows Phone. Putty was created and is maintained primarily by Simon Tatham, a British programmer. Putty supports a number of options on the secure remote terminal, including user control over the SSH encryption key and protocol

version, as well as other cyphers such as AES, 3DES, RC4, Blowfish, DES, and public-key authentication.

Putty has its own key file format known as pre-printed key (protected by Message Authentication Code). Putty supports GSSAPI, including user-supplied GSSAPI DLLs. It also supports SSH for local, remote, or dynamic port forwarding and can simulate control sequences from x-term, VT220, VT102, or ECMA-48 terminal emulation. The SSH protocol supports the delayed compression method, and the network communication layer supports IPv6. Connections to local serial ports are also supported.

7.3. REMOTE.IT

Remote.it makes it simple to manage cloud and field access to services and devices. With remote.it, you can re imagine cloud access. Manage VPC access via email and set resource access restrictions. Even when IP addresses are unavailable, you can monitor and connect to other devices without using open ports.

Applications for Linux and Raspberry Pi can be downloaded and installed quickly by using remote.it The Service Agent is only included in this package for inbound connections. Allows for the configuration of remote services. Supports Debian, Red Hat, OpenWrt, Axis, Jetson, and other Linux distributions. For a complete list, click "+ See More Options" below.

A GUI is included in the installation for monitoring, setting up services, initiating connections, and managing users. Create secure remote connections without using open ports using Windows RDP, Mac Screen Sharing, and other methods.

8. RESULTS AND CONCLUSION

The prototype created a better chance for security to detect the threat in real-time, potentially preventing a crime. The raspberry pi was used for surveillance at the camera side, eliminating the need for cloud computing. We also used the YOLOv3 algorithm for segmentation. On top of both models, a user-friendly interface was created to allow users to communicate with the system. This demonstrates that SV is appropriate for real-time surveillance.

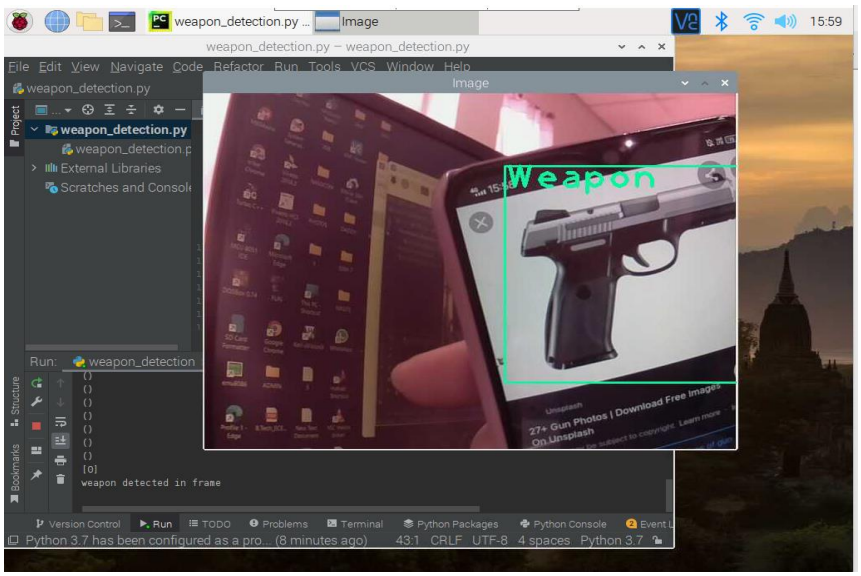


Figure 8.1: Gun detected in the frame as weapon and being alerted.

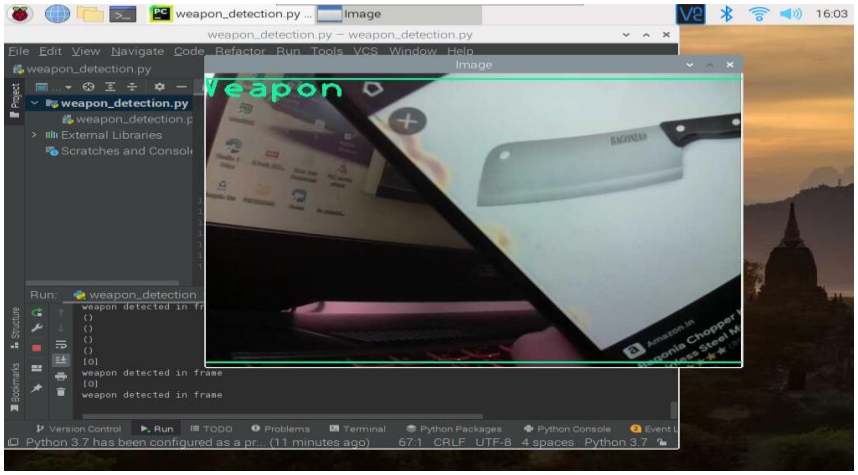


Figure 8.2: Knife detected in the frame as weapon and being alerted.

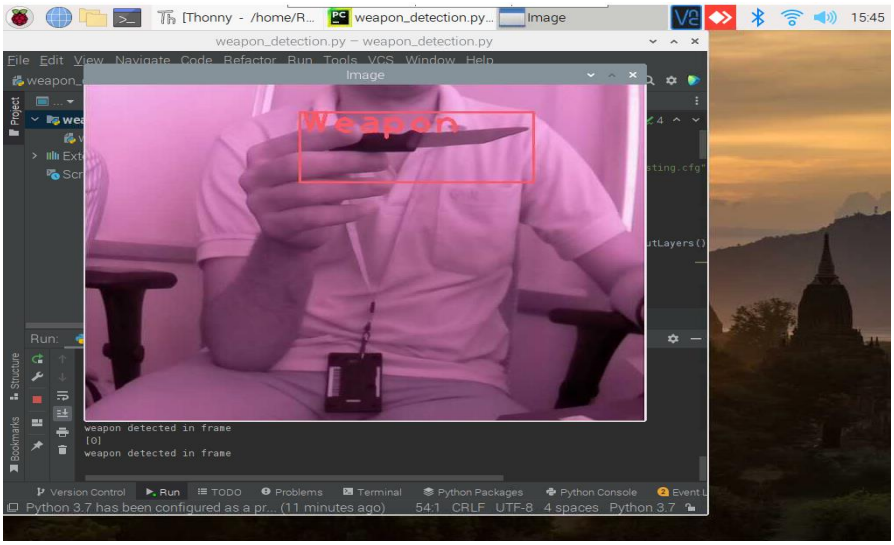


Figure 8.3: Physical Knife detected in the frame and being alerted.

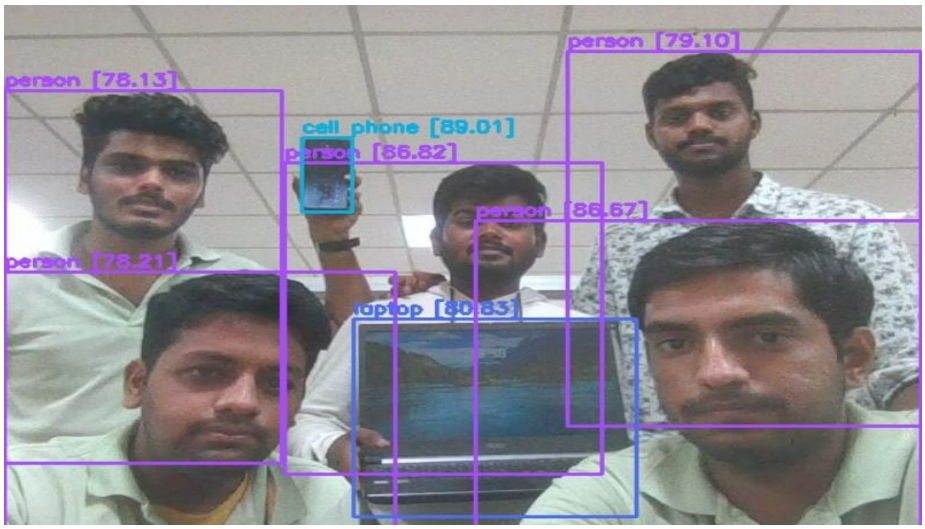


Figure 8.4: Final object detection system using YOLO V3.

Figure 8.5: Physical gun detected in the frame and being alerted.

Model performance parameters on different datasets

	UGR	IMFDB	COCO
Accuracy	0.8567	0.8174	0.8864
Precision	0.8396	0.7891	0.8234
F1 Score	0.8957	0.8221	0.8456
Recall	0.9563	0.8536	0.9345

Tabular Form 8.1: Comparison of Model performance parameters on different datasets

Figure 8.6: Surveillance Vehicle.

9. FUTURE SCOPE

This surveillance vehicle can be employed in areas without or less security or very remote locations. So, if we can increase the range of employment of deep learning and machine learning techniques then there is a chance of making the surveillance vehicle's functionality more dynamic.

X-ray scanning techniques can be used in order to detect the hidden objects. Ionizing radiation techniques help in achieving this feat. By computing most sophisticated algorithms it is even possible to find the percentage of metals present in the hidden objects. But usage of this technology must be with absolute care and precautions

In order to overcome the drawback of fixed camera motions, a telescopic arrangement can be made to increase the inclination angle and the height to which the camera can be moved upwards. Placing a servo motor helps in motion of camera in side wise direction.

The vehicle can be trained in such a way that it can lock on to a particular object. Employing the same design to drone can make surveillance much more effective.

Training with much more datasets increases the accuracy and precision. When trained with fire and water images it can alert people during calamities caused by bush fires, forest fires and floods etc.

REFERENCES

[1] T. Soni and B. Sridhar, “Modeling issues in vision-based aircraft navigation during landing," in Proc. IEEE Workshop Appl. Comput. Vision, Sarasota, FL, Dec. 5-7, 1994. pp. 89–96.

[2] I. Cohen and G. Medioni, "Detecting and tracking moving objects in video from and airborne observer," in Proc. DARPA Image Understanding Workshop, Monterey, CA, Nov. 20-23, 1998, pp. 217–222.

[3] R. Garcia, X. Cufi, and M. Carreras, "Estimating the motion of an underwater robot from a monocular image sequence,” in Proc, IEEE/RJS Int. Conf. Intell. Robots and Syst., Maui, HI, Oct. 29-Nov. 3, 2001, vol. 3,pp. 1682-1687.

[4] J. Rosenblatt, S. Willams, and H. Durrant -Whyte, "Behavior-based control for autonomous underwater exploration," in Proc: IEEE Int. Conf. Robor. Autom., San Francisco, CA. Apr. 24-28, 2000, PP. 920-925.

[5] A. Bennett and J. J. Leonard, "A behavior-based approach to adaptive feature mapping with autonomous underwater vehicles." IEEE J. Ocean.Eng., vol. 25. no. 2. pp. 213-226. Apr. 2000.

[6] M. Hebert, C. Thorpe, and A. Stentz, Intelligent Unmanned Ground Veliicles: Autonomous Navigation Research at Carnegie Mellon Norwell,MA: Kluwer, 1997.

[7] B. Southall, T. Hague, J. A. Marchant, and B. F. Buxton, "Vision-aided outdoor navigation of an autonomous horticultural vehicle," in Proc. 1st ICVS. Gran Canaria, Spain. Jan. 1,3-15, 1999. pp. 37–50.

[8] E. D. Dickmanns and B. Mysliwets, "Recursive 3D road and relative egoslate recognition," IEEE Trans. Pattern Anal. Mach. Intell., vol. 14.no. 2, pp. 199–213, Feb. 1992.

[9] E. D. Dickmanns, "Computer vision and highway automation," Veh. Syst. Dyn., vol. 31. no. 5. Pp. 325–343. Jun. 1999.

[10] D. A. Pomerlau and T. Jockem, “Rapidly adapting machine vision for automated vehicle steering," IEEE Intell. Susi.. vol. II. no. 2. pp. 19–27.Apr. 1996.

[11] H. I: Moravec, "The Standford Cart and the CMU rover," Proc. IEEE,vol. 71, no. 7. pr. 872–884, Jul. 1983.

[12] R. Collins, A. Lipton, H. Fujiyoshi, and T. Kanade, "Algorithms for cooperative multisensor surveillance, “ Proc, IEEE, vol, 89, no. 10, pp,1456–1477. Oct. 2001.

[13] C. Regazzoni, V. Ramesh, and G. Foresti, "Special issue on video communications, processing, and understanding for third generation surveillance systems." Proc. IEEC, vol. 89. no. 10. pp. 1.355–1539. Oct. 2001.

[14] L. Davis, R. Chellapa, Y. Yaccob, and Q. Zheng, "Visual surveillance and monitoring of human and vehicular activity." in Proc DARPA Image Understanding Workshop. New Orleans, LA. May 13–15, 1997.pp. 19–27.

[15] R. Howarthand and H. Buxton. "Visual surveillance monitoring and watching," in Proc. Eur. Conf. Comp. Vis., Cambridge, L.K.Apr. 13-14. 1996. pp. 3321-334.

[16] T. Kanade, R. Collins, A. Lipton. I! Burt, and L. Wixson, "Advance's in cooperative multi sensor video surveillance,” in Proc. DARPA Image Understanding Workshop, Monterey, CA, Nov. 20–2.3.1995, pp. 23-24.

[17] G. L. Foresti. "Object recognition and tracking for remote video surveillance." IEEE Trans, Circuits Syst. Video Technol. Vol. 9, no. 7. pp. 10445_1062, Out.1999.

[18] G. L. Foresti, P. Mahonen, and C. Regazzoni, Multimedia Video-Based Surveillance Systems: From User Requirements to Research Solutions. Norwell, MA: Kluwer. Sep. 2000.

[19] G. L. Foresti, P. Mahonen, and C. Regazzoni, Multimedia Video-Based Surveillance Systems: From User Requirements to Research Solutions. Norwell, MA: Kluwer. Sep. 2000.

[20] D. Koller, K. Daniilidis, and H. H. Nagel, "Model-based object tracking in monocular sequences of road traffic scenes," Int. J. Comput. Vis., vol. 10.no. pp. 257–281. Jun. 1993.

[21] Z. Zhu, G. Xu, B. Yung. D. Shi, and X. Lin, "VISATRAM: A real time vision system for automatic traffic monitoring." Image Vis. Comput.,vol. 18. no. 10. pp. 787–794. Jul. 2000.

[22] S. Araki, T. Matsuoka, N. Yokova, and H. Takemura, "Real-time tracking of multiple moving object contours in a moving-camera image sequences," JEICE Trans. Inf. Syst., vol. E83-D, no. 7. pp. 1583–1591, Jul. 2000.

[23] D. Murray and A. Basu, "Motion tracking with an active camera,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 19, no. 5, pp. 449-454, May 1994.

[24] C. Tomasi and T. Kanade, “Detection and tracking of point features, Carnegie Mellon Univ., Pittsburgh, PA, Tech. Rep. CMU-CS-91-132, Apr. 1991.

[25] M. J. Black and D. J. Fleet, "Probabilistic detection and tracking of motion boundaries, " Int. J. Comput. Vis., vol. 38, no. 3. pp. 231-245, Jul/Aug. 2000.

[26] M. Irani and P. Anandan, "A unified approach to moving object detection in 2D and 3D scenes," IEEE Trans. Pattern Anal. Mach. Intell., vol. 20, no. 6, pp. 577-589. Jun. 1998.

[27] P. Anandan, P. J. Burt, K. Dana, M. Hansen, and G. Van der Wal, “Realtime scene stabilization and mosaic construction," in Proc. DARPA Image Understanding Workshop, Monterey, CA, Nov. 13-16, 1994, pp. 457-465.

[28] G. L. Foresti. "Object recognition and tracking for remote video surveillance." IEEE Trans, Circuits Syst. Video Technol. Vol. 9, no. 7. pp. 10445_1062, Out.1999.

[29] L. Snidaro and G. L. Foresti, "Real-time thresholding with Euler numbers, " Pattern Recognit. Lett., vol. 24, no. 9/10, pp. 1533–1544. Jun. 2003.

[30] T. Tommasini, A. Fusiello, E. Trucco, and V. Roberto, “Making good features track better,” in Proc. IEEE Conf. Comput. Vis. und Pattern Recog., Santa Barbara, CA. Jun. 23–25. 1998, pp. 178–183.

[31] R. Garcia, X. Cufi, and M. Carreras, "Estimating the motion of an underwater robot from a monocular image sequence,” in Proc, IEEE/RJS Int. Conf. Intell. Robots and Syst., Maui, HI, Oct. 29-Nov. 3, 2001, vol. 3,pp.1682-1687.

[32] B. K. P. Horn and E. J. Weldon, "Direct methods for recovering motion." In. J. Comput. Vis., vol. 2, no. 1, pp. 51–76, Jun. 1988.

[33] G. P. Stein and A. Shashua, "Model-based brightness constraints: On direct estimation of structure and motion." IEEE Trans. Pat tern Anal.Mach. Intell.,vol. 22, no. 9, pp. 992-1015, Sep. 2000.

[34] M. Irani and P. Anandan, “About direct methods," in Proc. Int. Workshop Vis. Algorithms: Theory and Practice ICCV, Corfù, Greece, Sep. 21-22,1999, pp. 267–277.

[35] A. A. Moamen and N. Jamali, ‘‘Opportunistic sharing of continu- ous mobile sensing data for energy and power conservation,’’ *IEEE Trans. Services Comput.,* vol. 13, no. 3, pp. 503–514, May/Jun. 2020, doi: 10.1109/TSC.2017.2705685.

[36] J. Lim, M. I. Al Jobayer, V. M. Baskaran, J. M. Lim, K. Wong, and J. See, ‘‘Gun detection in surveillance videos using deep neural networks,’’ in *Proc. Asia–Pacific Signal Inf. Process. Assoc. Annu. Summit Conf. (APSIPA ASC)*, Nov. 2019, pp. 1998–2002.

[37] M. Grega, S. Lach, and R. Sieradzki, ‘‘Automated recognition of firearms in surveillance video,’’ in *Proc. IEEE Int. Multi-Disciplinary Conf. Cog- nit. Methods Situation Awareness Decis. Support (CogSIMA)*, Feb. 2013, pp. 45–50.

[38] U. V. Navalgund and P. K., ‘‘Crime intention detection system using deep learning,’’ in *Proc. Int. Conf. Circuits Syst. Digit. Enterprise Technol. (ICCSDET)*, Dec. 2018, pp. 1–6.

[39] Y. Zhou, L. Liu, L. Shao, and M. Mellor, ‘‘Fast automatic vehicle anno- tation for urban traffic surveillance,’’ *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 6, pp. 1973–1984, Jun. 2018.

[40] Y.-X. Liu, Y. Yang, A. Shi, P. Jigang, and L. Haowei, ‘‘Intelligent moni- toring of indoor surveillance video based on deep learning,’’ in *Proc. 21st Int. Conf. Adv. Commun. Technol. (ICACT)*, Feb. 2019, pp. 648–653.

[41] H. Cui, Z. Wei, P. Zhang, and D. Zhang, ‘‘A multiple granular cascaded model of object tracking under surveillance videos,’’ in *Proc. Int. Conf. Algorithms, Comput. Artif. Intell.*, Dec. 2018, pp. 1–8.

[42] Y. Wang, T. Bao, C. Ding, and M. Zhu, ‘‘Face recognition in real-world surveillance videos with deep learning method,’’ in *Proc. 2nd Int. Conf. Image, Vis. Comput. (ICIVC)*, Jun. 2017, pp. 239–243.

[43] C. Ding and D. Tao, ‘‘Trunk-branch ensemble convolutional neural net- works for video-based face recognition,’’ *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 4, pp. 1002–1014, Apr. 2018.

[44] R. Qian, X. Lai, X. Li, 3D object detection for autonomous driving: A survey, 2021, ArXiv preprint arXiv:2106.10823.

[45] G. K. Verma and A. Dhillon, “A Handheld Gun Detection using Faster R-CNN Deep Learning,” Proceedings of the 7th International Conference on Computer and Communication Technology – ICCCT-2017, 2017

[46] J. Redmon, S. Divvala, R. Girshick and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, 2016, pp. 779-788.

[47] J. Redmon, A. Farhadi, "Yolov3: An incremental improvement," arXiv 2018 arXiv:1804.02767.

[48] J. Redmon, “Darknet: Open-source neural networks in c,” 2013