

# **An Intelligent Intrusion Detection System for Surveillance Cameras**

Graduation Project I (Midterm Report)

By:

**Mohamed Nasser Hashem, ID: 372029063**

**Mustafa Ahmed Abdulsami, ID: 372029353**

**Husen Muhammad Kalaepih, ID: 352072163**

Supervised By:

**Dr. Emad Nabil**



الجامعة الإسلامية بالمدينة المنورة  
**ISLAMIC UNIVERSITY OF MADINAH**

**Faculty of Computer and Information Systems**

**Islamic University of Madinah**

---

October - 2021

## **DEDICATION**

This work is completely dedicated to my grandfather (whom my eyes have not seen), respectful Parents, beloved wife, and darling son. without whose constant support this was not possible.

Mohamed Nasser Hashem

## **ACKNOWLEDGMENT**

The success and results of this project have required a tremendous amount of guidance, and fortunately we have achieved that throughout our project.

All that we did was, firstly, by the grace of Allah Almighty for His bounty to complete the project, without his blessings, we would not be able to do anything.

Then, we like to express our special thanks and gratitude to our supervisor, Dr. Emad Nabil, who helped us and directed us throughout the period of work on this project then.

In the end, the support and help from people around us we think it is necessary to thank them, we thank our professors and colleagues who have been supportive of us throughout the project.

## DECLARATION

We hereby certify that this material, which we now submit for assessment on the program of study leading to the award of Bachelor of (*Computer Science*) is entirely my own work, that We have exercised reasonable care to ensure that the work is original, and does not to the best of my knowledge breach any law of copyright, and has not been taken from the work of others save and to the extent that such work has been cited and acknowledged within the text of my work.

**Stud ID.:** 372029063, **Stud Name:** Mohamed Nasser Hashem, **Signed:** \_\_Mohamed\_\_

**Stud ID.:** 372029353, **Stud Name:** Mustafa Ahmed Abdulsami, **Signed:** \_\_Mustafa\_\_

**Stud ID.:** 352072163, **Stud Name:** Husen Muhammad Kalaepheh, **Signed:** \_\_Husen\_\_

**Date:** 10/10/2021

## ABSTRACT

Security threats have become common and real in this time, only humans are no longer able to protect valuables. This may expose them to theft or total or partial damage at the very least. Ordinary human insurance is no longer sufficient currently. This is on the physical side.

On the other hand, the presence of humans has become more intense in public places. With so many criminals or menacing tactics it is almost impossible to discern them by visual ability, as they easily hide inside large crowds.

For this, the developers and the police are working to secure it both technically and realistically. Because security alone is unable to deal with such matters

Due to modern technology, the current era, to the researchers' background in computer science, we can work to create a detection system for surveillance.

In this project we will develop an intelligent system that is able to detect people and some weapons that is used by criminals. The weapons are gun, trifle, and knife. After a detection of one of the weapons an alarm will be raised. The system can be used in public places, squares, shopping malls, etc. we will use modern machine learning technologies for object detection to achieve that goal.

# TABLE OF CONTENTS

|  |             |
|--|-------------|
| <b>DEDICATION .....</b>  | <b>I</b>    |
| <b>ACKNOWLEDGMENT.....</b>                                     | <b>II</b>   |
| <b>DECLARATION .....</b>                                       | <b>III</b>  |
| <b>ABSTRACT .....</b>  | <b>IV</b>   |
| <b>TABLE OF CONTENTS.....</b>                                  | <b>V</b>    |
| <b>LIST OF FIGURES.....</b>                                    | <b>VII</b>  |
| <b>LIST OF TABLES.....</b>                                     | <b>VIII</b> |
| <b>1 INTRODUCTION .....</b>                                    | <b>1</b>    |
| 1.1 AIM.....   | 2           |
| 1.2 PROBLEM STATEMENT .....                                    | 2           |
| 1.3 OBJECTIVES:.....   | 2           |
| 1.4 SCOPE: .....   | 2           |
| 1.5 METHODOLOGY:.....  | 3           |
| 1.5.1 Waterfall:.....  | 3           |
| 1.6 TIMELINE: .....  | 4           |
| <b>2 LITERATURE REVIEW.....</b>                                | <b>5</b>    |
| 2.1 BACKGROUND.....  | 5           |
| 2.1.1 Image Classification.....                                | 5           |
| 2.1.2 Object Localization.....                                 | 5           |
| 2.1.3 Object Detection .....                                   | 6           |
| 2.2 ALGORITHMS .....   | 7           |
| 2.2.1 Histogram of Oriented Gradients (HOG) .....              | 8           |
| 2.2.2 Single Shot Detector (SSD).....                          | 9           |
| 2.2.3 Region-based Convolutional Neural Networks (R-CNN) ..... | 10          |
| 2.2.4 Fast R-CNN.....  | 11          |
| 2.2.5 Faster R-CNN.....  | 12          |
| 2.2.6 Region-based Fully Convolutional Network (R-FCN).....    | 13          |
| 2.2.7 You Just Look Once (YOLO).....                           | 15          |
| 2.2.8 Comparisons.....   | 17          |
| 2.2.9 Summary .....  | 18          |
| 2.3 DATASETS.....  | 19          |
| 2.3.1 Crime Detection – using Deep learning .....              | 19          |
| 2.3.2 Weapon detection datasets.....                           | 19          |
| 2.3.3 Weapon Detection and Classification .....                | 19          |
| 2.3.4 Handgun Dataset.....                                     | 19          |

|          |                                 |           |
|----------|---------------------------------|-----------|
| 2.3.5    | Knife Dataset.....              | 19        |
| 2.3.6    | Summary .....                   | 20        |
| <b>3</b> | <b>ANALYSIS AND DESIGN.....</b> | <b>21</b> |
| 3.1      | FUNCTION REQUIREMENT .....      | 21        |
| 3.2      | NON-FUNCTION REQUIREMENT .....  | 22        |
|          | <b>REFERENCES.....</b>          | <b>IX</b> |

## LIST OF FIGURES

|  |    |
|--|----|
| Figure 1-1: Project Methodology Model (Waterfall) .....                        | 3  |
| Figure 1-2: Giant Chart for Project.....                                       | 4  |
| Figure 2-1: Object Detection Algorithms series.....                            | 6  |
| Figure 2-2: VGG-16 Architecture .....  | 9  |
| Figure 2-3: R-CNN system overview .....  | 10 |
| Figure 2-4: Fast R-CNN system overview .....                                   | 11 |
| Figure 2-5: Faster R-CNN system overview .....                                 | 12 |
| Figure 2-6: Faster R-CNN system analysis.....                                  | 13 |
| Figure 2-7: Comparison of test-time speed of object detection algorithms ..... | 13 |
| Figure 2-8: R-FCN system overview .....  | 14 |
| Figure 2-9: Single vs. Multiple object (YOLO) .....                            | 16 |
| Figure 2-10: Summary of Datasets (Data Type %) .....                           | 20 |



## LIST OF TABLES

|  |    |
|--|----|
| Table 2-1: Summary of Algorithms.....              | 7  |
| Table 2-2: Algorithms Comparison (Speed) .....     | 17 |
| Table 2-3 Algorithms Comparison (Performance)..... | 17 |
| Table 2-4: Summary of Datasets .....               | 20 |

## *Chapter One*

# **1 INTRODUCTION**

In the age of technology, everything in this life has become connected to the world of technology. Coins, diamonds, gold, jewelry, antiques, and other valuables. Humans alone are no longer able to protect these precious things. This is on the physical side.

On the human side, the matter has become more different. The presence of humans has become more intense in public places, such as schools, universities, and parks. With so many, criminals or threatening tactics are virtually impossible to discern.

Security threats have become common and a reality in this time, threats may be from forced robbery to mass hostage-taking. For this, the developers and the police are working to secure it both technically and realistically. It is our duty as developers now to work seriously to obtain security monitoring systems in cooperation with security agencies.

In this project we will try to use the latest open-source object detection algorithms and datasets. To create an intelligent intrusion detection system for surveillance. to start in our work, we started by doing a detailed and meticulous research into the two main sections of this project: the algorithms that will be used as a model for the system, and open-source datasets related to our project (knife, pistol, rifle).

For algorithms, the literature review began examining each algorithm by: research paper, and source-code. Which led to the arrangement of algorithms according to strength and speed, which led us to define the YOLO algorithm. As for data sets, all open-source databases were compiled and filtered, resulting in more than 35,000 images.

## **1.1 AIM**

Create a system connected to a camera that can identify the criminal who enters the bank or the commercial store, by identifying if he is carrying a gun, trifle, or a knife. and to give a warning to all those present in this facility and warn them.

## **1.2 PROBLEM STATEMENT**

Robbery of banks and stores is spreading all over the world as well. It spreads in remote places or branches due to the lack of sufficient security. The thief always carries a gun, a knife, and an ordinary person cannot stop him or warn others about him. He threatens the victim and is taken with hatred and leaves behind many economic problems. The owner of this originator, and no one can defend himself in front of that thief because of the weapon he carries, and in many cases the police cannot reach that thief. Again, theft is repeated a lot without any deterrent to stop them, arrest them, or even flee from them.

## **1.3 OBJECTIVES:**

The main objective of Intrusion Detection System is recognized and find at least one viable focus from still picture or video information. It thoroughly incorporates an assortment of It comprehensively includes a variety of important techniques.[1] Following are the primary objectives:

1. Create model(s) for threat/intrusion detection
2. Train the model(s) using relevant data sets
3. Measure the performance of the model(s).
4. Enhance the system until it reaches an accepted accuracy rate.

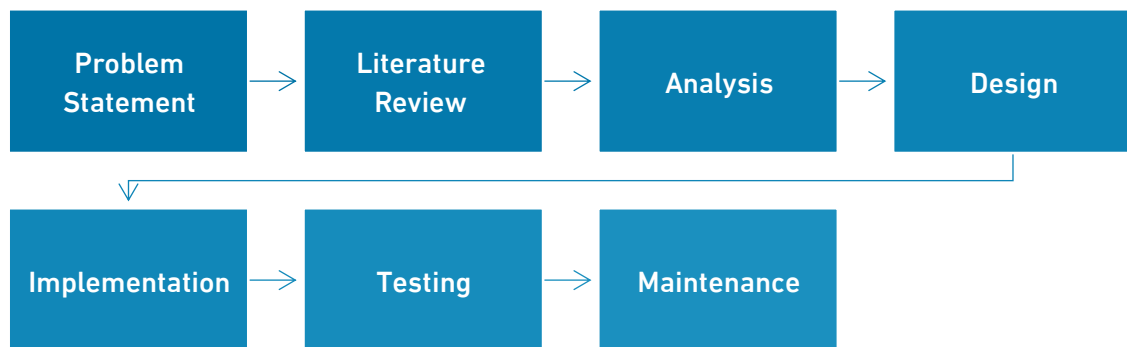
## **1.4 SCOPE:**

- Detecting persons(s) holds a gun, trifle, or knife from a camera video stream.
- The system will be in the form of web/desktop application
- The system will give an alarm in case of detection a threat with the mentioned properties.

## 1.5 METHODOLOGY:

A methodology is “a system that sets guidelines for solving a problem, with components including phases, tasks, methods, methods, and tools.” depending on the requirements and the project type.

In this system we have looked carefully for the requirements, and we have decided that the best approach to develop the system is using the Waterfall Methodology. We believe that using this approach can help us reach our goals and deliver the best quality.



**Figure 1-1: Project Methodology Model (Waterfall)**

In next lines, there is a brief about the Waterfall Methodology.

### 1.5.1 Waterfall:

The waterfall Methodology is “a design process model that used in software development processes commonly.” The workflow progress is in fixed form of pieces that start from top to down through these stages: The process includes the following steps: planning, analysis, design, construction, testing, production, implementation, and maintenance.

## 1.6 TIMELINE:

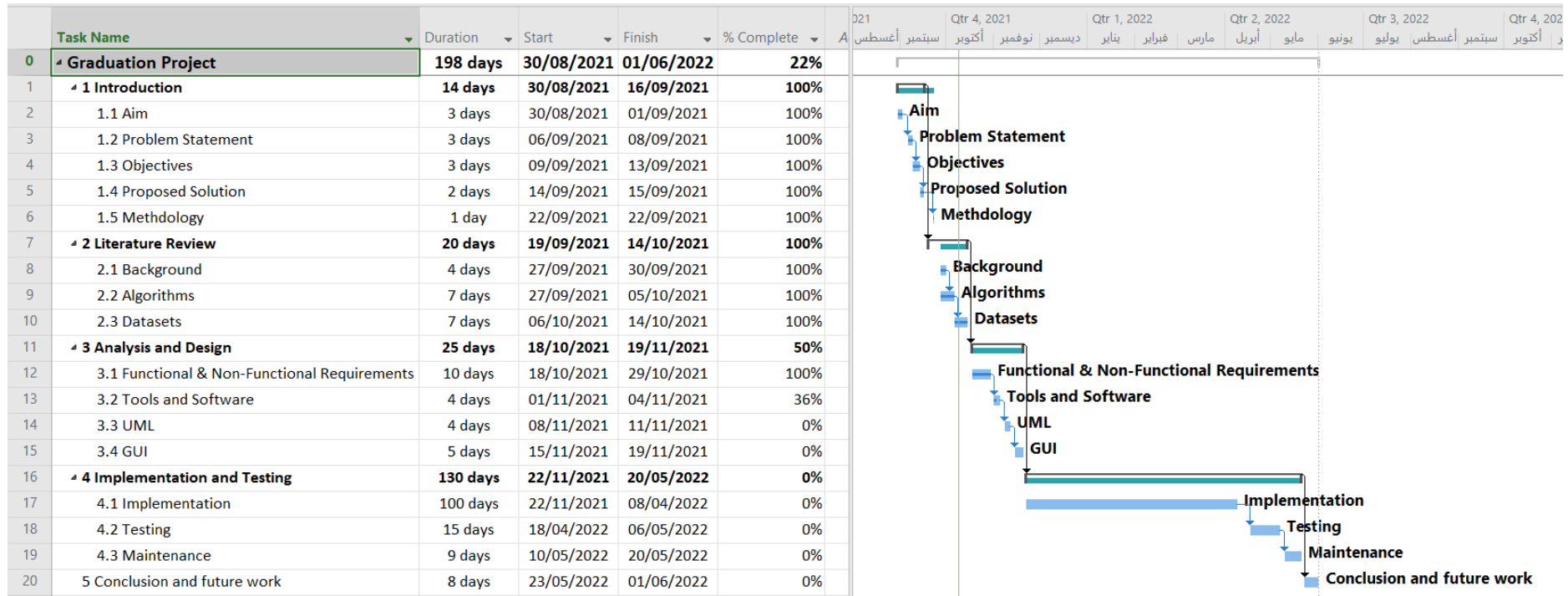


Figure 1-2: Giant Chart for Project

## **2 LITERATURE REVIEW**

Object detection is a computer vision task that detects instances of visual objects of specific classes (such as persons, animals, cars, or buildings) in digital pictures like photos or video frames. The purpose of object detection is to create computational models that give computer vision applications the most basic information they need, in our case, we will study it and we compare a set of algorithms trying to produce the best one or the one that works well on our idea

### **2.1 BACKGROUND**

Object recognition is a wide term that refers to a set of related computer vision tasks that include identifying objects in digital pictures. Image classification is defined as “predicting the class of one item in a picture.” Object localization is the process of determining the position of one or more things in a photograph and drawing a bounding box around their extent. Object detection combines these two tasks by identifying and categorizing one or more objects in a picture. Therefore, three distinct computer vision tasks may be identified:

#### **2.1.1 Image Classification**

Predict the type or class of an object based on a photograph.[2] As input, a single-object image, such as a photograph, is utilized. As a consequence, a class label is created (e.g., one or more integers that are mapped to class labels).[3]

#### **2.1.2 Object Localization**

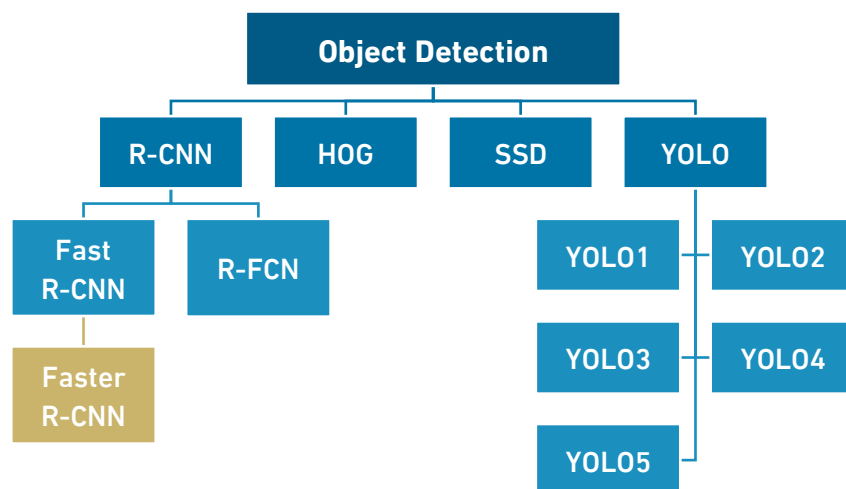
Determine the presence of items in a photograph and use a bounding box to pinpoint their location. As input, an image with one or more things, such as a photograph, is utilized. As an output, one or more bounding boxes (e.g., defined by a point, width, and height).

### 2.1.3 Object Detection

Using a bounding box, determine the presence of things in an image and the types or classes of the objects discovered.[3] As input, an image with one or more things, such as a photograph, is utilized. One or more bounding boxes (e.g., specified by a point, width, and height) are produced, together with a class label for each bounding box.

To understand more about Object detection what and what are the most famous types and how they are used Object detection is a computer vision task that detects instances of visual objects of specific classes (such as persons, animals, cars, or buildings) in digital pictures like photos or video frames. The purpose of object detection is to create computational models that give computer vision applications the most basic information they need, in our case, we will study it and we compare a set of algorithms trying to produce the best one or the one that works well on our idea.

In the upcoming pictures, best types of object detection help in a clear picture output, the difference between them, and their speed of identification and exploration.



**Figure 2-1: Object Detection Algorithms series**

## 2.2 ALGORITHMS

**Table 2-1: Summary of Algorithms**

| Algorithms   | Definition   | Creation |
|--|--|----------|
| Histogram of Oriented Gradients (HOG)              | Canny Edge Detector and SIFT are examples of feature descriptors (Scale Invariant and Feature Transform). It is used in computer vision and image processing for object detection. The method counts how many times a gradient orientation appears in a certain region of a picture. | 1986     |
| Single Shot Detector (SSD)                         | Detect several objects inside a picture using a single shot.   | 2016     |
| Region-based Fully Convolutional Network (R-FCN)   | Reduces the amount of work necessary for each ROI, which speeds up the process.  | 2016     |
| Fast R-CNN   | The CNN is fed the input picture, which creates a convolutional feature map. As a starting point, use the convolutional feature map.   | 2015     |
| Faster R-CNN                                       | deep convolutional network for object detection that appears to the user as a single, end-to-end, unified network.   | 2015     |
| Region-based Convolutional Neural Networks (R-CNN) | Combination of region recommendation and Convolutional Neural Networks (CNNs)  | 2014     |
| You Only Look Once (YOLO)                          | A single neural network is used in an object detection method. In contrast to several other object detection algorithms that do a bit-by-bit scan of the picture.  | 2016     |

In the next lines, we are going to have a brief for each algorithm.



### **2.2.1 Histogram of Oriented Gradients (HOG)**

In 2005, Navneet Dalal and Bill Triggs presented highlights from Histogram of Oriented Gradients (HOG). The Histogram of Arranged Inclinations (Hoard) is a component descriptor used in image processing, mostly for object detection. A representation of an image or a picture repair that enhances the picture by isolating useful data from it is called an element descriptor.[4], [5]

The histogram of organized inclinations descriptor is based on the idea that the dispersion of force angles or edge bearings might represent the look and form of nearby objects inside an image. Because the extent of angles is limited, the x and y subsidiaries of an image (Inclinations) are useful. Because of the abrupt shift in power near edges and corners, is enormous, and we understand that edges and corners pack in far more info about object form than level locations. As a result, the histograms of inclination headings are used as items in this descriptor.

#### **Object detection workflow with HOG**

We will go on to how we compute the histograms and how the element vectors obtained from the Hoard descriptor are used by the classifier such as SVM to identify the concerned article now that we understand the fundamental idea of Histogram of Oriented Slopes.

#### **How does it work?**

Normalizing the image is part of the preprocessing process, but it is entirely optional. Its purpose is to improve the Hoard descriptor's execution. We do not use any preprocessing standards because we are only creating a simple descriptor.

### 2.2.2 Single Shot Detector (SSD)

The Single Shot Detector (SSD) is an engineering improvement for the VGG16 that calculates article discovery. It was released at the end of November 2016 and achieved new standards for object identification accuracy and execution, reaching over 74% Guide (mean Normal Accuracy) at 59 edges per second on common datasets such as PascalVOC and COCO. [6], [7]

#### Architecture

The SSD architecture builds on the well-known VGG-16 engineering but eliminates all the layers that go with it.

VGG-16 was chosen as the basic organization for the following reasons: a reputation for excellent accomplishment in top-notch picture order endeavors situations in which move learning aids in the development of results Instead of the initial VGG fully associated layers, several auxiliary convolutional layers (from conv6 onwards) were introduced, allowing for the separation of components at different scales and the logical fall in the amount of the contribution to each resultant layer.

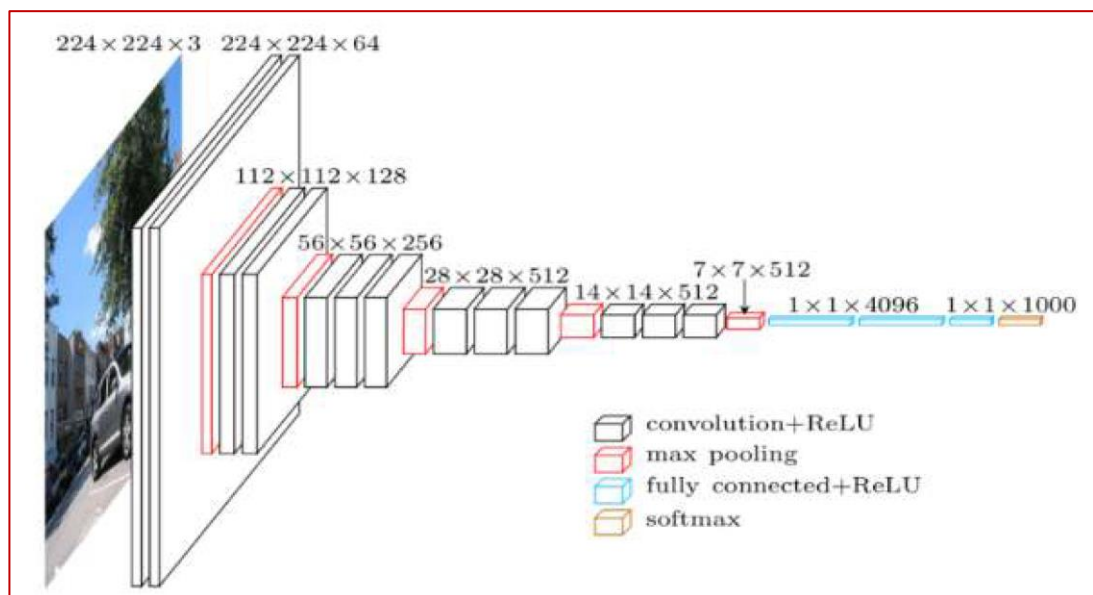
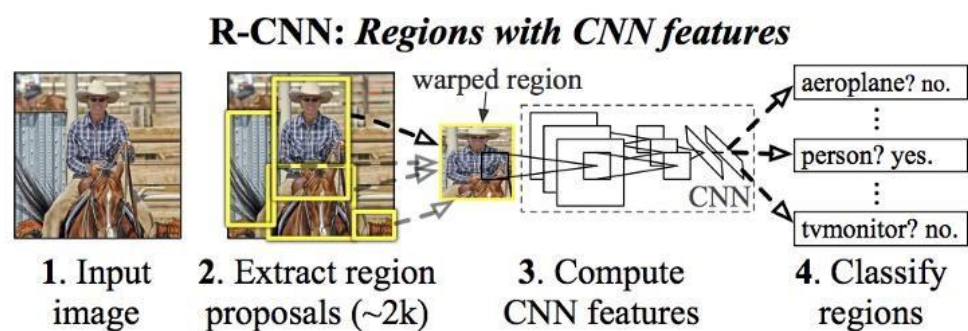


Figure 2-2: VGG-16 Architecture

### 2.2.3 Region-based Convolutional Neural Networks (R-CNN)

The Region-based Convolutional Network procedure (R-CNN) is a blend of district proposal with Convolution Neural Networks (CNNs). R-CNN helps in confining articles with a huge affiliation and drawing up a high-line model with a smidgen of proportion of explained region information. It accomplishes shocking article conspicuous verification exactness by utilizing a critical ConvNet to organize object proposition. R-CNN can scale to a significant number of thing classes without going to procedures, including hashing.[4]–[8]



**Figure 2-3: R-CNN system overview**

1. Takes an image as input.
2. Input image as input.
3. Uses a massive convolutional neural network to compute features for each proposal (CNN).
4. Classify each area using class specific linear SVMs.

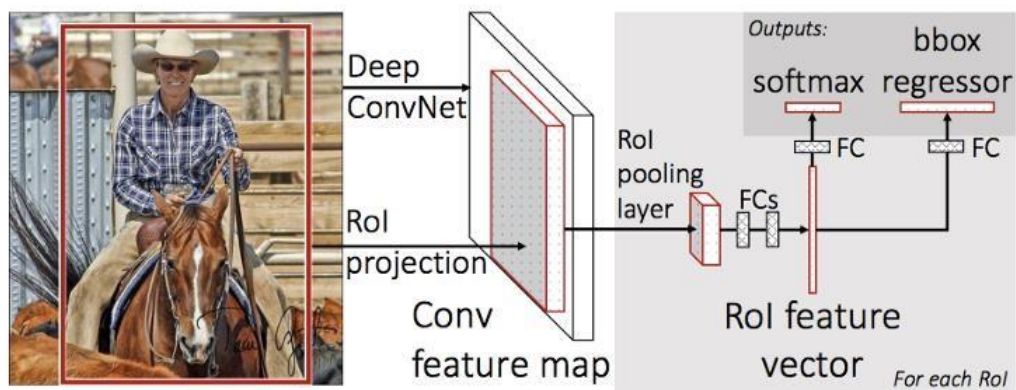
#### **Drawbacks**

- It finds Regions of Interest using the Selective Search Algorithm, which is a slow and time-consuming
- procedure that requires each image is used to categorize 2000 region proposals. As a result, training takes a long time. Detecting objects in a picture on GPU takes 49 seconds.
- A large amount of disk space is also required to store the region proposal's feature map.

## 2.2.4 Fast R-CNN

Fast R-CNN was a similar producer to the previous paper (R-CNN) that addressed a few problems of R-CNN to build a quicker article affirmation computation. The method looks to be like the R-CNN assessment. In any case, rather than controlling the region notion using CNN, we feed the information image to CNN to send a convolutional include map. We take the region of suggestion from the convolutional highlight map and curve it into squares, then reshape it into an acceptable size using a RoI pooling layer, so it can be managed into a completely relevant overall image.[8]–[13]

Based on the RoI highlight vector, we utilize a delicate max layer to forecast the class of the proposed zone and the offset respects for the jumping box. The reason (Fast R-CNN) is faster than R-CNN is because you do not have to manage 2000 locale concepts to the convolutional neural connection when in doubt. Taking everything into consideration, the convolution activity is done just once per picture, resulting in the generation of a section map.



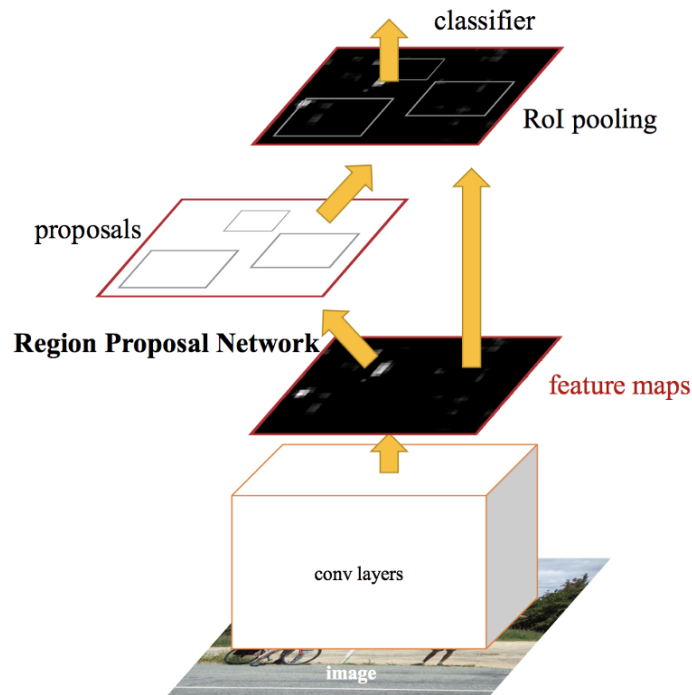
**Figure 2-4: Fast R-CNN system overview**

### Benefits

- Higher identification quality (mAP) than R-CNN, SPPnet
- Training is single stage, utilizing a perform multiple tasks misfortune
- Training can refresh all organization layers
- No plate stockpiling is needed for include storing

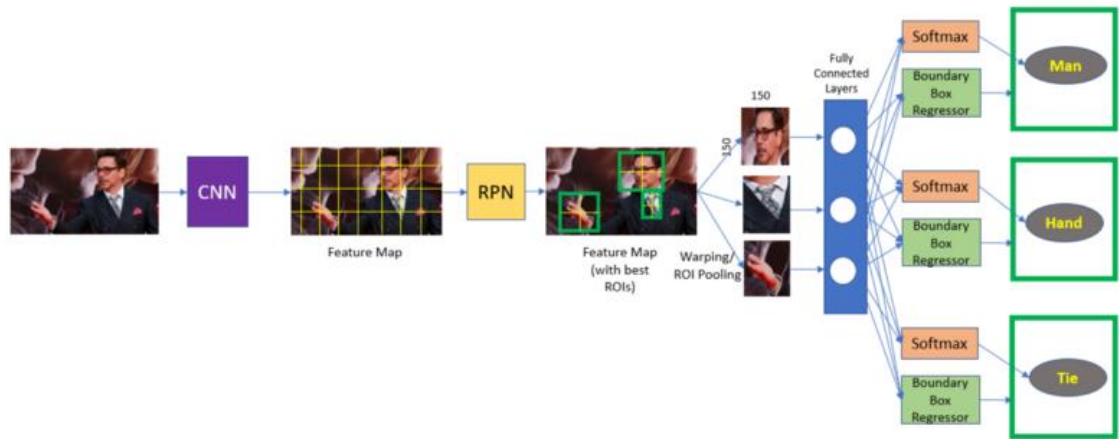
### 2.2.5 Faster R-CNN

Both preceding algorithms (R-CNN and Fast R-CNN) employ requests to find area suggestions. Specific request is a torpid and monotonous cycle, impacting the introduction of the association. Accordingly, Shaoqing Ren, thought about a thing area computation that wipes out the pursuit estimation and permits the association to acquire capability with the region suggestion.[8]–[11], [14]–[16]



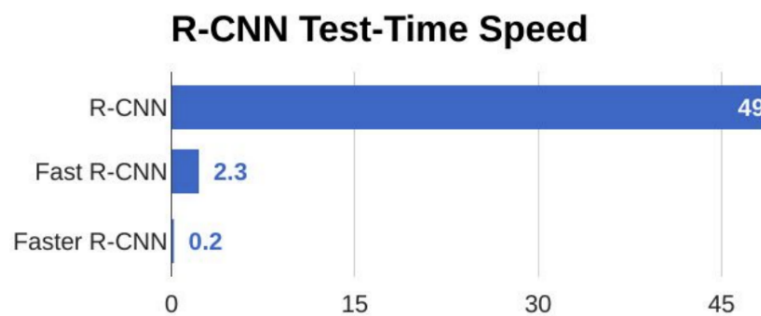
**Figure 2-5: Faster R-CNN system overview**

The picture is supplied as a commitment to a convolutional network that produces a convolutional incorporate guide, like Fast R-CNN. An alternate association is utilized to anticipate the district suggestions by utilizing special request calculation on the part manual for recognizing the region suggestion. A RoI pooling layer is then utilized to characterize the picture inside the suggested region and predict the offset respect for the skipping boxes, reshaping the expected region recommendations.



**Figure 2-6: Faster R-CNN system analysis**

Instead of Selective Search algorithm, it uses RPN (Region Proposal Network) to select the best ROIs automatically to be passed for ROI Pooling.



**Figure 2-7: Comparison of test-time speed of object detection algorithms**

Quicker R-CNN is clearly faster than its predecessors, as shown in the graph above. As a result, it may even be used to identify objects in real time.

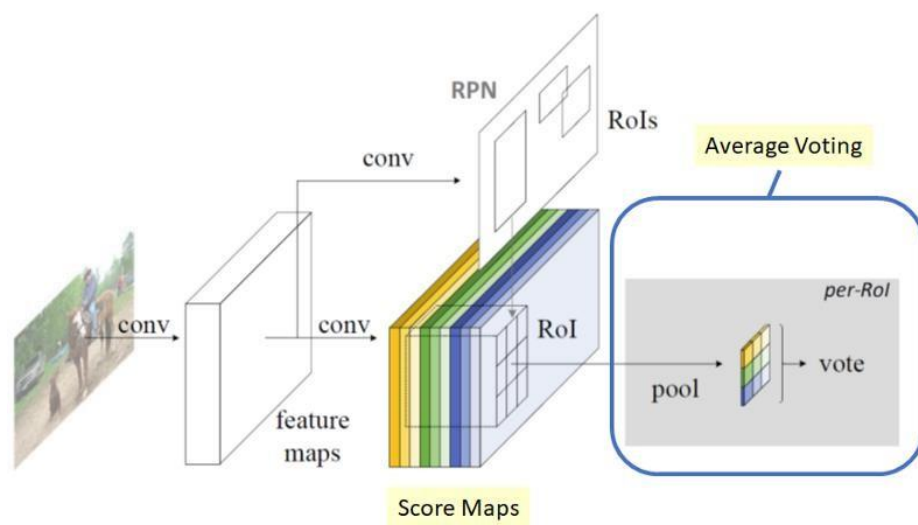
## 2.2.6 Region-based Fully Convolutional Network (R-FCN)

District based Fully Convolutional Networks or R-FCN is an area-based locator for object identification. Not at all like other district-based locators that apply an exorbitant per-locale subnetwork like Fast R-CNN or Faster R-CNN, this area-based indicator is completely convolutional with all calculation shared on the whole picture.

R-FCN comprises of shared, completely convolutional designs just like the instance of FCN that is known to yield a preferred outcome over the Faster R-CNN. In this

calculation, all learnable weight layers are convolutional and are intended to characterize the ROIs into object classifications and foundations.[7]

For conventional Region Proposal Network (RPN) approaches like R-CNN, Fast RCNN and Faster R-CNN, region suggestions are delivered by RPN first. Then ROI pooling is done and going through totally related (FC) layers for portrayal and bobbing box backslide. The collaboration (FC layers) after ROI pooling does not split among ROI, and saves time, which makes RPN approaches slow. Also, the FC layers increase the number of affiliations (limits) which moreover increase the multifaceted nature.



**Figure 2-8: R-FCN system overview**

In R-FCN, we use RPN to generate district recommendations, however unlike the R-CNN series, FC layers are removed after ROI pooling. All substantial complexity is shifted before ROI pooling to create the score maps, assuming all other factors are identical. Following ROI pooling, every district proposal will use a same set of score guidelines to conduct normal democratic, which is a fundamental estimation. As a result, there is no learnable layer after the ROI layer, which is costless. R-FCN is faster than Faster R-CNN with smaller mAP as a result.

### 2.2.7 You Just Look Once (YOLO)

A convincing continuous article acknowledgement calculation, originally described in Joseph Redmon et al.'s 2015 study. In this post, we will go through the concept of item location, the YOLO calculation, and one of the open-source implementations of the computation: Darknet. Read our explanation of why PP-YOLO (or PaddlePaddle YOLO) is faster than YOLOv4 to learn more about PP-YOLO (or PaddlePaddle YOLO), which is an improvement for YOLOv4.[17]–[20]

One of the many exciting applications of convolutional neural networks is picture grouping. Aside from simple picture grouping, there are many exciting difficulties in PC vision, with object identification being one of the most intriguing. It is most associated with self-driving cars, where systems combine PC vision, LIDAR, and other technologies to generate a multidimensional representation of the street with all its inhabitants. Video reconnaissance frequently makes use of article discovery as well.

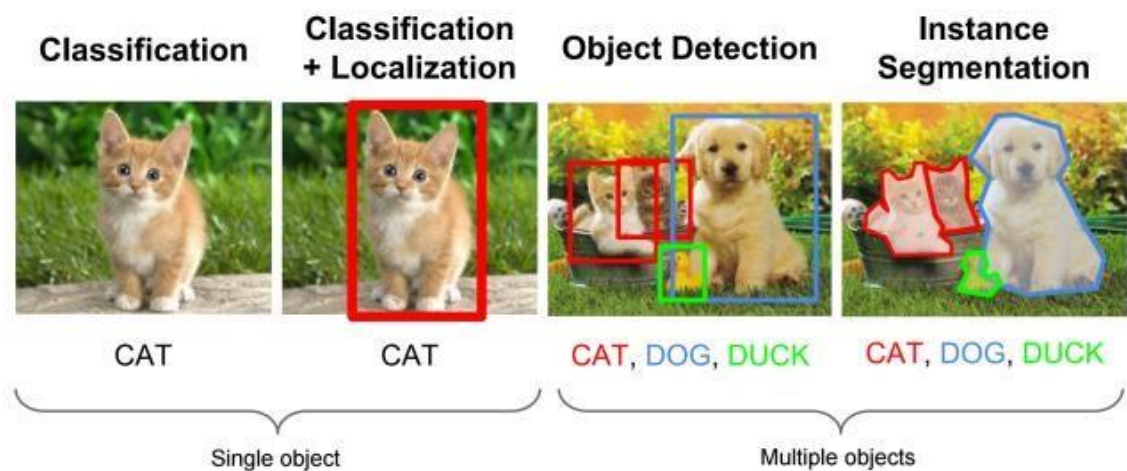
Especially in swarm checking to prevent psychological militant attacks, counting people for general metrics, or analyzing customer experience with mall walking paths. YOLO ("You Only Look Once") is a powerful continuous article acknowledgment calculation initially described by Joseph Redmon et al. in a seminal 2015 work. In this post, we will go through the concept of item location, the YOLO calculation, and one of the open-source implementations of the computation: Darknet. Read our explanation of why PP-YOLO (or PaddlePaddle YOLO) is faster than YOLOv4 to learn more about PP-YOLO (or PaddlePaddle YOLO), which is an upgrade for YOLOv4.

Article discovery is also commonly used in video reconnaissance, particularly in swarm checking to prevent psychological militant attacks, count people for general metrics, or evaluate customer experience with walking paths inside malls.

It is beneficial in the first-place picture grouping to study the idea of article discovery. The process of grouping pictures develops in complexity. Picture grouping (1) aims to classify a photograph into one of several categories (for example, automobile, canine, feline, human, and so on), answering the question "What is in this image?" A single class has been assigned to one image. Article confinement (2) then allows us to locate our item



in the image, changing our question to "What is it and where is it?". In a real circumstance, we would need to look for multiple items in a single image rather than just one. For example, a self-driving vehicle must discover the location of other vehicles, traffic signals, signs, and pedestrians, and then make an appropriate move based on this information. Article discovery (3) provides the tools to accomplish just that - locate all the things in a photograph and draw the bounding boxes around them. There are a few instances where we need to determine the exact limits of our articles via the process known as example division, but that is a topic for another essay.



**Figure 2-9: Single vs. Multiple object (YOLO)**

## 2.2.8 Comparisons

While YOLOv3 is quite high and far to the left, you can know it is well. Is it possible for you to cite your own work? Guess who is going to give it a shot, this man. Oh, and we also fixed a data loading error in YOLOv2, which saved us about 2 mAP. I'm just slipping this in here to keep the layout from being thrown off.[21], [22]

**Table 2-2: Algorithms Comparison (Speed)**

| Algorithm  | mAP  | ms  |
|------------|------|-----|
| YOLOv2     | 21.6 | 25  |
| R-FCN      | 29,9 | 85  |
| SSD513     | 31.2 | 125 |
| FPN FRCN   | 36.2 | 172 |
| YOLOv3-329 | 28.2 | 22  |
| YOLOv3-416 | 31   | 29  |
| YOLOv3-608 | 33   | 51  |

While 63.4 mAP (mean average prediction) and 45 FPS in YOLO mode. YOLO can achieve real-time performance with similar mAP as R-CNN, Fast R-CNN, and Faster R-CNN. After we know all the algorithms and study their properties, we will do a comparison between them, which one is faster and takes more evidence.[23]

**Table 2-3 Algorithms Comparison (Performance)**

| Algorithm       | mAP  | FPS |
|-----------------|------|-----|
| R-CNN           | 53.5 | 6   |
| Fast R-CNN      | 70   | 0.5 |
| Faster R-CNN    | 73.2 | 7   |
| Faster R-CNN ZF | 62.1 | 18  |
| YOLO VGG-16     | 66.4 | 21  |

### 2.2.9 Summary

We learned a light introduction to the topic of object identification in this post, as well as state-of-the-art deep learning models meant to solve it.

Also compare these algorithms to see which one is better

- The phrase "object recognition" refers to a set of actions that are used to recognize objects in digital pictures.
- HOG (Histogram of Oriented Gradients), like the Canny Edge Detector and SIFT, is a feature descriptor (Scale Invariant and Feature Transform).
- Single Shot Detector (SSD) To detect several items within an image, just one shot is required.
- "Region-Based Convolutional Neural Networks", are a class of approaches for tackling object identification and localization problems that are optimized for model performance.
- "YOLO", group of object identification algorithms that are geared for speed and real-time application.

Finally, we can say that the difference between the two most important methods of detecting that a group R-CNN is characterized by superior performance and a group YOLO characterized by real-time object detection

## **2.3 DATASETS**

As mentioned in the previous chapter, we searched for open-source datasets and filtered them to the ones in this report.

### **2.3.1 Crime Detection – using Deep learning**

This project used YOLO Darknet framework. Project's Datasets is about Crime, and it is manually obtained from Google photos as well as the ImageNet database for crime detection. It has 3150 photos, txt files. It's categories: Gun, Knife, Person. [24], [25]

### **2.3.2 Weapon detection datasets**

These datasets used converted YOLOV5 to practice. Project's Datasets is Primarily concerned with the development of intelligent video surveillance automatic systems. [26], It is from: Andalusian Research Institute in Data Science and Computational Intelligence (DaSCI). It has 18097 photos, txt files. It's categories: Pistol, Knife, Weapons, and similar handled objects [27]

### **2.3.3 Weapon Detection and Classification**

These datasets used Deep Learning CNNs to practice. Weapon Detection & Classification through CCTV surveillance. It has 13215 photos. It's categories: Knife, Small Gun.[28]

### **2.3.4 Handgun Dataset**

This dataset was used on the paper titled "Firearm Detection from Surveillance Cameras Using Image Processing and Machine Learning Techniques." It has Consists of positive (Handgun) Images and negative images (Images of various objects), in total 1900 photos. It's categories: Gun. [29], [30]

### **2.3.5 Knife Dataset**

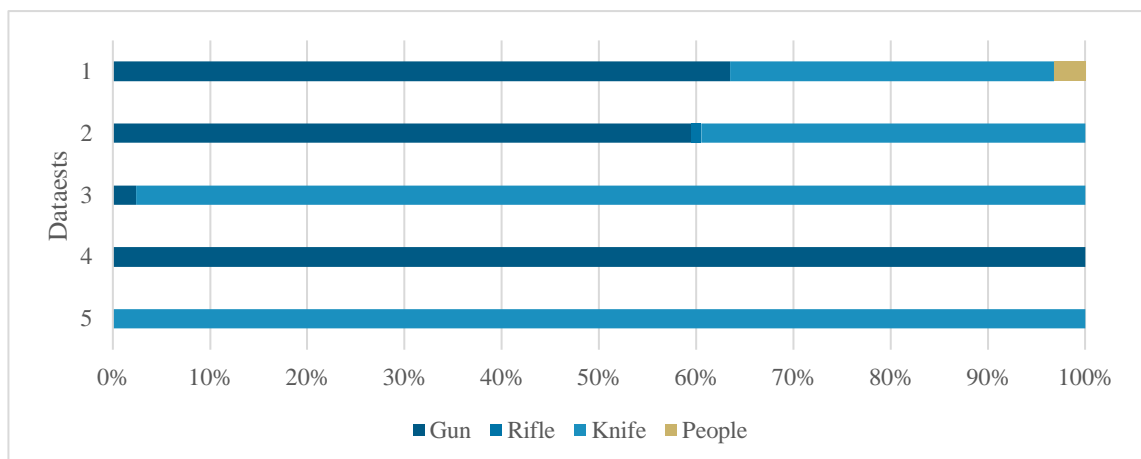
There are 400 training photos and 100 test images of knives in the dataset. It has 500 photos. It's categories: knife [31]

### 2.3.6 Summary

**Table 2-4: Summary of Datasets**

| Dataset                  | Gun   | Rifle | Knife | People | Total |
|--------------------------|-------|-------|-------|--------|-------|
| 1. Crime Detection       | 2000  | -     | 1050  | 100    | 3150  |
| 2. Weapon detection      | 10770 | 187   | 7140  | -      | 18097 |
| 3. Weapon Classification | 315   | -     | 12900 | -      | 13215 |
| 4. Handgun Dataset       | 1900  | -     | -     | -      | 1900  |
| 5. Knife Dataset         | -     | -     | 500   | -      | 500   |
| Total Datasets           |       |       |       |        | 36862 |

Below, there is a percentage representation for each category in each dataset



**Figure 2-10: Summary of Datasets (Data Type %)**

## **3 ANALYSIS AND DESIGN**

### **3.1 FUNCTION REQUIREMENT**

1. Login
  - To allow the user to enter the system and browse the camera, a login must be made to the system
2. Create account
  - If the user is new, you must create an account for the system to allow that to log in to the system
3. Identify the person holding the gun
  - One of the most crucial functions of the system is how to identify any person carrying a gun by taking a picture of him and determining the type of detection
4. Identify the person holding the knives
  - In many cases, the criminal is carrying our knife and passing through many monitoring devices, so it is a function in the system to identify anyone who carries a knife for the safety of customers.
5. Identify the person holding the Trifle
  - The type of detection differs, and one of the most important types of guns is that one of the functions of the system is to identify the Trifle, if it is found with the criminal, and show it to the user.
6. Create an alarm if the model knows that there is something that triggers an alarm
  - When the system recognizes the detection, it must send a warning to inform the users of the presence of danger and to take the necessary measures also to inform the customers to immunize themselves
7. Display camera stream
  - The camera content of the system must be displayed to determine the detection of the user who is following the system in action
8. Logout
  - The user must be logged out for the safety of the system if he finished his work or if he will leave the place

### 3.2 NON-FUNCTION REQUIREMENT

1. The system security
  - Protect data from external attacks
2. Visibility all  $24/7$ 
  - Live information that allows us to monitor the possible danger and their exact location at any given point.
3. Accuracy in recognizing object detection
  - Full accuracy in taking out the image or detection clearly
4. Speed in recognizing object detection
  - One of the most key features that must be in the system is that it can quickly detect images to eliminate the danger
5. Preserve the privacy of the place
  - The system must preserve its data and not allow anyone who is not responsible for it or those who have the right to see the system

## REFERENCES

- [1] D. Cao, Z. Chen, and L. Gao, “An improved object detection algorithm based on multi-scaled and deformable convolutional neural networks,” *Hum.-Centric Comput. Inf. Sci.*, vol. 10, no. 1, p. 14, Apr. 2020, doi: 10.1186/s13673-020-00219-9.
- [2] O. Russakovsky *et al.*, “ImageNet Large Scale Visual Recognition Challenge,” *ArXiv14090575 Cs*, Jan. 2015, Accessed: Oct. 10, 2021. [Online]. Available: <http://arxiv.org/abs/1409.0575>
- [3] J. Brownlee, “A Gentle Introduction to Object Recognition With Deep Learning,” *Machine Learning Mastery*, May 21, 2019. <https://machinelearningmastery.com/object-recognition-with-deep-learning/> (accessed Oct. 10, 2021).
- [4] K. Mittal, “A Gentle Introduction Into The Histogram Of Oriented Gradients,” *Analytics Vidhya*, Dec. 21, 2020. <https://medium.com/analytics-vidhya/a-gentle-introduction-into-the-histogram-of-oriented-gradients-fdee9ed8f2aa> (accessed Oct. 10, 2021).
- [5] “Histogram of Oriented Gradients and Object Detection,” *PyImageSearch*, Nov. 10, 2014. <https://www.pyimagesearch.com/2014/11/10/histogram-oriented-gradients-object-detection/> (accessed Oct. 10, 2021).
- [6] “Single Shot Detection (SSD) Algorithm,” *OpenGenus IQ: Computing Expertise & Legacy*, Jan. 21, 2019. <https://iq.opengenus.org/single-shot-detection-ssd-algorithm/> (accessed Oct. 10, 2021).
- [7] “13.7. Single Shot Multibox Detection — Dive into Deep Learning 0.17.0 documentation.” [https://d2l.ai/chapter\\_computer-vision/ssd.html](https://d2l.ai/chapter_computer-vision/ssd.html) (accessed Oct. 10, 2021).
- [8] A. Choudhury, “Top 8 Algorithms For Object Detection,” *Analytics India Magazine*, Jun. 16, 2020. <https://analyticsindiamag.com/top-8-algorithms-for-object-detection/> (accessed Oct. 10, 2021).
- [9] S. Majumder, “Object Detection Algorithms-R CNN vs Fast-R CNN vs Faster-R CNN,” *Analytics Vidhya*, Jul. 04, 2020. <https://medium.com/analytics-vidhya/object-detection-algorithms-r-cnn-vs-fast-r-cnn-vs-faster-r-cnn-3a7bbaad2c4a> (accessed Oct. 10, 2021).
- [10] “R-CNN vs Fast R-CNN vs Faster R-CNN | ML,” *GeeksforGeeks*, Feb. 28, 2020. <https://www.geeksforgeeks.org/r-cnn-vs-fast-r-cnn-vs-faster-r-cnn-ml/> (accessed Oct. 10, 2021).



- [11] R. Gandhi, “R-CNN, Fast R-CNN, Faster R-CNN, YOLO — Object Detection Algorithms,” *Medium*, Jul. 09, 2018. <https://towardsdatascience.com/r-cnn-fast-r-cnn-faster-r-cnn-yolo-object-detection-algorithms-36d53571365e> (accessed Oct. 10, 2021).
- [12] “Object Detection / Fast R-CNN - Convolutional Neural Networks for Image and Video Processing - TUM Wiki.” <https://wiki.tum.de/pages/viewpage.action?pageId=22578448#ObjectDetection/FastRCNN-Weblinks> (accessed Oct. 11, 2021).
- [13] R. Girshick, “Fast R-CNN,” *ArXiv150408083 Cs*, Sep. 2015, Accessed: Oct. 11, 2021. [Online]. Available: <http://arxiv.org/abs/1504.08083>
- [14] S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks,” *ArXiv150601497 Cs*, Jan. 2016, Accessed: Oct. 11, 2021. [Online]. Available: <http://arxiv.org/abs/1506.01497>
- [15] R. Girshick, *py-faster-rcnn has been deprecated. Please see Detectron, which includes an implementation of Mask R-CNN.* 2021. Accessed: Oct. 11, 2021. [Online]. Available: <https://github.com/rbgirshick/py-faster-rcnn>
- [16] “Object Detection / Fast R-CNN - Convolutional Neural Networks for Image and Video Processing - TUM Wiki.” <https://wiki.tum.de/pages/viewpage.action?pageId=22578448#ObjectDetection/FastRCNN-Weblinks> (accessed Oct. 11, 2021).
- [17] A. Singh, “YOLO: You Only Look Once,” *Nerd For Tech*, May 29, 2021. <https://medium.com/nerd-for-tech/yolo-you-only-look-once-65ea86104c51> (accessed Oct. 11, 2021).
- [18] “Papers with Code - You Only Look Once: Unified, Real-Time Object Detection.” <https://paperswithcode.com/paper/you-only-look-once-unified-real-time-object> (accessed Oct. 11, 2021).
- [19] “Publications.” <https://pjreddie.com/publications/> (accessed Oct. 11, 2021).
- [20] J. Redmon and A. Farhadi, “YOLOv3: An Incremental Improvement,” Apr. 2018, Accessed: Oct. 11, 2021. [Online]. Available: <https://arxiv.org/abs/1804.02767v1>
- [21] “Papers with Code - PASCAL VOC 2007 Benchmark (Object Detection).” <https://paperswithcode.com/sota/object-detection-on-pascal-voc-2007> (accessed Oct. 11, 2021).
- [22] “YOLO: Real-Time Object Detection.” <https://pjreddie.com/darknet/yolo/> (accessed Oct. 12, 2021).
- [23] J. Du, “Understanding of Object Detection Based on CNN Family and YOLO,” *J. Phys. Conf. Ser.*, vol. 1004, p. 012029, Apr. 2018, doi: 10.1088/1742-6596/1004/1/012029.

- [24] “Crime-Detection--using-Deep-learning/data at master · sowmyadvn/Crime-Detection--using-Deep-learning,” *GitHub*. <https://github.com/sowmyadvn/Crime-Detection--using-Deep-learning> (accessed Sep. 28, 2021).
- [25] “GitHub - sowmyadvn/Crime-Detection--using-Deep-learning: Developed an object detection model using YOLO Darknet framework to prevent crimes. Collected the crime data manually from Google images as well as Imagenet database.” <https://github.com/sowmyadvn/Crime-Detection--using-Deep-learning> (accessed Sep. 28, 2021).
- [26] *Weapon detection datasets*. ARI-DaSCI, 2021. Accessed: Sep. 28, 2021. [Online]. Available: <https://github.com/ari-dasci/OD-WeaponDetection>
- [27] “Weapon Detection – DaSCI.” <https://dasci.es/transferencia/open-data/24705/> (accessed Sep. 28, 2021).
- [28] “Weapon-Detection-And-Classification/train at master · ivaibhavkr/Weapon-Detection-And-Classification,” *GitHub*. <https://github.com/ivaibhavkr/Weapon-Detection-And-Classification> (accessed Sep. 28, 2021).
- [29] F. Gelana and A. Yadav, “Firearm Detection from Surveillance Cameras Using Image Processing and Machine Learning Techniques,” in *Smart Innovations in Communication and Computational Sciences*, Singapore, 2019, pp. 25–34. doi: 10.1007/978-981-13-2414-7\_3.
- [30] F. Gelana, *Handgun-Dataset*. 2021. Accessed: Sep. 28, 2021. [Online]. Available: <https://github.com/FrexG/Handgun-Dataset>
- [31] “Knife Dataset | Kaggle.” <https://www.kaggle.com/shank885/knife-dataset> (accessed Sep. 28, 2021).