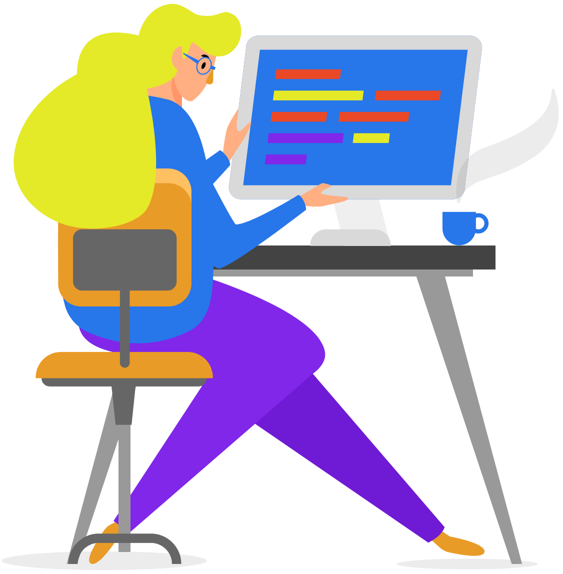
GIDC Degree Engineering College
Managed by GIDC Education Society
Gujarat's Public Private Partnership(PPP) Model Institute

Weapon Detection in Images using YOLOv3 & YOLOv4

Submitted By:
Priya Yadav(201100107524)

Project Guide:
Prof. Archana Naik

INTRODUCTION



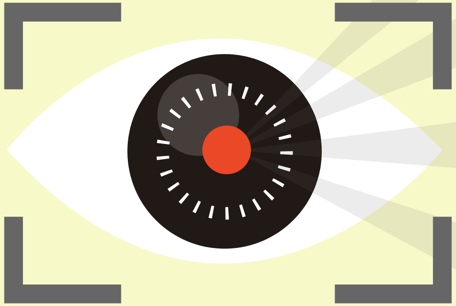
Machine Learning Infographics

The focus of our mini project is to develop a security system which detects weapons like guns, knife, etc . in no time with less computational time and high accuracy.

It is hoped that through the design and implementation of such system, we can come closer to ensuring safety and security of everyday citizens in both public and private level.

PROJECT SCOPE

Weapon Detection



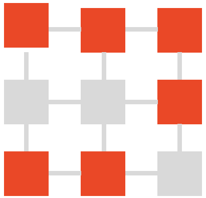
Label objects	We will label objects of weapon
YOLOv3 architecture	YOLO v3 architecture for our gun detection.
YOLOv4 architecture	YOLO v3 architecture for our knife detection
Improving model performance	We will try to get high test set accuracy.
Testing	We will test the model with real world images

PROJECT SCOPE

01

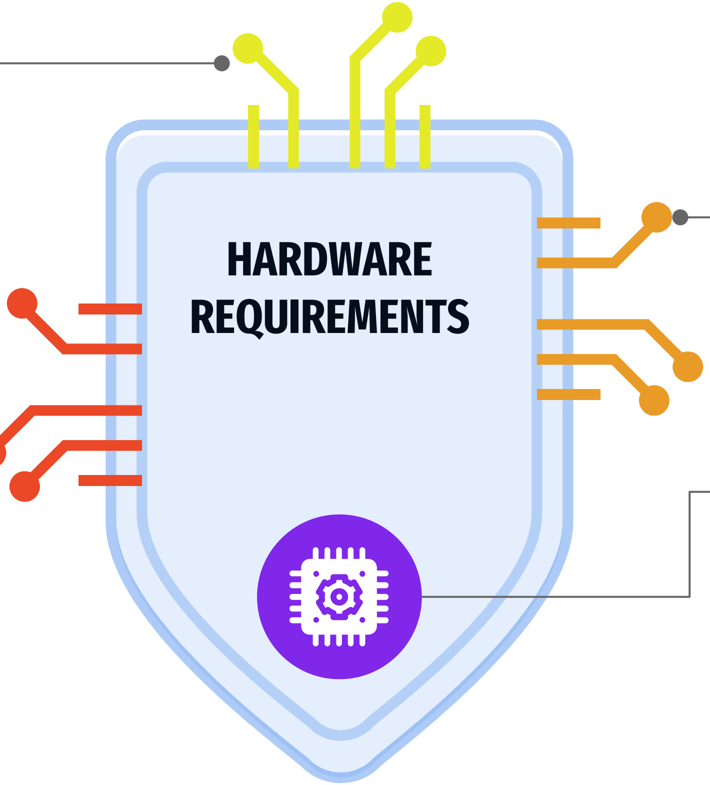
Scope of the project is to build a ML model which can detect weapons from captured images and videos.

02



1) Acquiring images containing weapons and non-weapons.
2) Label the images for weapon detection.
3) Use YOLOv3 architecture to validate our dataset to get high accuracy.
4) Use YOLOv4 architecture to train and validate our dataset to get high accuracy.

PROJECT REQUIREMENTS



01

Google Colab's

GPU machines to train a ML model

02

SOFTWARE REQUIREMENTS

We will use Python and various open source Python based package to train a model

03

ASSUMPTIONS AND CONSTRAINTS

04

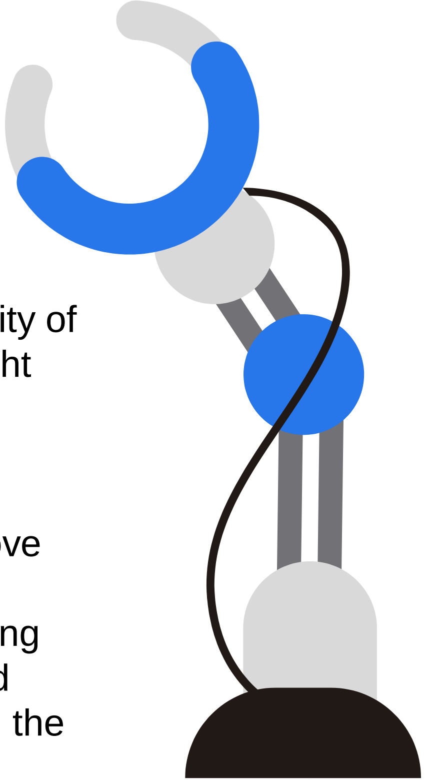
DATASET SPECIFICATIONS

We have used this labeled dataset for detecting weapons in images and videos by using pretrained model

ASSUMPTIONS AND CONSTRAINTS

Weapon Detection

- The gun is in line of sight of camera and fully/partially exposed to the camera.
- There is enough background light to detect the ammunition.
- Generally the model accuracy depends on quality of training data, in real life scenario the model might fail on different lights settings.
- We will target accuracy of a model greater than 85% but real time accuracy might increase or decrease based on training data. We can improve such model by additional training.
- This project requires high performance computing power to train the model, So GPU with high-end computation power were used, to remove lag in the ammunition detection.

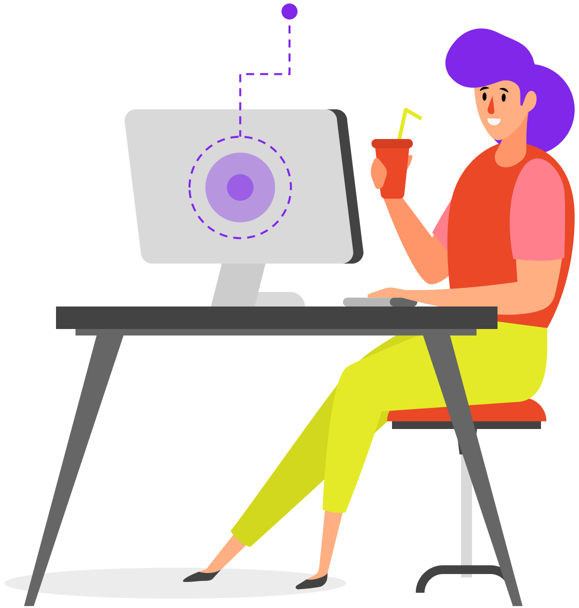


LITERATURE REVIEW

01 **Anuj Mahajan and Simran Gupta**

“Weapon Detection in Video Surveillance using Computer Vision Techniques”,

ML



05 **Michal Grega, Andrzej Matiolanski, Piotr Guzik and Mikolaj Leszczuk**

“Automated Detection of Firearms and Knives in a CCTV Image”,

03 **Dr. N Geetha, Akash Kumar, K.S. Akshita**

“Weapon Detection in Surveillance System”

04 **Tahir Bhatti, Gufran Khan and Masood Aslam**

“ Weapon Detection in Real Time CCTV Videos using Deep Learning”

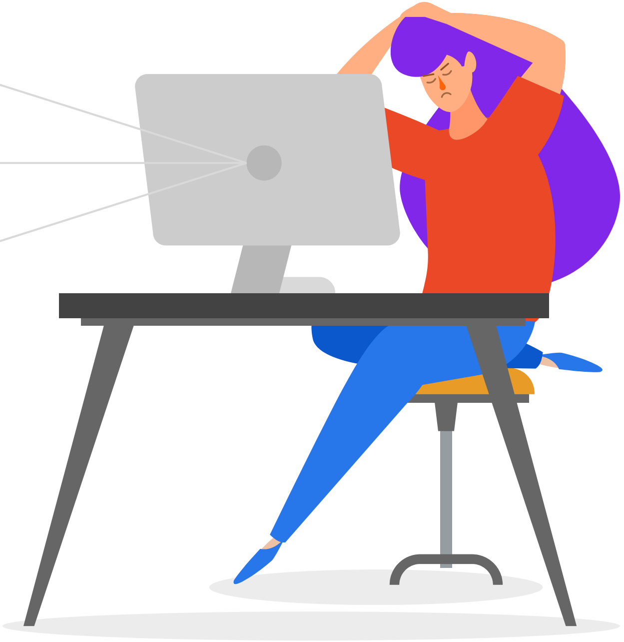
PROJECT FEASIBILITY STUDY

Technical Feasibility

01 Our mini project is Technical feasible because the technologies needed in our project are readily available in market.

02 We have opted Google Colab for storage and good processing speed.

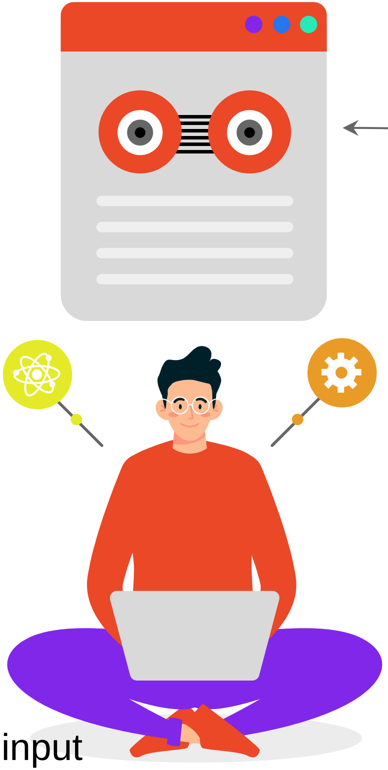
03 We have used Python and its libraries like Jupyter Notebook to run code



Economic Feasibility

Our project is economically feasible because the system is developed by the team of 2 people

The development tools used are free and open source, hence the cost of development only accounts for electricity bills and internet bills for laptops which is very minimum



Operational Feasibility

Our system will take pictures or video as input and detect weapons using ML algorithms

Operational Feasibility

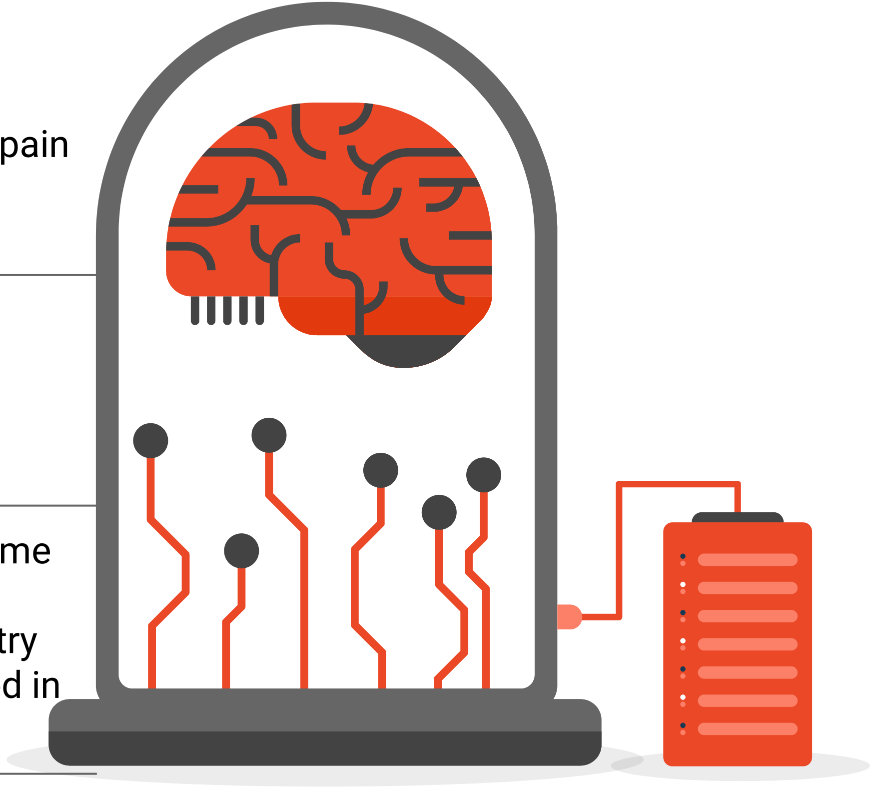
The system will also classify the type of weapons

PROJECT REQUIREMENT GATHERING

01 We need to work with various stockholders in both
private and public services to understand the pain Points

02 There is a long challenge in adoption of new technology and privacy, we will work with stockholders to overcome the same.

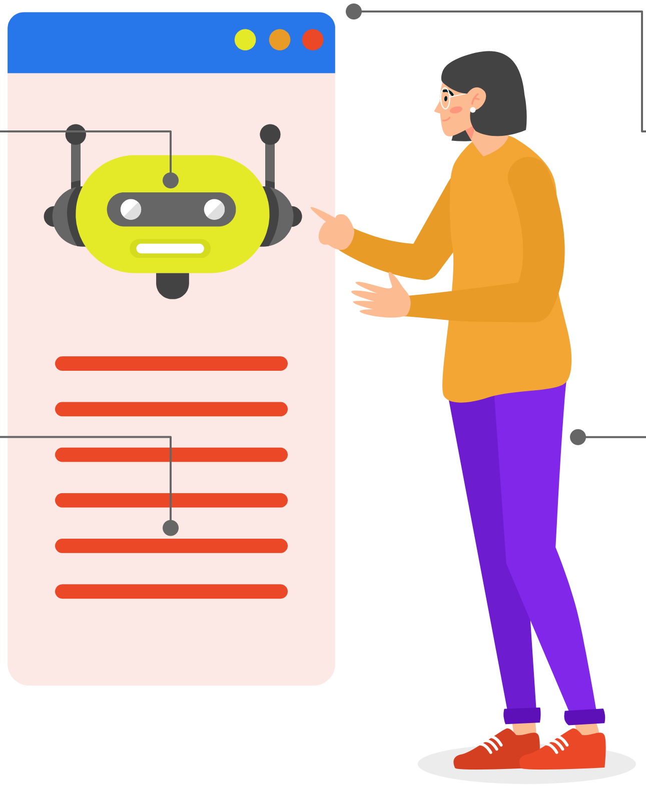
03 To get this project working, we might need some real time setting like Low visibility areas, bad camera positions, bad resolution etc. We will try and procure additional such images to be used in training data



PROJECT SRS

01 USE CASE DIAGRAM

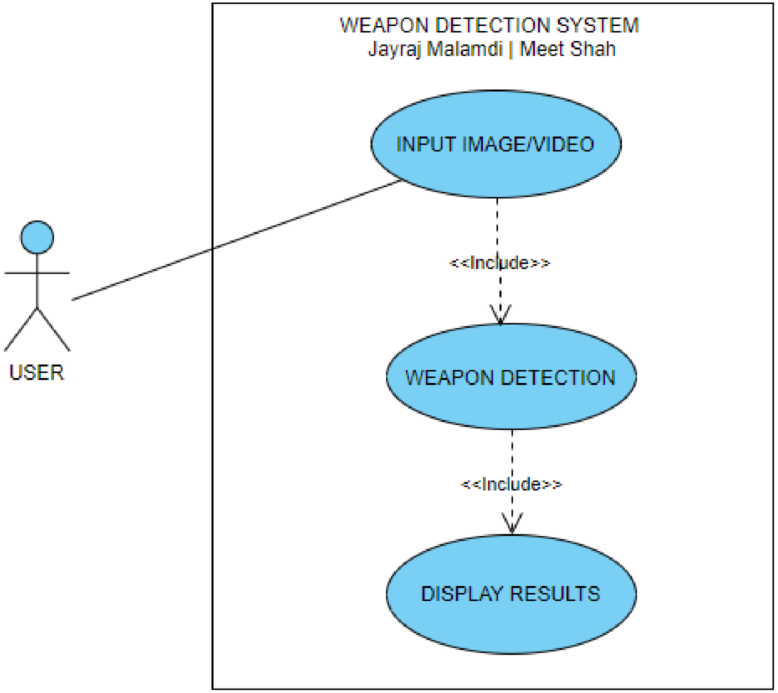
02 YOLOv4 ARCHITECTURE



03 YOLOv3 ARCHITECTURE

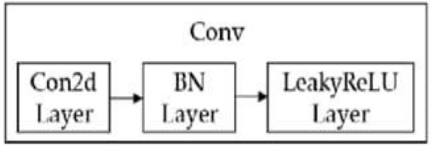
04 MODEL TRAINING FLOW

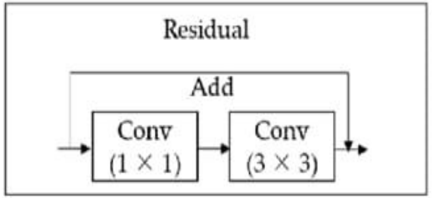
USE CASE DIAGRAM



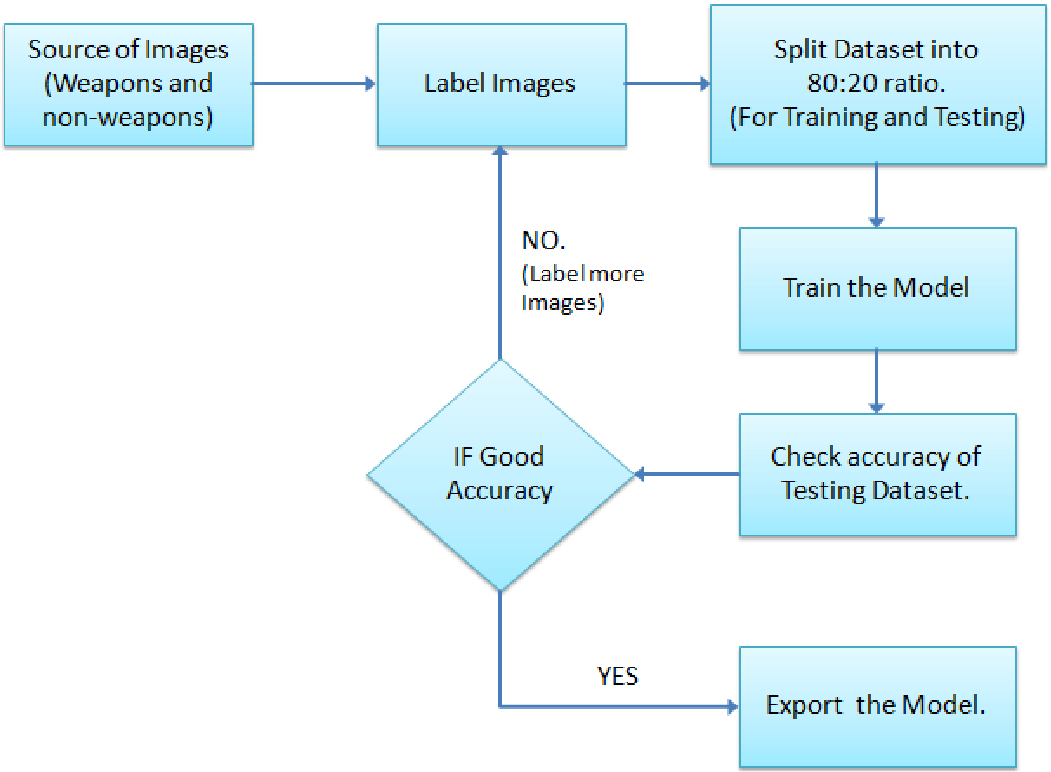
YOLOv3 ARCHITECTURE

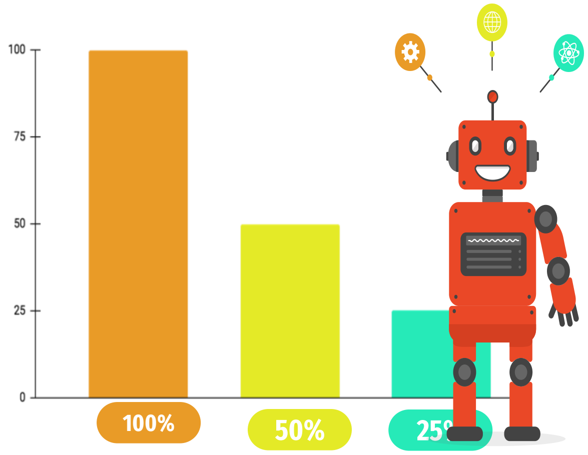
Layer	Filters size	Repeat	Output size
Image			416 × 416
Conv	32 3 × 3/1	1	416 × 416
Conv	64 3 × 3/2	1	208 × 208
Conv Conv Residual	32 1 × 1/1 64 3 × 3/1	Conv Conv Residual × 1	208 × 208 208 × 208 208 × 208
Conv	128 3 × 3/2	1	104 × 104
Conv Conv Residual	64 1 × 1/1 128 3 × 3/1	Conv Conv Residual × 2	104 × 104 104 × 104 104 × 104
Conv	256 3 × 3/2	1	52 × 52
Conv Conv Residual	128 1 × 1/1 256 3 × 3/1	Conv Conv Residual × 8	52 × 52 52 × 52 52 × 52
Conv	512 3 × 3/2	1	26 × 26
Conv Conv Residual	256 1 × 1/1 512 3 × 3/1	Conv Conv Residual × 8	26 × 26 26 × 26 26 × 26
Conv	1024 3 × 3/2	1	13 × 13
Conv Conv Residual	512 1 × 1/1 1024 3 × 3/1	Conv Conv Residual × 4	13 × 13 13 × 13 13 × 13



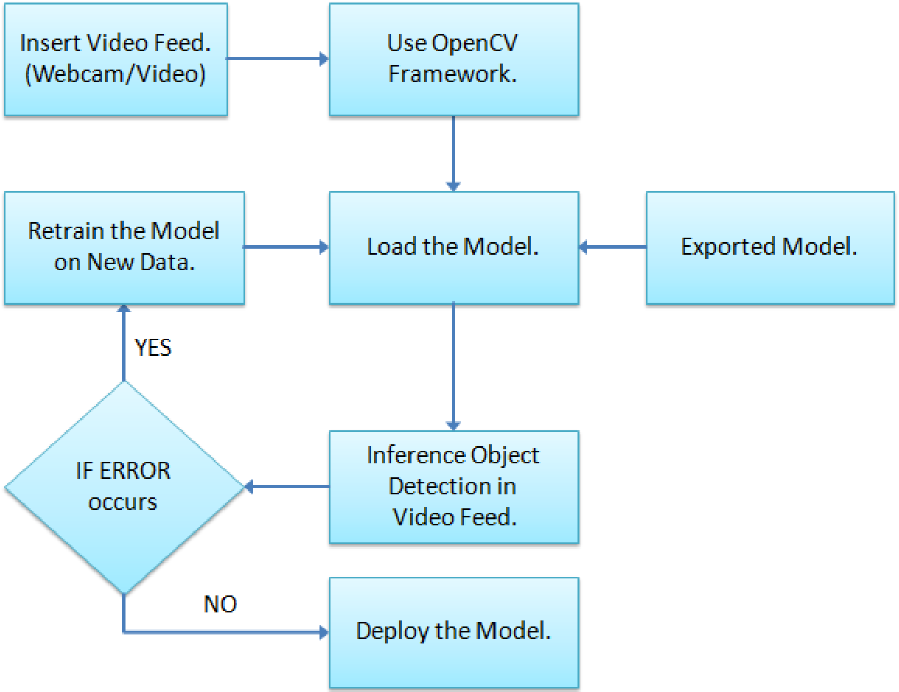


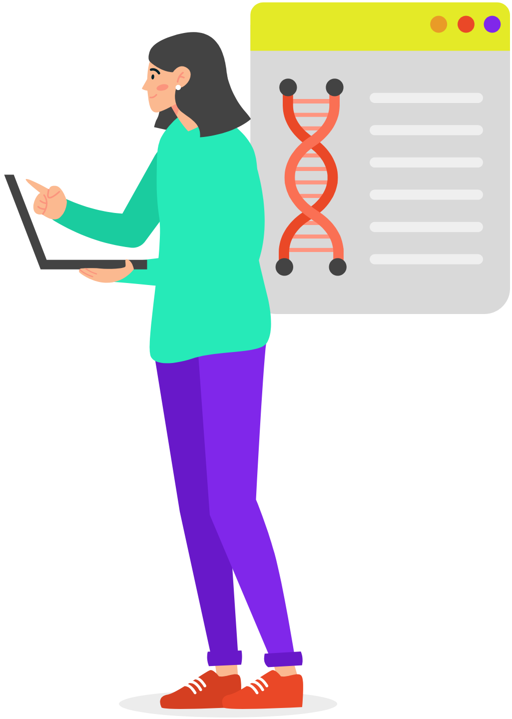
MODEL TRAINING FLOW



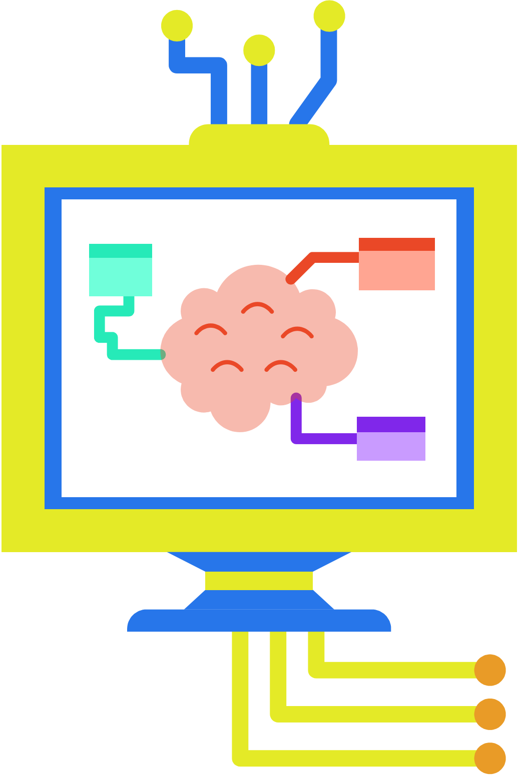


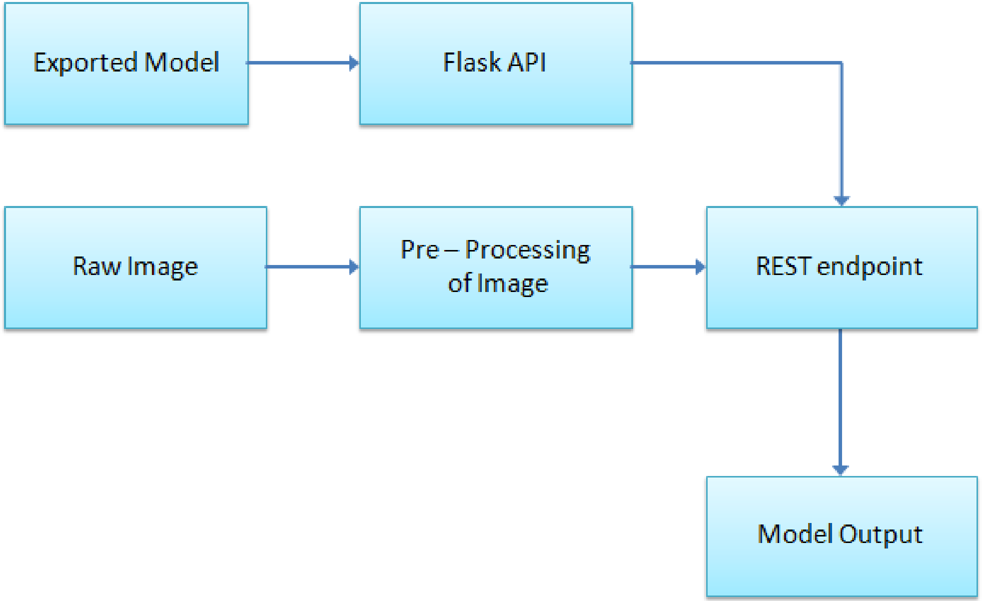
MODEL DEPLOYMENT





MODEL VALIDATION ON IMAGES





PROJECT IMPLEMENTATION FOR GUN DETECTION

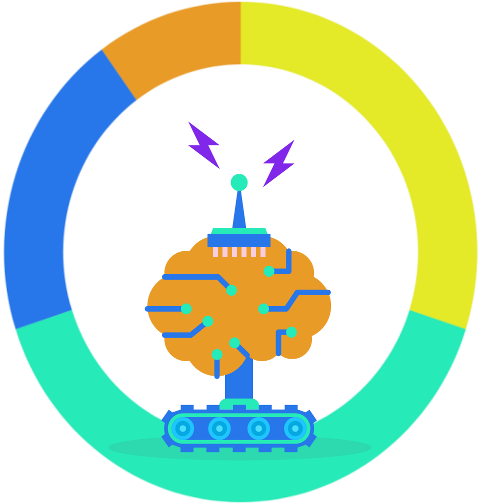
Importing Libraries

```
In [19]: # import libraries

         import numpy as np
         import time
         import cv2
         import os
         import matplotlib.pyplot as plt
         %matplotlib inline
```

Matplotlib for plotting

```
# using matplotlib here for ploting and show image in output

def display_img(img,cmap=None):
    fig = plt.figure(figsize = (12,12))
    plt.axis(False)
    ax = fig.add_subplot(111)
    ax.imshow(img,cmap)
```

Load YOLOv3 architecture & pre-trained weights



```python
labelsPath =(r"C:\Users\DELL\weapon detection inference\yolo_gun.names")
LABELS = open(labelsPath).read().strip().split("\n")
print("----Label to predict---",LABELS)
```

```
----Label to predict--- ['gun']
```

```python
weightsPath = os.path.join(r"C:\Users\DELL\weapon detection inference\yolov3_900.weights")
configPath = os.path.join(r"C:\Users\DELL\weapon detection inference\yolov3_custom_train.cfg.txt")
```

```python
# Loading the neural network framework Darknet (YOLO was created based on this framework)
net = cv2.dnn.readNetFromDarknet(configPath,weightsPath)
```

Detecting & Creating Bounding Box

```
# Create the function which predict the frame input
def predict(image):

    # initialize a list of colors to represent each possible class label
    np.random.seed(42)
    COLORS = np.random.randint(0, 255, size=(len(LABELS), 3), dtype="uint8")
    (H, W) = image.shape[:2]

    # determine only the "ouput" layers name which we need from YOLO
    ln = net.getLayerNames()
    ln = [ln[i - 1] for i in net.getUnconnectedOutLayers()]

    # construct a blob from the input image and then perform a forward pass of the YOLO object detec
    # giving us our bounding boxes and associated probabilities
    blob = cv2.dnn.blobFromImage(image, 1 / 255.0, (416, 416), swapRB=True, crop=False)
    net.setInput(blob)
    layerOutputs = net.forward(ln)

    boxes = []
    confidences = []
    classIDs = []
    threshold = 0.2
```

output

02

REAL TIME CCTV IMAGE

01

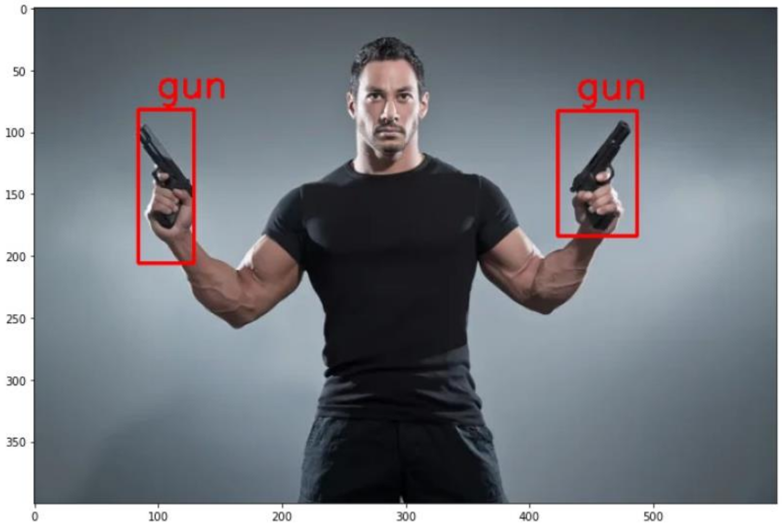
HD IMAGE

03

WEAKNESSES

HD IMAGE

In [29]:
```
# Execute prediction on a single image
img = cv2.imread(r"C:\Users\DELL\weapon detection inference/sample images/sample3.jpg")
#img = cv2.resize(img, (416, 416))
img = cv2.cvtColor(img,cv2.COLOR_BGR2RGB)
display_img(predict(img))
```

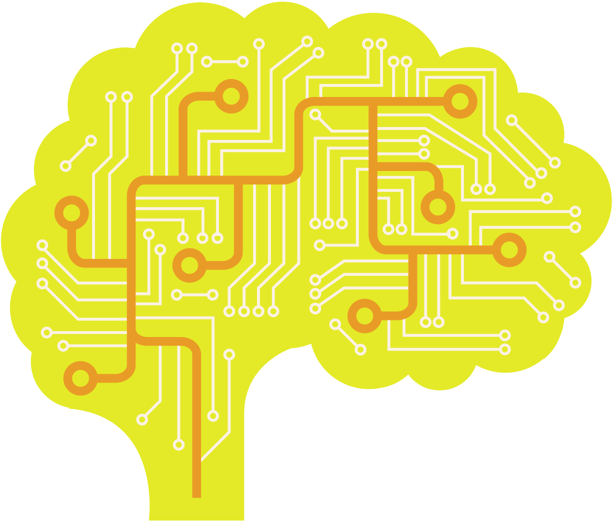


REAL TIME CCTV IMAGE

In [28]:
```
# Execute prediction on a single image
img = cv2.imread(r"C:\Users\DELL\weapon detection inference/cctv_sample1.jpg")
#img = cv2.resize(img, (416, 416))
img = cv2.cvtColor(img,cv2.COLOR_BGR2RGB)
display_img(predict(img))
```

WEAKNESSES

First Weakness



```
In [9]: # Execute prediction on a single image
        img = cv2.imread(r"C:\Users\DELL\weapon detection inference/sample images/sample5.jpg")
        #img = cv2.resize(img, (416, 416))
        img = cv2.cvtColor(img,cv2.COLOR_BGR2RGB)
        display_img(predict(img))
```

WEAKNESSES

Second Weakness

In [30]:
```
# Execute prediction on a single image
img = cv2.imread(r"C:\Users\DELL\weapon detection inference/cctv_sample2.jpg")
#img = cv2.resize(img, (416, 416))
img = cv2.cvtColor(img,cv2.COLOR_BGR2RGB)
display_img(predict(img))
```

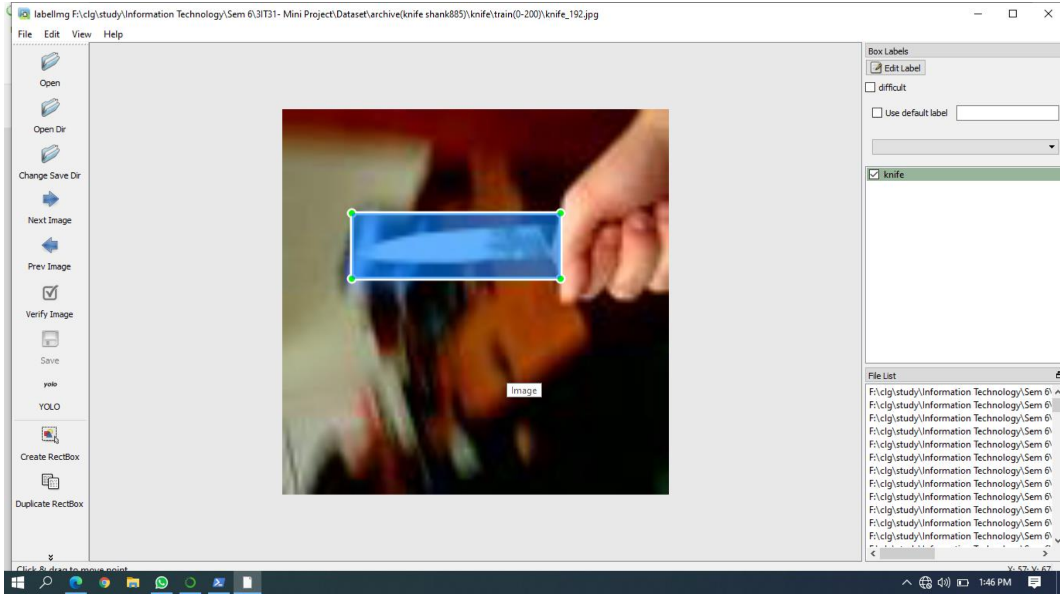
WEAKNESSES

Third Weakness

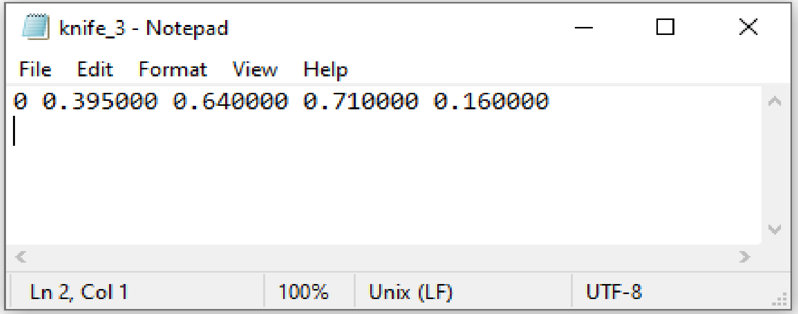
In [10]:
```
# Execute prediction on a single image
img = cv2.imread(r"C:\Users\DELL\weapon detection inference/sample images/sample6.jpg")
#img = cv2.resize(img, (416, 416))
img = cv2.cvtColor(img,cv2.COLOR_BGR2RGB)
display_img(predict(img))
```

PROJECT IMPLEMENTATION FOR KNIFE DETECTION

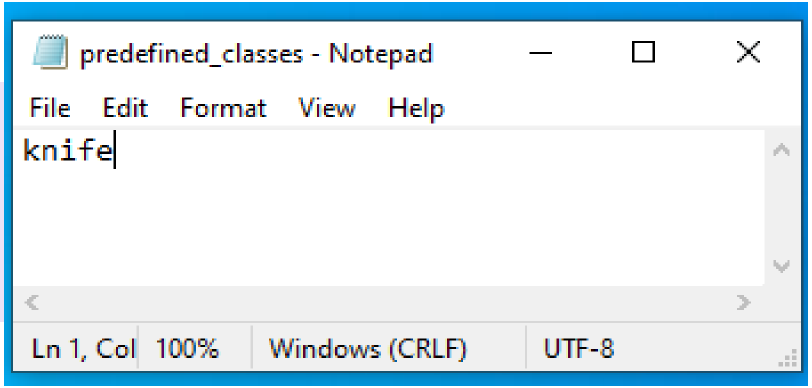
Label The Knife Dataset using LabelImg Tool



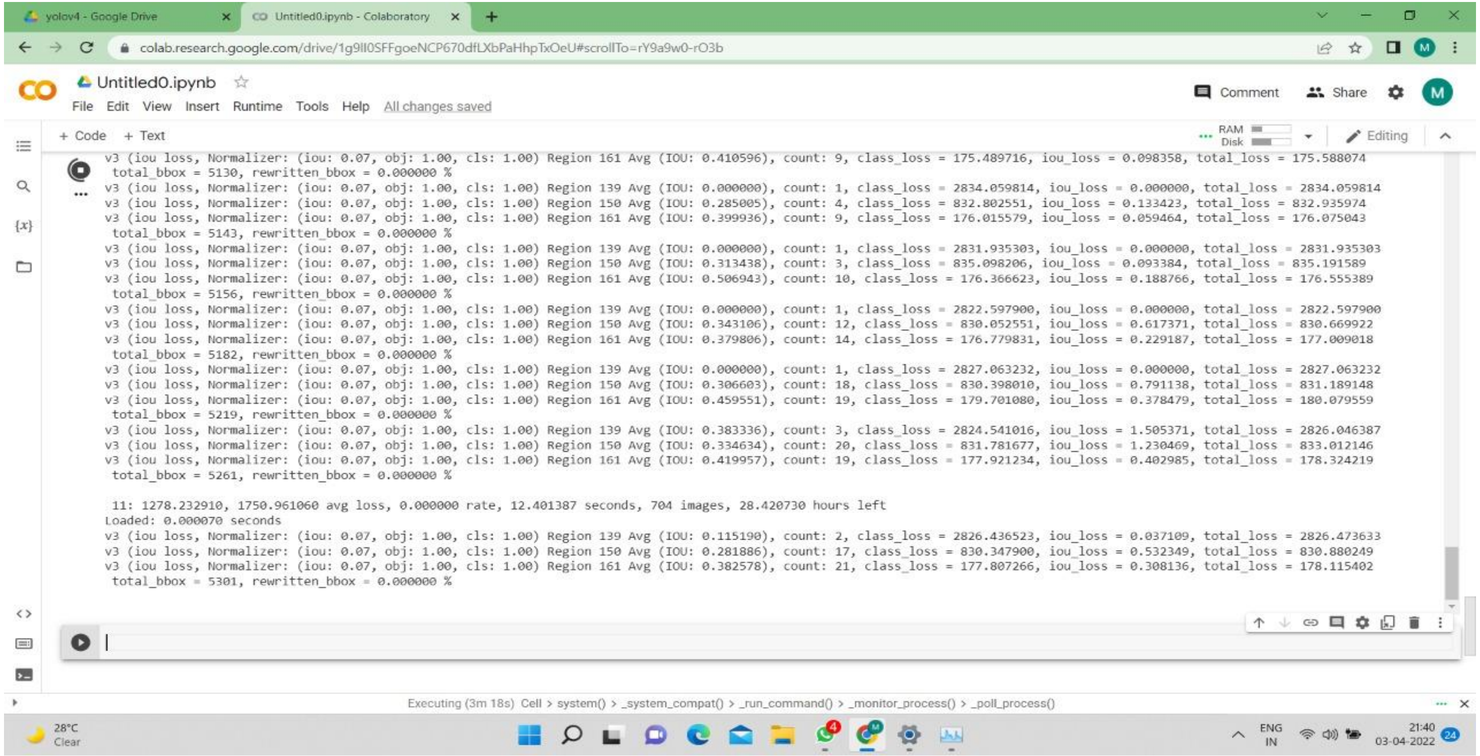
Coordinates of the bound Image



The classes used while labeling



Train the Model on Google Colab using YOLOv4 architecture.



Thank You ☺