Project Report

INTRODUCTION

Project Overview

To build a YOLO-based machine learning model for detecting safety-gear on construction workers and by using this model to build a web application using flask, JavaScript, CSS etc. This web application will input video/live video from device connected to the host computer to be analyzed by the YOLO model and give the analyzed output.

Purpose

This web application will be useful to site manager at construction site who monitor and keep check of workers wearing safety gear or not. This web application will make it a lot easier and improve performance of this type of tasks.

LITERATURE SURVEY

Existing problem

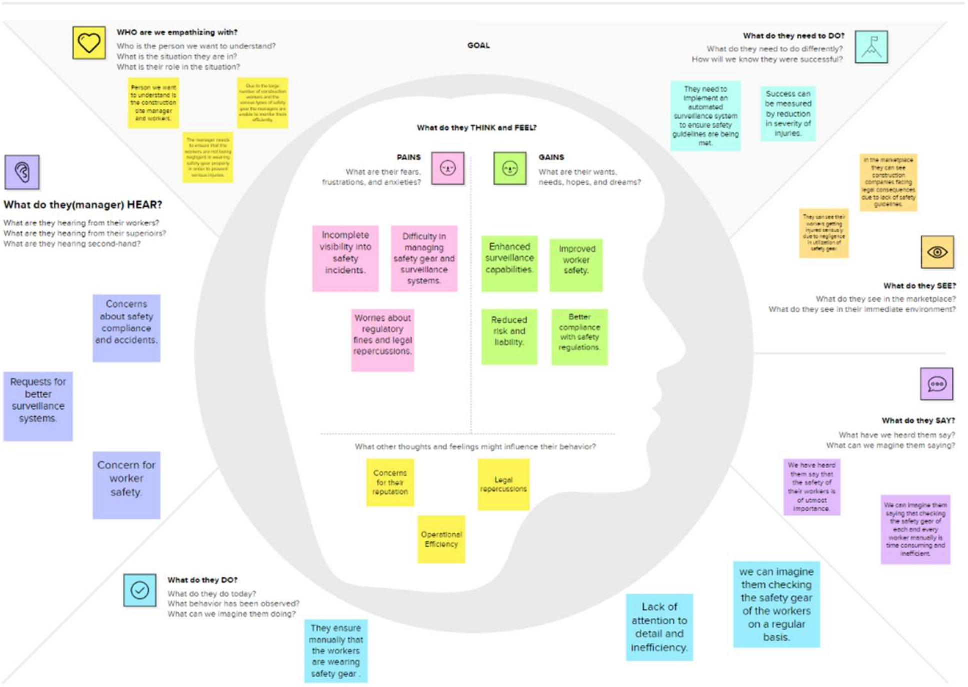
Safety of the construction workers is of the utmost importance when it comes to construction work. To ensure the safety of these worker construction site managers ensure that the workers are appropriately equipped with safety gear at all times. However, this task is done manually and is hence inefficient and repetitive. The mundane nature of this task can lead to negligence and serious injury

Problem Statement Definition

The aim is to develop a machine learning model that analyses live surveillance footage of construction sites in order to ensure that the workers are appropriately equipped with safety gear.

IDEATION & PROPOSED SOLUTION

Empathy Map Canvas:



WHO are we empathizing with?
Who is the person we want to understand?
What is the situation they are in?
What is their role in the situation?

Person we want to understand is the construction site manager and workers.

Due to the large number of construction workers and the various types of safety gear the managers are unable to monitor them efficiently.

The manager needs to ensure that the workers are not being negligent in wearing safety gear properly in order to prevent serious injuries.

GOAL

What do they need to DO?
What do they need to do differently?
How will we know they were successful?

They need to implement an automated surveillance system to ensure safety guidelines are being met.

Success can be measured by reduction in severity of injuries.

What do they(manager) HEAR?
What are they hearing from their workers?
What are they hearing from their superioirs?
What are they hearing second-hand?

Concerns about safety compliance and accidents.

Requests for better surveillance systems.

Concern for worker safety.

What do they THINK and FEEL?

PAINS
What are their fears, frustrations, and anxieties?

Incomplete visibility into safety incidents.

Difficulty in managing safety gear and surveillance systems.

Worries about regulatory fines and legal repercussions.

GAINS
What are their wants, needs, hopes, and dreams?

Enhanced surveillance capabilities.

Improved worker safety.

Reduced risk and liability.

Better compliance with safety regulations.

What other thoughts and feelings might influence their behavior?

Concerns for their reputation

Legal repercussions

Operational Efficiency

What do they SEE?
What do they see in the marketplace?
What do they see in their immediate environment?

In the marketplace they can see construction companies facing legal consequences due to lack of safety guidelines.

They can see their workers getting injured seriously due to negligence in utilization of safety gear.

What do they SAY?
What have we heard them say?
What can we magine them saying?

We have heard them say that the safety of their workers is of utmost importance.

We can imagine them saying that checking the safety gear of each and every worker manually is time consuming and inefficient.

we can imagine them checking the safety gear of the workers on a regular basis.

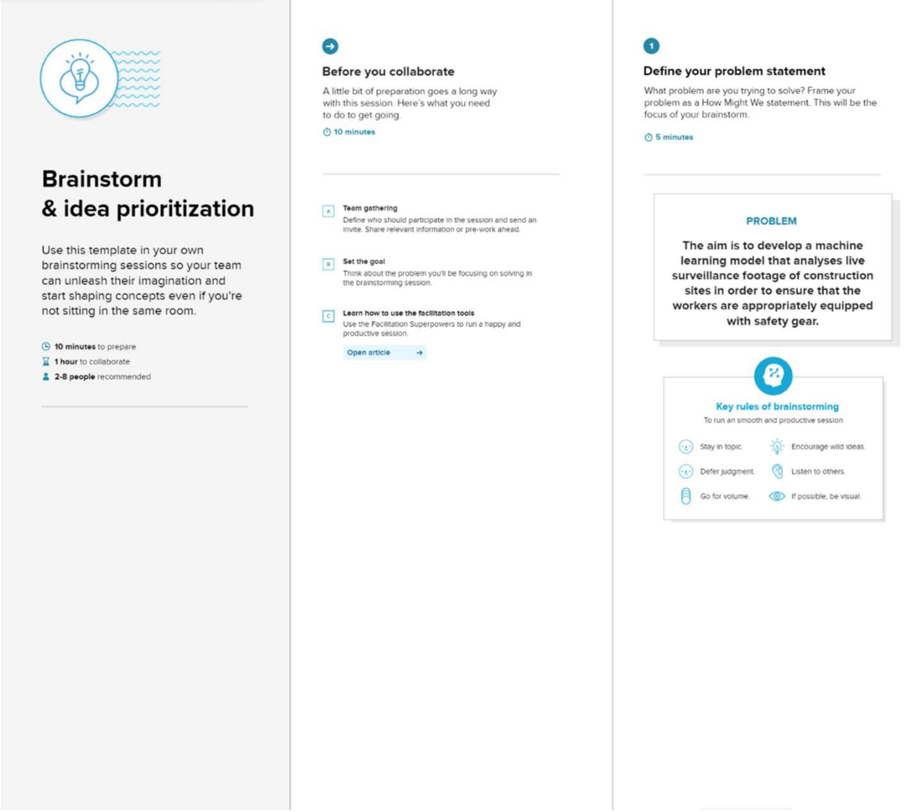
Lack of attention to detail and inefficiency.

What do they DO?
What do they do today?
What behavior has been observed?
What can we imagine them doing?

They ensure manually that the workers are wearing safety gear .

Ideation & Brainstorming

Team Gathering, Collaboration and Select the Problem Statement



Brainstorm & idea prioritization

Use this template in your own brainstorming sessions so your team can unleash their imagination and start shaping concepts even if you're not sitting in the same room.

10 minutes to prepare
1 hour to collaborate
2-8 people recommended

Before you collaborate

A little bit of preparation goes a long way with this session. Here's what you need to do to get going.

10 minutes

A **Team gathering**
Define who should participate in the session and send an invite. Share relevant information or pre-work ahead.

B **Set the goal**
Think about the problem you'll be focusing on solving in the brainstorming session.

C **Learn how to use the facilitation tools**
Use the Facilitation Superpowers to run a happy and productive session.

1 **Define your problem statement**

What problem are you trying to solve? Frame your problem as a How Might We statement. This will be the focus of your brainstorm.

5 minutes

PROBLEM

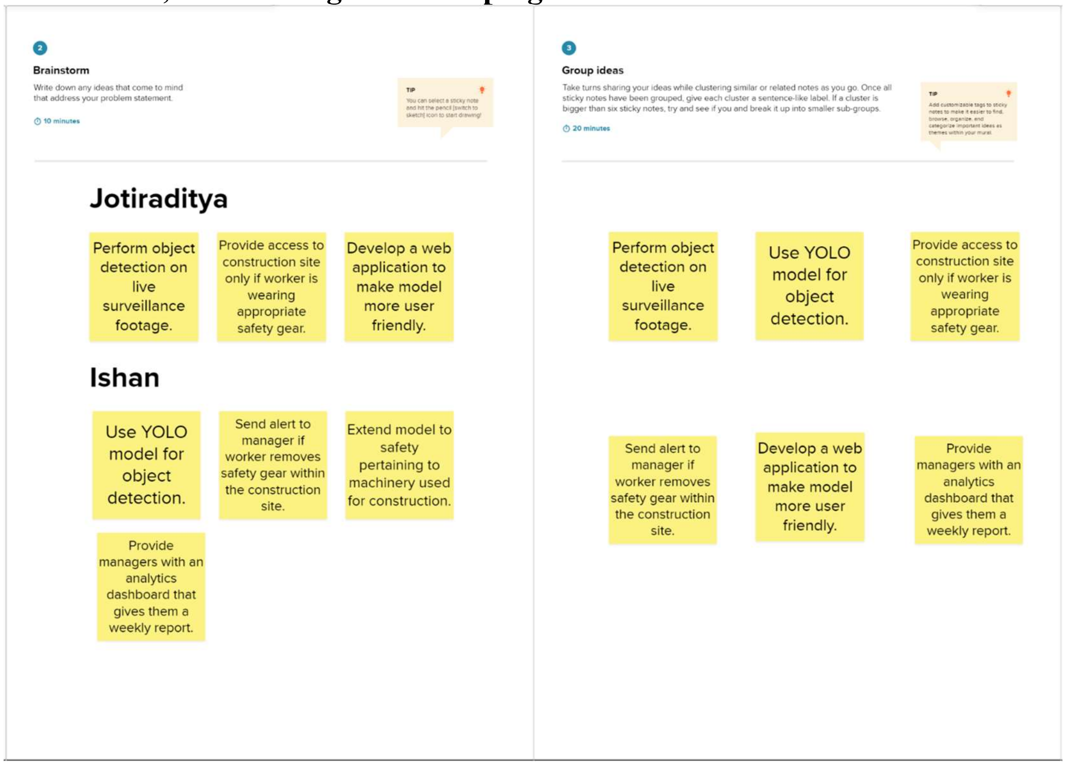
The aim is to develop a machine learning model that analyses live surveillance footage of construction sites in order to ensure that the workers are appropriately equipped with safety gear.

Key rules of brainstorming

To run an smooth and productive session

- Stay in topic.
- Encourage wild ideas.
- Defer judgment.
- Listen to others.
- Go for volume.
- If possible, be visual.

Brainstorm, Idea Listing and Grouping



2

Brainstorm

Write down any ideas that come to mind that address your problem statement.

10 minutes

TIP
You can select a sticky note and hit the pencil [switch to sketch] icon to start drawing!

Jotiraditya

- Perform object detection on live surveillance footage.
- Provide access to construction site only if worker is wearing appropriate safety gear.
- Develop a web application to make model more user friendly.

Ishan

- Use YOLO model for object detection.
- Send alert to manager if worker removes safety gear within the construction site.
- Extend model to safety pertaining to machinery used for construction.
- Provide managers with an analytics dashboard that gives them a weekly report.

3

Group ideas

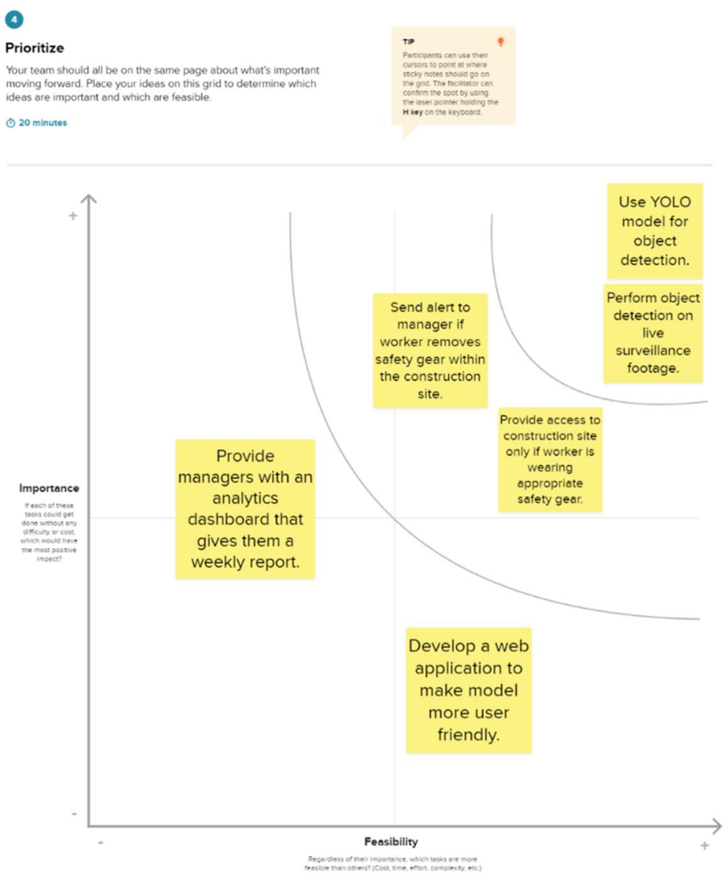
Take turns sharing your ideas while clustering similar or related notes as you go. Once all sticky notes have been grouped, give each cluster a sentence-like label. If a cluster is bigger than six sticky notes, try and see if you and break it up into smaller sub-groups.

20 minutes

TIP
Add customizable tags to sticky notes to make it easier to find, browse, organize, and categorize important ideas as themes within your mural.

- Perform object detection on live surveillance footage.
- Use YOLO model for object detection.
- Provide access to construction site only if worker is wearing appropriate safety gear.
- Send alert to manager if worker removes safety gear within the construction site.
- Develop a web application to make model more user friendly.
- Provide managers with an analytics dashboard that gives them a weekly report.

Idea Prioritization



4

Prioritize

Your team should all be on the same page about what's important moving forward. Place your ideas on this grid to determine which ideas are important and which are feasible.

20 minutes

TIP
Participants can use their cursors to point at where sticky notes should go on the grid. The facilitator can confirm the spot by using the laser pointer holding the **H key** on the keyboard.

+
Importance
If each of these tasks could get done without any difficulty or cost, which would have the most positive impact?
-
-
Feasibility
Regardless of their importance, which tasks are more feasible than others? (Cost, time, effort, complexity, etc.)
+

- Use YOLO model for object detection.
- Perform object detection on live surveillance footage.
- Send alert to manager if worker removes safety gear within the construction site.
- Provide access to construction site only if worker is wearing appropriate safety gear.
- Provide managers with an analytics dashboard that gives them a weekly report.
- Develop a web application to make model more user friendly.

REQUIREMENT ANALYSIS

Functional Requirements: -

1. site manager can access the web applications and obtain a clear understanding of all the features it provides.
2. site manager can upload video of construction site on the web application.
3. site manager can view the analyzed video and gain the understanding of the presence or absence of safety gear equipped by the workers.
4. site manager can upload live surveillance footage of the construction site to the web application.
5. site manager can view real time object detection done on the live surveillance footage to check for presence or absence of safety gear equipped by the workers.

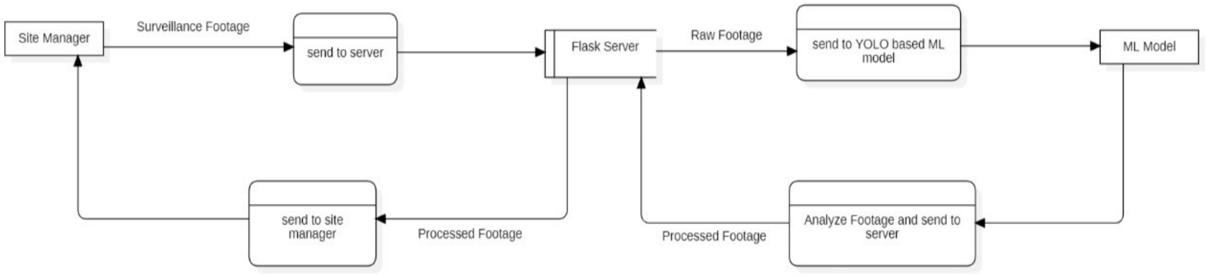
Non-Functional Requirements

1. users must have basic knowledge on operation of computers ad web applications.
2. users must have access to computers with at least 2GB RAM and 200MB of free space.
3. users must have access to decent internet connection.
4. user's surveillance hardware must be compatible with the web application.
5. Users must understand basic English.

PROJECT DESIGN

Data Flow Diagrams & User Stories

Data Flow Diagram:



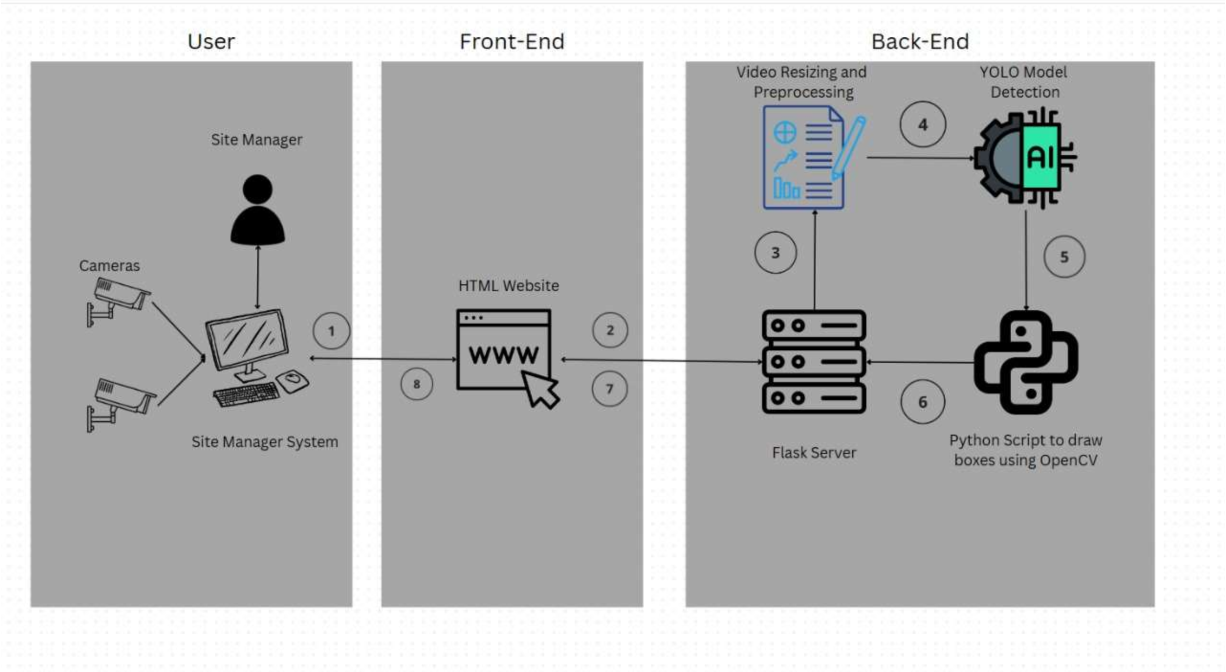
User Stories

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance Citeria	Priority	Release
Site Manager (Web user)	Upload Surveillance	USN-1	As a user, I can upload the live surveillance footage of the construction site through the web application	The flask sever is receiving the footage successfully.	**High**	Sprint-1

	Receive Analyzed footage	USN-2	As a user, I should be able to receive the analyzed footage that explicitly indicates whether the workers are appropriately equipped with safety gear or not.	The flask sever should send the analyzed footage to the users.	**High**	Sprint-1
	Receive alerts/notifications	USN-3	As a user, I should receive automatic alerts when a worker is not wearing safety gear for a specified duration.	The user should receive alert.	**Medium**	Sprint-2

PROJECT PLANNING & SCHEDULING

Technical Architecture



Sprint Planning & Estimation

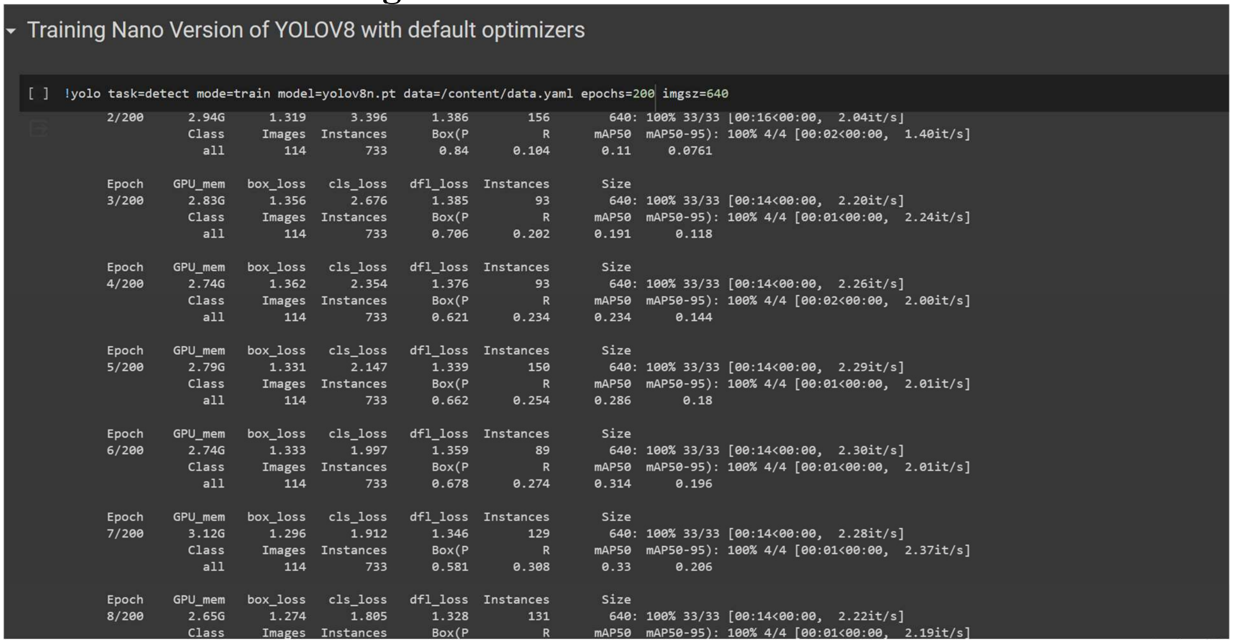
Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Upload Surveillance	USN-1	As a user, I can upload the live surveillance footage of the construction site through the web application	10	High	Jotiraditya and Ishan
Sprint-1	Receive Analyzed footage	USN-2	As a user, I should be able to receive the analyzed footage that explicitly indicates whether the workers are appropriately equipped with safety gear or not	10	High	Jotiraditya and Ishan
Sprint-2	Optimized version of YOLO	USN-3	The object detection should be faster to process video better	5	Medium	Jotiraditya
Sprint-2	Visually informing the presence or absence of safety-gear	USN-4	As a user, I can register for the application through Gmail	5	Medium	Ishan

Sprint Delivery Schedule

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	12 Days	18 Oct 2023	30 Oct 2023	20	1 Nov 2023
Sprint-2	10	8 Days	31 Oct 2023	08 Nov 2023	10	9 Nov 2023

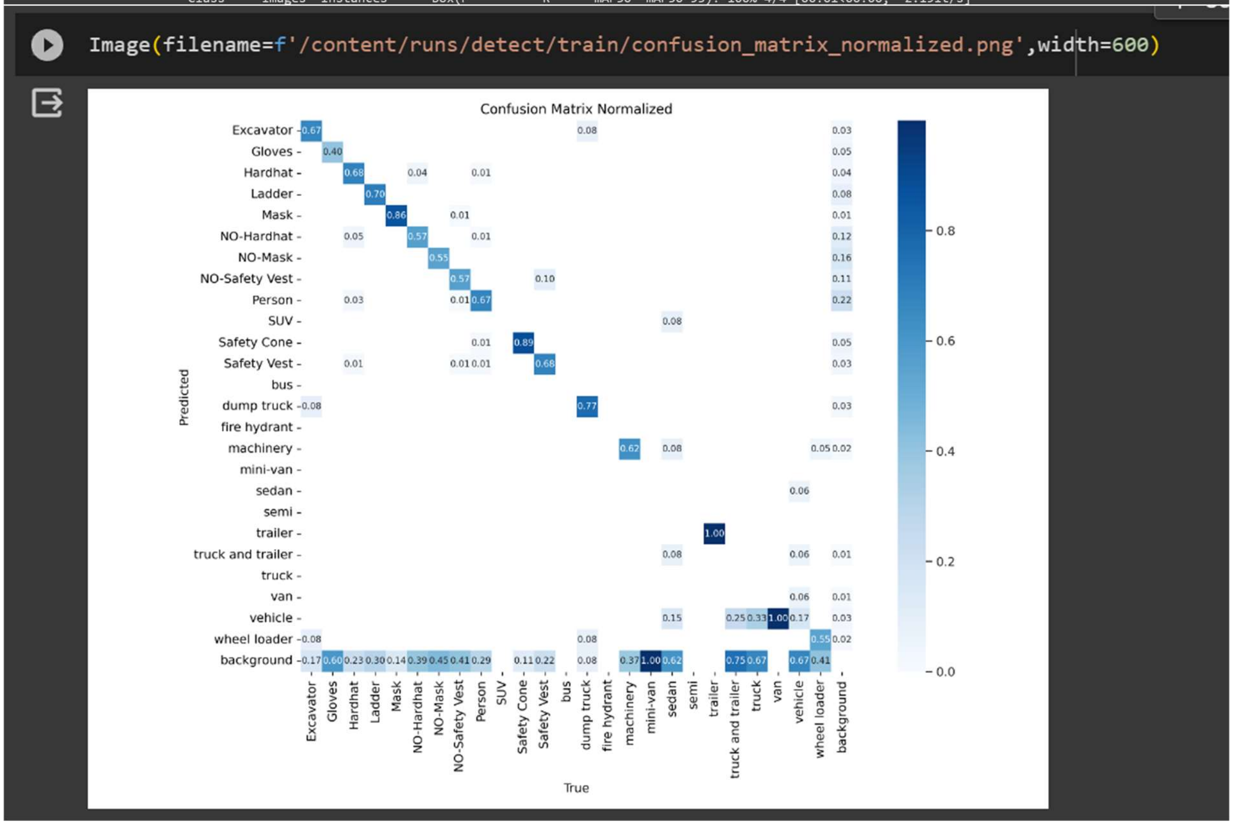
CODING & SOLUTIONING (Explain the features added in the project along with code)

YOLO Model Training:



Training Nano Version of YOLOV8 with default optimizers

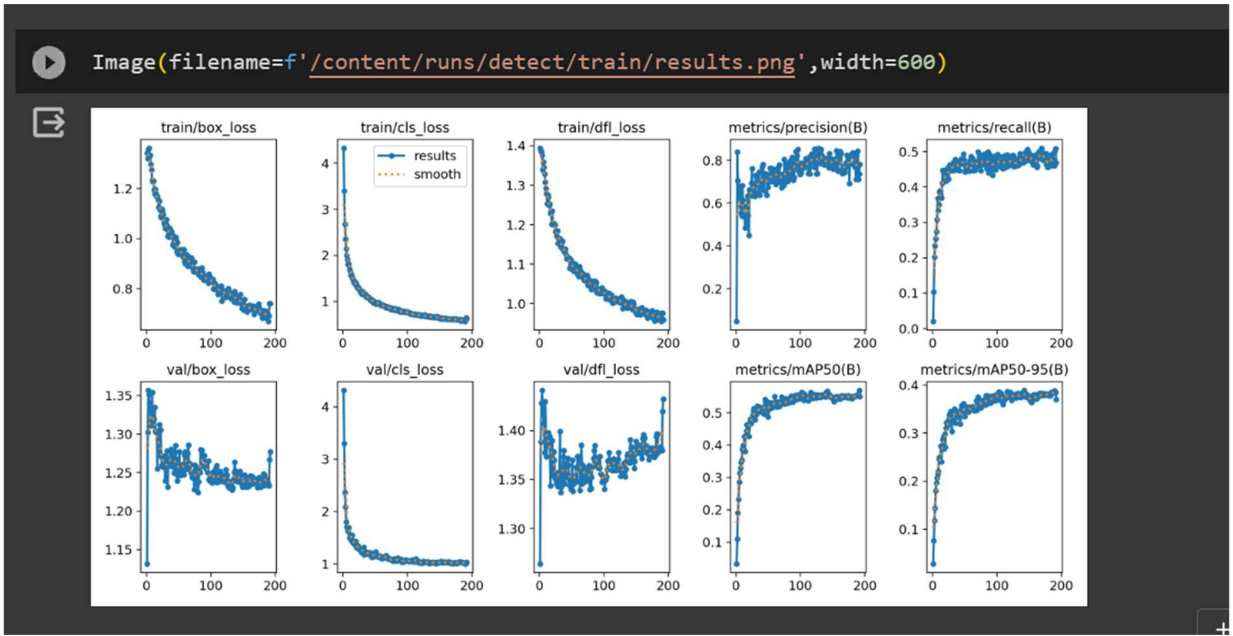
```
[ ] !yolo task=detect mode=train model=yolov8n.pt data=/content/data.yaml epochs=200 imgsz=640
       2/200      2.94G      1.319      3.396      1.386        156        640: 100% 33/33 [00:16<00:00,  2.04it/s]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% 4/4 [00:02<00:00,  1.40it/s]
                   all        114        733       0.84      0.104       0.11     0.0761

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
       3/200      2.83G      1.356      2.676      1.385         93        640: 100% 33/33 [00:14<00:00,  2.20it/s]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% 4/4 [00:01<00:00,  2.24it/s]
                   all        114        733      0.706      0.202      0.191      0.118

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
       4/200      2.74G      1.362      2.354      1.376         93        640: 100% 33/33 [00:14<00:00,  2.26it/s]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% 4/4 [00:02<00:00,  2.00it/s]
                   all        114        733      0.621      0.234      0.234      0.144

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
       5/200      2.79G      1.331      2.147      1.339        150        640: 100% 33/33 [00:14<00:00,  2.29it/s]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% 4/4 [00:01<00:00,  2.01it/s]
                   all        114        733      0.662      0.254      0.286       0.18

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
       6/200      2.74G      1.333      1.997      1.359         89        640: 100% 33/33 [00:14<00:00,  2.30it/s]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% 4/4 [00:01<00:00,  2.01it/s]
                   all        114        733      0.678      0.274      0.314      0.196

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
       7/200      3.12G      1.296      1.912      1.346        129        640: 100% 33/33 [00:14<00:00,  2.28it/s]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% 4/4 [00:01<00:00,  2.37it/s]
                   all        114        733      0.581      0.308       0.33      0.206

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
       8/200      2.65G      1.274      1.805      1.328        131        640: 100% 33/33 [00:14<00:00,  2.22it/s]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% 4/4 [00:01<00:00,  2.19it/s]
```



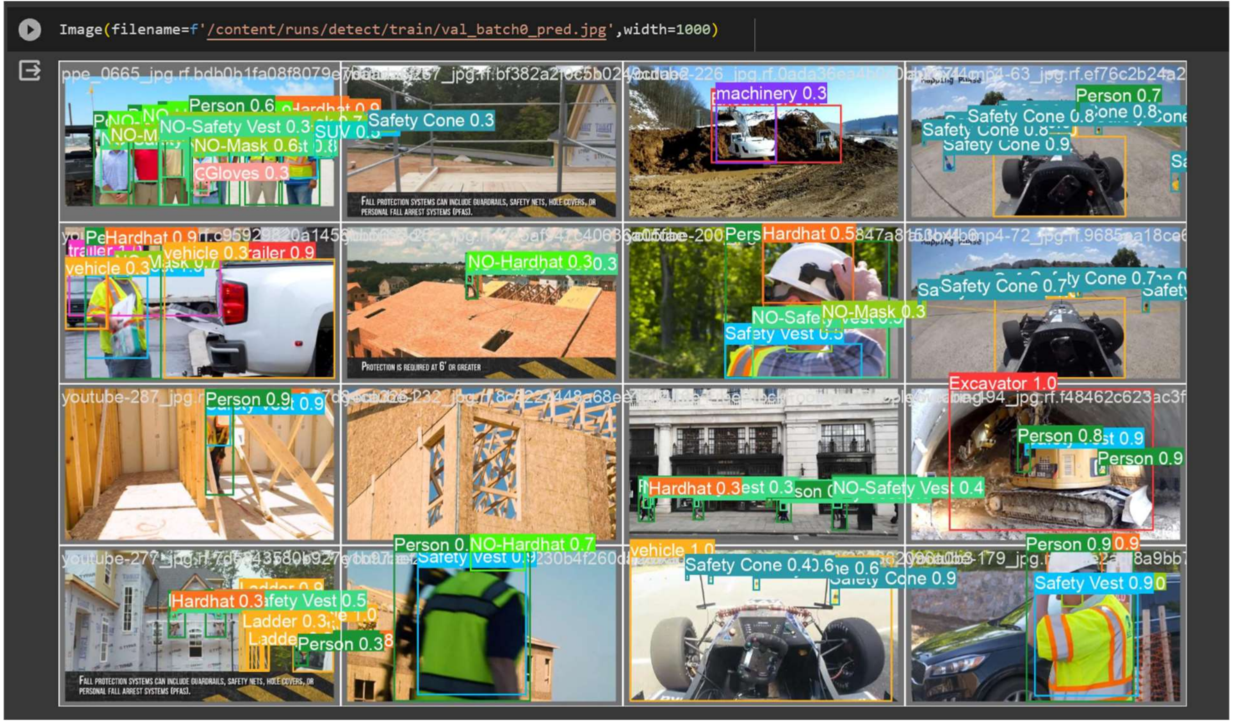
```
Image(filename=f'/content/runs/detect/train/confusion_matrix_normalized.png',width=600)
```

Confusion Matrix Normalized

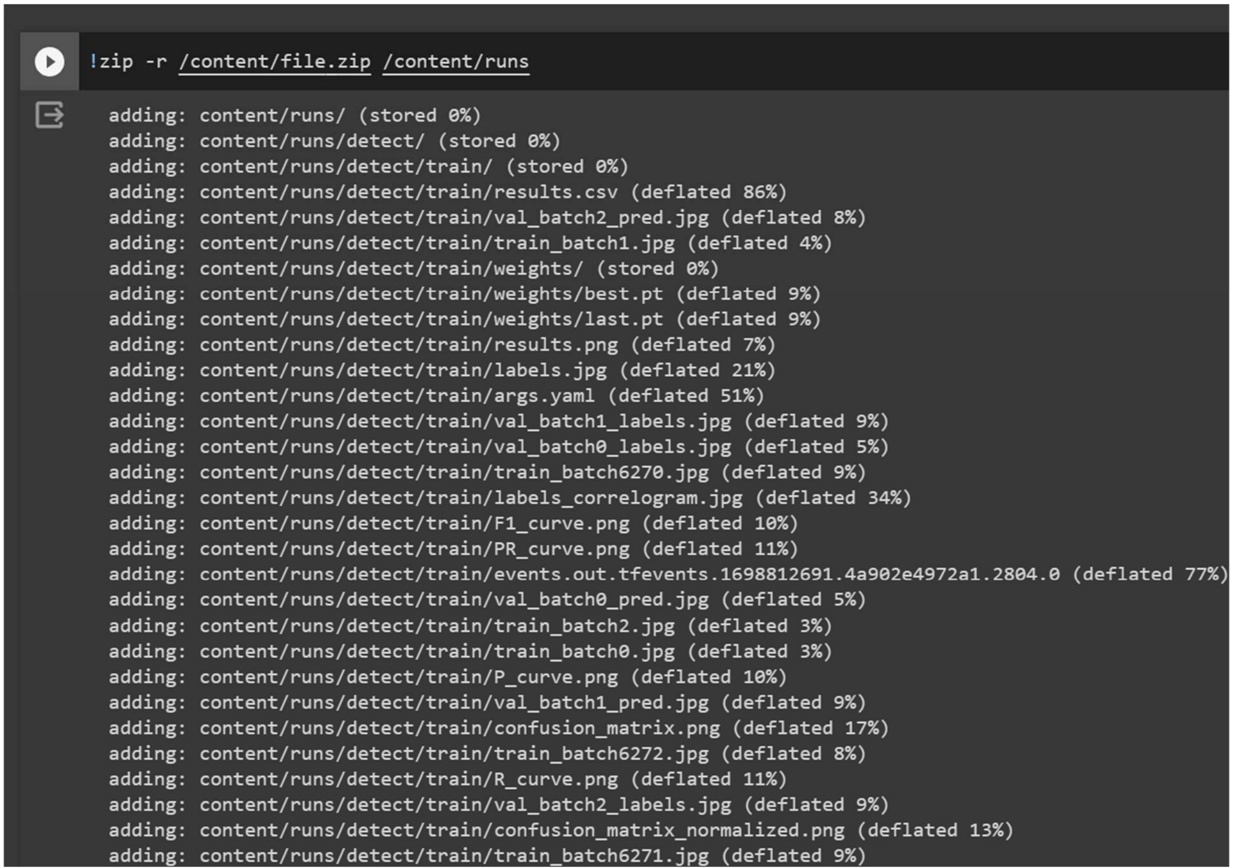
	Excavator	Gloves	Hardhat	Ladder	Mask	NO-Hardhat	NO-Mask	NO-Safety Vest	Person	SUV	Safety Cone	Safety Vest	bus	dump truck	fire hydrant	machinery	mini-van	sedan	semi	trailer	truck and trailer	truck	van	vehicle	wheel loader	background
Excavator	0.67													0.08												0.03
Gloves		0.40																								0.05
Hardhat			0.68			0.04			0.01																	0.04
Ladder				0.70																						0.08
Mask					0.86			0.01																		0.01
NO-Hardhat			0.05			0.57			0.01																	0.12
NO-Mask							0.55																			0.16
NO-Safety Vest								0.57				0.10														0.11
Person			0.03					0.01	0.67																	0.22
SUV																		0.08								
Safety Cone									0.01		0.89															0.05
Safety Vest			0.01					0.01	0.01			0.68														0.03
bus																										
dump truck	0.08													0.77												0.03
fire hydrant																										
machinery																0.62		0.08							0.05	0.02
mini-van																										
sedan																								0.06		
semi																										
trailer																				1.00						
truck and trailer																		0.08						0.06		0.01
truck																										
van																								0.06		0.01
vehicle																		0.15			0.25	0.33	1.00	0.17		0.03
wheel loader	0.08													0.08											0.55	0.02
background	0.17	0.60	0.23	0.30	0.14	0.39	0.45	0.41	0.29		0.11	0.22		0.08		0.37	1.00	0.62			0.75	0.67		0.67	0.41	



```
Image(filename=f'/content/runs/detect/train/results.png',width=600)
```

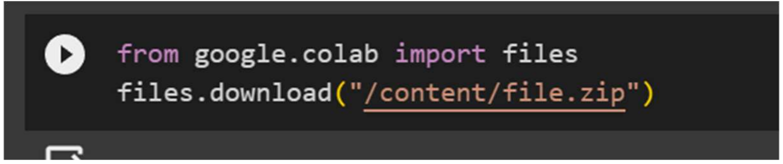


```
Image(filename=f'/content/runs/detect/train/val_batch0_pred.jpg',width=1000)
```



```
!zip -r /content/file.zip /content/runs
```

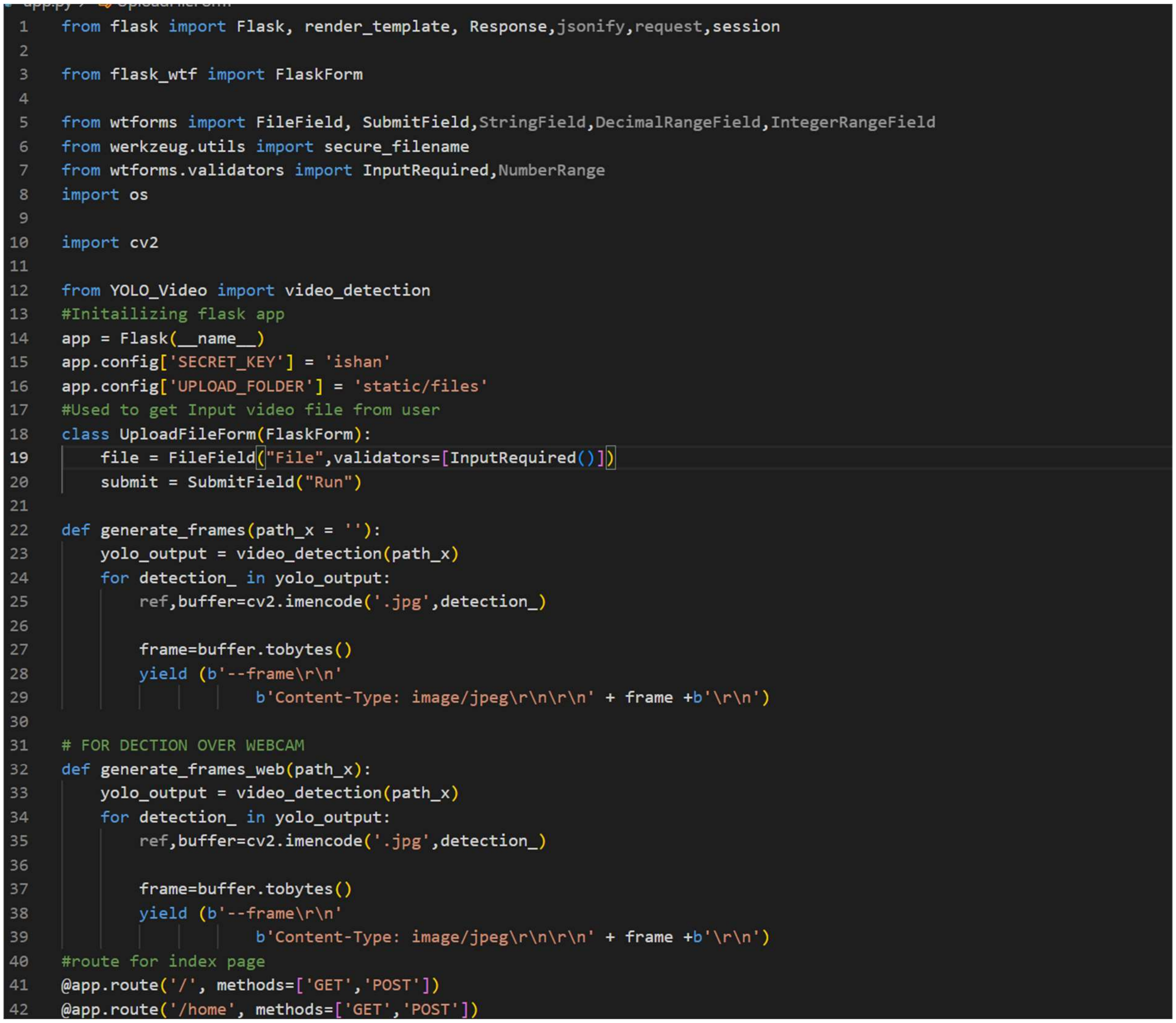
```
  adding: content/runs/ (stored 0%)
  adding: content/runs/detect/ (stored 0%)
  adding: content/runs/detect/train/ (stored 0%)
  adding: content/runs/detect/train/results.csv (deflated 86%)
  adding: content/runs/detect/train/val_batch2_pred.jpg (deflated 8%)
  adding: content/runs/detect/train/train_batch1.jpg (deflated 4%)
  adding: content/runs/detect/train/weights/ (stored 0%)
  adding: content/runs/detect/train/weights/best.pt (deflated 9%)
  adding: content/runs/detect/train/weights/last.pt (deflated 9%)
  adding: content/runs/detect/train/results.png (deflated 7%)
  adding: content/runs/detect/train/labels.jpg (deflated 21%)
  adding: content/runs/detect/train/args.yaml (deflated 51%)
  adding: content/runs/detect/train/val_batch1_labels.jpg (deflated 9%)
  adding: content/runs/detect/train/val_batch0_labels.jpg (deflated 5%)
  adding: content/runs/detect/train/train_batch6270.jpg (deflated 9%)
  adding: content/runs/detect/train/labels_correlogram.jpg (deflated 34%)
  adding: content/runs/detect/train/F1_curve.png (deflated 10%)
  adding: content/runs/detect/train/PR_curve.png (deflated 11%)
  adding: content/runs/detect/train/events.out.tfevents.1698812691.4a902e4972a1.2804.0 (deflated 77%)
  adding: content/runs/detect/train/val_batch0_pred.jpg (deflated 5%)
  adding: content/runs/detect/train/train_batch2.jpg (deflated 3%)
  adding: content/runs/detect/train/train_batch0.jpg (deflated 3%)
  adding: content/runs/detect/train/P_curve.png (deflated 10%)
  adding: content/runs/detect/train/val_batch1_pred.jpg (deflated 9%)
  adding: content/runs/detect/train/confusion_matrix.png (deflated 17%)
  adding: content/runs/detect/train/train_batch6272.jpg (deflated 8%)
  adding: content/runs/detect/train/R_curve.png (deflated 11%)
  adding: content/runs/detect/train/val_batch2_labels.jpg (deflated 9%)
  adding: content/runs/detect/train/confusion_matrix_normalized.png (deflated 13%)
  adding: content/runs/detect/train/train_batch6271.jpg (deflated 9%)
```



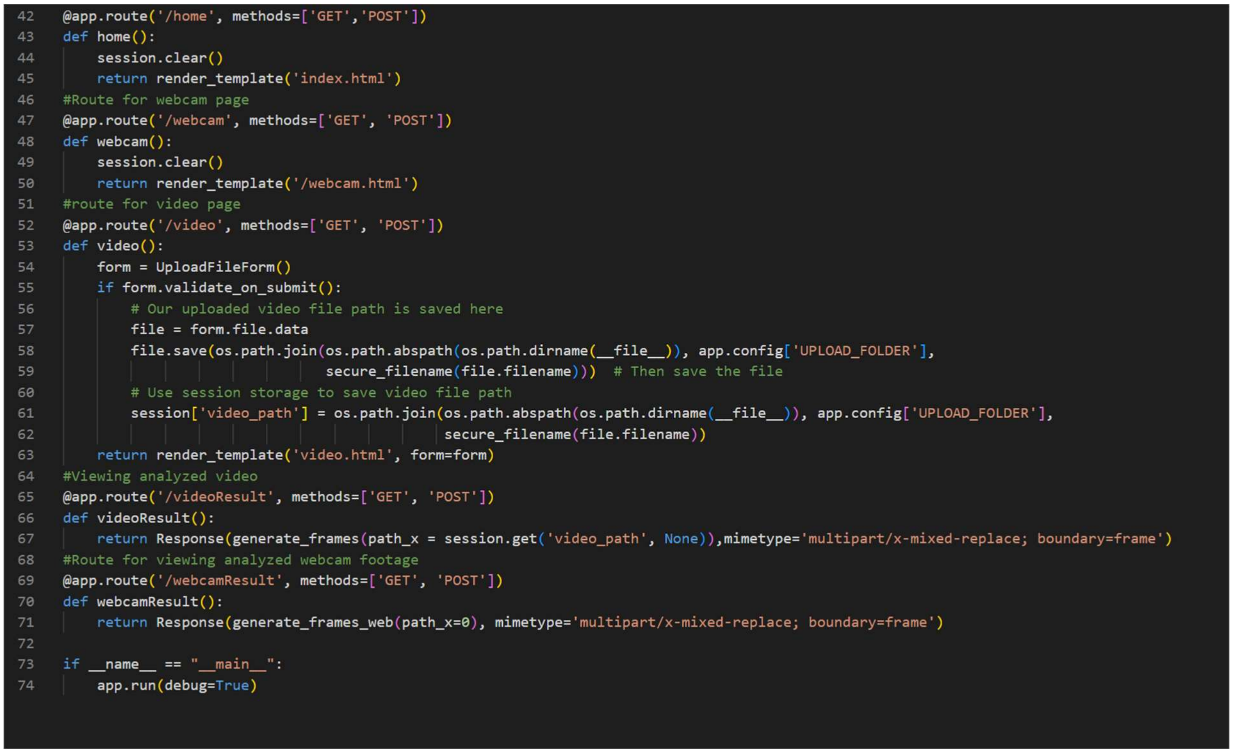
```python
from google.colab import files
files.download("/content/file.zip")
```

Flask Application:

App.py:

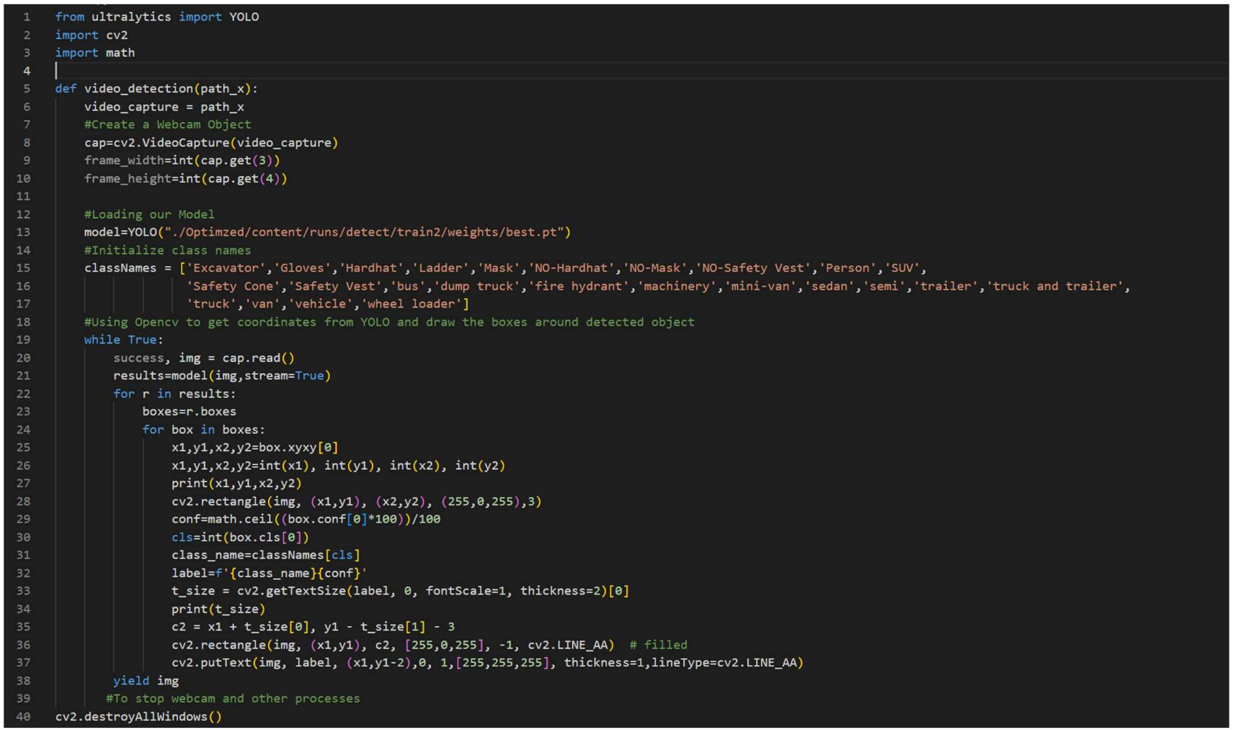


```python
from flask import Flask, render_template, Response,jsonify,request,session

from flask_wtf import FlaskForm

from wtforms import FileField, SubmitField,StringField,DecimalRangeField,IntegerRangeField
from werkzeug.utils import secure_filename
from wtforms.validators import InputRequired,NumberRange
import os

import cv2

from YOLO_Video import video_detection
#Initailizing flask app
app = Flask(__name__)
app.config['SECRET_KEY'] = 'ishan'
app.config['UPLOAD_FOLDER'] = 'static/files'
#Used to get Input video file from user
class UploadFileForm(FlaskForm):
    file = FileField("File",validators=[InputRequired()])
    submit = SubmitField("Run")

def generate_frames(path_x = ''):
    yolo_output = video_detection(path_x)
    for detection_ in yolo_output:
        ref,buffer=cv2.imencode('.jpg',detection_)

        frame=buffer.tobytes()
        yield (b'--frame\r\n'
                    b'Content-Type: image/jpeg\r\n\r\n' + frame +b'\r\n')

# FOR DECTION OVER WEBCAM
def generate_frames_web(path_x):
    yolo_output = video_detection(path_x)
    for detection_ in yolo_output:
        ref,buffer=cv2.imencode('.jpg',detection_)

        frame=buffer.tobytes()
        yield (b'--frame\r\n'
                    b'Content-Type: image/jpeg\r\n\r\n' + frame +b'\r\n')
#route for index page
@app.route('/', methods=['GET','POST'])
@app.route('/home', methods=['GET','POST'])
```

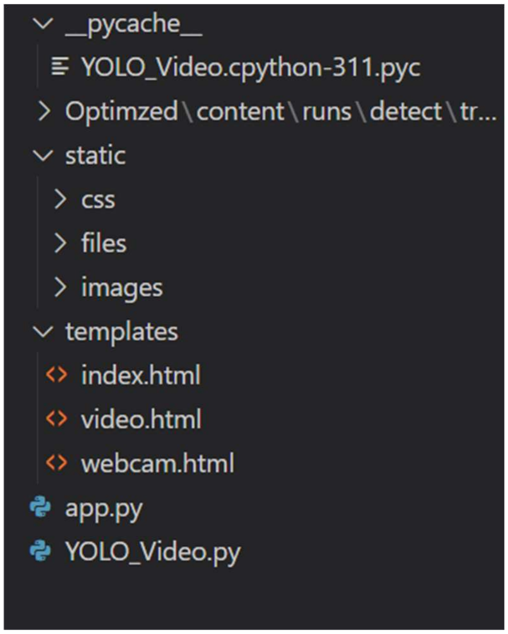


```python
@app.route('/home', methods=['GET','POST'])
def home():
    session.clear()
    return render_template('index.html')
#Route for webcam page
@app.route('/webcam', methods=['GET', 'POST'])
def webcam():
    session.clear()
    return render_template('/webcam.html')
#route for video page
@app.route('/video', methods=['GET', 'POST'])
def video():
    form = UploadFileForm()
    if form.validate_on_submit():
        # Our uploaded video file path is saved here
        file = form.file.data
        file.save(os.path.join(os.path.abspath(os.path.dirname(__file__)), app.config['UPLOAD_FOLDER'],
                               secure_filename(file.filename)))  # Then save the file
        # Use session storage to save video file path
        session['video_path'] = os.path.join(os.path.abspath(os.path.dirname(__file__)), app.config['UPLOAD_FOLDER'],
                                             secure_filename(file.filename))
    return render_template('video.html', form=form)
#Viewing analyzed video
@app.route('/videoResult', methods=['GET', 'POST'])
def videoResult():
    return Response(generate_frames(path_x = session.get('video_path', None)),mimetype='multipart/x-mixed-replace; boundary=frame')
#Route for viewing analyzed webcam footage
@app.route('/webcamResult', methods=['GET', 'POST'])
def webcamResult():
    return Response(generate_frames_web(path_x=0), mimetype='multipart/x-mixed-replace; boundary=frame')

if __name__ == "__main__":
    app.run(debug=True)
```

YOLO_video.py:

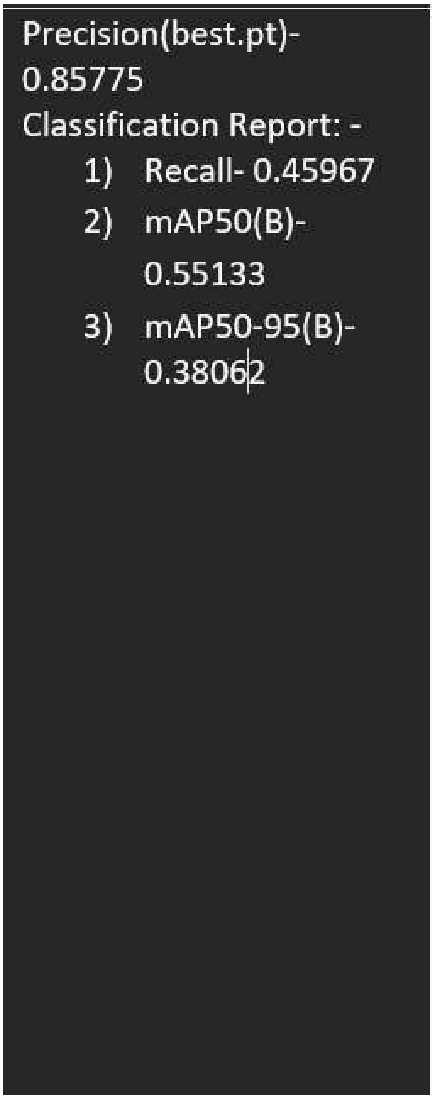


```python
from ultralytics import YOLO
import cv2
import math

def video_detection(path_x):
    video_capture = path_x
    #Create a Webcam Object
    cap=cv2.VideoCapture(video_capture)
    frame_width=int(cap.get(3))
    frame_height=int(cap.get(4))

    #Loading our Model
    model=YOLO("./Optimzed/content/runs/detect/train2/weights/best.pt")
    #Initialize class names
    classNames = ['Excavator','Gloves','Hardhat','Ladder','Mask','NO-Hardhat','NO-Mask','NO-Safety Vest','Person','SUV',
                  'Safety Cone','Safety Vest','bus','dump truck','fire hydrant','machinery','mini-van','sedan','semi','trailer','truck and trailer',
                  'truck','van','vehicle','wheel loader']
    #Using Opencv to get coordinates from YOLO and draw the boxes around detected object
    while True:
        success, img = cap.read()
        results=model(img,stream=True)
        for r in results:
            boxes=r.boxes
            for box in boxes:
                x1,y1,x2,y2=box.xyxy[0]
                x1,y1,x2,y2=int(x1), int(y1), int(x2), int(y2)
                print(x1,y1,x2,y2)
                cv2.rectangle(img, (x1,y1), (x2,y2), (255,0,255),3)
                conf=math.ceil((box.conf[0]*100))/100
                cls=int(box.cls[0])
                class_name=classNames[cls]
                label=f'{class_name}{conf}'
                t_size = cv2.getTextSize(label, 0, fontScale=1, thickness=2)[0]
                print(t_size)
                c2 = x1 + t_size[0], y1 - t_size[1] - 3
                cv2.rectangle(img, (x1,y1), c2, [255,0,255], -1, cv2.LINE_AA)  # filled
                cv2.putText(img, label, (x1,y1-2),0, 1,[255,255,255], thickness=1,lineType=cv2.LINE_AA)
        yield img
       #To stop webcam and other processes
cv2.destroyAllWindows()
```



PERFORMANCE TESTING

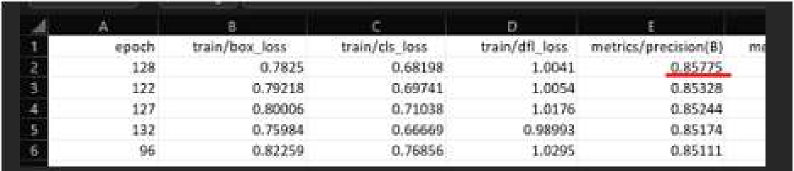
Performance Metrics



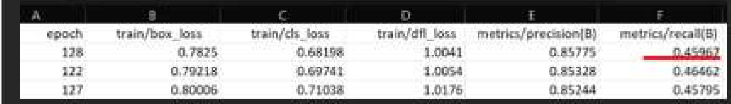
Precision(best.pt)- 0.85775

Classification Report: -

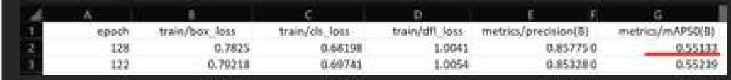
1) Recall- 0.45967
2) mAP50(B)- 0.55133
3) mAP50-95(B)- 0.38062



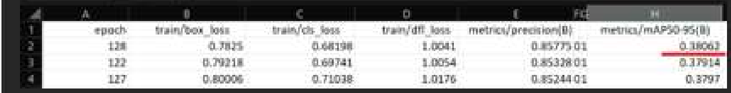
	A	B	C	D	E
1	epoch	train/box_loss	train/cls_loss	train/dfl_loss	metrics/precision(B)
2	128	0.7825	0.68198	1.0041	0.85775
3	122	0.79218	0.69741	1.0054	0.85328
4	127	0.80006	0.71038	1.0176	0.85244
5	132	0.75984	0.66669	0.98993	0.85174
6	96	0.82259	0.76856	1.0295	0.85111



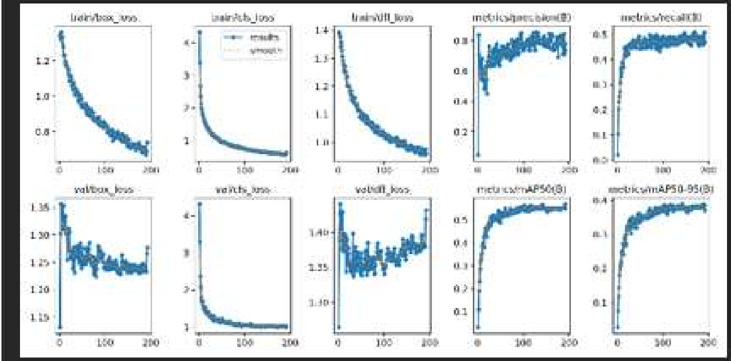
A	B	C	D	E	F
epoch	train/box_loss	train/cls_loss	train/dfl_loss	metrics/precision(B)	metrics/recall(B)
128	0.7825	0.68198	1.0041	0.85775	0.45967
122	0.79218	0.69741	1.0054	0.85328	0.46462
127	0.80006	0.71038	1.0176	0.85244	0.45795



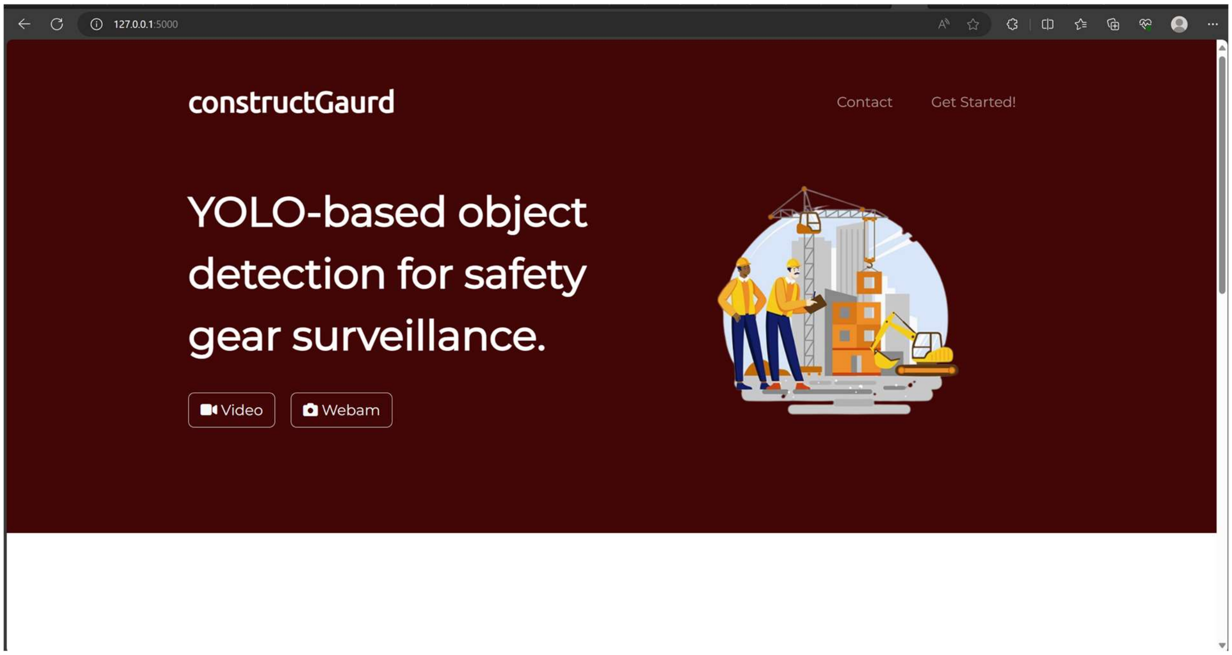
	A	B	C	D	E	F	G
1	epoch	train/box_loss	train/cls_loss	train/dfl_loss	metrics/precision(B)		metrics/mAP50(B)
2	128	0.7825	0.68198	1.0041	0.85775	0	0.55133
3	122	0.79218	0.69741	1.0054	0.85328	0	0.55239



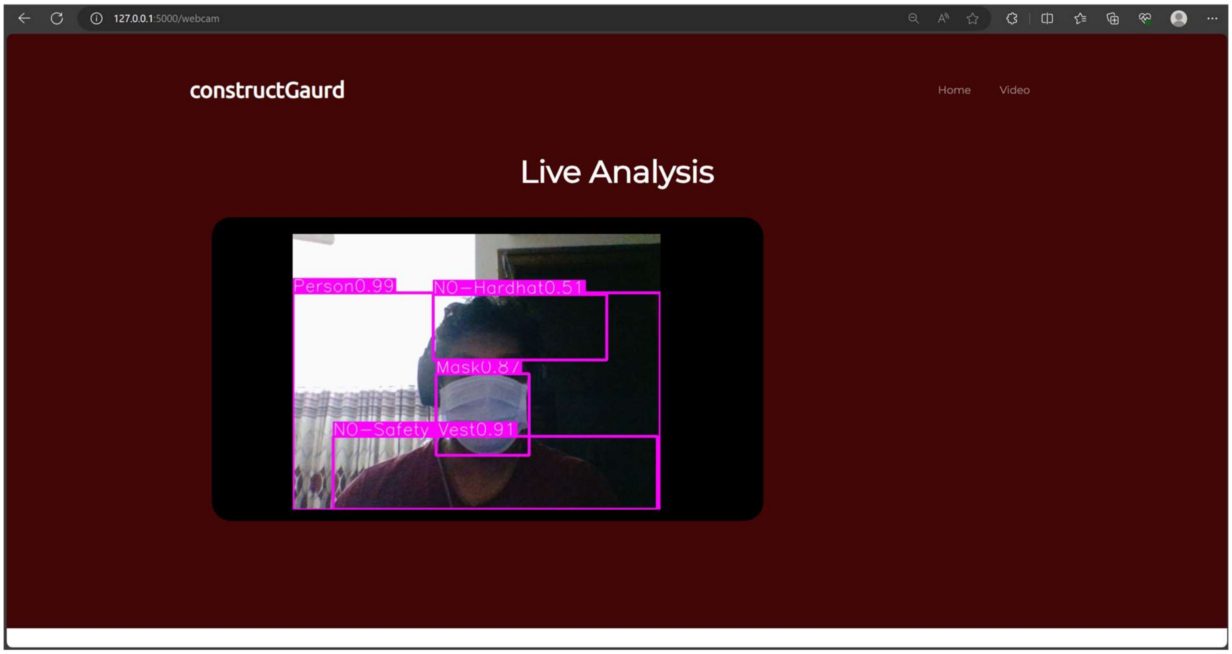
	A	B	C	D	E	FG	H
1	epoch	train/box_loss	train/cls_loss	train/dfl_loss	metrics/precision(B)		metrics/mAP50-95(B)
2	128	0.7825	0.68198	1.0041	0.85775	01	0.38062
3	122	0.79218	0.69741	1.0054	0.85328	01	0.37914
4	127	0.80006	0.71038	1.0176	0.85244	01	0.3797



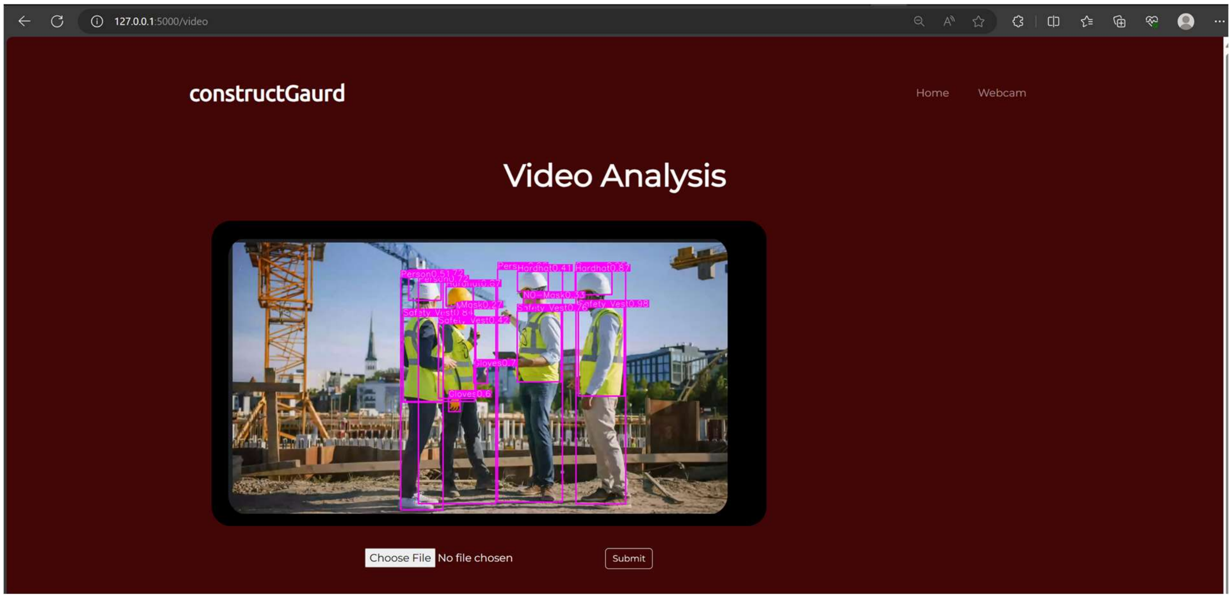
RESULTS

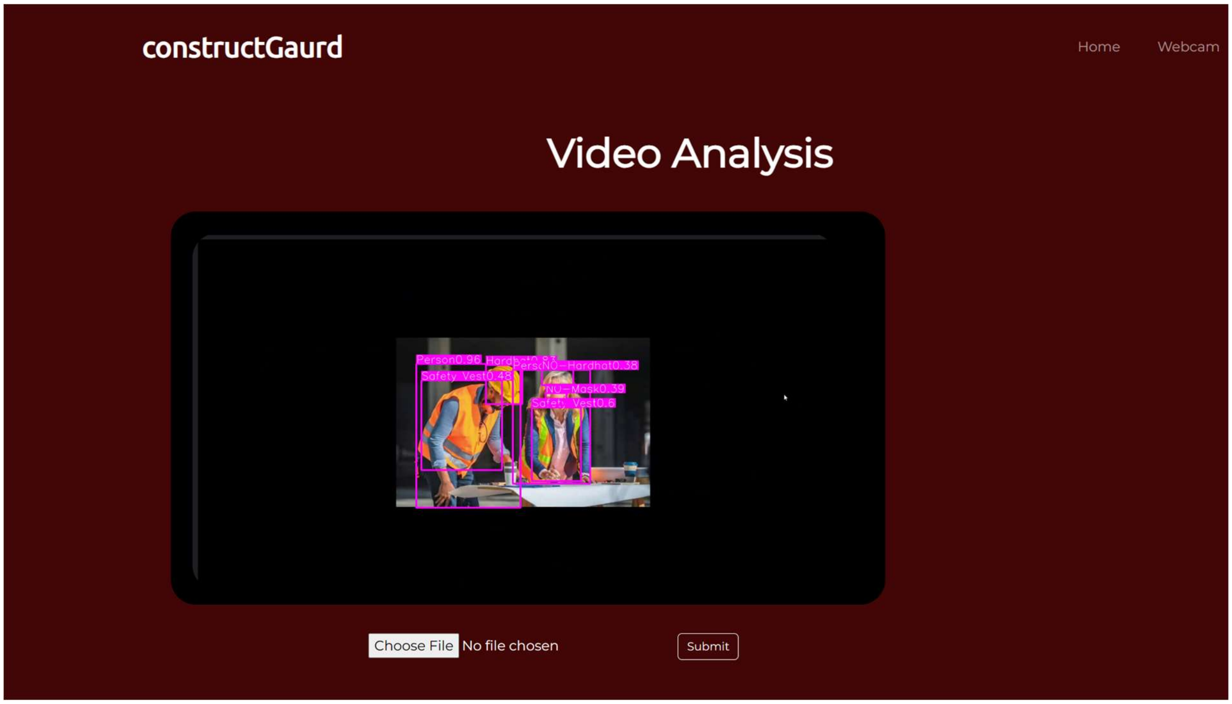


YOLO-based object detection for safety gear surveillance.



Live Analysis


127.0.0.1:5000/video
constructGaurd
Home
Webcam
Video Analysis
Choose File
No file chosen
Submit


constructGaurd
Home
Webcam
Video Analysis
Choose File
No file chosen
Submit

ADVANTAGES & DISADVANTAGES

Advantages: -

- Site Managers do not need to manually check safety gear or workers.
- Reduction in number of serious injuries.
- Safety guidelines in construction sites are enforced more easily.

Disadvantages: -

- There might be occurrences of false positive or negatives.

CONCLUSION

Using our application construction site managers will be able to mitigate safety risks easily and efficiently thereby making this a viable and desirable product.

FUTURE SCOPE

- **This model can be extended to enforce safety guidelines in fields other than construction such as medicine, engineering, manufacturing etc.**
- We can extend the functionality of the model to other tasks performed by site managers essentially developing an Artificial Intelligence.

APPENDIX

Source Code: smartinternz02/SI-GuidedProject-592725-1697204161 (github.com)

Project Demo Link:
https://drive.google.com/file/d/1KpLfnQNU9VM0ME7oGpWqjzZJxToQUkmJ/view?usp=sharing