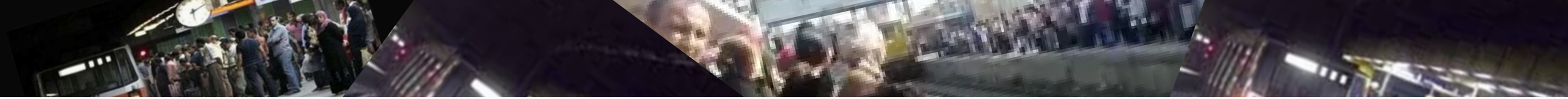
METRO SAFETY SYSTEM

WHY ?!



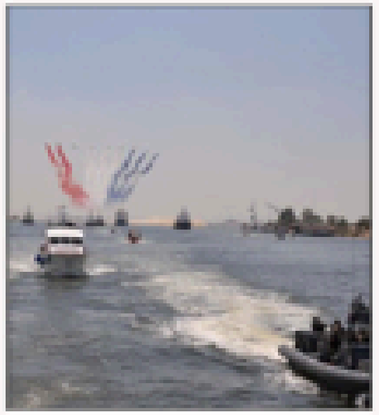
رئيس مجلس الإدارة
محمد بركات
رئيس التحرير
ياسر رزق

الأخبار

تأسست سنة ١٩٥٢
أسسها مصطفى أمين وعلي أمين سنة ١٩٥٢

٢٢ صفحة
١٥٠ قرش
الطبعة الأولى
مؤسسة أخبار اليوم
٦ شارع الصحافة_ القاهرة

١٠ الجمعة ١ شوال ١٤٣٦ هـ ١٧ يوليو (أيار) ٢٠١٥ م ١٠ أبيب ٧٤٣١ ش العدد ١٩٢٨١ السنة ٦٢
AL AKHBAR 17 July 2015



حَادِثةٌ مُرعِبَةٌ أسفلَ عَجَلَاتِ المِتْرُو

Overview:
Metro Safety System

A real-time system to detect when a person crosses or touches the yellow line on the subway (metro) platform, triggering an alert to ensure safety

Training Datasets

Dataset 1: Station Platform and Detecting Braille Blocks 2

- **Source:** Roboflow
- **Classes:** railway track, braille blocks, platform, stop braille blocks
- **Focus for Project:** railway track, stop braille blocks
- **Purpose:** Defines stop braille blocks or safety line that must not be crossed

Training Datasets

Dataset 2: Person

- **Source:** Roboflow
- **Classes:** person
- **Purpose:** Enables detection of individuals on the subway platform, ensuring the system accurately monitors human activity

Inference Data

For real-time operation, we used video footage from a CCTV camera placed at the metro station to run inference on our trained model

Data Preprocessing

Image Resize

Roboflow outputs all images in one size 640

Augmentation

Yolo auto augmentation

Line & Track Detection

People Detection

Annotated Video Processing

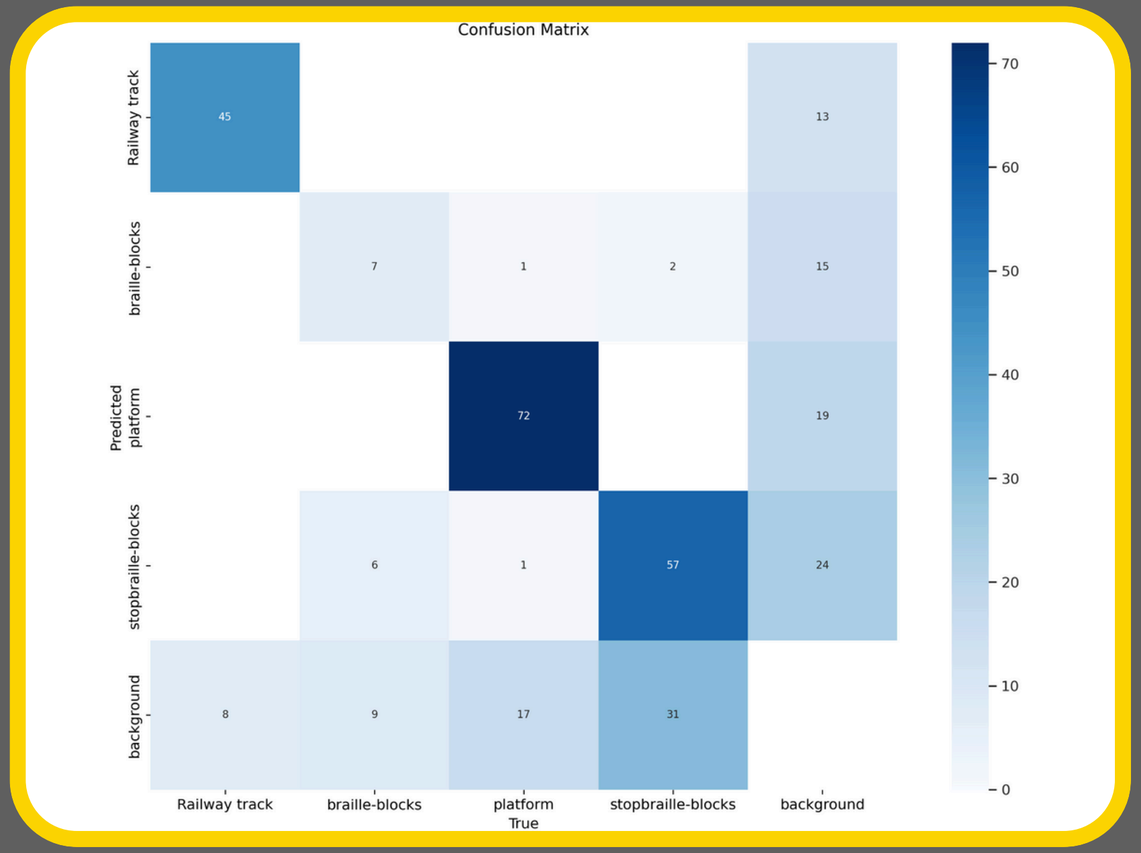
Methodology

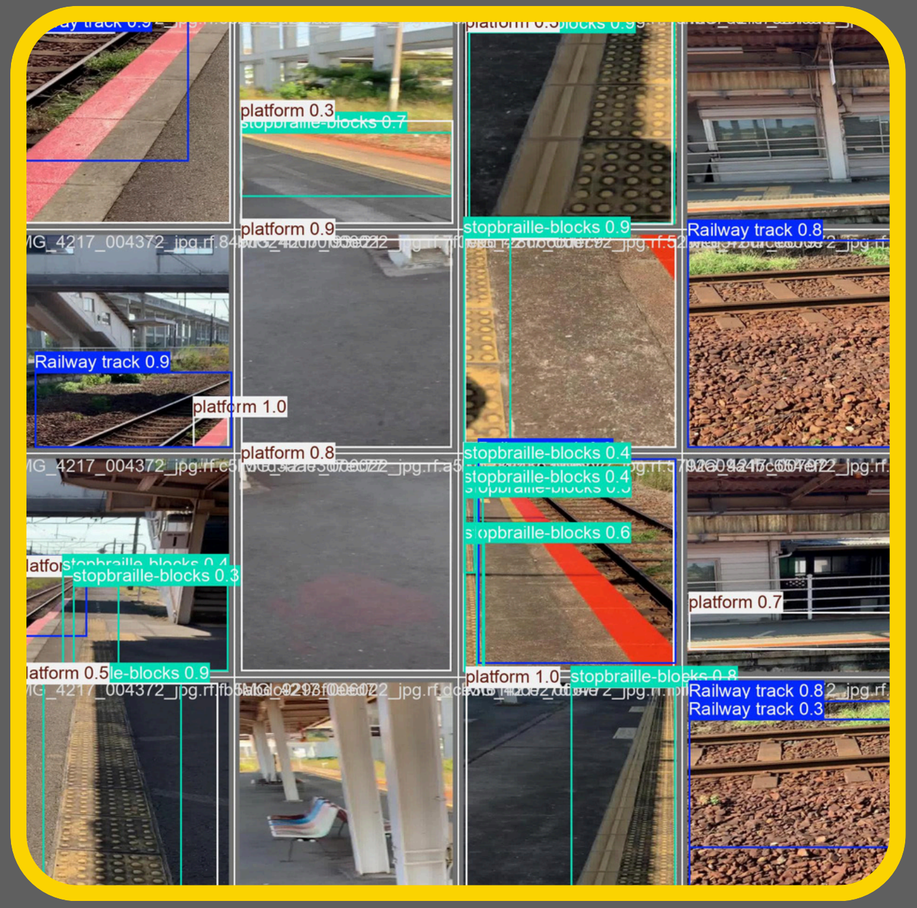
Methodology:
Line and Track Detection

Dataset: Station Platform and Detecting Braille Blocks 2
Model : Yolov8m

Methodology: Line & Track Detection

Visualize Yolo Results





Methodology: People Detection

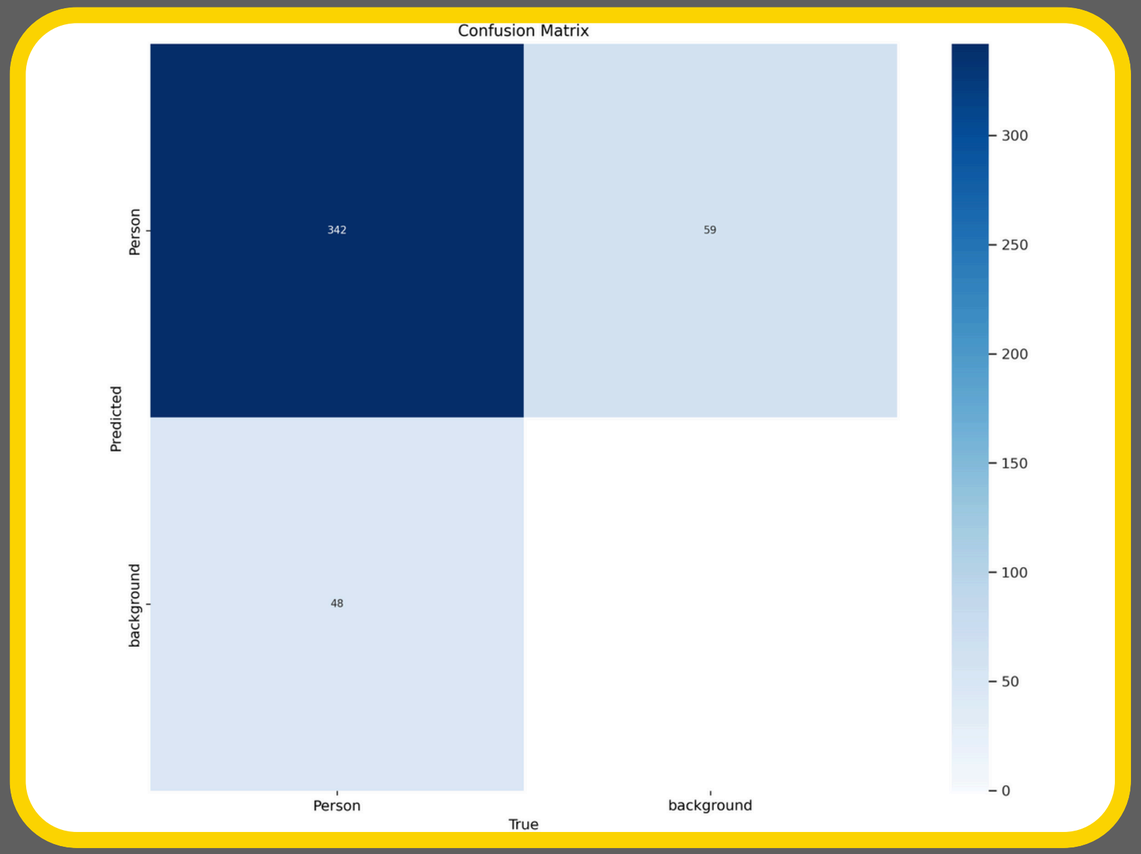
Dataset: Person

Model : Yolov8s

Trained on very well annotated person dataset to crowd on platform

Methodology: People Detection

Visualize Yolo Results



Methodology:
Annotated Video Processing

1. Color Segmentation

Inside the boundaries of the bounding box, color segmentation is applied to detect the precise location of the yellow line.

2.Masking

A line is drawn on the color segmentation masks to make the detected yellow line more visible.

3 .Line Detection

The algorithm identifies the start and end points of the yellow line within the bounding box.

4.Railway Detection

The algorithm checks if the railway is detected using the YOLO model.

Methodology:
Annotated Video Processing

5.Danger Zone Identification

If the railway is detected, it signals that the danger area is active.

6.Platform Side Specification

Based on the relative positions of the railway and yellow line bounding boxes, the algorithm determines on which side the platform is located.

7.Foot Position Detection

The algorithm finds the foot position of each person based on their bounding box.

8.Line Position Comparison:

The foot position is compared to the yellow line's position.

9.Danger Zone Check

If the person crosses the line while the danger zone is active (i.e., railway detected), the person is marked as being in a hazardous area.

Methodology:
Annotated Video Processing

10.Annotation

Annotation Color:

- **Red:** If the person passes the line during the active danger zone.
- **Green:** If the person stays behind the line or the danger zone is inactive.

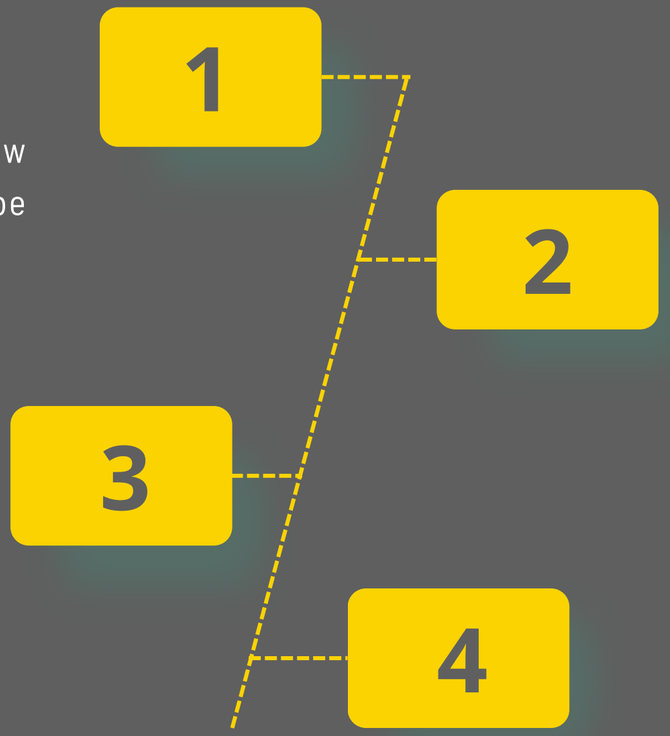
Key Features

Handling Unclear Yellow Line Detection:

A variable saves the last known yellow line coordinates when the line can't be clearly detected.

Railway Detection Validation:

A condition ensures the railway is considered real only if detected in 3 consecutive frames, reducing false positives.



Comprehensive Masking for Metro Platforms:

A large masking range is applied to cover most metro platform line colors, with auto-completion using the detected start and end points.

Alert System Output:

- Red Annotation: Persons detected in the danger zone are marked with a red border.
- Counter: A counter updates frame by frame, tracking the number of persons in the danger zone.

Deployment

Large Model Sizes:

- Models were uploaded to Dropbox in a downloadable format.
- The code was updated to download the models to specified paths before execution.

Input/Output Handling:

- Utilized Google Drive for input video storage, simplifying the process.
- The input video link is received from the website, downloaded into the workspace, and processed.
- Output video (with alerts) is saved to a constant path that the website reads.

Deployment

Endpoints:

- Implemented simple POST and GET methods.
- Takes a video URL as input, processes it, and returns the output video file.

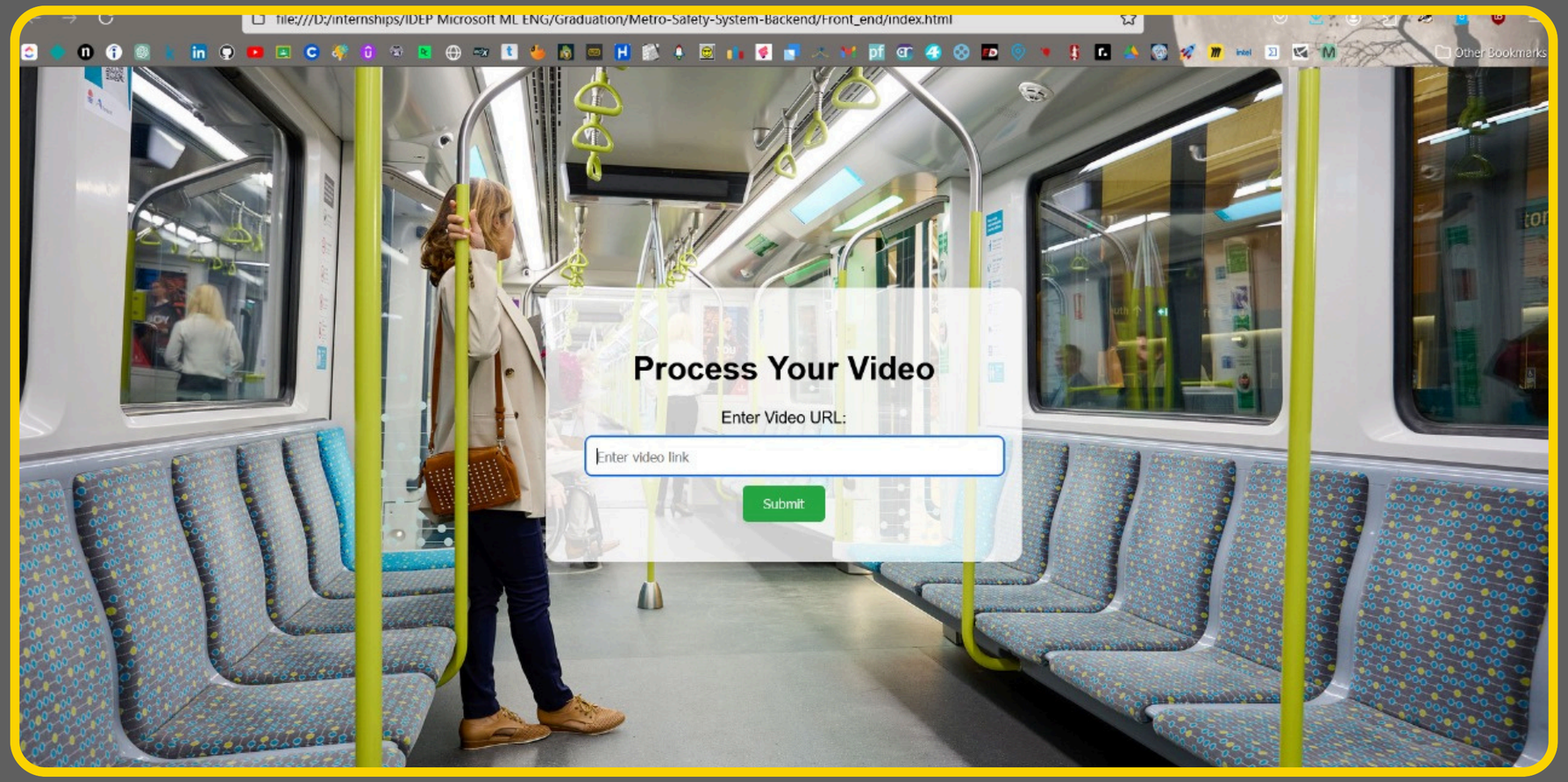
Project Hosting and APIs:

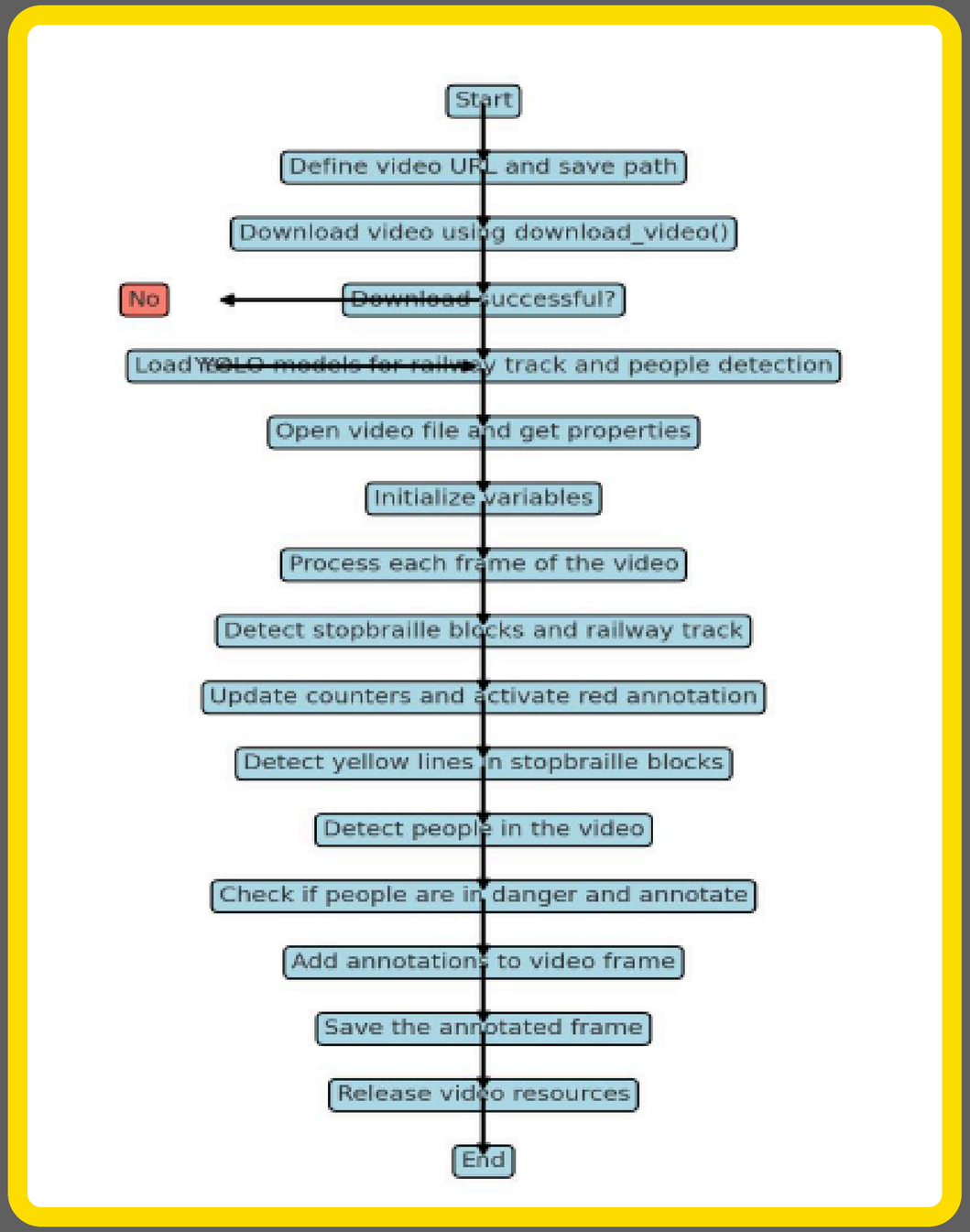
- The project was uploaded to GitHub.
- Used Microsoft Azure Virtual Machine for deployment and API creation.

Deployment

GUI:

- A simple website was created to explain the project and its features.





System Flow Chart

Our Way to Success:

Approaches Tried but Didn't Work

Approach 1: Fixed Line with Only Train and Person Detections

- User was asked to draw a fixed line on the screen.
- Problems:
  - Hand-drawing inaccuracies.
  - Line perspective issues from the camera view (not always straight).
  - Camera movement caused false alerts.
  - Train detection was unreliable due to its fast movement, confusing the models.

Our Way to Success:

Approaches Tried but Didn't Work

Approach 2: Person, Train, and Line Detection

- Added detection for persons, trains, and line bounding box.
- Problems:
  - Train detection was still unreliable.
  - Line detection only provided a large bounding box without precise location.
  - Even manual annotations through RoboFlow did not improve results.

Our Way to Success:
Approaches Tried but Didn't Work

Approach 3: ResNet Segmentation for Line with Object Detection for Train and Person

- Achieved better results with line detection using ResNet segmentation.
- Problems:
  - Limited detection success due to insufficient data.
  - Training required large datasets (300 photos weren't enough).
  - Tried using a pre-trained model but didn’t achieve desired accuracy.

Our Way to Success:

Approaches Tried but Didn't Work

Approach 4: YOLO Model for Normal and Danger Zone Person Detection

- Annotated data into two classes: normal person and danger zone person, along with train detection.
- Trained YOLO model to differentiate between normal and danger zone situations.
- Problems:
  - Dataset was too small for the model to learn distinguishing features.
  - The model couldn't reliably detect when people were in the danger zone or normal zones.

MEET OUR TEAM!

Hanin Sherif

Ahmed Fawzy

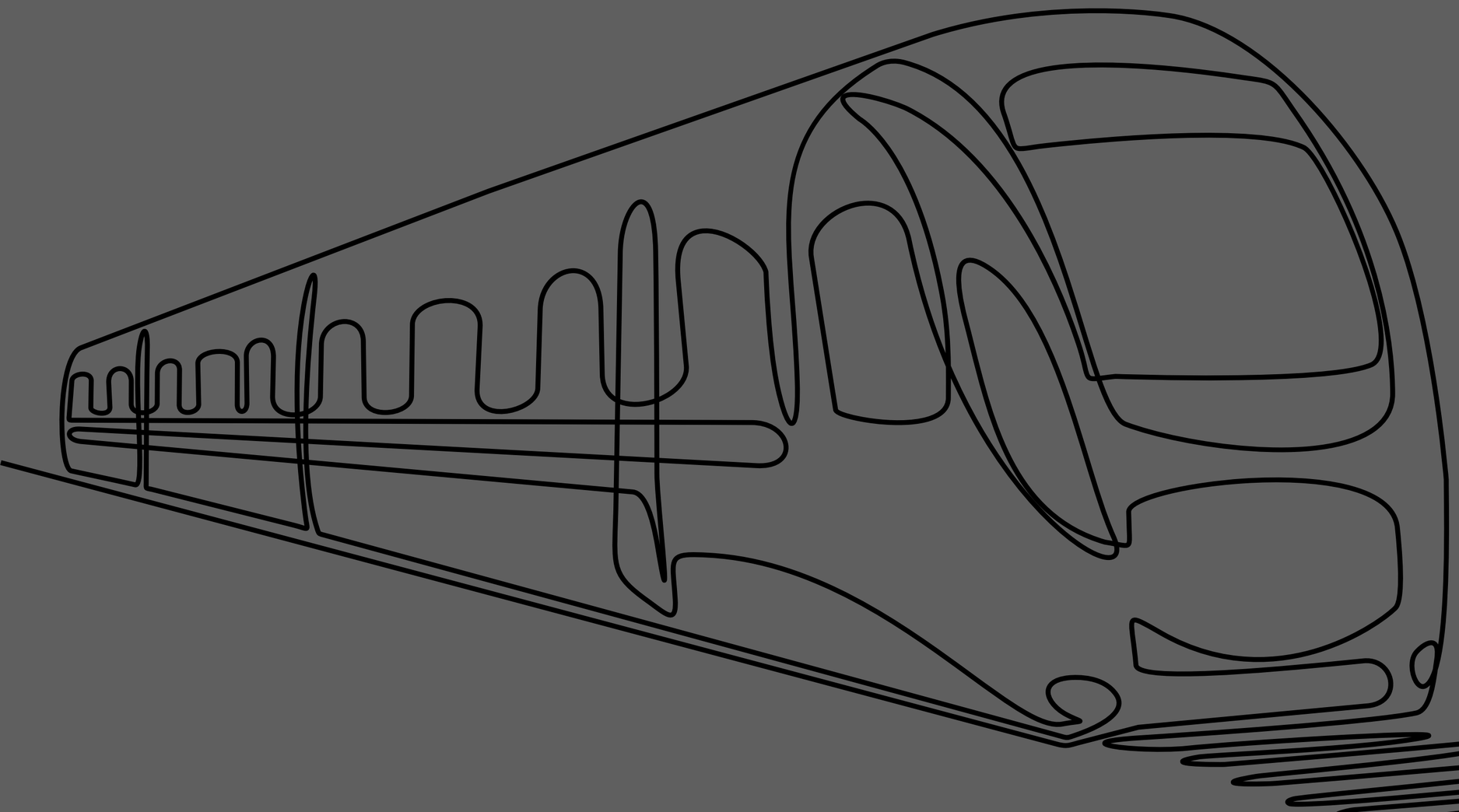
Mahmoud Elsayes

Ibrahim Ahmed

Ahmed Gabr

Yousef Elzahaby

THANK YOU!



STAY SAFE