IE Bag Detector

Deep Learning Project

IE University

Professor Ruben Zazo Candil

Group A

Cecilia Crepaldi

Joel Prithvi Ignatius

Juan Sebastian Chombo Quiñones

Kangjie Yu

Mikel Albisu Astigarraga

Nina Tensak

Vinay Palankar

Contents

Abstract

The IE Bag Detector project addresses the significant issue of theft targeting IE University students in Madrid, who are easily identifiable by their distinctive IE-branded bags. Utilizing advanced deep learning techniques, our application monitors entrances and exits of key locations through security cameras, detecting IE bags and alerting security personnel to potential thefts.

The project involved meticulous data collection and annotation, model training using YOLO v8, and deployment on an Android platform via TensorFlow Lite. With promising initial results and plans for dataset expansion and enhanced testing, the IE Bag Detector aims to significantly enhance campus security and reduce theft incidents, setting a foundation for future technological advancements in security measures.

Colab Notebook link:
https://colab.research.google.com/drive/1ecNawDMLb9F1k4nPL3r7gFi7KnJs9Suv

I. Introduction

The IE Bag Detector project emerged as a response to the rising issue of theft targeting IE University students in Madrid. This problem has become particularly pressing due to the distinctive nature of IE-branded bags carried by a significant number of international students, who are perceived as high-net-worth individuals. These students, easily identifiable by their branded bags, have become prime targets for thieves.

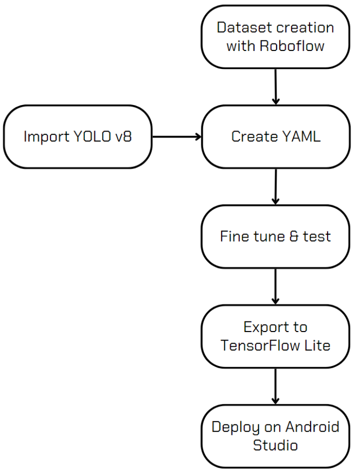
Addressing this issue, our project aims to leverage advanced deep learning techniques to detect IE bags through security camera feeds, thereby enhancing security measures at key university locations and reducing theft incidents.

Our solution is based on an object detection application designed to monitor the entrances and exits of specific locations, comparing the data to detect any inconsistencies that might indicate theft. The application employs state-of-the-art object detection technology to recognize IE bags.

When a bag is detected upon entry but not upon exit, the system triggers an alert to notify security personnel of a potential theft. This approach not only increases the efficiency of security operations but also acts as a deterrent to would-be thieves, knowing that advanced surveillance technology is in place.

II. Methodology

The workflow involves several key steps to develop and deploy an object detection model for mobile application. We created a dataset with Roboflow, which was used to annotate the images and create the dataset for our model training. We then fine-tuned and tested the YOLO v8 model with our dataset to ensure it meets our performance criteria. After fine-tuning, the model is exported to TensorFlow Lite format for compatibility with mobile platforms. Finally, the TensorFlow Lite model is deployed on Android Studio to enable real-time object detection on mobile devices, ensuring practical and accessible use for end users.

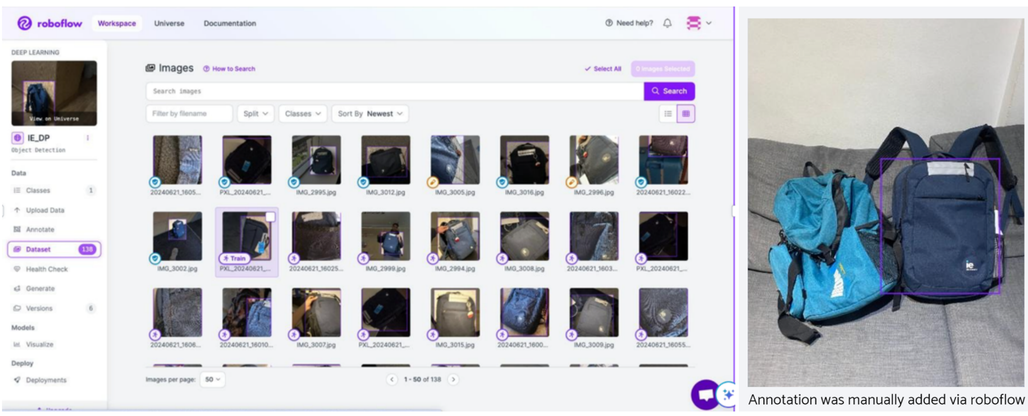


workflow of the project

III. Data Collection and Annotation

To create our own dataset, we took 138 images in total of the IE bag manually, placed in different conditions, including being covered by other object and different lightings.

Using Roboflow, we manually annotated the images, ensuring a diverse dataset that included various lighting conditions, positions, and dimensions of the bags. We augmented the rest of the images using techniques such as rotation, greyscaling, and adding noise. In total, we had 332 images for our model, which we divided into training, validation, and test datasets, containing 70%, 20%, and 10% of the images, respectively.



Annotating the images

IV. Model Development and Deployment

The chosen model for this task was YOLOv8s, a highly efficient and accurate object detection framework known for its performance in real-time applications.

1. Transfer Learning

We utilized transfer learning with YOLOv8s to train a model for detecting IE bags and differentiating the IE bags from other similar objects. The process involved importing the dataset from Roboflow, configuring paths and creating a data.yaml file for training, and loading a pre-trained YOLOv8 model.

```python
from ultralytics import YOLO
import torch

model = YOLO('yolov8n.pt')

def freeze_layer(trainer):
    model = trainer.model
    num_freeze = 10
    print(f"Freezing {num_freeze} layers")
    freeze = [f'model.{x}.' for x in range(num_freeze)]  # layers to freeze
    for k, v in model.named_parameters():
        v.requires_grad = True  # train all layers
        if any(x in k for x in freeze):
            print(f'freezing {k}')
            v.requires_grad = False
    print(f"{num_freeze} layers are freezed.")

model.add_callback("on_train_start", freeze_layer)
model.train(data="/content/datasets/Yolo-2/data.yaml",epochs=10)
```

the code snippet above demonstrates loading the YOLOv8 model

2. Model Training

```
!yolo task=detect \
mode=train \
model=yolov8s.pt \
data={dataset.location}/data.yaml \
epochs=50 \
imgsz=640
```

Training the model with customised dataset

The model was fine-tuned by trying different parameters and ways. For instance, we froze the initial layers to retain pre-trained weights while training the remaining layers on the new dataset, but it does not make a dramatic difference while deploying it into the application. The model training was configured with specific parameters, and post-training, the model was saved. Finally, we tested the model using a test dataset to evaluate its performance.

3. Results

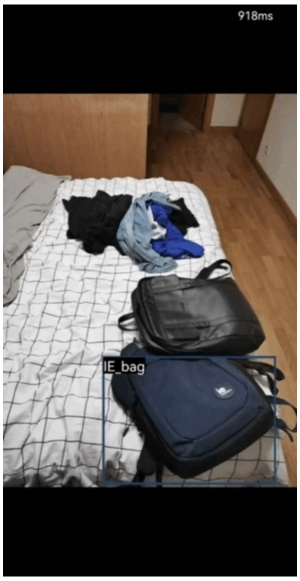
We obtained the following results from testing the YOLOv8 model in the validation dataset:

Images	Box(P)	Recall	mAP50	mAP50-95
28	0.929	0.964	0.968	0.637

The model demonstrated strong object detection capabilities with a balanced performance in precision and recall, indicating effective detection and localization of objects.

4. Deployment

Once the model was trained and validated, we exported it to TensorFlow Lite format, suitable for mobile deployment. The next phase involved integrating this model into an Android application.



Deploying the model in Android Studio

We cloned a YOLOv8 repository for Android Studio from Github and replaced the default model file with our custom TensorFlow Lite model and the txt file which indicates the number and name of the classes.

Ensuring compatibility and proper functioning, we meticulously configured the application's dependencies and tested it on an Android device. This deployment phase was critical to ensure that the model could operate effectively in real-world conditions, providing real-time detection and alerts.

V. Enhancing Model Performance

The performance of the IE Bag Detector has been promising, showing high accuracy in detecting IE bags under various conditions. However, we recognize the potential for further improvements. Expanding the dataset to include more images and varied scenarios will enhance the model's robustness. Using a larger model of yolov8, like yolov8x, might also improve the performance of the model and decrease the latency when deploying it in an application.

Extensive real-world testing, particularly in frequently visited areas by IE students, along with a systematic feedback loop, will refine the model based on practical challenges. Additionally, deploying the model on edge devices for real-time detection and establishing a robust monitoring system will ensure continuous performance improvement and reliability.

VI. Conclusion and Future Directions

In conclusion, the IE Bag Detector project represents a significant advancement in campus security technology. By integrating deep learning and real-time object detection into a user-friendly mobile application, we provide a practical solution to the pressing issue of bag theft among IE students.

The project not only demonstrates the potential of AI in enhancing security measures but also sets a foundation for future developments that can adapt to evolving security needs. A particularly promising direction is integrating the bag detector with a face recognition feature, linked to a repository of known faces in IE records. This would enable the system to identify individuals associated with the detected bags and cross-reference them with the university's database.

Additionally, implementing an alerting system that notifies security personnel in real-time about unauthorized bag movements or unidentified individuals could significantly enhance campus safety. Extending the application to monitor other valuable personal items and integrating it into a broader security app for students could further increase its utility. With continuous improvements and rigorous testing, the IE Bag Detector aims to become an indispensable tool in ensuring the safety and security of the university community.