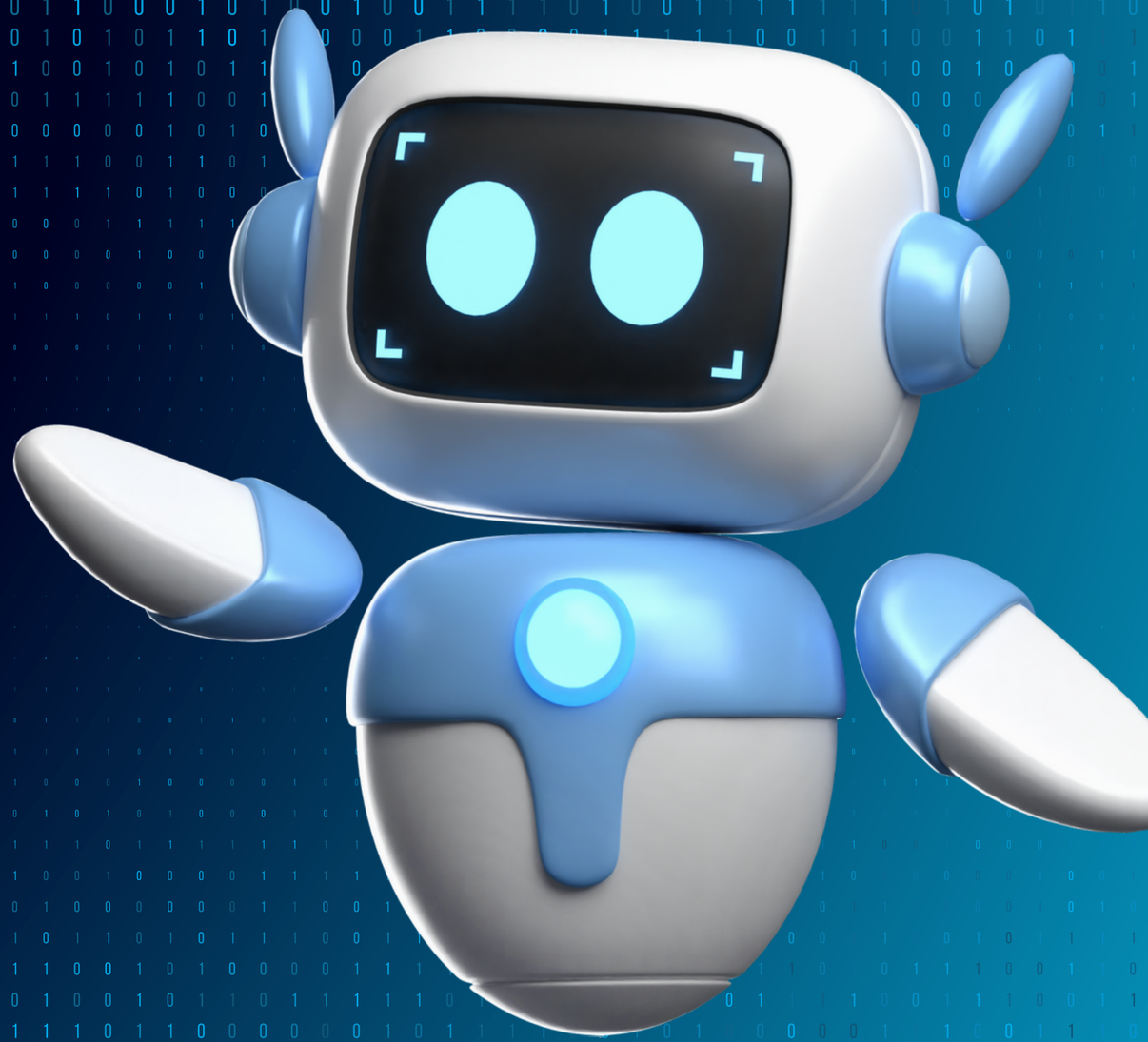
YVR HACKATHON PRESENTATION

FearlessTrailblazers

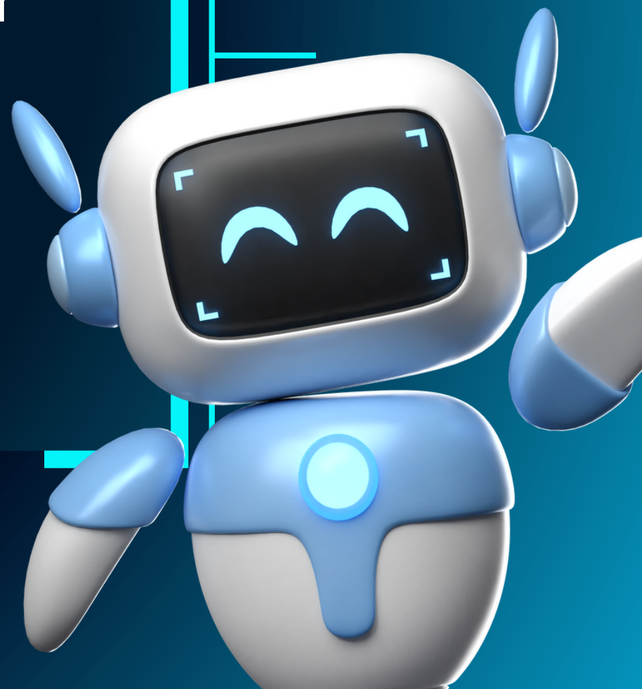
Team: Shikha, Abdul, Bruce, Pahul



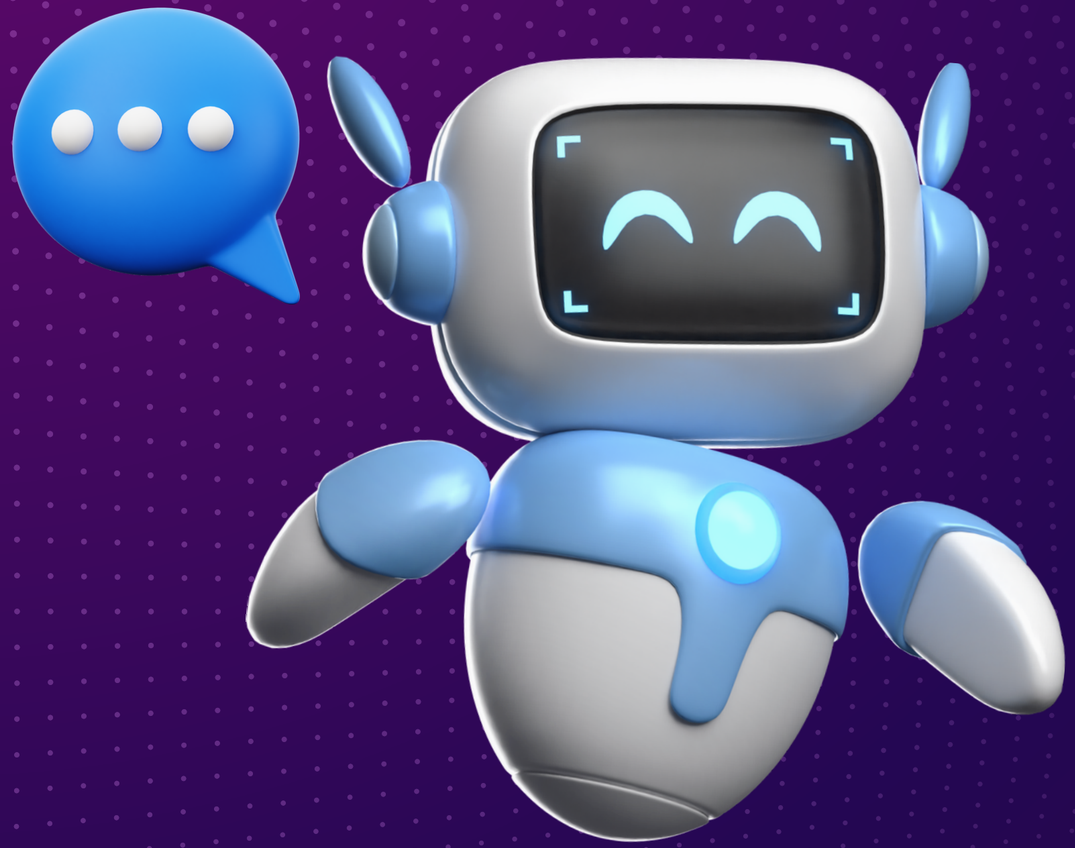
CONTENTS

1 OBJECTIVE

Utilize intelligent sensors, computer vision, and machine learning for real-time insights into cleanliness, unattended baggage, and passenger volume.



2 PROPOSED SOLUTION



LEVERAGE YOLOV7

TRAIN SPECIALIZED MODELS

TARGETED DATASETS

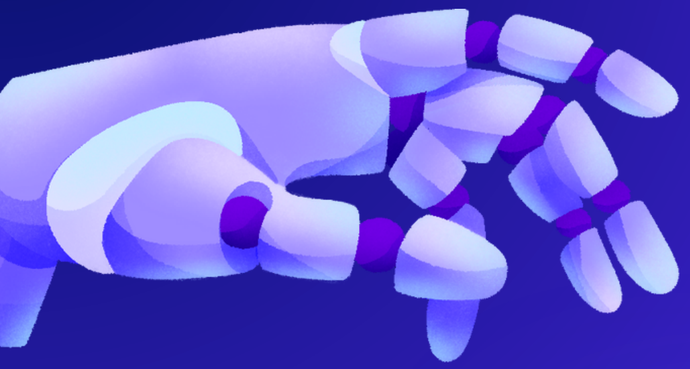
3 THE FEARLESSTRAILBLAZERS PROPOSE A SOLUTION LEVERAGING ADVANCEMENTS IN OBJECT DETECTION TO ENHANCE THE CROW'S NEST SYSTEM

MACHINE LEARNING MODEL

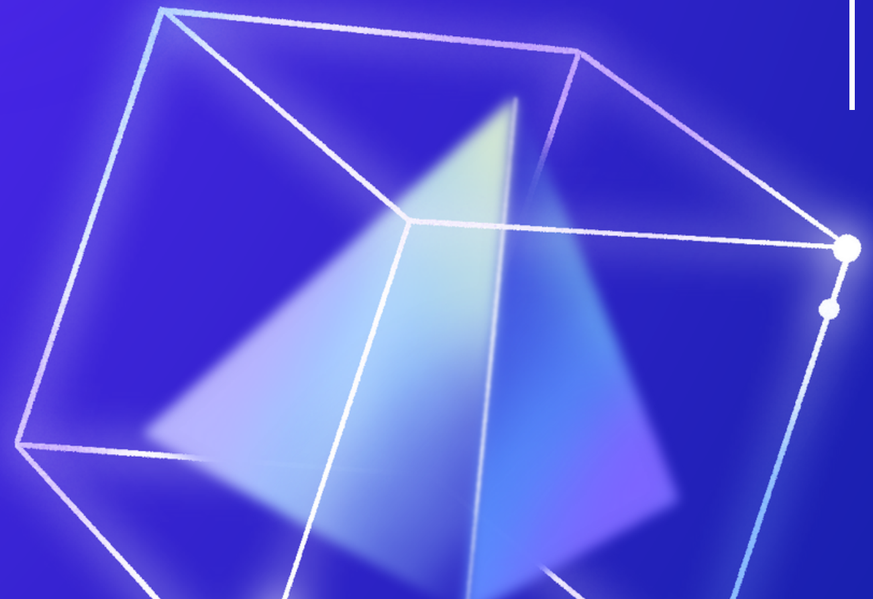
We will train a YOLOv7 object detection model for trash detection using the TACO dataset and a similar approach for other datasets.

TARGETED DATASETS

TACO: Trash Images (Fine-grained)
COCO: Objects (Bags/Luggage)
AVIDBAGS: Attended/Unattended Bags



CODE OVERVIEW



Set Up Environment

This section installs necessary Python packages for the project, including PyTorch and the Ultralytics YOLOv5 library, which are essential for machine learning and object detection tasks

```
!pip install torch torchvision
!pip install ultralytics
```

Download the TACO Dataset

This part clones the TACO dataset repository from GitHub, changes the current directory to the TACO directory, and then runs a script to download the dataset images and annotations.

```
!git clone https://github.com/pedropro/TACO.git
%cd TACO
!python download.py
```

5 PREPARE DATASET FOR TRAINING

SETTING THE DATASET PATH

```
dataset_path = './data/annotations.json'
dataset_dir = os.path.dirname(dataset_path)
```

LOADING ANNOTATIONS

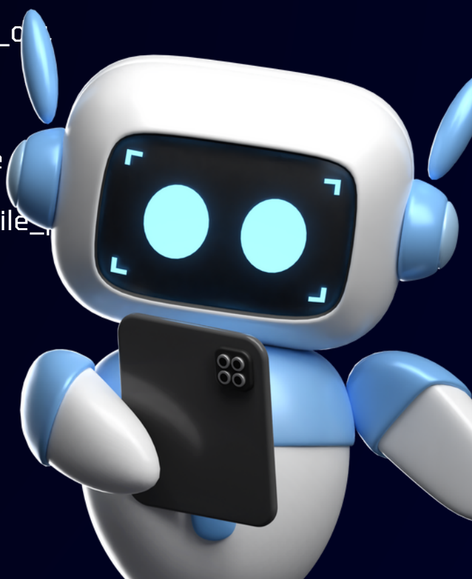
```
# Load annotations with open(dataset_path, 'r') as f:
annotations = json.load(f) nr_images =
len(annotations['images'])
```

ITERATING OVER IMAGES

```
for i, image in enumerate(annotations['images']):
file_name = image['file_name'] url_original =
image['flickr_url'] file_path = os.path.join(dataset_d
file_name)
```

DOWNLOADING AND SAVING IMAGES

```
if not os.path.isfile(file_path):# Load and Save Image
= requests.get(url_original) img =
Image.open(BytesIO(response.content)) img.save(file_
```



GENERATE YOLO-FORMATTED LABELS

5

LOADING ANNOTATIONS

```
with open('/content/TACO/data/annotations.json') as f: data = json.load(f)
```

CREATING CATEGORY NAMES LIST

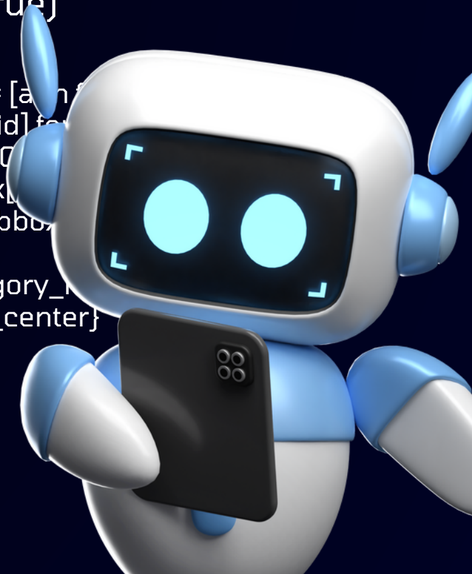
```
category_names = [cat['name'] for cat in sorted(data['categories'], key=lambda x: x['id'])]
```

CREATING LABELS DIRECTORY

```
labels_base_dir = '/content/TACO/labels'
os.makedirs(labels_base_dir, exist_ok=True)
```

WRITING YOLO-FORMAT LABELS

```
with open(label_path, 'w') as label_file: annotations = [a
ann in data['annotations'] if ann['image_id'] == img_id]
in annotations: bbox = ann['bbox'] x_center = (bbox[0
bbox[2] / 2) / img['width'] y_center = (bbox[1] + bbox[
img['height'] width = bbox[2] / img['width'] height = bbo
img['height'] class_id =
category_names.index(data['categories'][ann['category_
1]['name']) label_file.write(f"{class_id} {x_center} {y_center}
{width} {height}\n")
```



CONFIGURE YOLOV5 FOR TRAINING

5

GENERATING CLASS NAMES

```
class_names = [f'class{i+1}' for i in range(60)]
```

CONVERTING CLASS NAMES TO YAML FORMAT

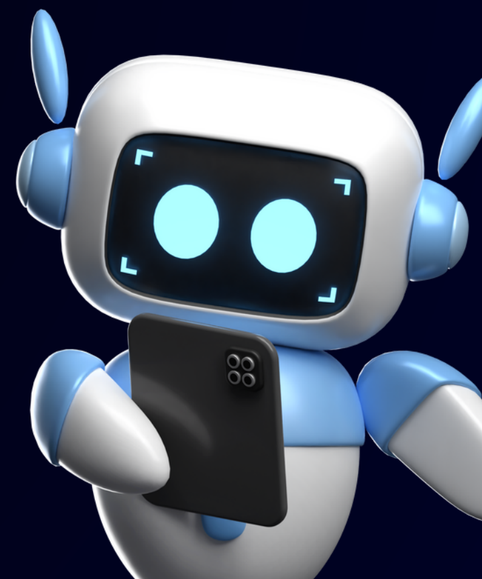
```
class_names_str = str(class_names)
```

CREATING YAML CONTENT

```
yaml_content = f""" train: /content/TACO/train val:
/content/TACO/val nc: 60 names:
{class_names_str} """
```

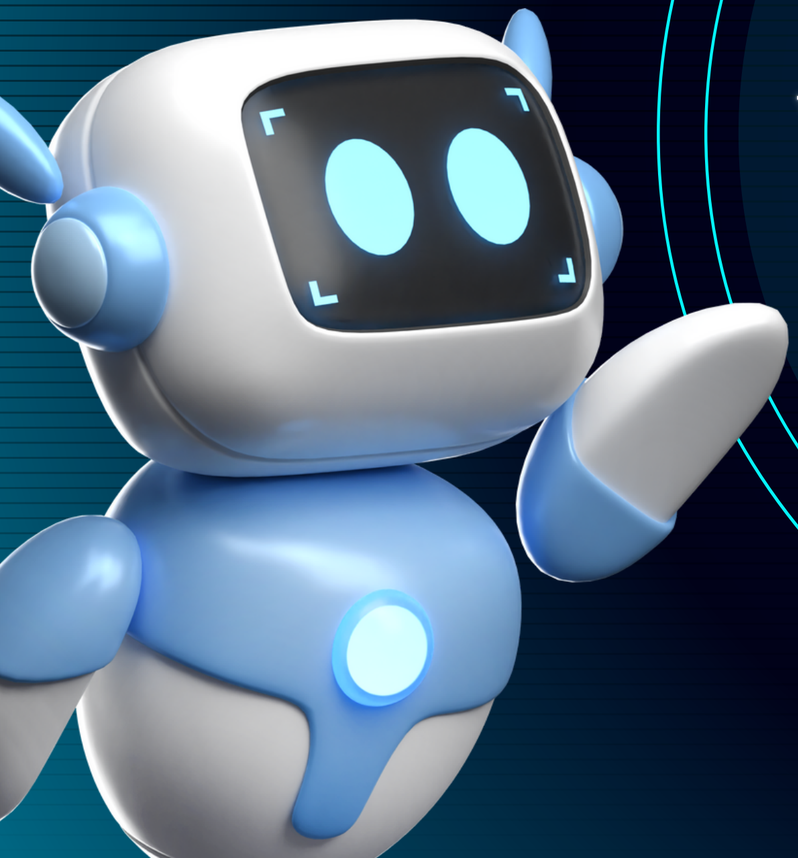
WRITING YAML CONTENT TO FILE

```
with open('taco.yaml', 'w') as f:
f.write(yaml_content.strip())
```



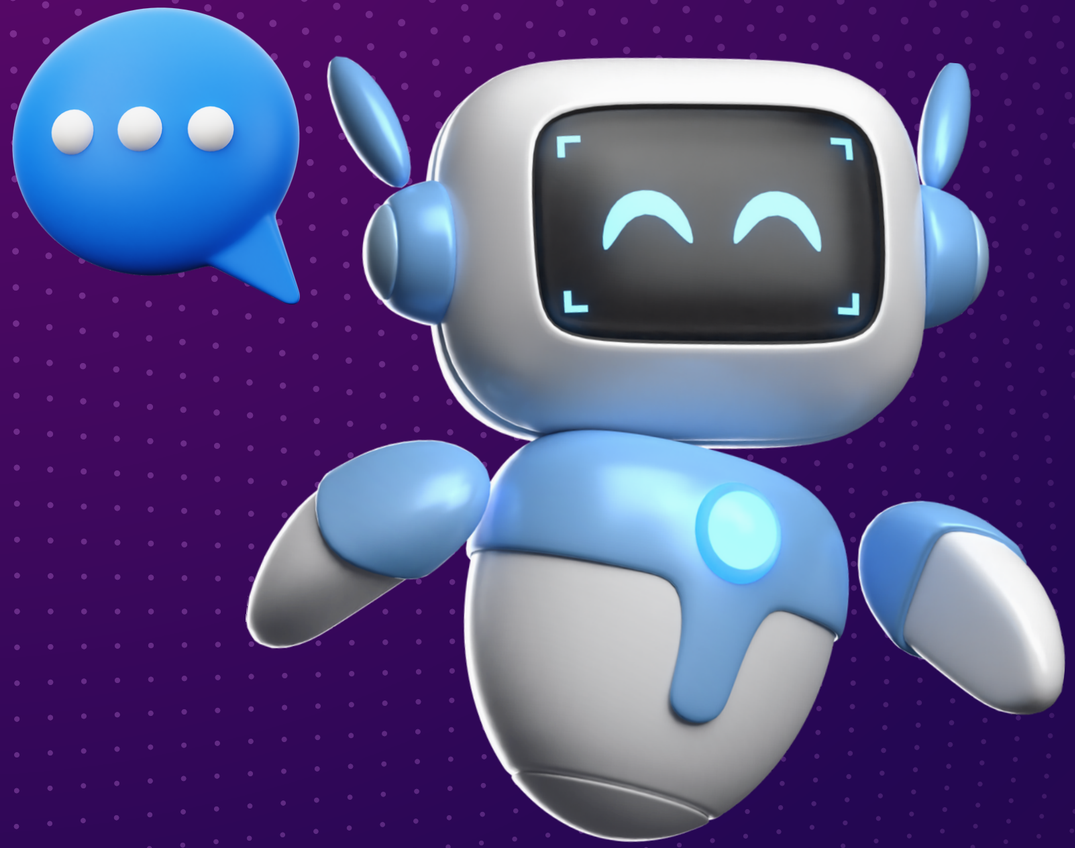
INITIATE MODEL TRAINING

```
!PYTHON TRAIN.PY --IMG 640 --BATCH 16 --EPOCHS 10 --DATA
TACO.YAML --CFG MODELS/YOLOV5S.YAML --WEIGHTS YOLOV5S.PT
--NAME TACO_MODEL
```



Example Output- by using YOLOv7

2 BENEFITS



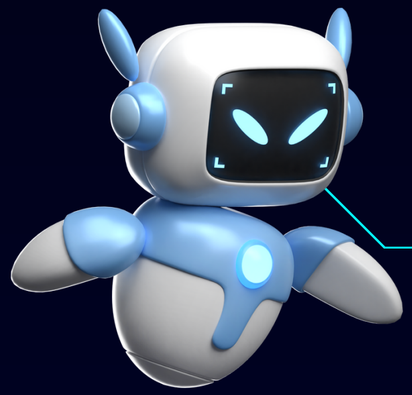
REAL-TIME AWARENESS: DETECT ISSUES & VOLUME INSTANTLY.

AUTOMATED MONITORING: FREE CROW'S NEST FOR CRITICAL TASKS.

SUPERIOR ACCURACY: MACHINE LEARNING OUTPERFORMS MANUAL METHODS.

4 NEXT STEPS

REFINE THE YOLOV7 MODEL FOR OPTIMAL PERFORMANCE ON YVR-SPECIFIC DATA.



INTEGRATE THE MODEL AND CONDUCT PILOT TESTING WITHIN A DESIGNATED AREA

THANK YOU