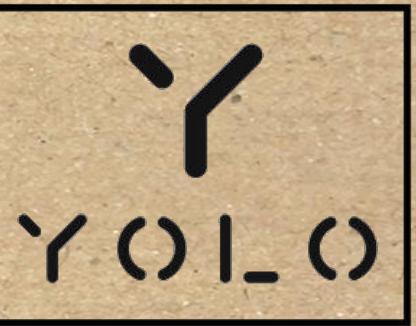


You Only Label Once (YOLO)

// April 2021 Capstone Project Presentation
Alex Li, Azwin Lam, Jason Lee



You Only Label Once

TABLE OF CONTENTS

PROJECT OVERVIEW 01

About Clearbot
Project Aim
Business Value

DATA HANDLING 02

Sources
Preprocessing
Data Overview

STRATEGY 03

Modelling
GUI Workflow

CHALLENGES 04

Data Gathering
Material Detection

WHAT'S NEXT 05

Improvement Ideas

CONCLUSION 06

Project Evaluation
References

01

PROJECT OVERVIEW



Clearbot

Clearbot is a swarm of trash collecting robots that use AI-Vision to detect and collect trash from water bodies. These robots are fully autonomous, solar-powered and work as a team to remove trash. In comparison to any current solution, Clearbot is 15x cheaper, has 5x more reach and removes 2x more trash daily (24x7x365).

PROJECT AIM

LABEL

Create a model that can label images of trash at ~80% precision & recall to improve on manual label speed.

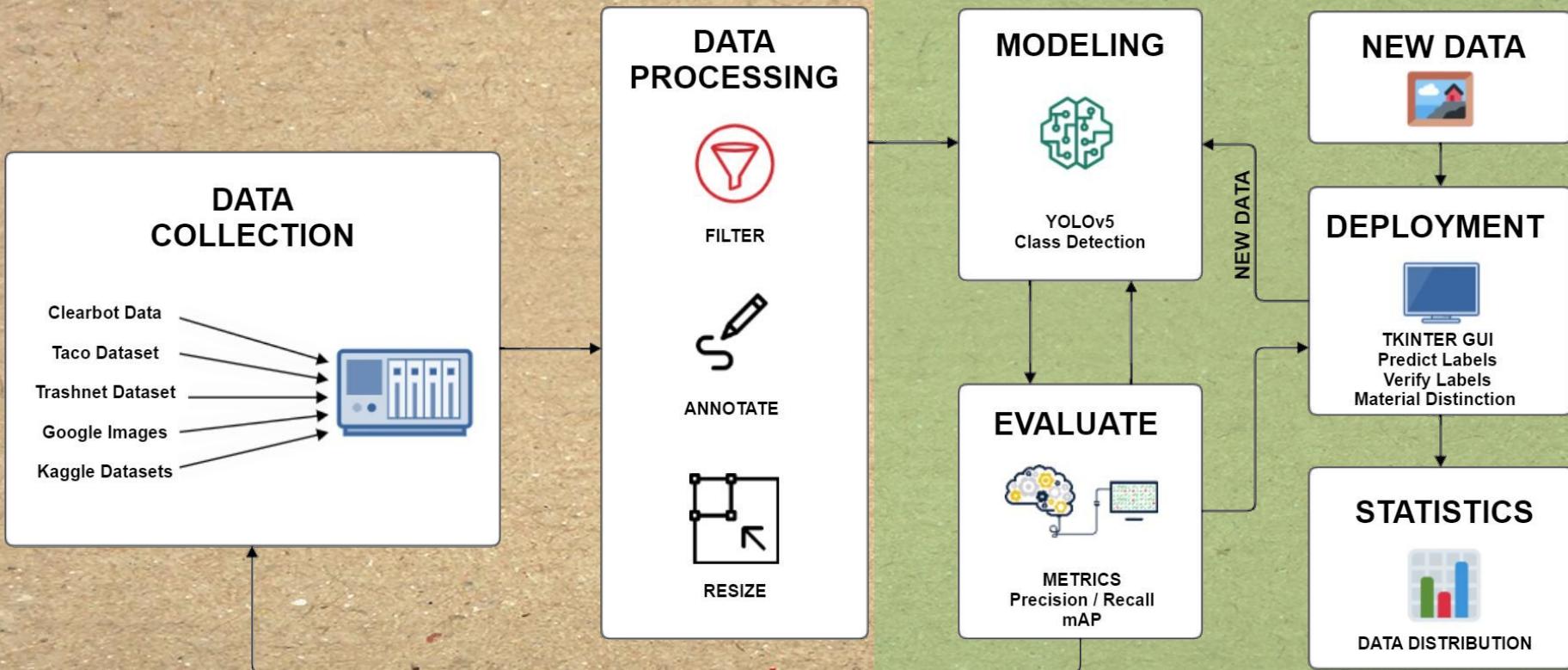
COUNT

Collect and output statistical information about distribution of input images.

BUSINESS VALUE

- Our solution will allow Clearbot to significantly reduce annotation time.
- Countless man hours saved via the tool will mean long term cost savings and allow more efficient deployment of manpower.

SYSTEM ARCHITECTURE



02

DATA HANDLING

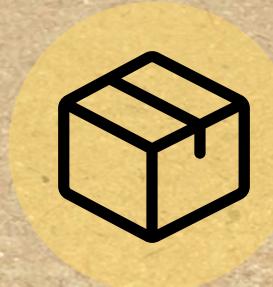
DATA SOURCES

GOOGLE IMAGES

- Python Webscraper found on GitHub

KAGGLE DATASETS

- Plastic Images
- Waste Classification



CLEARBOT DATA

- Data Collection provided by Clearbot

OTHER DATASETS

- TACO Dataset
- Trashnet Dataset

DATA PROCESSING

COLLECT DATA

Gather from various sources



ANNOTATE DATA

Use LabelImg to (re)annotate



FILTER DATA

Remove duplicate images and images that are too small

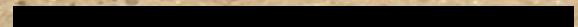
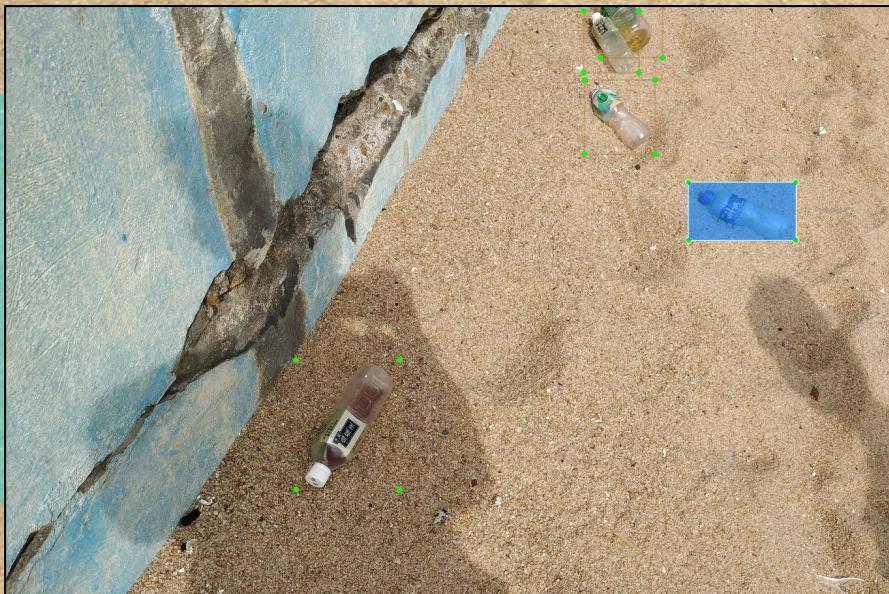


RESIZE DATA

Images resized for faster modeling time



SAMPLE DATA



DATA OVERVIEW

6

Classes

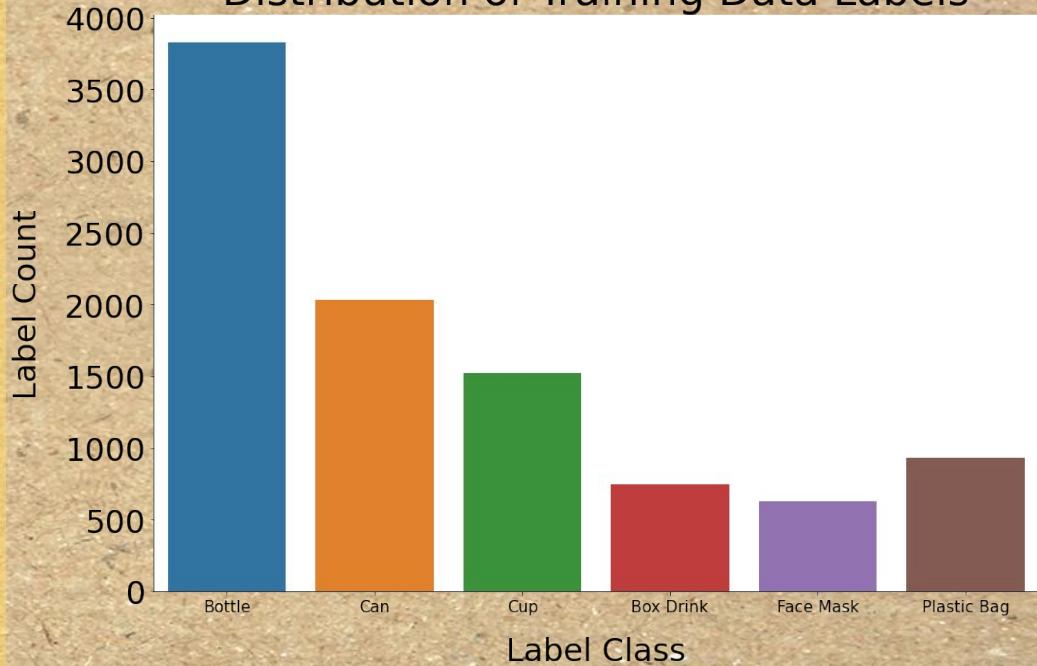
5,210

Unique Images

9,675

Labels

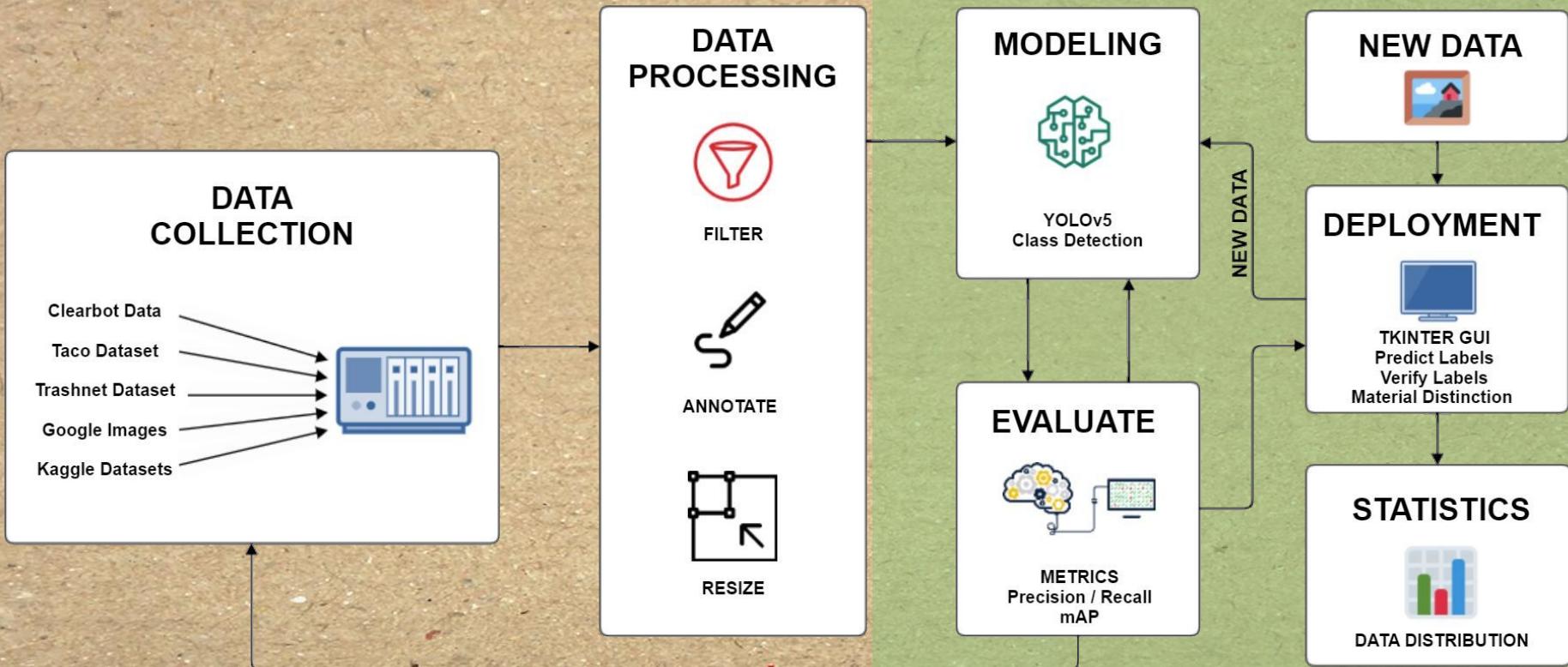
Distribution of Training Data Labels



03

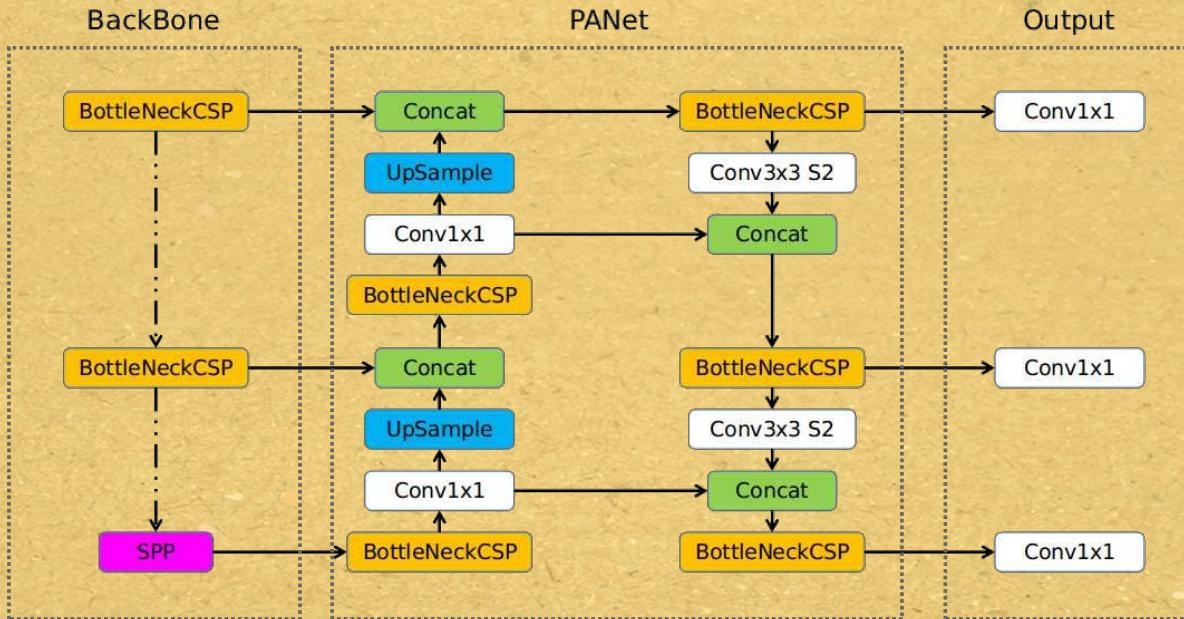
STRATEGY

SYSTEM ARCHITECTURE



MODEL ARCHITECTURE

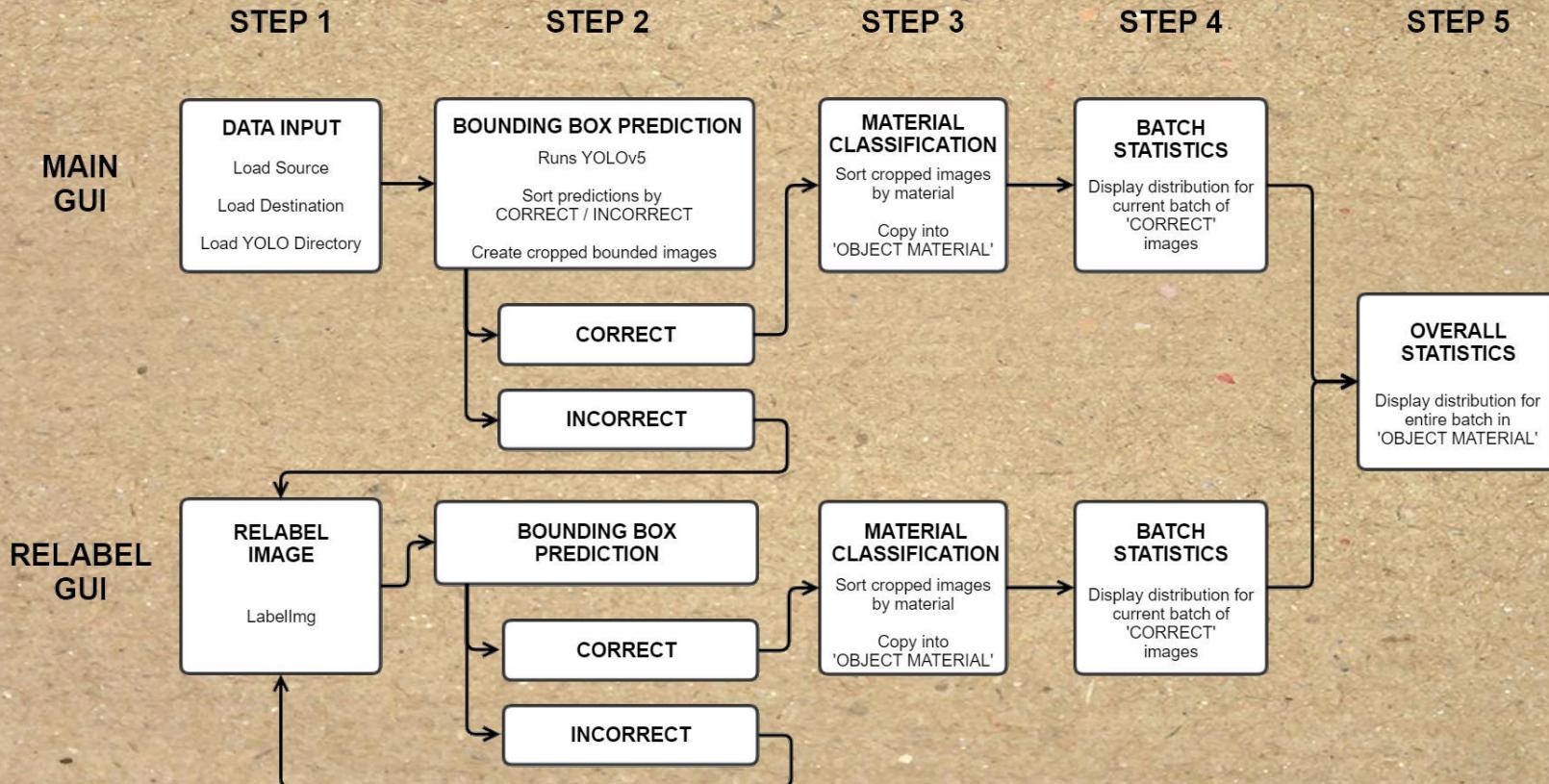
Overview of YOLOv5



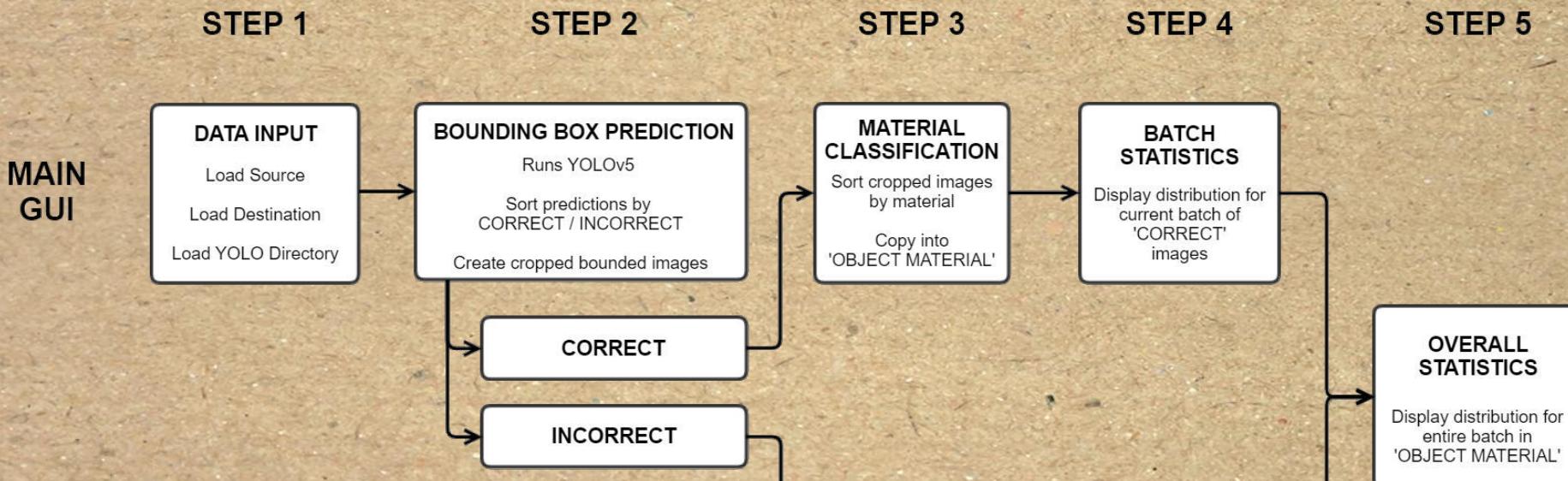
FINAL MODEL SELECTION

	PRECISION	RECALL	mAP	DETECT TIME (sec / image)	TRAINING TIME (per epoch)
YOLO5s	0.806	0.816	0.863	~ 0.2 - 0.3	~ 1 min
YOLO5m	0.836	0.84	0.899	~ 0.4 - 0.6	~ 2 min
YOLO5l	0.813	0.829	0.875	~ 0.8 - 1.2	~ 7 min

GUI WORKFLOW

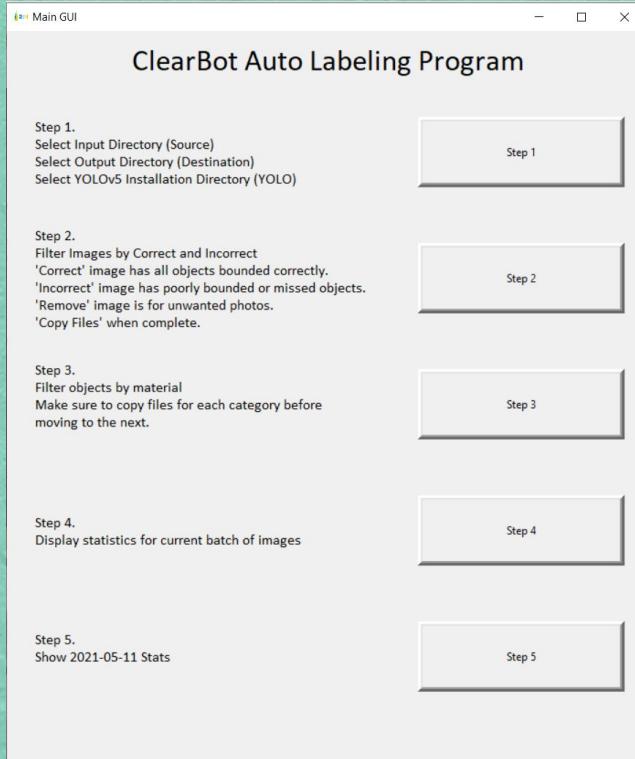


GUI WORKFLOW

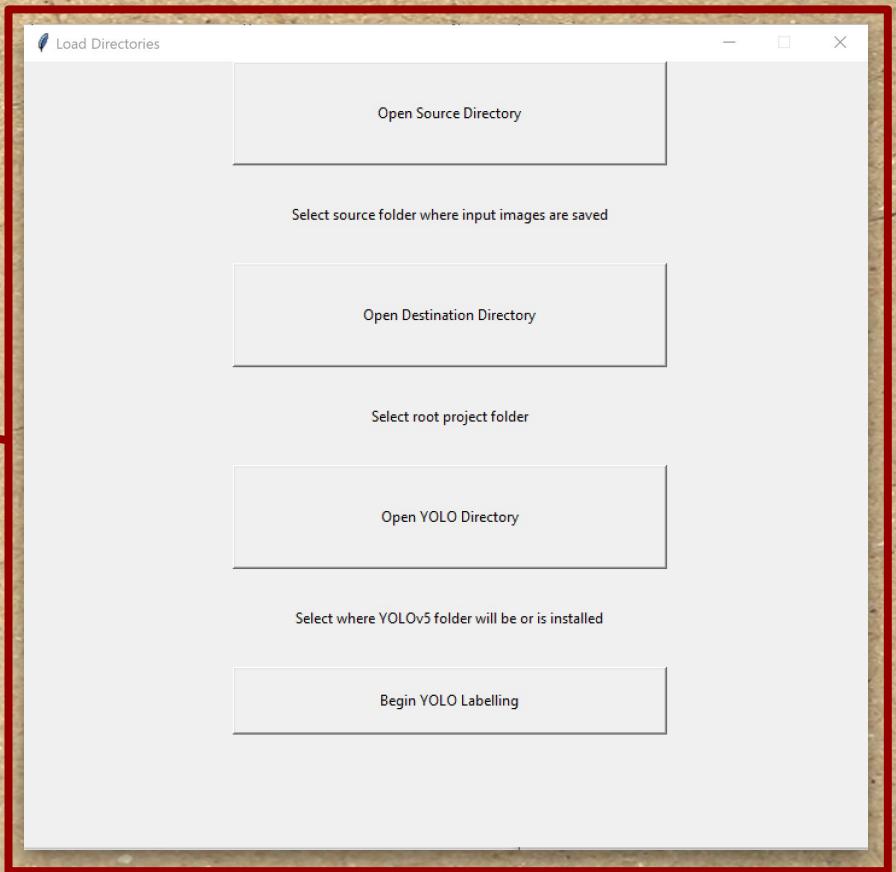
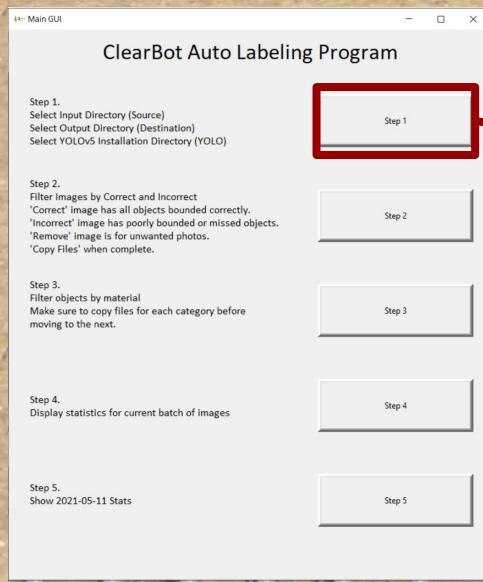


GUI WORKFLOW

5 Step Process



Step 1 - Load Folders



Step 1 - Load Folders

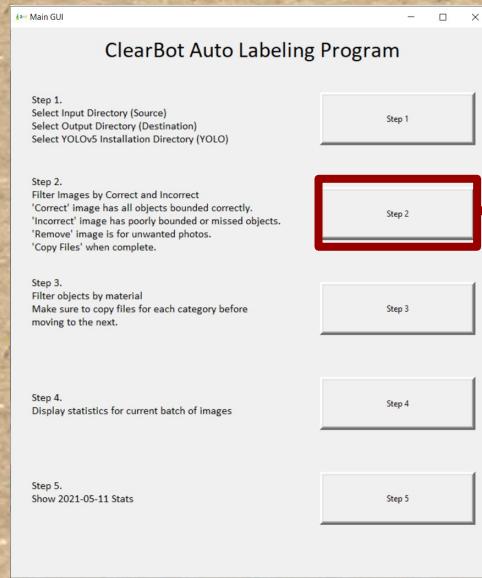
```
C:\Program Files\Python38\python.exe
load finished
YOLOv5 folder found!

-----
Renaming all source images for processing
100%|██████████| 8/8 [00:00<00:00, 10.78it/s]
-----
Converting all images to JPG format for processing
100%|██████████| 8/8 [00:00<00:00, 19.05it/s]
Source location is: C:/Users/raeky/Documents/python_work/group_project_clearbot/test/SOURCE
8 images found in source folder.

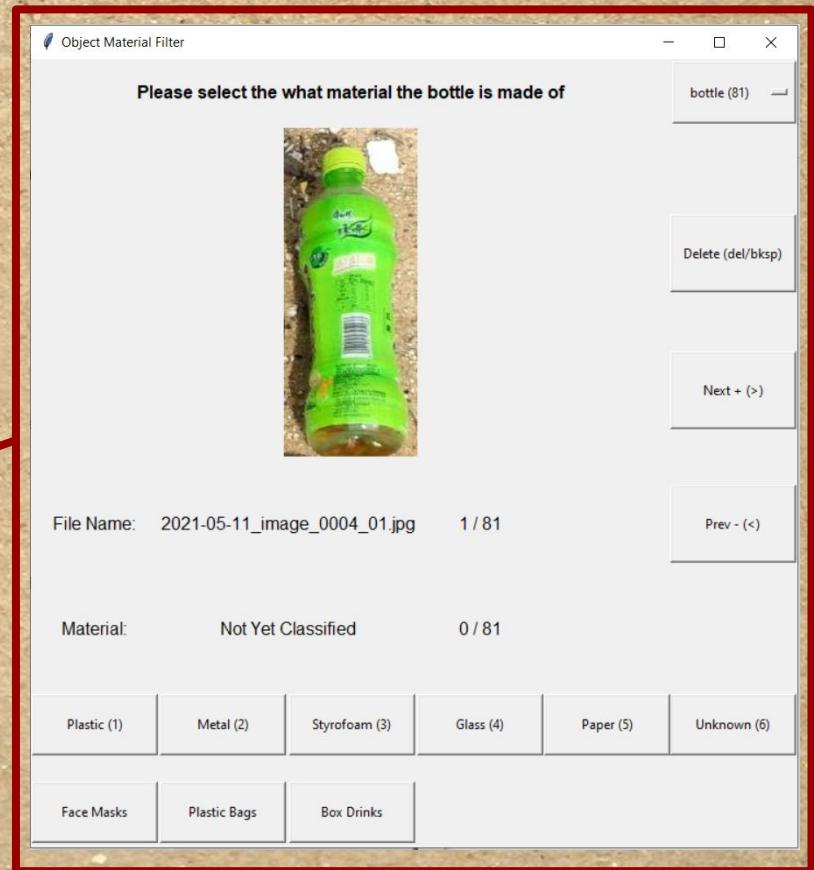
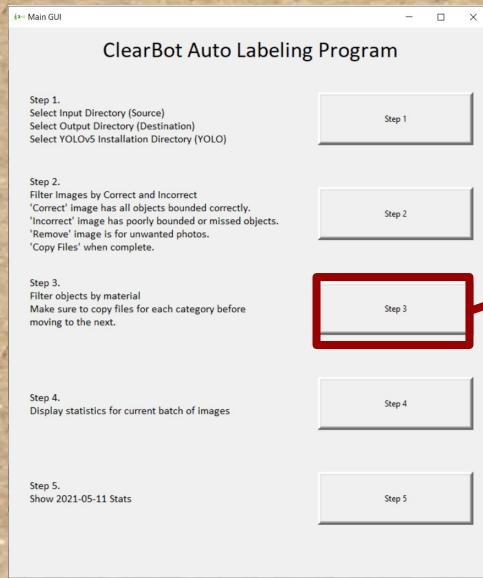
Namespace(agnostic_nms=False, augment=False, classes=None, conf_thres=0.6, device='', exist_ok=False, hide_conf=False, hide_labels=False, img_size=640, iou_thres=0.45, line_thickness=1, name='predictions_16_57', nosave=False, project='C:/Users/raeky/Documents/python_work/group_project_clearbot/test/2021-05-11/', save_conf=False, save_crop=False, save_txt=True, source='C:/Users/raeky/Documents/python_work/group_project_clearbot/test/2021-05-11/images/', update=False, view_img=False, weights=['C:\\\\Users\\\\raeky\\\\Documents\\\\python_work\\\\group_project_clearbot\\\\clearbot\\\\Prediction_Pipeline\\best.pt'])
YOLOv5 v5.0-68-gb18ca31 torch 1.8.1+cpu CPU

Fusing layers...
Model Summary: 308 layers, 21057843 parameters, 0 gradients, 50.4 GFLOPS
image 1/8 C:\\Users\\raeky\\Documents\\python_work\\group_project_clearbot\\test\\2021-05-11\\images\\2021-05-11_image_0001.jpg: 640x480 1 can, Done. (0.474s)
image 2/8 C:\\Users\\raeky\\Documents\\python_work\\group_project_clearbot\\test\\2021-05-11\\images\\2021-05-11_image_0002.jpg: 640x480 1 plastic bag, Done. (0.446s)
image 3/8 C:\\Users\\raeky\\Documents\\python_work\\group_project_clearbot\\test\\2021-05-11\\images\\2021-05-11_image_0003.jpg: 640x480 3 cups, Done. (0.424s)
```

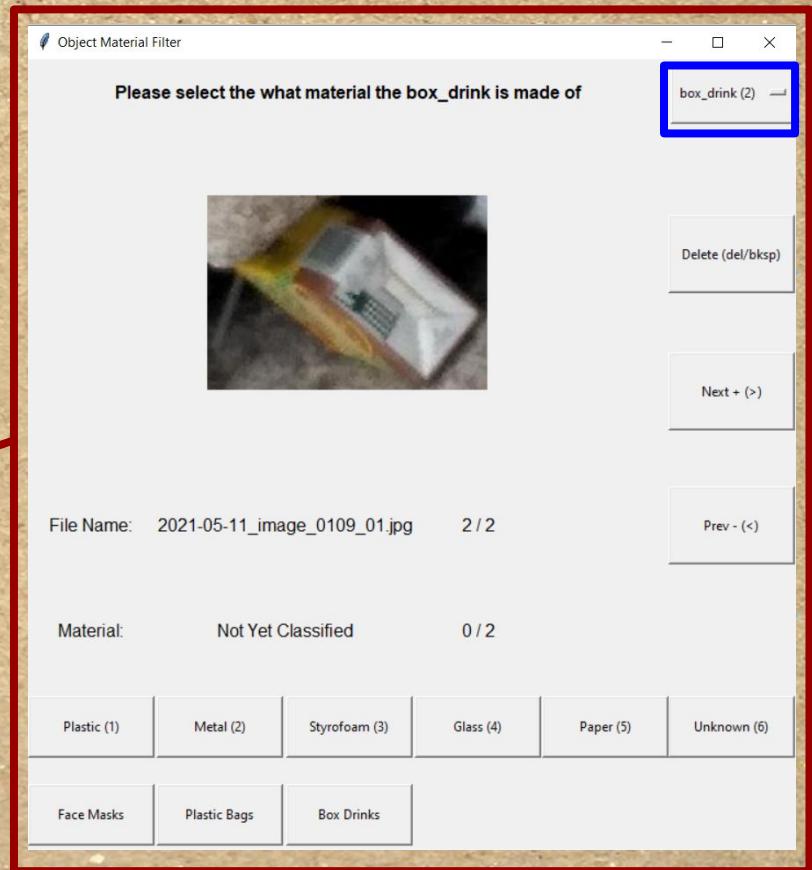
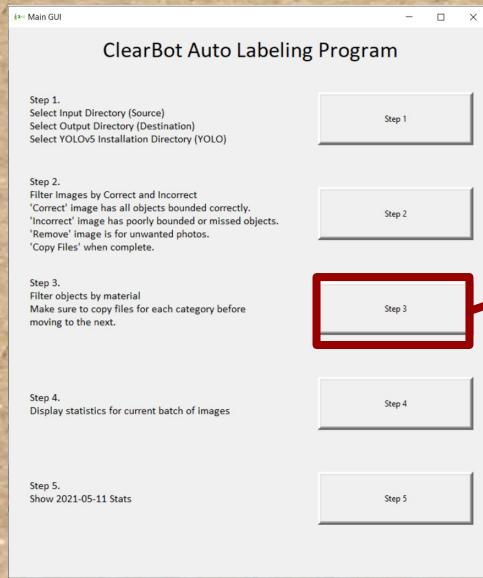
Step 2 - Image Review



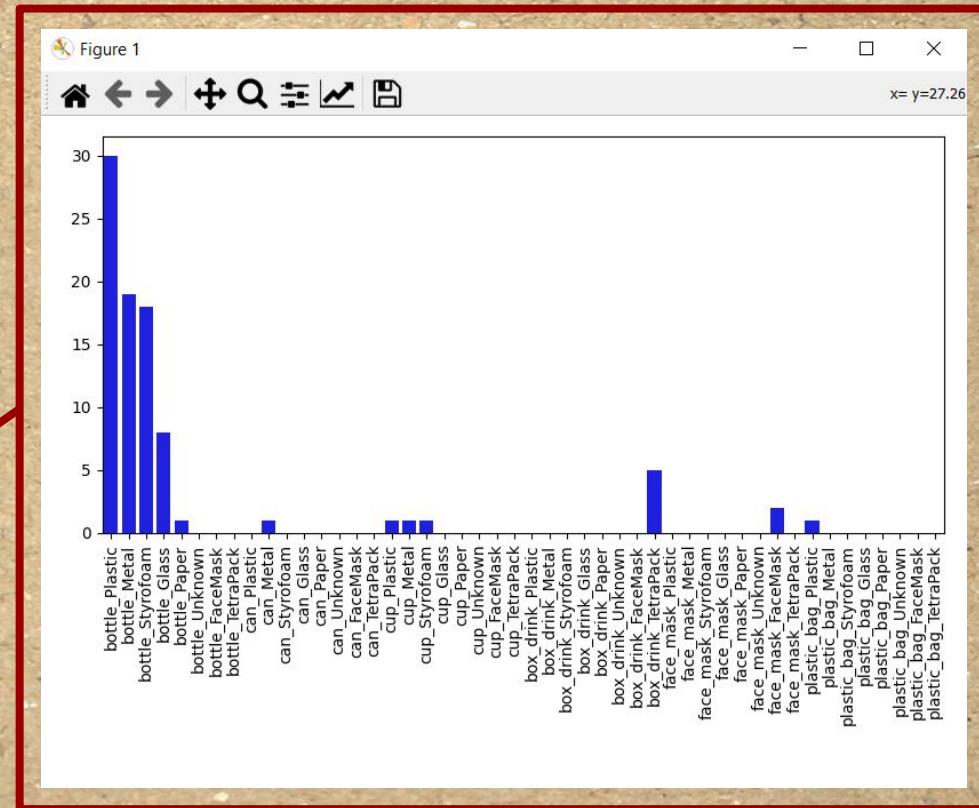
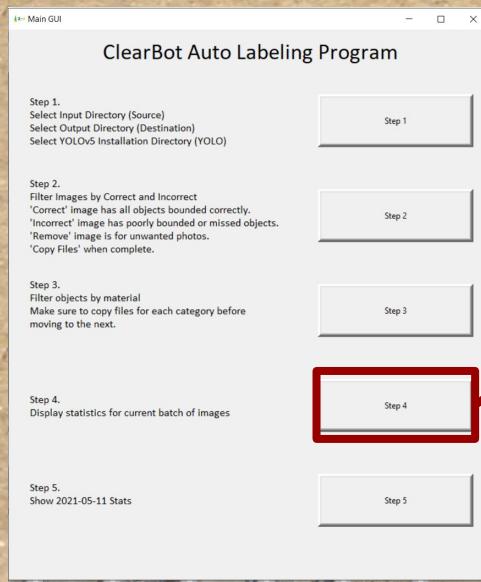
Step 3 - Material Filter



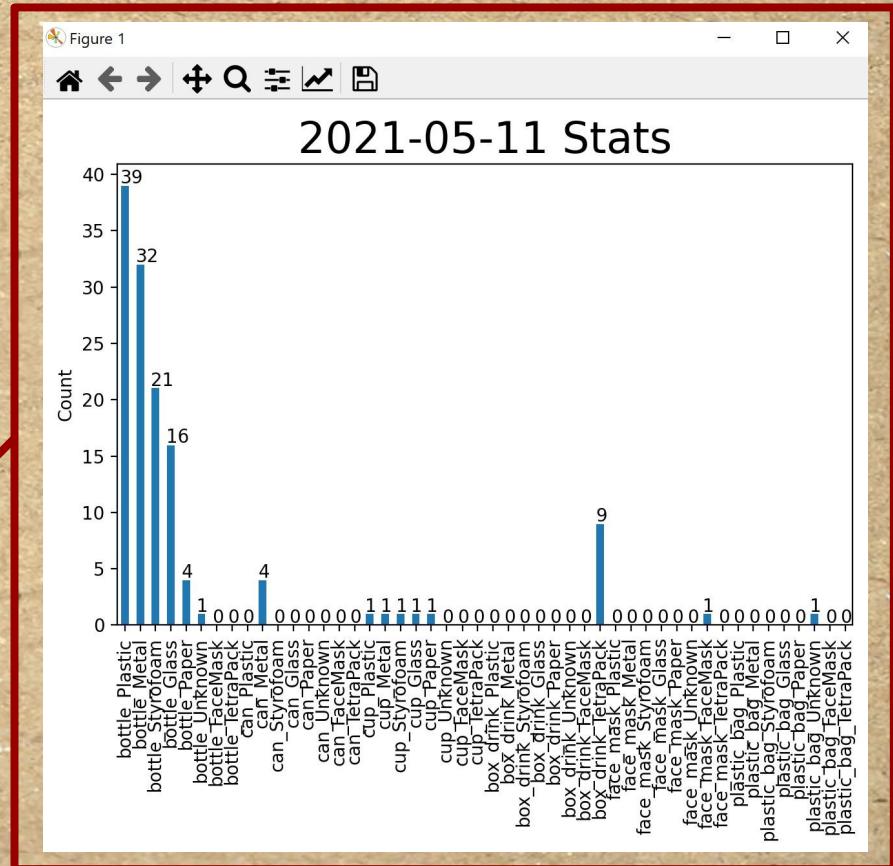
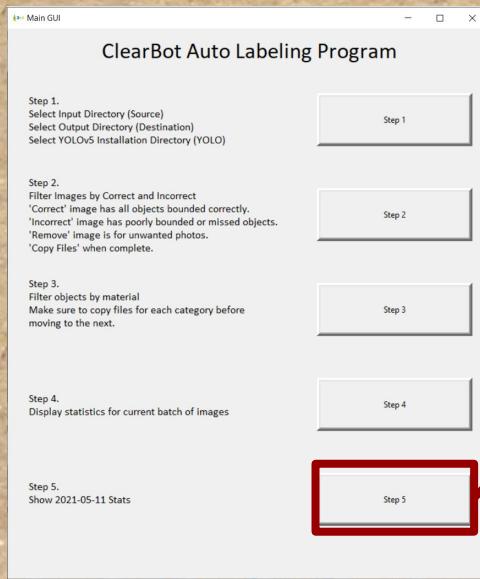
Step 3 - Material Filter



Step 4 - Display Stats



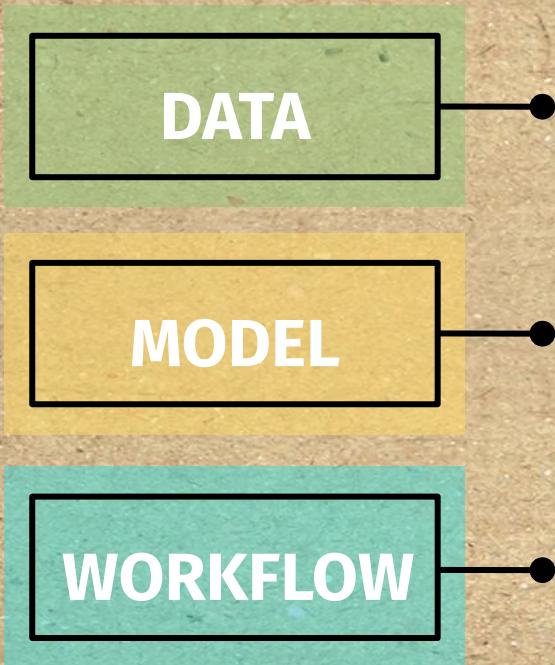
Step 5 - Display Stats



04

CHALLENGES

MAIN CHALLENGES



Difficult to find quality images of trash in sufficient volume. Limitations on variety of images found via search engines.

Material detection is not a mature technology yet. A lot of theoretical approaches that are difficult to replicate.

Not UI/UX or software engineers. Using relatively primitive module that focuses on getting the job done.

05

**WHAT'S
NEXT**

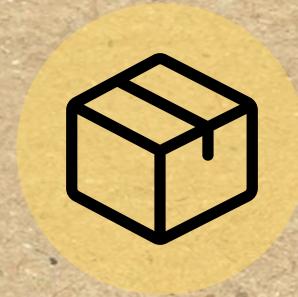
MOVING FORWARD

ALT MODEL

Material detection currently done manually. Develop, test and integrate a proven detection model.

RELABELING TOOL

Currently reliant on an external relabelling tool for incorrectly identified images. Build an in-workflow labelling tool as part of the GUI.



MORE CLASSES

Limited to 6 main classes. Given more time, can expand selection to include others like utensils.

MORE DATA

When Clearbot is able to gather more data with their boat, any collected data can be re-fed into the model for improved overall performance.

06 CONCLUSION

PROJECT AIM RECAP

LABEL

Create a model that can label images of trash at ~80% precision & recall to improve on manual label speed.

COUNT

Collect and output statistical information about distribution of input images.

PROJECT EVALUATION

LABEL

Using YOLOv5, our object detection model predicts bounding boxes at ~84% precision and ~84% recall.

COUNT

Successfully deployed a GUI for counting batch statistics and overall statistics on input images.

WORKFLOW EFFICIENCY

TEST DATA



289 images



105 fully correct
74 to relabel
110 to remove



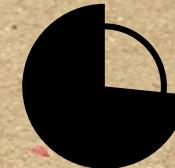
~15 minutes to
process and
predict

TIME SAVINGS



58%

20s per picture
~35 minutes



72%

30s per picture
~53 minutes

KEY MESSAGE

Our aim is to create a tool that can help Clearbot save on labelling time.
Time saved is money saved.

Our created workflow can label and process photos at least 2x faster.
Workflow can be adapted for other datasets.

REFERENCES

Modelling

YOLOv5: <https://github.com/ultralytics/yolov5>

Webscraper

Image Scraper: <https://github.com/debadridtt/Scraping-Google-Images-using-Python>

Labelling Tools

LabelImg: <https://github.com/tzutalin/labelImg>

Datasets

Plastic Images: <https://www.kaggle.com/nandinibagga/plastic-images>

Waste Classification Data: <https://www.kaggle.com/techsash/waste-classification-data>

Taco Dataset: <http://tacodataset.org/>

TrashNet Dataset: <https://github.com/garythung/trashnet>

Research Papers

Material Recognition:

https://openaccess.thecvf.com/content_cvpr_2015/papers/Bell_Material_Recognition_in_2015_CVPR_paper.pdf

OUR TEAM



Alex Li

- Data Collection
- Data Preprocessing
- Model Build: YOLOv5
- Code Review



Azwin Lam

- Data Preprocessing
- Model Deployment: Tkinter GUI
- GitHub repository management
- Code Review



Jason Lee

- Data Collection
- Data Preprocessing
- Model Research
- Model Deployment: Tkinter GUI



You Only Label Once

A labelling workflow created for Clearbot, a local startup, that speeds up productivity by accurately predicting bounding boxes for trash collected by their robot.



Business Value

Time saved is money saved. Our solution provides an easily adaptable workflow for any company that needs to annotate large volumes of data that can speed up their overall process by at least a factor of 2.

Data Collection

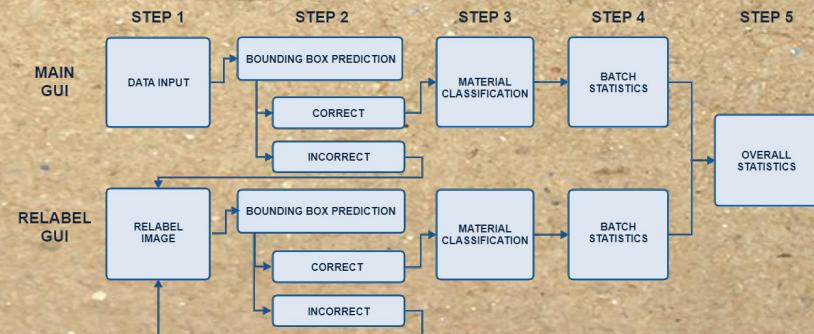
5200+ images split into six classes, with 9000+ total labels. Data gathered through web scraping and online datasets.

What's Next

Build an in-house labelling tool to increase performance and have a smoother workflow.
Deploy workflow on a more aesthetically pleasing GUI.



GUI Snapshot



YOLÖv5

THANKS

Do you have any questions?

Project GitHub:
www.github.com/ahhhlexli/clearbot

CREDITS: This presentation template was
created by **Slidesgo**, including icons by
Flaticon, infographics & images by **Freepik**

END!