

# Individual Report

---

## # Contribution

---

Since our group was three people, we were required by the task to each find at least three papers and then choose an algorithm to complete the task. Based on this assignment, we started by looking for algorithms that could satisfy our assignment, and then selected one each to study in depth.

In the beginning, we often asked each other questions in our whatsapp group because we didn't understand many basic conceptual things, such as what COCO and ASCAL VOC are.

Later, after some discussions and after determining the main directions and ideas of everyone, we started to go to Github and the code carried in the appendix of the paper to find code that could be run. Since many of the models require data in XML format, I wrote a python script, `txt2xml.py`, to convert the dataset carried by the job into XML format.

Regarding the panel report, we have divided it into the following main sections:

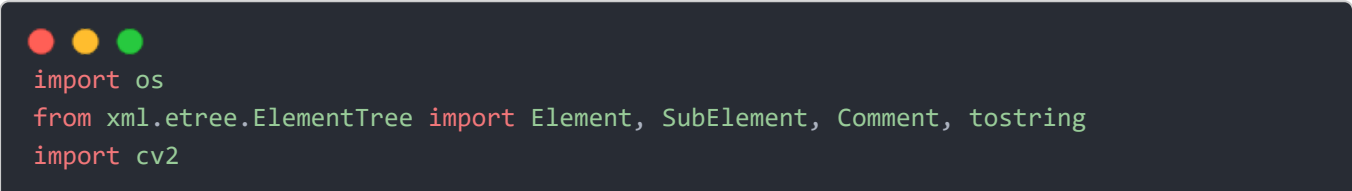
- Introduction
- Background
- Implementation
- Results
- Limitations, Conclusion and Future Work

Among them, **Introduction**, **Limitations**, **Conclusion and Future Work** were written by the three of us together during the online meeting. The rest of the section was divided into three subsections each, with the respective group members putting their chosen method and the results of their respective training on the report. One of them, 4.2.4 section, has a comparison of three methods that I wrote.

After we worked together on the formatting of the report on Google Docs, I finally reviewed the report and adjusted the font, line spacing, image size, etc. Our group report was then packaged and uploaded.

## # txt2XML.py

---



```
import os
from xml.etree.ElementTree import Element, SubElement, Comment, tostring
import cv2
```

```

from glob import glob

def get_img_size(img):
    img = cv2.imread(img)
    return img.shape[0], img.shape[1]

def txt_to_xml(txt_path, xml_path):
    # Read txt file
    with open(txt_path, 'r') as txt_file:
        lines = txt_file.readlines()
    # Extracting Information
    ripeness_class = lines[0].split()[0]
    x_center = lines[0].split()[1]
    y_center = lines[0].split()[2]
    x_width = lines[0].split()[3]
    y_width = lines[0].split()[4]
    ripeness_text = 'unripe' if ripeness_class == '0' else 'partially ripe' if
    ripeness_class == '1' else 'fully ripe'

    # Create XML elements
    annotation = Element('annotation')
    SubElement(annotation, 'filename').text =
os.path.basename(txt_path).replace('.txt', '.png')
    size = SubElement(annotation, 'size')
    SubElement(size, 'width').text = str(W)
    SubElement(size, 'height').text = str(H)
    SubElement(size, 'depth').text = '3'
    object_ = SubElement(annotation, 'object')
    SubElement(object_, 'name').text = ripeness_text
    bndbox = SubElement(object_, 'bndbox')
    SubElement(bndbox, 'xmin').text = str(int(float(x_center) * W - float(x_width) * W
/ 2))
    SubElement(bndbox, 'ymin').text = str(int(float(y_center) * H - float(y_width) * H
/ 2))
    SubElement(bndbox, 'xmax').text = str(int(float(x_center) * W + float(x_width) * W
/ 2))
    SubElement(bndbox, 'ymax').text = str(int(float(y_center) * H + float(y_width) * H
/ 2))
    # 写入 XML 文件
    with open(xml_path, 'w') as xml_file:
        xml_file.write(tostring(annotation).decode())

H, W = get_img_size('train_data/1.png')

folder_path = 'train_data'
file_names = [os.path.basename(x) for x in glob(folder_path + '/*')]

# Convert all txt files
txt_dir = '01_bounding_box/'
xml_dir = 'train_xml/'
for txt_filename in file_names:
    txt_path = os.path.join(txt_dir, txt_filename.replace('.png', '.txt'))

```

```
xml_path = os.path.join(xml_dir, txt_filename.replace('.png', '.xml'))  
# print(txt_path, xml_path)  
txt_to_xml(txt_path, xml_path)
```