EE/CprE/SE 491 - sddec23-10

Developing a Deep Learning Model to Automatically Detect Microscale Objects in Images and Videos

Week 3 Report

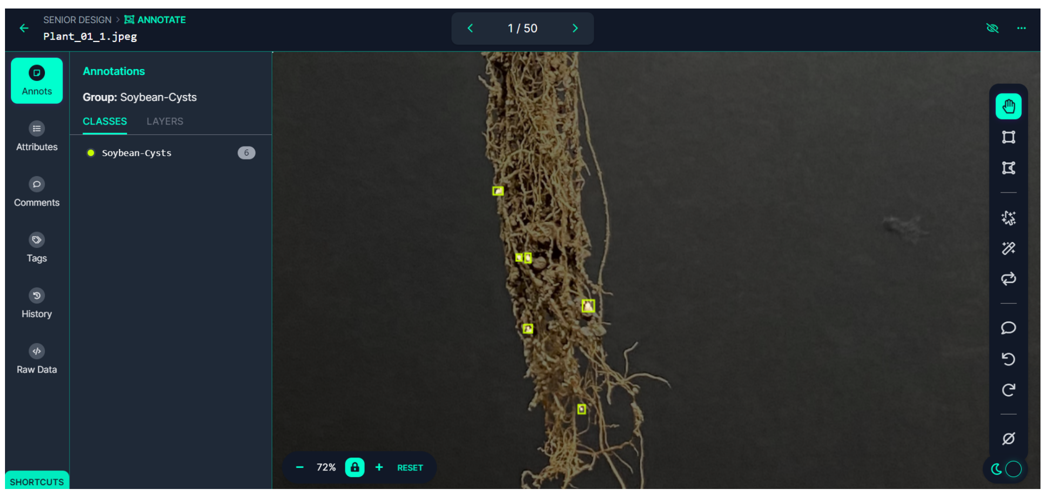
02/13/2023 – 02/19/2023
Client : Professor. Santosh Pandey
Group number: 10

Team Members:

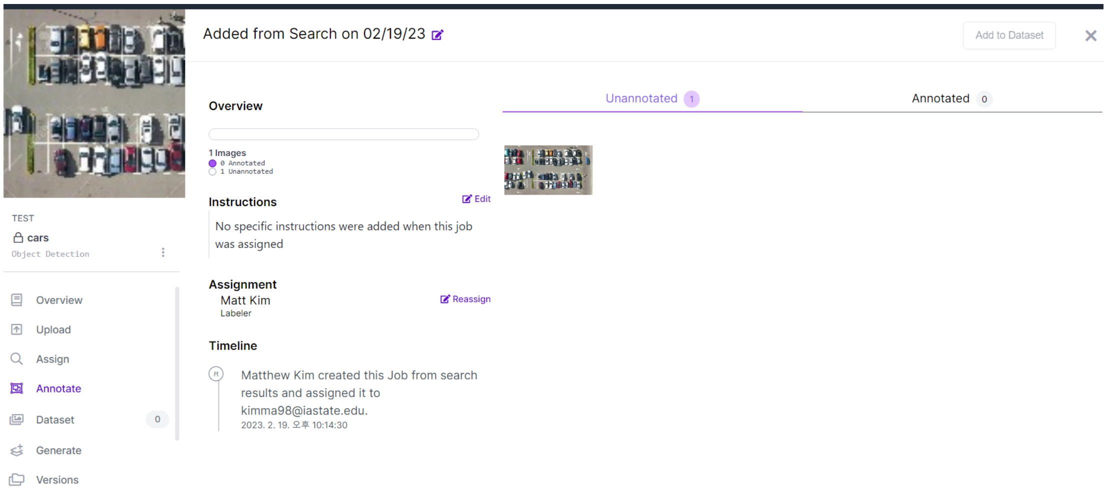
Katherine Moretina
Ethan Baranowski
Chris Cannon
Matthew Kim

Past week Accomplishments

Test Labeling Tools- Katie

- Currently on a waiting list for V7. Said it would take about 2-3 weeks for student liscense to come in
- Roboflow
    - Easiest to set up and navigate
    - Good user interface
    - Uploaded entire image folder for labeling
    - Easy bounding boxes
    - Option for polygon tools- I think we will use bounding box, but it's nice to have the option
    - Didn't have to install anything on computer- nice since I doubt we have admin privileges on the lab computers
    - 

    - Cons of Roboflow
        - Projects can only be public unless we want to pay for the liscense
        - I'm not sure if Dr. Pandey wants our projects to be online because current images are password protected
    - More research on the Roboflow - Matthew

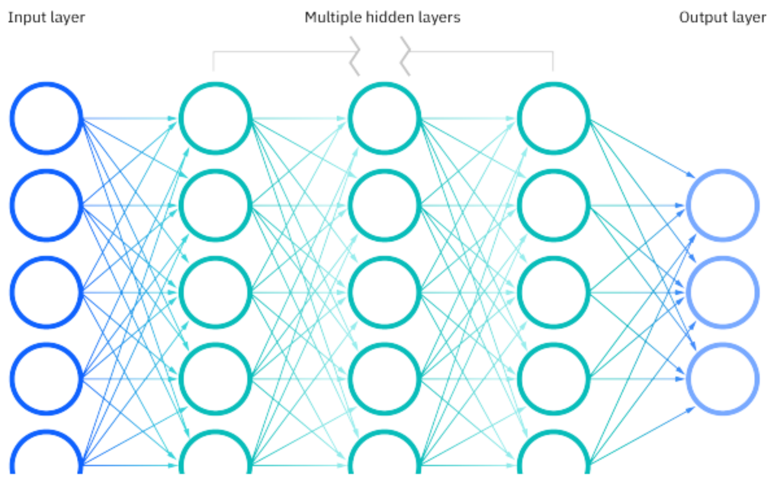


    - Possible to upload multiple images then assign different images to each other.
    - Possible to export into any different format.

- Label Studio
    - Not as easy to set up but more customizable
    - Option to use template or build out our labeling method
    - Did have an option for machine learning that can be explored more
    - Image below shows results- bounding boxes are pink
    - 
    - Cons of Label Studio
        - Way harder to set up
        - Not as intuitive user interface

- Both have the ability to share our project with multiple people, which will come in handy
- If Dr. Pandey does not want public access to our project, I suggest Label Studio

What is a Neural Network?
Used in deep learning algorithms.
Structure mimics the way neurons communicate.
Series of algorithms that recognize underlying relationships in a set of data, similar to neurons in the brain.



Single Shot Detector (SSD)

- Breaks down bounding box of object into several smaller boxes that identify different feature locations relative to the object.
- Has reduced accuracy performance for small object detection
- 300x300 input 72.1% accuracy
- 500x500 input 75.8% accuracy
- Single Stage detector

Method	mAP	FPS	batch size	# Boxes	Input resolution
Faster R-CNN (VGG16)	73.2	7	1	~ 6000	$\sim 1000 \times 600$
Fast YOLO	52.7	155	1	98	448×448
YOLO (VGG16)	66.4	21	1	98	448×448
SSD300	74.3	46	1	8732	300×300
SSD512	76.8	19	1	24564	512×512
SSD300	74.3	59	8	8732	300×300
SSD512	76.8	22	8	24564	512×512

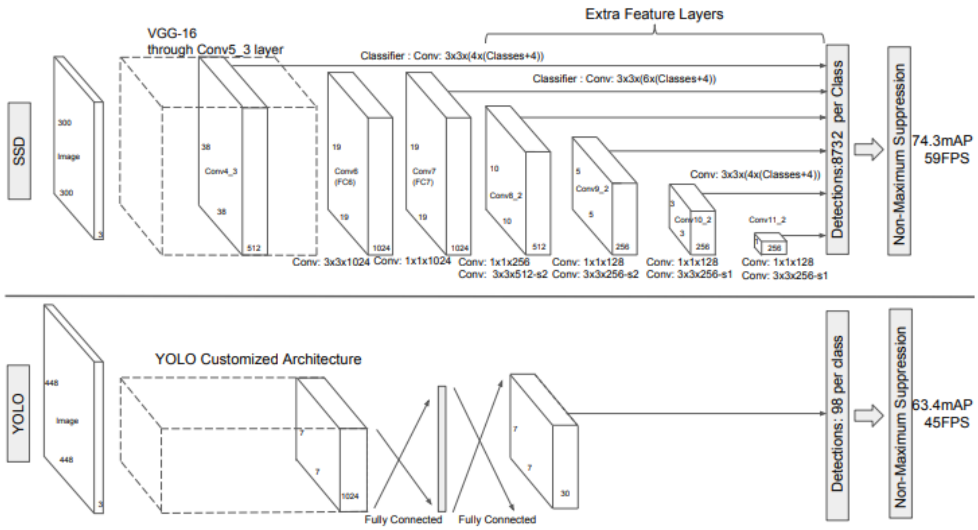


Fig. 2: A comparison between two single shot detection models: SSD and YOLO [5].

3.6 Data Augmentation for Small Object Accuracy

Without a follow-up feature resampling step as in Faster R-CNN, the classification task for small objects is relatively hard for SSD, as demonstrated in our analysis (see Fig. 4). The data augmentation strategy described in Sec. 2.2 helps to improve the performance dramatically, especially on small datasets such as PASCAL VOC. The random crops

PASCAL VOC, MS COCO, and ILSVRC datasets
https://arxiv.org/abs/1512.02325
[1512.02325] SSD: Single Shot MultiBox Detector

YOLO (You Only Look Once) - Chris

- Known for speed and accuracy - Introduced in 2016
- High accuracy while maintaining a low model size.
- Single-shot object detection can be less effective in detecting small objects (source)
- SSD is typically used to detect objects in real-time in resource-constrained environments.
    - Real-time is not necessary for our use case, though resource constraints will be strict.
- Uses a fully convolutional neural network to process an image.

What is YOLO?

- End-to-end neural network that makes predictions of bounding boxes and class probabilities all at once.
- Divides an input image into an SxS grid. If the center of an object falls into a grid cell, that cell is responsible for detecting that object.
- Non-maximum suppression (NMS)
  - Post-processing step used to improve accuracy and efficiency of object detection.
  - Identifies and removes redundant or incorrect bounding boxes to output a single bounding box for each object in the image.

Improvements to YOLO

YOLO v2 (YOLO9000)

- Designed to be faster, more accurate, and able to detect a wider range of object classes.
- Uses a different CNN backbone - Darknet-19
  - Variant of VGGNet architecture using simple progressive convolution and pooling layers
- Main improvement is the use of anchor boxes.
  - Set of predefined bounding boxes of different aspect ratios and scales.
  - When predicting bounding boxes, YOLO v2 uses a combination of anchor boxes and the predicted offsets to determine the final bounding box.
  - Allows the algorithm to handle a wider range of object sizes and aspect ratios
- Batch Normalization
  - Helps to improve the accuracy and stability of the model.
  - Multi-scale training strategy
    - Involves training the model on images at multiple scales and averaging the predictions - **improves detection performance of small objects**
- Loss function better suited to object detection tasks.

YOLO v3

- New CNN architecture called Darknet-53
  - Variant of the ResNet architecture designed specifically for object detection tasks.
  - Has 53 convolutional layers.
- Anchor boxes can now be scaled and have different aspect ratios.
- Feature pyramid Networks
  - Helps to improve detection of small objects
  - Designed for objects of multiple scales

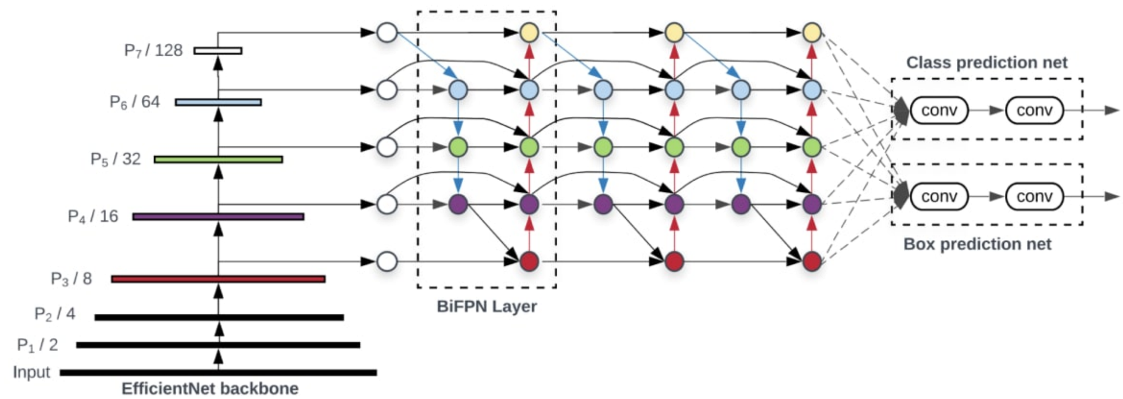
YOLO v4

- Primary improvement is the use of new CNN architecture called CSPnet

- Cross Stage Partial Network
    - Variant of ResNet architecture
    - Relatively shallow structure, only 54 convolutional layers - still achieves state-of-the-art results
- Anchor boxes are generated using k-means clustering
    - A clustering algorithm to group the ground truth bounding boxes into clusters and then using the centroids of the clusters as anchor boxes.
    - Allows anchor boxes to be more closely aligned with detected objects' size and shape
- Improved FPN architecture

YOLOv5

- Uses a more complex architecture called EfficientDet



EfficientDet architecture – It employs EfficientNet [39] as the backbone network, BiFPN as the feature network, and shared class/box prediction network. Both BiFPN layers and class/box net layers are repeated multiple times based on different resource constraints as shown in Table 1.

- Uses a new method for generating anchor boxes called "dynamic anchor boxes"
    - Involves using a clustering algorithm to group ground truth bounding boxes into clusters and using the centroids of the clusters as the anchor boxes.
- Introduced the concept of spatial pyramid pooling (SPP).
    - A type of pooling layer used to reduce the spatial resolution of feature maps
    - **Improves detection performance on small objects** as it allows the model to see the objects at multiple scales

YOLO v6

- Proposed by Li et al. in 2022
- Uses a variant of the EfficientNet architecture called EfficientNet-L2
    - More Efficient, with fewer parameters and higher computational efficiency

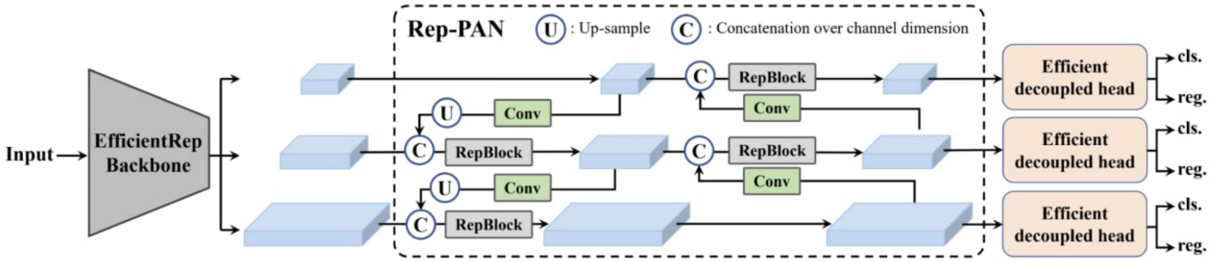


Figure 2: The YOLOv6 framework (N and S are shown). Note for M/L, RepBlocks is replaced with CSPStackRep.

- New method of generating anchor boxes, called “dense anchor boxes”

YOLO v7

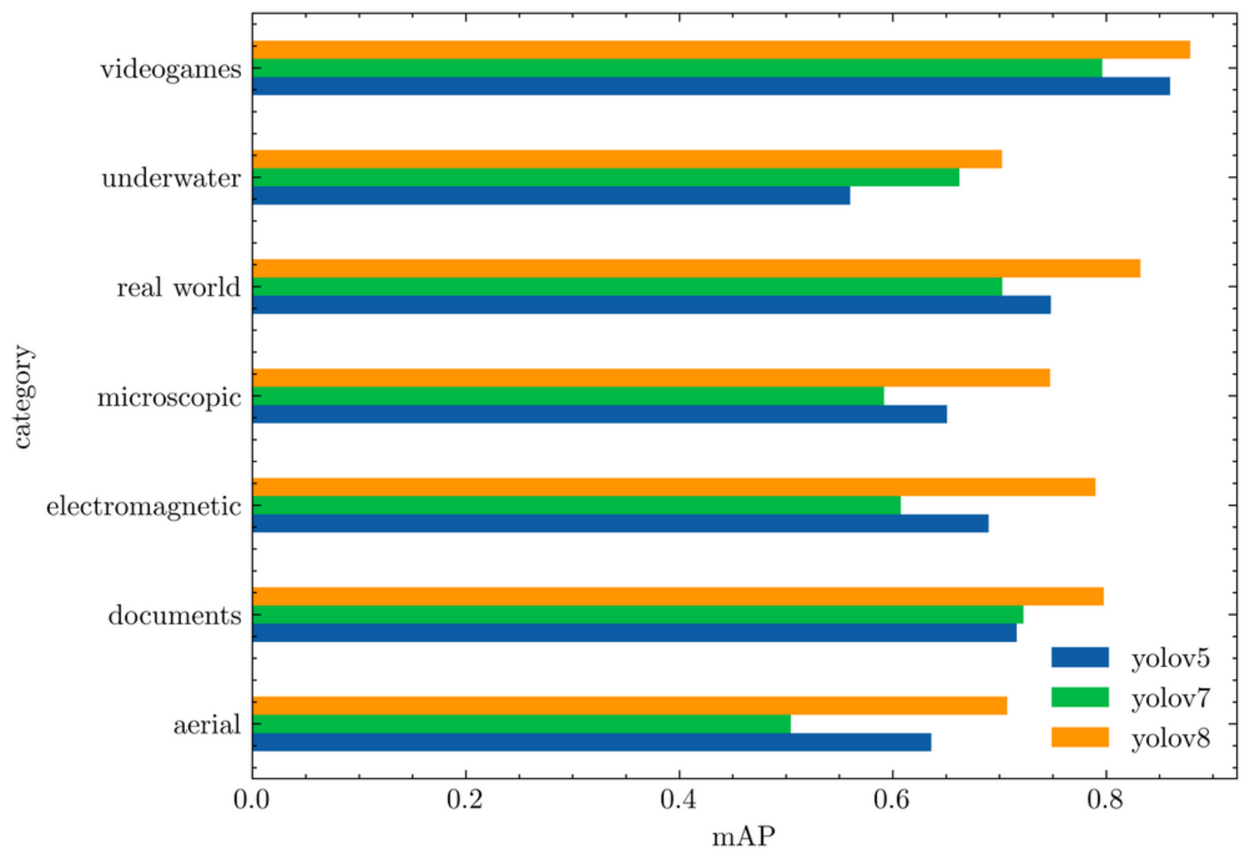
- Uses 9 anchor boxes allowing it to detect a wider range of object and sizes.
  - Reduces false positives
- Uses a new loss function called “focal loss”
  - Previous versions used a standard cross-entropy loss function, which is known to be less effective at detecting small objects
  - Focal loss battles this issue by down-weighting the loss for well-classified examples and focusing on the hard examples.
- Uses a higher resolution than previous versions. Processes images at a resolution of 608x608 pixels, rather than the previous 416x416
  - Detects smaller objects, has a higher accuracy.
- Accuracy is comparable to other state-of-the-art object detection algorithms
- Less accurate than two-stage detectors such as Faster R-CNN and Mask R-CNN

Limitations of YOLO v7

1. YOLO v7, like many object detection algorithms, struggles to detect small objects. It might fail to accurately detecting objects in crowded scenes or when objects are far away from the camera.
2. YOLO v7 is also not perfect at detecting objects at different scales. This can make it difficult to detect objects that are either very large or very small compared to the other objects in the scene.
3. YOLO v7 can be sensitive to changes in lighting or other environmental conditions, so it may be inconvenient to use in real-world applications where lighting conditions may vary.
4. YOLO v7 can be computationally intensive, which can make it difficult to run in real-time on resource-constrained devices like smartphones or other edge devices.

YOLO v8 - Most recent version

- Has not had a published paper yet
- **Notably** YOLOv8 is an anchor-free model. It directly predicts the center of an object instead of the offset from a known anchor box.
    - Reduces the number of box predictions, which speeds up the Non-Maximum Suppression (NMS) discussed above.
- Training routine is crucial -
    - YOLOv8 augments images during training. At each epoch, the model sees a slightly different variation of the images it has been provided.
    - Mosaic augmentation:
        - This training technique involves stitching four images together, forcing the model to learn objects in new locations, in partial occlusion, and against different surrounding pixels.
        - This augmentation has been shown to degrade performance when performed during the whole training routine - it is advantageous to turn it off for the last ten training epochs.
- YOLOv8 COCO accuracy is state of the art for models at comparable inference latencies.
- It also outperforms every previous iteration of YOLO in each Roboflow100 category



- YOLOv8 also includes several developer-friendly features, like an easy-to-use CLI and a well-structured Python package.

Different Labeling Formats- Matthew

https://www.edge-ai-vision.com/2022/04/exploring-data-labeling-and-the-6-different-types-of-image-annotation/

There are different types of labeling format that each of them are used for different algorithms. However, it is possible to convert one into another using programs, or by implementing one.

For example git provides conversion from the label studio into another format.
https://github.com/heartexlabs/label-studio-converter

COCO(json file)
Coco sets the left top as a reference point. From there, it forms its box with width and height.

```
annotation{
    "id" : int,
    "image_id": int,
    "category_id": int,
    "segmentation": RLE or [polygon],
    "area": float,
    "bbox": [x,y,width,height],
    "iscrowd": 0 or 1,
}
categories[{
    "id": int,
    "name": str,
    "supercategory": str,
}]
```

PASCAL VOC (XML)

```xml
<annotation>
  <folder>Train</folder>
  <filename>01.png</filename>
  <path>/path/Train/01.png</path>
  <source>
    <database>Unknown</database>
  </source>
  <size>
    <width>224</width>
    <height>224</height>
    <depth>3</depth>
  </size>
  <segmented>0</segmented>
  <object>
    <name>36</name>
    <pose>Frontal</pose>
    <truncated>0</truncated>
    <difficult>0</difficult>
    <occluded>0</occluded>
    <bndbox>
      <xmin>90</xmin>
      <xmax>190</xmax>
      <ymin>54</ymin>
      <ymax>70</ymax>
    </bndbox>
  </object>
</annotation>
```

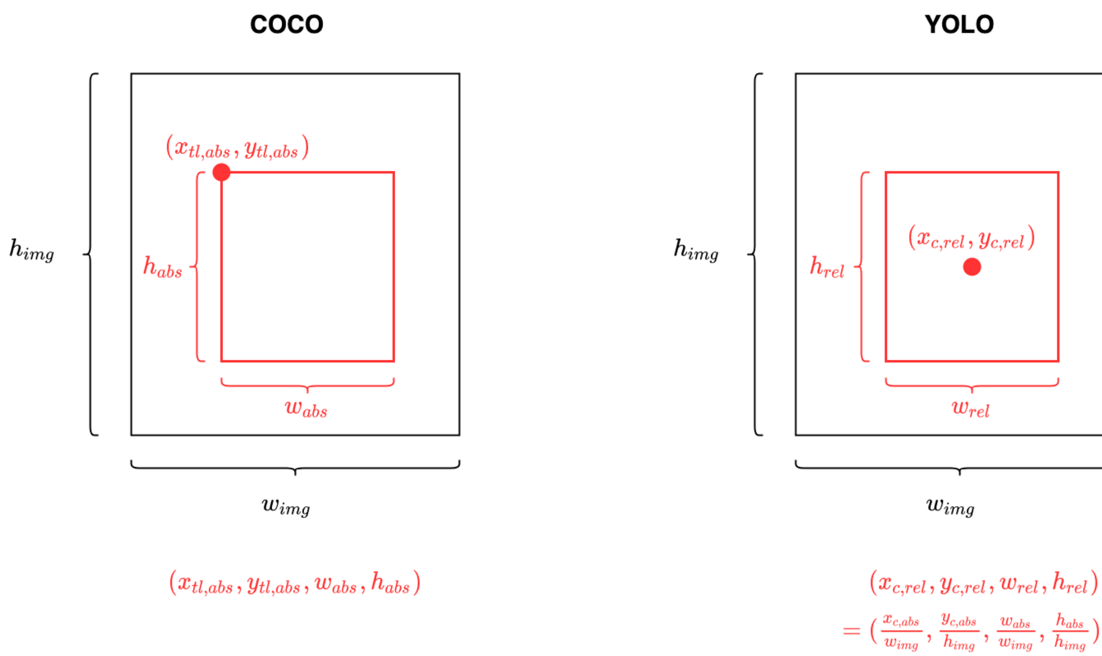
YOLO

Yolo sets Center as the reference point. Then it forms the box point to the top, bottom, right and left.

```
<object-class><x><y><width><height>
```

When there are more than 2 objects, it is recorded as line by line.

```
0 45 55 29 67
1 99 83 28 44
```



https://haobin-tan.netlify.app/ai/computer-vision/object-detection/coco-json-to-yolo-txt/

Individual Contributions

Member	Tasks Completed	Hours This Week	Total Hours
Katherine Moretina	Continued to gain background information on the algorithms we've talked about in the past two weeks. Read through labeling methods. Tested all labeling methods I could access this week.	4	11
Matthew Kim	Attended regular meetings to check a phase, and further research on R-CNN and labeling tools.	2	8
Chris Cannon	Attended regular meetings, tested label studio labelling software, continued background research. I also did in-depth	5	10

	research on YOLO, including YOLOv8		
Ethan	Finished guided research task on types of machine learning algorithms. Organized regular meetings on mondays with Prof. Pandey and/or TA Yunsoo Park. Started research on different ways to label training data. Got access to image repository. Attended meetings, where Yunsoo Park walked us through the phases of the small object detection and discussed possible strategies.	5	13

Plans for Coming Week

- Compile research results of Neural Networks, Object Detection, and Machine Learning.
- Discuss evaluation methods for ranking the researched algorithms.
- Continue discussion on whether to implement multiple algorithms or single most applicable algorithm.
- Group research on the highest ranked algorithms.
- Test labeling tools and evaluate which method is easiest for us to implement and use and allows customized data sets.
- Have Yunsoo Park walk us through coding on the lab computer.
- Setup Jupyter Notebooks server for student collaboration
- Compare current algorithm findings- this week we will decide on a top 3-5 algorithms we want to use and do more research on those.