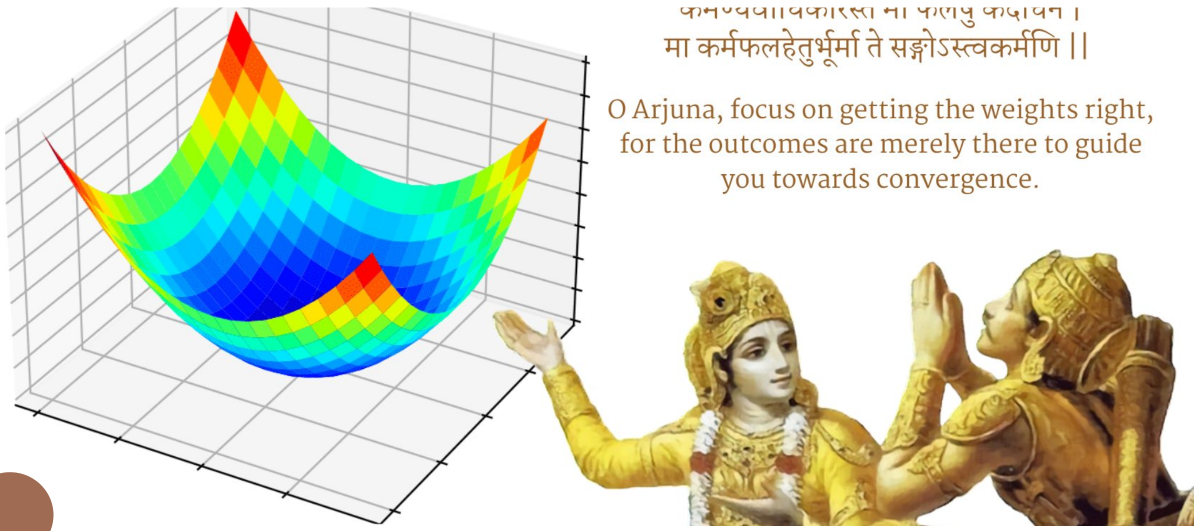

मा कर्मफलहेतुर्भूर्मा ते सङ्गोऽस्त्वकर्मणि ||
O Arjuna, focus on getting the weights right, for the outcomes are merely there to guide you towards convergence.

Project Report: PPE Detection Model

Problem Statement:

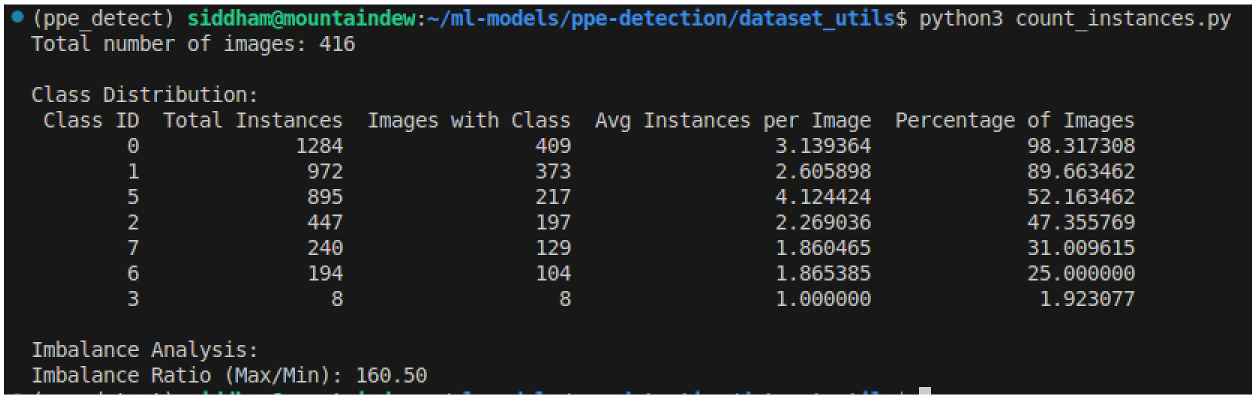
To design a workflow that will have two `YOLO` models. One will detect the person in the full image and pass the cropped image of the person to the second model. The second model will run inference on the cropped image which will convey information about the protective equipment the person is wearing and save the output image as a full image in the output directory.

Objectives:

1. to convert `PascalVOC` annotation to `YOLOv8` format
2. to modify labels for the training of person detection model
3. train person detection model
4. generate a new dataset of cropped images of persons with labels modified accordingly
5. train ppe detection model using a stratified k-fold cross-validation technique
6. work on `inference.py`

Dataset Overview

Full image dataset:-



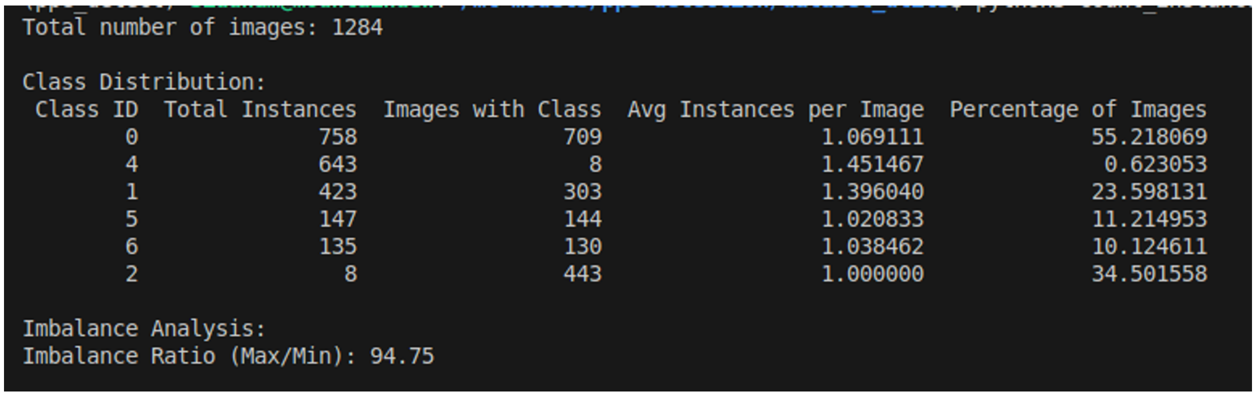
```
(ppe_detect) siddham@mountaindew:~/ml-models/ppe-detection/dataset_utils$ python3 count_instances.py
Total number of images: 416

Class Distribution:
 Class ID  Total Instances  Images with Class  Avg Instances per Image  Percentage of Images
        0             1284                409                 3.139364             98.317308
        1              972                373                 2.605898             89.663462
        5              895                217                 4.124424             52.163462
        2              447                197                 2.269036             47.355769
        7              240                129                 1.860465             31.009615
        6              194                104                 1.865385             25.000000
        3                8                  8                 1.000000              1.923077

Imbalance Analysis:
Imbalance Ratio (Max/Min): 160.50
```

- The dataset contains total of 416 images with a total of 4040 instances across 8 classes
- Class id 0 i.e. 'Person' class has the most number of images in the dataset
- Class mapping:

```
{
    "person": 0,
    "hard-hat": 1,
    "gloves": 2,
    "mask": 3,
    "glasses": 4,
    "boots": 5,
    "vest": 6,
    "ppe-suit": 7,
    "ear-protector": 8,
    "safety-harness": 9
}
```

- There are no instances of class id 8 and 9 i.e. `ear-protector` and `safety-harness`

Cropped image dataset:-



```
Total number of images: 1284

Class Distribution:
 Class ID  Total Instances  Images with Class  Avg Instances per Image  Percentage of Images
        0              758                709                 1.069111             55.218069
        4              643                  8                 1.451467              0.623053
        1              423                303                 1.396040             23.598131
        5              147                144                 1.020833             11.214953
        6              135                130                 1.038462             10.124611
        2                8                443                 1.000000             34.501558

Imbalance Analysis:
Imbalance Ratio (Max/Min): 94.75
```

- This is the dataset obtained by cropping person's image from full image dataset
- Total number of images are 1284 with a total of 2114 instance across 7 classes. I have excluded the person's class from it
- To tackle imbalance in class distribution I used stratified K-fold cross validation method to train PPE detection model
- Class mapping:-

```
{
  "hard-hat": 0,
 "gloves": 1,
 "mask": 2,
 "glasses": 3,
 "boots": 4,
 "vest": 5,
 "ppe-suit": 6,
 "ear-protector": 7,
 "safety-harness": 8
}
```

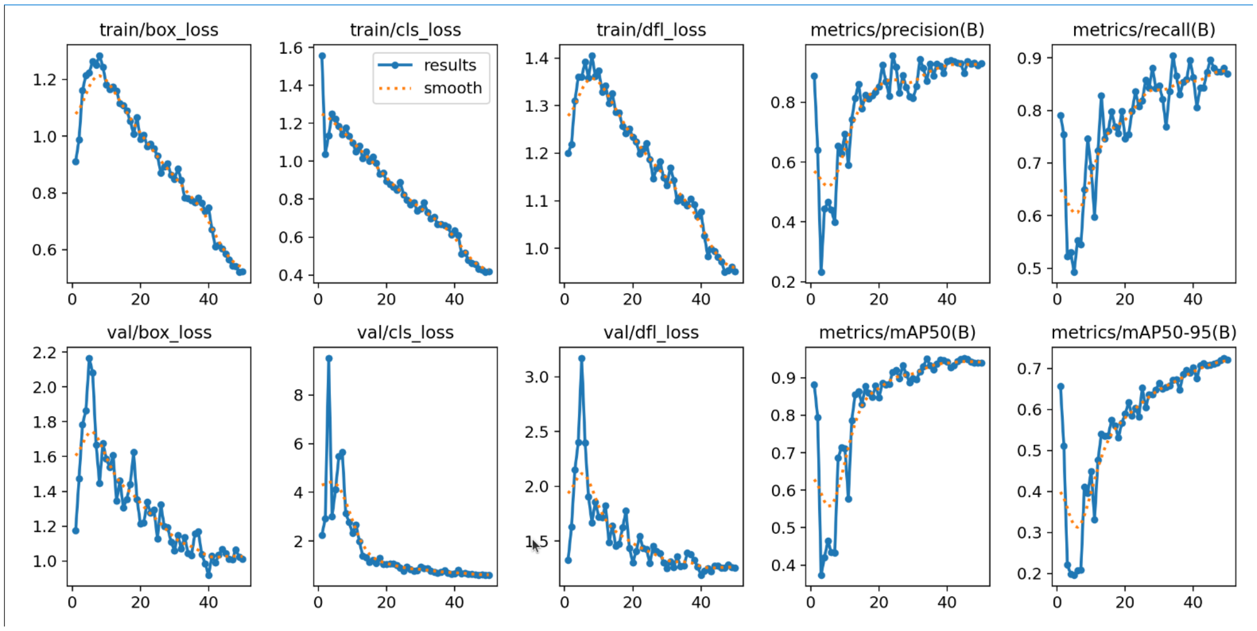
Approach

Person detection model:

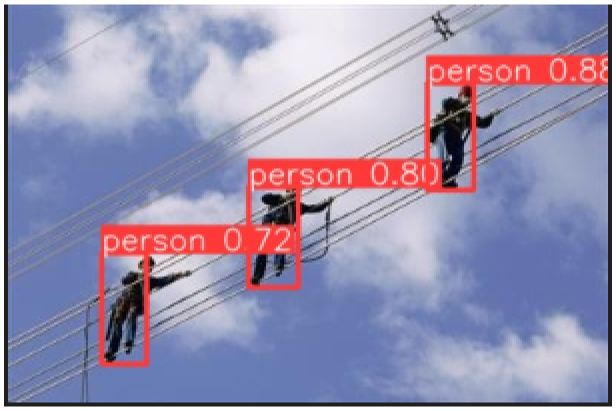
For training the person detection model, I used a script to divide the dataset into three splits: train, test, and validation. I allocated 80% of the data to the training set and 10% each to the test and validation sets. I used a straightforward approach to divide the dataset and trained the model directly, as there was only one class, 'person,' in which I needed to train the model.

Metrics:

- Precision: 0.921
- Recall: 0.881
- mAP50: 0.939



- All loss graphs show a general downward trend showing that model is improving and learning after every epoch
- The validation loss graphs show occasional spikes, which could be due to my model encountering potential overfitting.
- However, overall, it performs quite well when I test it on the test data. Below are some test images on which i ran my model



PPE detection model:

To tackle class imbalance i used stratified K-fold cross validation technique to train this model. During training i divided my dataset into 5 fold and trained different `YOLOV8` model for each fold and used majority voting for final classification.

I aggregated the predictions from multiple models by using majority voting. Each model casts a vote for a particular class, and the class with the most votes is selected as the final prediction.

Metrics (all classes):

Fold #	Precision	Recall	mAP50	mAP50-95
1	0.956	0.94	0.969	0.904
2	0.974	0.937	0.966	0.916

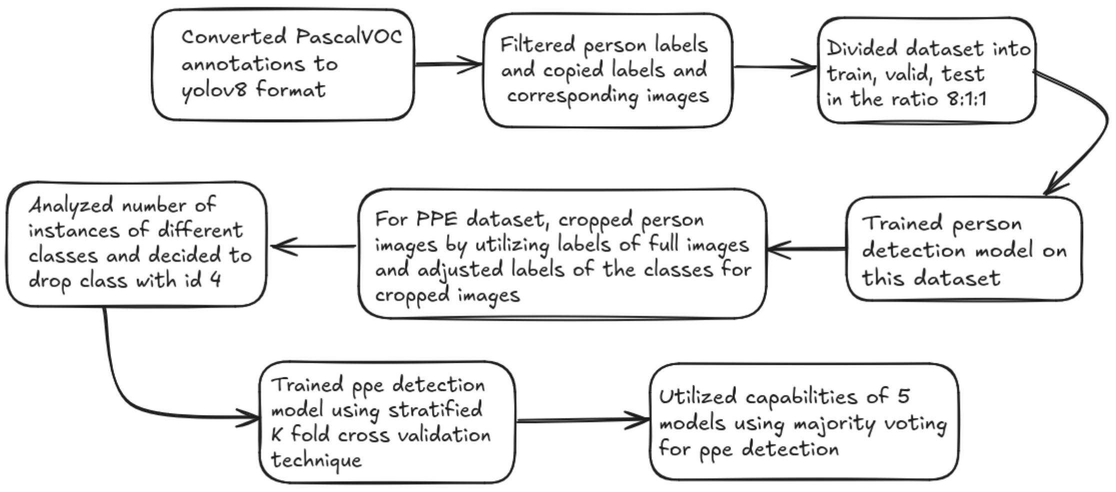
Fold #	Precision	Recall	mAP50	mAP50-95
3	0.96	0.906	0.949	0.89
4	0.954	0.951	0.973	0.932
5	0.967	0.943	0.966	0.93

- The precision value ranging from 0.954 to 0.974 indicates model is working well with identifying true positive cases, minimizing false negative effectively
- The recall values demonstrates the model's strong ability to detect true positives, while keeping false negatives relatively low
- The mAP@50 consistently exceeded 0.949 across all folds, with a peak performance of 0.973 which indicates model performs well in terms of both precision and recall when evaluated under this threshold.

`inference.py` :

- I detect persons in each image using the person detection model, extracting bounding boxes and confidence scores.
- For each detected person, crop the image to the bounding box and run PPE detection models on these cropped images. Aggregate results from all PPE models and apply non-maximum suppression to filter redundant detections.
- After detecting PPE items in the cropped image, I adjust the bounding box coordinates of the detected PPE items to the original image coordinates. This is done by adding the coordinates of the person's bounding box back to the PPE detection results
- Draw bounding boxes around detected PPE items on the original image, based on their class and confidence score. The annotated image is then saved to the specified output directory.

Detailed Workflow:


Converted PascalVOC annotations to yolov8 format
Filtered person labels and copied labels and corresponding images
Divided dataset into train, valid, test in the ratio 8:1:1
Trained person detection model on this dataset
For PPE dataset, cropped person images by utilizing labels of full images and adjusted labels of the classes for cropped images
Analyzed number of instances of different classes and decided to drop class with id 4
Trained ppe detection model using stratified K fold cross validation technique
Utilized capabilities of 5 models using majority voting for ppe detection