**Working Progress**

I have started this competition with yolov5. With an image size of 1024, I have trained model 1. On improvement, I have changed the pre-trained weights and trained model 2. On looking for improving results, I have trained model 3 with increasing epochs to 199, and other things were kept the same. On analysis of discussion on the Kaggle competition, I have prepared the report of things that I have done and things that first rank holder did. After looking for that analysis, I found that initially, I had given an empty txt file for false images. However, then I have passed full boxes of the whole image in a text file, and I have also added one more annotation as no findings (class 14) for these full boxes. I have trained model 4 with a batch size of 28, keeping other hyperparameters the same. I have trained model 5 with increasing epoch to 299 with decreasing batch size to 16.

To improve the results, I have trained model 6 with decreasing image size to 512 and increasing batch size to 64, keeping full boxes for false images. I changed full boxes to again empty boxes, and with the same hyperparameters, I have trained model 7.

I have removed these false images and trained model 8, keeping other hyperparameters the same. I took ten partial class images from the previous dataset having an absence of false images; I have trained model 9. Similarly, for the rest of 4 partial classes, I have trained model 10. With the same things as model 9 and model 10, I have changed the image size to 1024 and batch size to 28. I have trained model 11 and model 12, respectively.

Based on further analysis, from the partial ten classes, according to the number of instances, I have trained model 13 with classes that include 1, 4, 12 having a batch size of 40 and images size of 640. Model 14 with classes includes 2,5,6 with hyperparameters as the previous model. Model 15 with classes 7,8,9,10. For models 13, 14, and 15, I have randomly chosen 10 % (corresponding number of images) false images during training which helps to build good confusion metrics.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Model | Training complete | Image Size | Epochs | No Findings | False Image | Other Info | Result  Folder |
| Model1 | Yes | 1024 | 49 | exclude | include | Bz 16, Adam, error 8,  Yolov5x.pt, Run/exp 6 | Output1 |
| Model2 | Yes | 1024 | 49 | exclude | include | Bz 16, Adam, error 13,  Yolov5x6.pt, Run/exp 9 | Output2 |
| Model3 | Yes | 1024 | 199 | exclude | include | Bz 16, Adam, error 21,  Yolov5x6.pt, Run/exp 12 | Output3 |
| Model4 | Yes | 1024 | 199 | include | - | Bz 28, Adam, error 24,  Yolov5x6.pt, Run/exp 15  train\_ .cache,  full boxes for false images | Output4 |
| Model5 | No | 1024 | 202/  299 | include | - | Bz 16, Adam, error 25,  Yolov5x6.pt, Run/exp 16  train\_ .cache,  full boxes for false images | Output5 |
| Model6 | Yes | 512 | 199 | include | - | Bz 64, Adam, error 26,  Yolov5x6.pt, Run/exp 17,  train\_ .cache,  full boxes for false images | Output6 |
| Model7 | Yes | 512 | 199 | exclude | include | Bz 64, Adam, error 27,  Yolov5x6.pt, Run/exp 18,  train\_1.cache,  empty txt for false images | Output7 |
| Model8 | Yes | 512 | 199 | exclude | exclude | Bz 64, Adam, error 28,  Yolov5x6.pt, Run/exp 38,  train\_2.cache,  remove false images txt | Output8 |
| Model9 | Yes | 512 | 199 | exclude | exclude | Bz 64, Adam, error 29,  Yolov5x6.pt,  yolov5/run/exp 44,  train\_3.cache  only 10 classes from 1to 12  (1,2,4,5,6,7,8,9,10,12),  dataset\_1, yolo5x\_1, vinbig\_1 while txt data is train\_3 and val\_3 | Output9 |
| Model10 | Yes | 512 | 199 | exclude | exclude | Bz 64, Adam, error 30,  Yolov5x6.pt, Run/exp 46,  train\_4.cache  only 4 classes 0, 3, 11, 13  dataset\_2, yolo5x\_2 and vinbig\_2 while txt data is train\_4 and val\_4 | Output10 |
| Model11 | Yes | 1024 | 99 | exclude | exclude | Bz 28, Adam, error 31,  Yolov5x6.pt,  yolov5/run/exp 47,  train\_3.cache  only 10 classes from 1to 12  (1,2,4,5,6,7,8,9,10,12) ),  dataset\_1, yolo5x\_1, vinbig\_1 while txt data is train\_3 and val\_3 | Output11 |
| Model12 | No | 1024 | 31/99 | exclude | exclude | Bz 28, Adam, error 32,  Yolov5x6.pt,  Yolov5/run/exp 48,  train\_4.cache  only 4 classes 0, 3, 11, 13  dataset\_2, yolo5x\_2 and vinbig\_2 while txt data is train\_4 and val\_4 | Output12 |
| Model13 | Yes | 640 | 79 | exclude | include | Bz 40, SGD, error 34,  Yolov5x6.pt,  yolov5/run/exp 58,  only 3 classes 1,4,12  with 10% false image  dataset\_3, yolo5x\_3 and vinbig\_3 while txt data is train\_5 and val\_5 | Output13 |
| Model14 | Yes | 640 | 79 | exclude | include | Bz 40, SGD, error 35,  Yolov5x6.pt,  yolov5/run/exp 55,  only 3 classes 2,5,6  with 10% false image  dataset\_4, yolo5x\_4 and vinbig\_4 while txt data is train\_6 and val\_6 | Output14 |
| Model15 | Yes | 640 | 79 | exclude | include | Bz 40, SGD, error 36,  Yolov5x6.pt,  yolov5/run/exp 61,  only 4 classes 7,8,9,10  with 10% false image  dataset\_5, yolo5x\_5 and vinbig\_5 while txt data is train\_7 and val\_7 | Output15 |

**K Fold Method**

Split the dataset into five folds having 3000 images in each fold. I have trained 5 Yolo models having every four folds for training and one fold for validation purposes. Using these five models, I have generated results.

* Fold0
* Fold1
* Fold2
* Fold3
* Fold4

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Model | Training complete | Image Size | Epochs | No Findings | False Image | Other Info | Result  Folder |
| Kmodel1 | Yes | 640 | 199 | exclude | include | Bz 40, SGD, error 0,  Yolov5x.pt, fold0/exp 4 | Fold0 |
| Kmodel2 | Yes | 640 | 79,  199 | exclude | include | Bz 40, SGD, error 1,  Yolov5x.pt, fold0/exp 2,  Fold0/exp 8 | Fold1  (output1-exp8,  Output2 -exp2) |
| Kmodel3 | Yes | 640 | 199 | exclude | include | Bz 40, SGD, error 2,  Yolov5x.pt, fold0/exp 3 | Fold2 |
| Kmodel4 | Yes | 640 | 199 | exclude | include | Bz 40, SGD, error 3,  Yolov5x.pt, fold0/exp 2 | Fold3 |
| Kmodel5 | Yes | 640 | 99 | exclude | include | Bz 40, SGD, error 4,  Yolov5x.pt, fold0/exp 4 | Fold4 |

**Results generation**

Output generated files are saved into the Results folder. Convert labels into CSV files. Use multiple CSV files to generated final output using the ensemble method.

Using Kmodel 1 to 5, generated CSV files. Also, using a mirror image of the test folder, generated CSV files were saved in the results folder. Generated ensemble standard (ensemble\_yolo\_standard) and mirror (ensemble\_yolo\_mirror) using corresponding CSV files respectively. Two above ensemble files are used to generate the final ensemble file (ensemble\_yolo\_final).

Generated labels result of model 6, 8, 9, 10 (train 26, 28, 29, 30). Converted into csv files (test 26, 28, 29, 30), that are used to generated final ensemble csv file (test\_26\_to\_30). test\_26\_to\_30 and ensemble\_yolo\_final used to generated final ensemble

Generated labels result of model 8, 9, 10, 13, 14, 15 (train 28, 29, 30, 34, 35, 36). Converted into csv files (test 28, 29, 30, 34, 35, 36), that are used to generated final ensemble csv file (test\_28\_to\_36). I have used threshold values 0.01 common. I have used different threshold values according to best f1 score (corresponding confidence values as threshold values) and generated test 34\_1, 35\_1, 36\_1. Used these csv and generated test\_28\_36\_1 with ensemble method.

Used test\_28\_to\_36 and ensemble\_yolo\_final to generate ensemble\_final\_1. Used test\_28\_to\_36\_1 and ensemble\_yolo\_final to generate ensemble\_final\_2.