

#### Report 4

1. Explain what you have implemented during this week. (Specify what each person has done)
  - a. What were the steps? (Explain the details)

Ans: For this week, I complete the machine learning pipeline including data extraction, data annotation, model training, and model evaluation. Below is the list of scripts and their purposes.

- **videos.py** -> download various videos from YouTube and save them in a folder.
- **extract.py** -> randomly extract frames from the train video without duplication and save it in a folder.
- **yolov5.py** -> download yolov5 source code from GitHub, save it to a folder, and modify parts of it.
- **dataset.py** -> download **dataset.zip** from [Mega.io](https://mega.io), perform some modifications, split the dataset into multiple datasets of various sizes, save new datasets to a folder.
- **train.py** -> for each dataset, train a model using the train dataset and keep on training until its performance no longer increases when validate against the validation dataset. Lastly, evaluate the model performance with the test dataset. Save all results to a folder.
- **detect.py** -> use trained models to detect characters in the train and the test videos and save results to a folder.

- b. Any outputs or results?

Ans:

- **Hardware Benchmark:** To start with, I need to determine how capable are my hardware in training YOLOv5 models and thus, I decide to train all models for 3 epochs with 100% dataset and examine the max batch size. Below is the result.

| <i>Name</i> | <i>Size (pixels)</i> | <i>Params (M)</i> | <i>Epoch</i> | <i>Max Batch Size</i> |
|-------------|----------------------|-------------------|--------------|-----------------------|
| YOLOv5n     | 640                  | 1.9               | 3            | 96                    |
| YOLOv5s     | 640                  | 7.2               | 3            | 54                    |
| YOLOv5m     | 640                  | 21.2              | 3            | 24                    |
| YOLOv5l     | 640                  | 46.5              | 3            | 16                    |
| YOLOv5x     | 640                  | 86.7              | 3            | 8                     |
| YOLOv5n6    | 1280                 | 3.2               | 3            | 26                    |
| YOLOv5s6    | 1280                 | 16.8              | 3            | 14                    |
| YOLOv5m6    | 1280                 | 35.7              | 3            | 6                     |
| YOLOv5l6    | 1280                 | 76.8              | 3            | 4                     |
| YOLOv5x6    | 1280                 | 140.7             | 3            | 1                     |

From the data collected, I decide on YOLOv5n, YOLOv5s, and YOLOv5m models for further testing.

- **Batch Size:** The next decision is the batch size as it will determine how well a model will converge. Generally, increasing batch size will improve the training time but at the cost of performance. Thus, I want to see the performance difference between models trained with max batch size and models trained with a normalized batch size of 16 when they train on the 100% dataset and a normalized epoch. Below is the result.

| <i>Name</i> | <i>Epoch</i> | <i>Batch Size</i> | <i>Val mAP@.5</i> | <i>Test mAP@5</i> |
|-------------|--------------|-------------------|-------------------|-------------------|
| YOLOv5n     | 200          | 96                | 0.982             | 0.988             |
| YOLOv5s     | 200          | 54                | 0.985             | 0.975             |
| YOLOv5m     | 200          | 26                | 0.990             | 0.979             |
| YOLOv5n     | 200          | 16                | 0.972             | 0.968             |
| YOLOv5s     | 200          | 16                | 0.992             | 0.971             |
| YOLOv5m     | 200          | 16                | 0.985             | 0.995             |

From the data collected, training models with a normalized batch size seem to produce models that can generalize better and thus, I decided to go with a batch size of 16 for further testing.

- **Limited Dataset:** After deciding on models and the batch size, I want to see how well do all models perform on incomplete data. Thus, I decided to split the dataset into four new datasets of various sizes and train models on such datasets until no further improvement can be observed. Below is the result.

| <i>Name</i> | <i>Batch Size</i> | <i>Dataset</i> | <i>Epoch</i> | <i>Val mAP@.5</i> | <i>Test map@.5</i> |
|-------------|-------------------|----------------|--------------|-------------------|--------------------|
| YOLOv5n     | 16                | 25%            | 980          | 0.964             | 0.989              |
| YOLOv5n     | 16                | 50%            | 980          | 0.969             | 0.953              |
| YOLOv5n     | 16                | 75%            | 980          | 0.977             | 0.982              |
| YOLOv5n     | 16                | 100%           | 980          | 0.989             | 0.977              |
| YOLOv5s     | 16                | 25%            | 734          | 0.989             | 0.921              |
| YOLOv5s     | 16                | 50%            | 734          | 0.983             | 0.935              |
| YOLOv5s     | 16                | 75%            | 734          | 0.977             | 0.980              |
| YOLOv5s     | 16                | 100%           | 734          | 0.980             | 0.985              |
| YOLOv5m     | 16                | 25%            | 461          | 0.946             | 0.914              |
| YOLOv5m     | 16                | 50%            | 461          | 0.984             | 0.982              |
| YOLOv5m     | 16                | 75%            | 859          | 0.971             | 0.992              |
| YOLOv5m     | 16                | 100%           | 651          | 0.983             | 0.986              |

From the data collected, YOLOv5n seems to be too small of a network for the dataset as it performs well on the 25% dataset but it suffers performance degradation as the dataset size increased. YOLOv5s performs as expected since its performance is improving as the size of the dataset increased. Unfortunately, the dataset might be too small for YOLOv5m since it performs better on a smaller dataset than the full dataset.

c. Any errors?

Ans: I re-check the result above and there don't seem to be any error observed so far.

2. Any challenge you faced during this week? If so, how are you planning to resolve it? Any solutions or ideas?

Ans: For this week, the major problem is time. It can take up to 12 hours to complete the training process of a model and there is always the possibility of something going wrong. Thus, I have to implement the “resume” function into the training step so that any model that has its training process stop prematurely can continue from the latest epoch. Unfortunately, it is not possible to save and reload parameters of the early stopping algorithm and thus, any resumed training process has to train at least a number of epochs equivalent to the patient parameter of the algorithm (currently 100).