**Team: Adam Optimizers**

The final approach for our submission includes the use of **Florence** model. The primary purpose of using this model is **Optical Character Recognition**. Initially it was decided to fine-tune it on the given dataset but it works fine as it is. As the given dataset does not contain all the text present in the image.

1. Optical Character Recognition

Notebook contains the code to download images and use Florence model for OCR. The data is processed in chunks of 2000 images and stored in CSV files.

1. Text Cleaning

Notebook contains the code to clean the OCR text generated in the previous step. The goal is to get the given entity type from the image.

1. ML for reducing ambiguity

`*neural\_net.ipynb*` notebook also contains the code to reduce ambiguity when we have multiple values generated by OCR for length. **Neural Network** is used if we have multiple values of length and we have to decide which one is height, width or depth.

Comparison of other models compared to random forest:

|  |  |
| --- | --- |
| Model | Accuracy |
| Random Forest | 0.64 |
| Neural Network | 0.66 |

Before finalizing this approach, we tried using the Tesserect and PaddlePaddle OCRs but the Florence seems to outperform these easily.

With Tesserect and PaddlePaddle OCRs we tried making a dynamic filter for the images so the it is easier for these models to work. But even after applying filter Florence outperformed others. Also, the generalization of this dynamic filter was not very good and was hence dropped.

Before this we tried to train a neural network with images as input and the recognized number as output. Using entity\_type we could tell the unit of the number. But this also didn’t have very good accuracy score and was hence dropped.