Deep Learning for Computer Vision | Project: Smart Waste Management in Modern Cities| Report – Phase 1

By,
Chandresh J. Sutariya (**21f3001415**)

Project Choosen:

Smart Waste Management in Modern Cities

Statistical Analysis:

	Training data	**Testing data**
Total images	907 (93.79 %)	60 (6.20 %)

Table 1 Total available training and testing data

Label	Training data	Testing data
1	337 (37.15 % of training data)	43 (71.66 % of testing data)
0	148 (16.32 % of training data)	17 (28.33 % of testing data)
None	422 (46.52 % of training data)	-

Table 2 label for training and testing data

- There are a total of **620 different sizes of images**.
- I have also used p*andas_profiling* library to get insights : https://drive.google.com/file/d/1nGQunorVzzy3gtYoVWa-ocNWr-3JZenC/view?usp=sharing
- Every *query* has 43-50 images except "city garbage" which has only 11 images.

Approach:

Data loading & Pre-processing:

- First I'll separate all training & test images as per the labels, and use *keras.utils.image_dataset_from_directory* to load labeled images.
- While loading images itself I'll resize the images as per my model pipeline requirement. I am currently doing 256 x 256.
- I'll keep the color format to RGB.
- As the all images are of larger size then 256 x 256, augmentation won't be needed..

Approach for non-labelled data:

There are two ways I have thought of approaching non-labeled data. One - I will assume that those are 0 labelled, and can be used to train classification model, but as the bounding box is not given, those non-labelled data I won't use for localization. The second approach is – I'll remove all non-labeled data.

I'll try these two approaches and see which gives the better score.

Training & testing on different models:

- For now I am planning to use two different models, one for classification and once for localization.
- As Localization itself is many classifications, for both classification and localization single pre-trained model MAY be used which can be fine-tuned on our data. I would also like to check as per this approach.

Baseline Model:

For Classification : EfficientNetV2B0 – pretrained on imagenet

For Localization : YOLOV8Detector – pretrained on pascalvoc

LearningModel:

I would try the following CNN model architecture to fine-tune and see how they perform for classification and localization tasks:

EfficientNetV2B0, YOLOV8Detector