Model Training and Evaluation

- After researching and selecting suitable models, the team conducted model training and evaluation on the collected datasets. The team performed image data preprocessing before training and created a framework for training in a specific flow. Finally, the collected parameters after training were summarized.

*** Training Environment:**

+ Operating System: Linux
+ Node Name: iec-sv
+ Kernel Version: 5.15.0-112-generic
+ Architecture: x86_64
+ Python Version: 3.10.14
+ CPU (Central Processing Unit)
+ Number of Physical Cores: 6
+ Total Number of Threads: 12
+ Maximum Frequency: 3.6 GHz
+ Minimum Frequency: 2.2 GHz
+ Total RAM (Memory): 62.72 GB
+ Main Storage Drive: (/dev/mapper/ubuntu--vg-ubuntu--lv)
+ Mountpoint: File system: ext4, total capacity: 1820.92 GB
+ Boot Drive (/dev/nvme0n1p2)
+ Mountpoint /boot: File system ext4
+ Loop Devices: Mostly snap packages, with small sizes (0.04 GB to 0.09 GB).
+ Large storage capacity with the main drive being 1.82 TB. Boot drive has a capacity of 1.9 GB.
+ GPU (Graphics): NVIDIA GeForce RTX 3060, NVIDIA GeForce RTX 3060 Ti

*** Statistical Indicators:**

+ Desired number of epochs: Desired training epochs
+ Actual number of epochs: Actual epochs after the training process
+ Patient: The maximum number of epochs to stop training, meaning if the model cannot
+ be optimized further after this number of epochs, the training will stop
+ Average Precision: Measures the proportion of true positive predictions made by the model. High precision means that when the model predicts a positive result, it is often correct.
+ Average Recall: Measures the model's ability to detect true positive samples. High recall means the model can find most of the positive samples.
+ Average F1-score: The harmonic mean of Precision and Recall, providing a balanced measure between these two metrics. It is useful when a single measure is needed to evaluate the model, especially when there is an imbalance between Precision and Recall.
+ Accuracy: Measures the overall proportion of correct predictions made by the model. It shows how many percent of the total samples the model predicted correctly.

- To achieve accurate classification of different types of waste from collected images, an excellent classification model is required. Therefore, the team carefully compared and selected the baseline model. The popular image classification models today are CNN, Resnet, VGG, Densenet, CoAtNet, EfficientNet, Pyramid-Net, ViTB/16 (Pytorch), with VGG and ResNet being typical models. To choose a suitable image classification model, the team used existing public datasets to compare CNN models and some pretrained models such as Resnet, VGG, Densenet, CoAtNet, EfficientNet, Pyramid-Net, and ViTB/16 (Pytorch). 200 training steps were performed. After training, the accuracy and model evaluation metrics were compared as shown in the statistics table below.

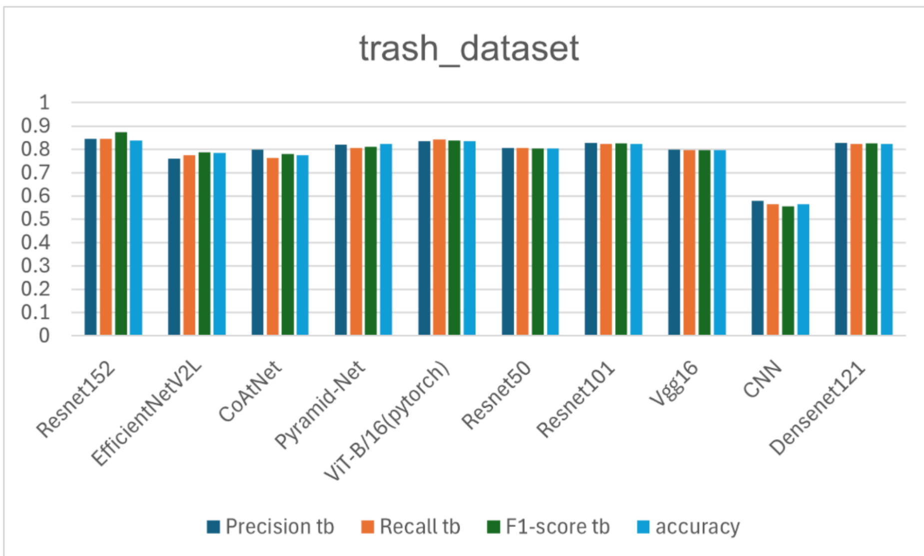

Figure 1: Comparison of evaluations between models on the trash_dataset

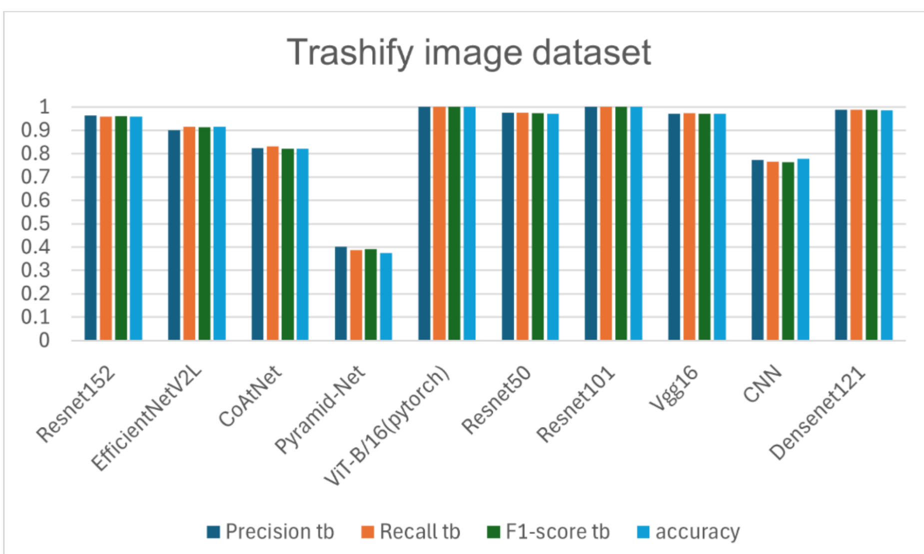

Figure 2: Comparison of evaluations between models on the Trashify image dataset

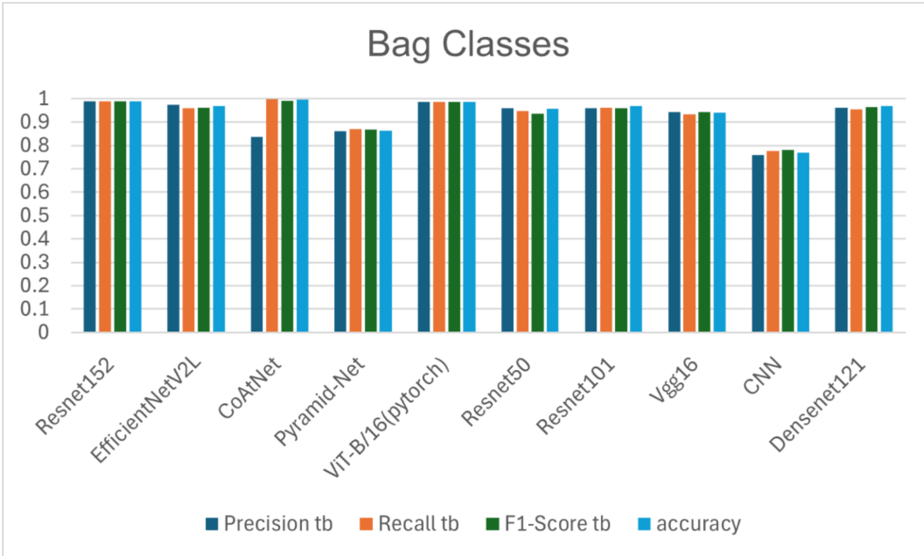

Figure 3: Comparison of evaluations between models on the Bag Classes dataset

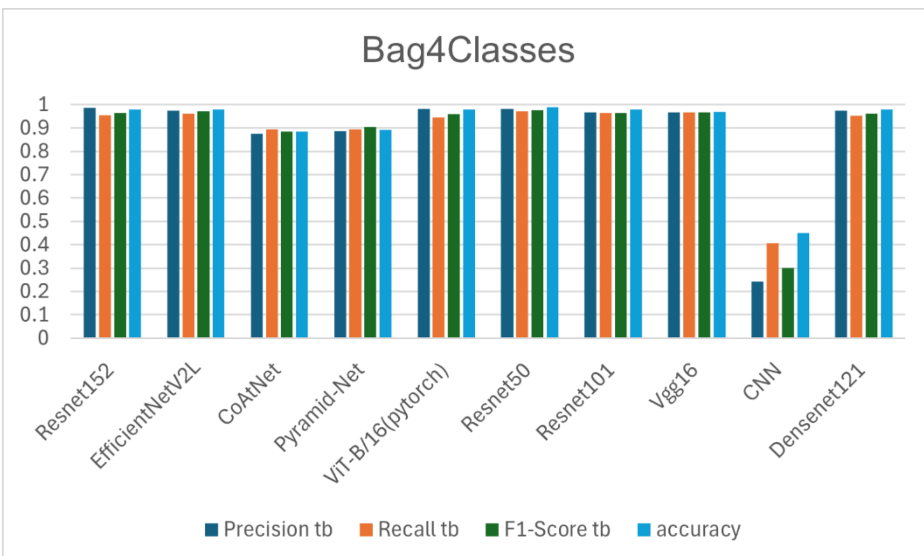

Figure 4: Comparison of evaluations between models on the Bag4Classes dataset

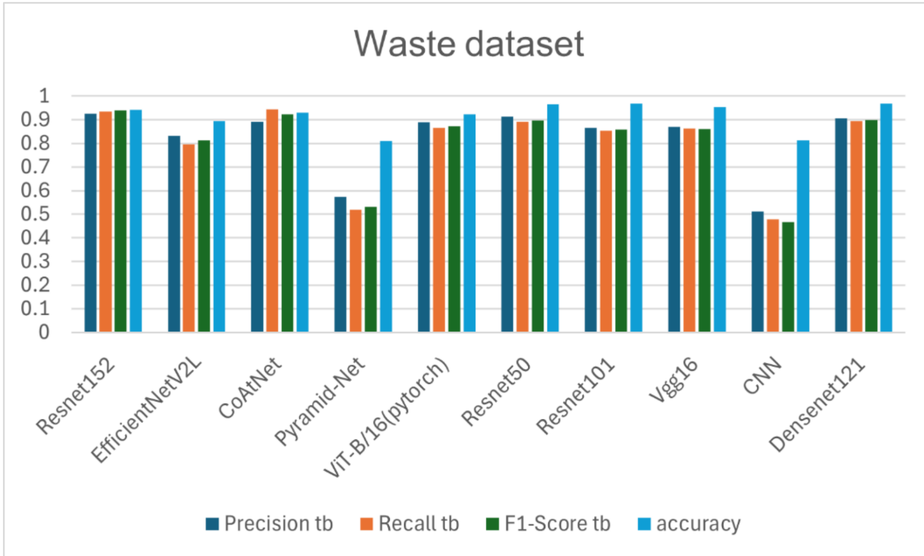

Figure 5: Comparison of evaluations between models on the Waste dataset

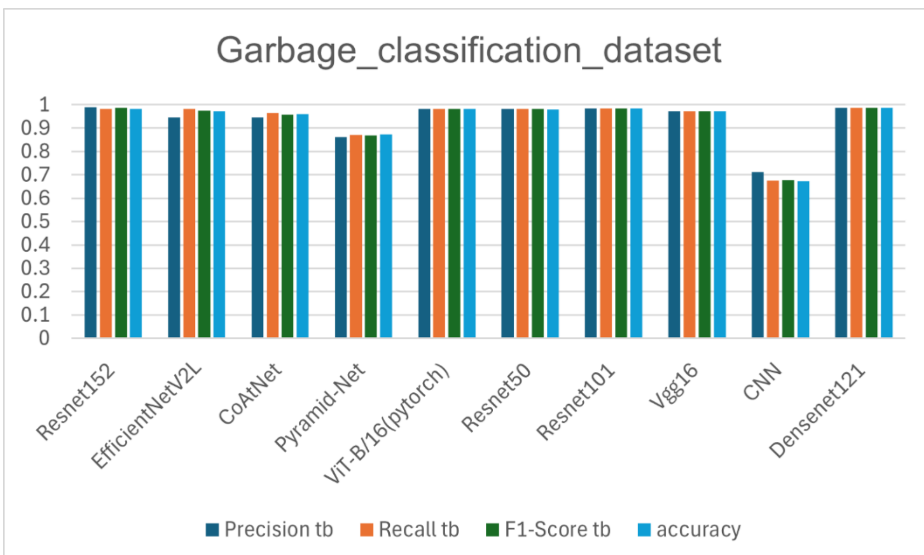

Figure 6: Comparison of evaluations between models on the Garbage_classification_dataset

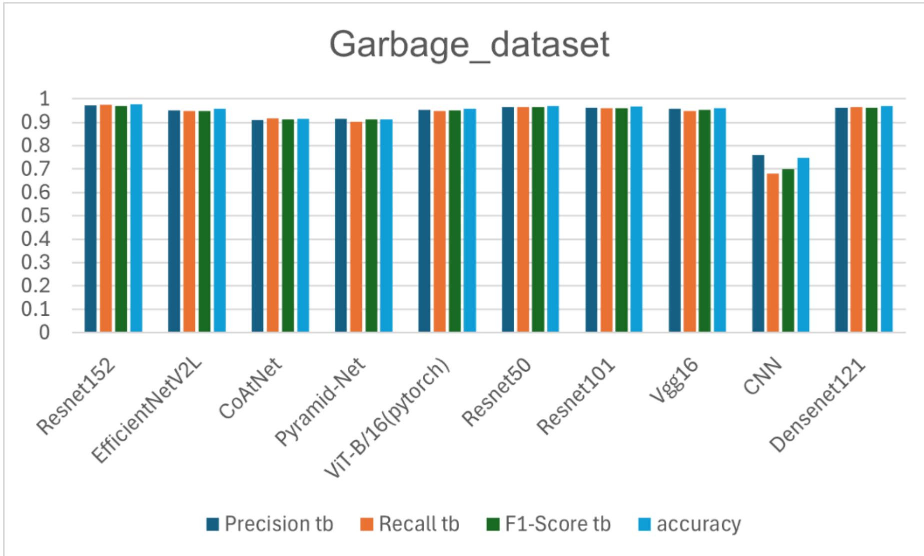

Figure 7: Comparison of evaluations between models on the Garbage_dataset

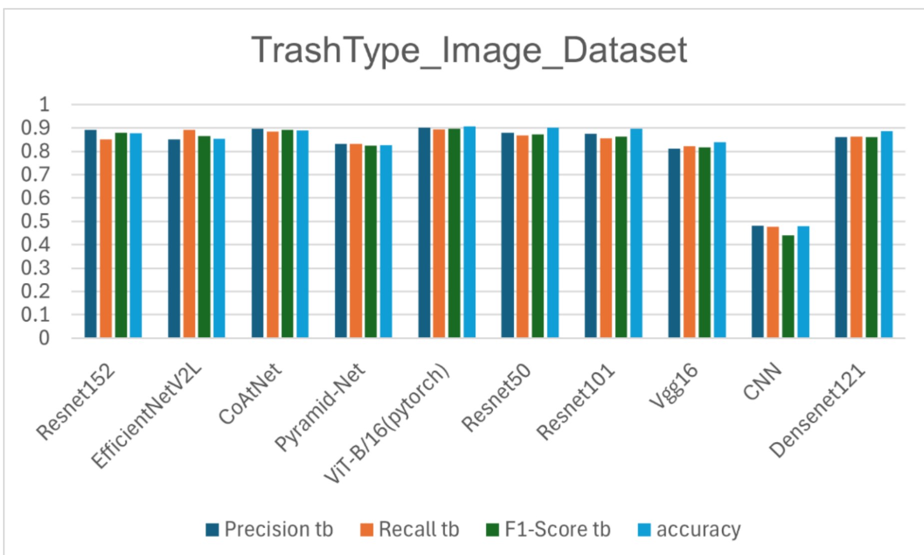

Figure 8: Comparison of evaluations between models on the TrashType_Image_Dataset

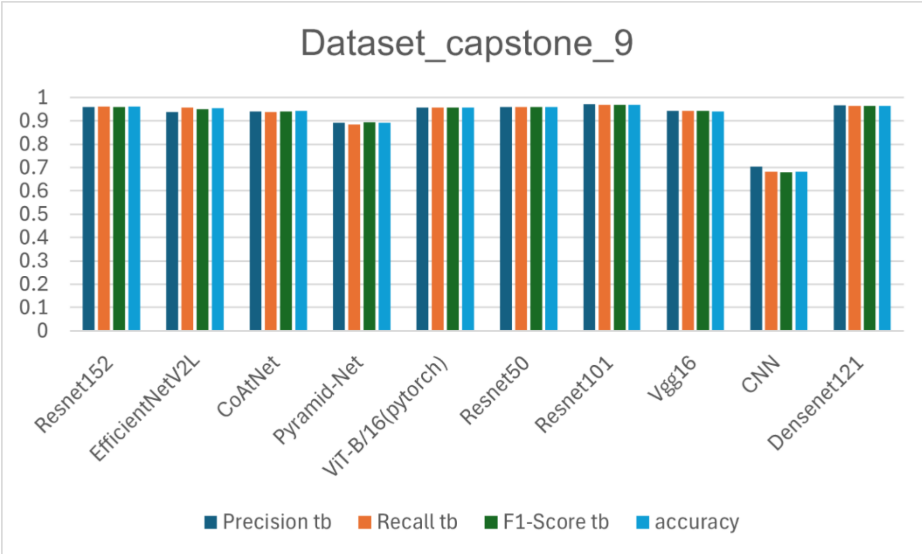

Figure 9: Comparison of evaluations between models on the Dataset_capstone_9

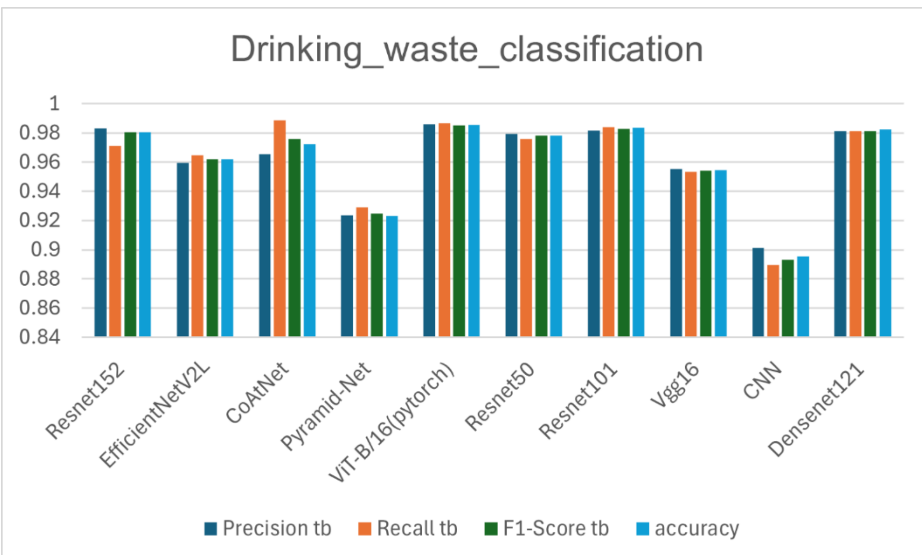

Figure 10: Comparison of evaluations between models on the Drinking_waste_Classification dataset

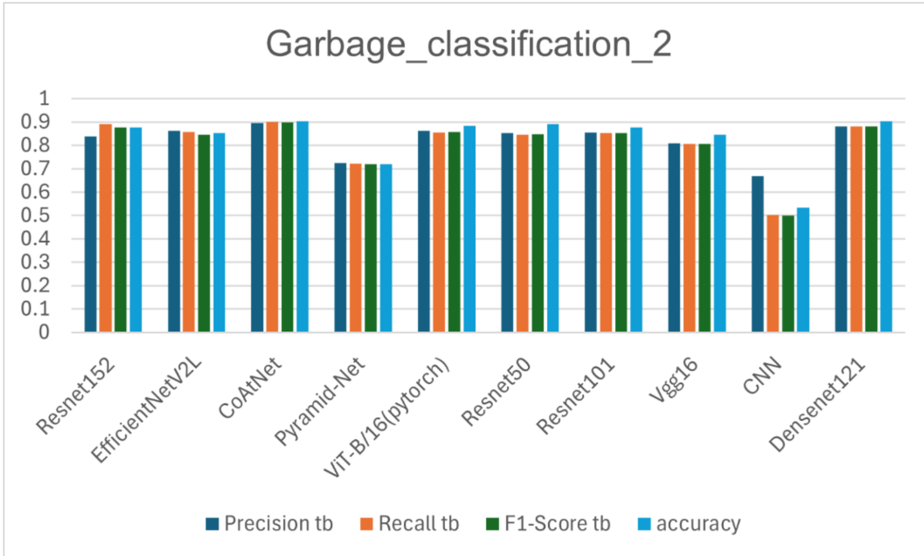

Figure 11: Comparison of evaluations between models on the Garbage_classification_2 dataset

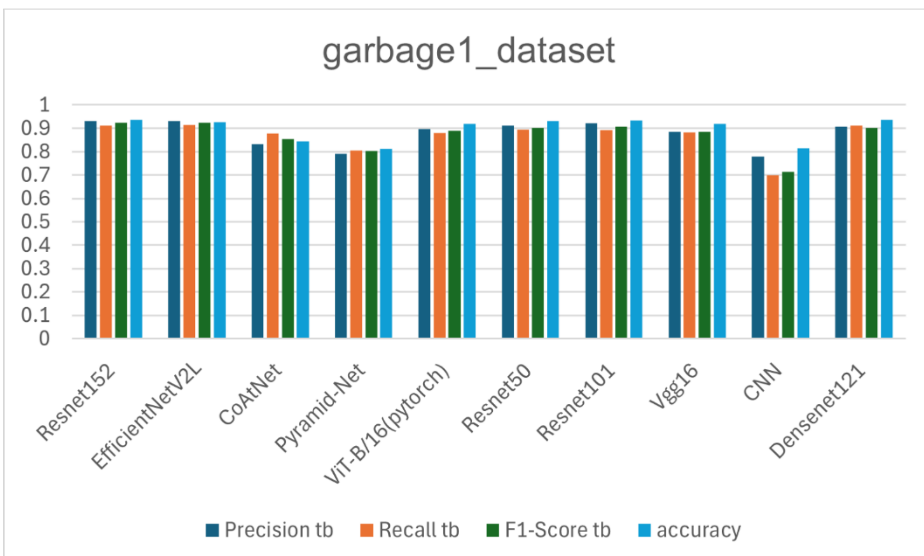

Figure 12: Comparison of evaluations between models on the garbage1_dataset

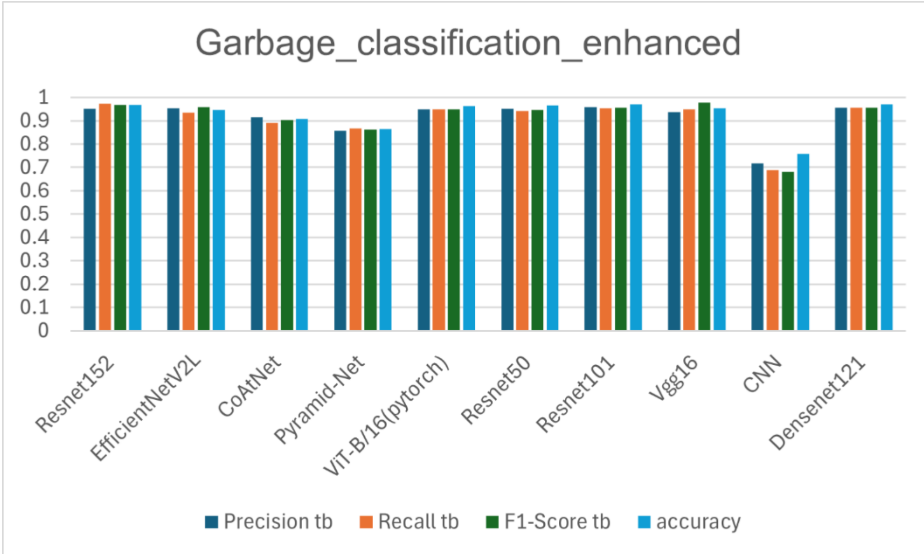

Figure 13: Comparison of evaluations between models on the Garbage_classificaion_enhanced dataset

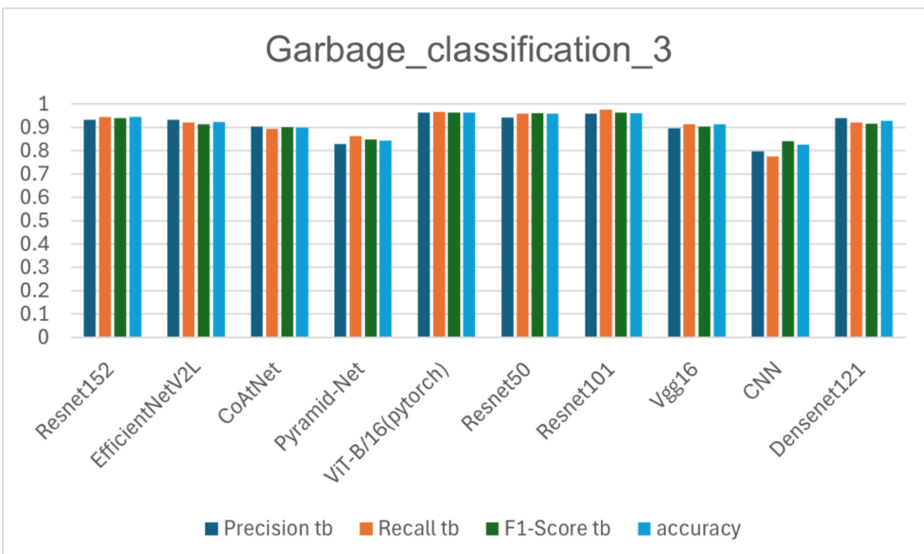

Figure 14: Comparison of evaluations between models on the Garbage_classificaion_3 dataset

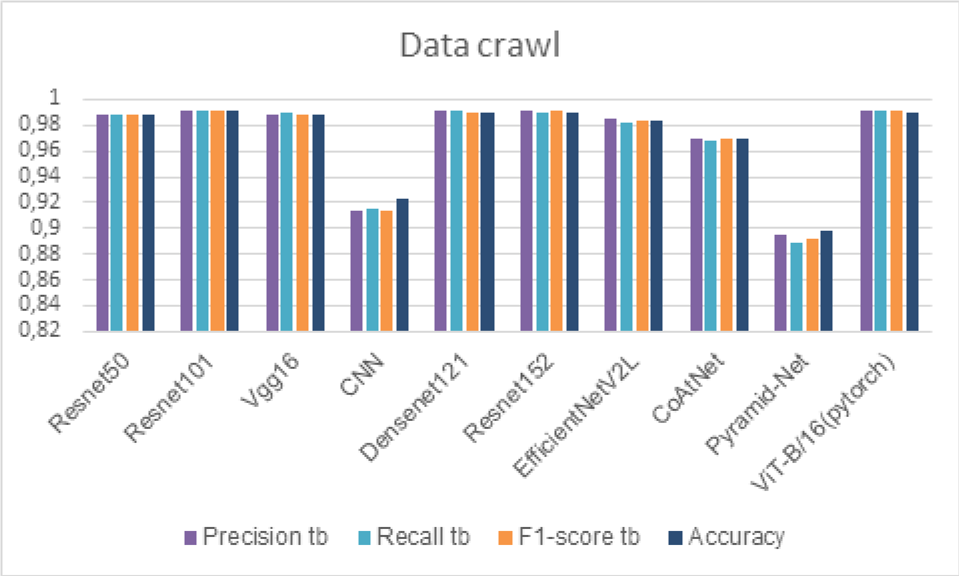

Figure 15: Comparison of model evaluations on the Data crawl dataset

Remarks:

- Resnet152:

+ Advantages:

- High performance on most datasets, achieving the highest accuracy and F1 in many cases.
- Moderate number of training epochs compared to other models.

+ Disadvantages:

- May be less efficient on certain datasets (e.g., TrashType Image).

- EfficientNetV2L:

+ Advantages:

- Achieves high accuracy and F1 on most datasets, only slightly behind Resnet152 in some cases.
- Requires fewer computational resources than models with similar performance (e.g., Resnet152), saving training time and costs.

+ Disadvantages:

- May be less efficient than Resnet152 on certain datasets.
- May require more training epochs to achieve desired accuracy compared to Resnet152.

- May be harder to interpret predictions compared to some other models (e.g., Vgg16).

- CoAtNet:

+ Advantages:

- High computational efficiency, requiring the fewest training epochs.

+ Disadvantages:

- Inconsistent performance may struggle on some datasets.
- Lower accuracy and F1 compared to Resnet152 and EfficientNetV2L in many cases.

- Pyramid-Net:

+ Advantages:

- May require fewer training epochs than some other models (e.g., Resnet152, EfficientNetV2L).

+ Disadvantages:

- Overall, the poorest performance among compared models.
- Requires the highest number of training epochs.
- Lowest accuracy and F1 on most datasets.

- ViT-B/16 (Pytorch):

+ Advantages:

- Shows high potential, achieving high accuracy on many datasets.

+ Disadvantages:

- Requires a high number of training epochs.
- Performance may be unstable on some datasets.

- Resnet50:

+ Advantages:

- Good choice for accuracy and F1 with a reasonable number of training epochs.
- Comparable performance to Resnet152 on some datasets.

+ Disadvantages:

- May have lower performance compared to more advanced models (e.g., Resnet152, EfficientNetV2L) on some datasets.
- Limited interpretability compared to some other models (e.g., Vgg16).

- Resnet101:

+ Advantages:

- Similar performance to Resnet50 but may require slightly more training epochs.

+ Disadvantages:

- Requires more training epochs than Resnet50, leading to longer training times and higher computational costs.
- Slightly lower performance compared to more advanced models (e.g., Resnet152, EfficientNetV2L) on some datasets.
- Limited interpretability compared to some other models (e.g., Vgg16).

- Vgg16:

+ Advantages:

- Simple and understandable architecture: Compared to more modern CNN models, Vgg16 has a simpler structure, making it easier to understand and modify.
- Good performance on smaller datasets: Vgg16 can perform well on smaller datasets with limited training samples, where more complex models may struggle.
- Lower computational requirements: Vgg16 requires fewer computational resources than more advanced models, making it suitable for lower-spec devices.

+ Disadvantages:

- Generally poorer performance compared to Resnet models in terms of accuracy and F1.

May be more suitable for less complex datasets.

- CNN (Custom):

+ Advantages:

- Customizable: The CNN model can be tailored specifically for the task, potentially offering better performance compared to pretrained models on specialized datasets.
- Adjustable architecture: The CNN can be modified to fit specific computational constraints or performance requirements.

+ Disadvantages:

- Difficult to evaluate performance without detailed information about the architecture.
- Current results show lower performance compared to pretrained models.

- Densenet121:

+ Advantages:

- Good performance on most datasets, achieving accuracy and F1 close to Resnet152.

+ Disadvantages:

- Requires slightly more training epochs than Resnet152.