



CLOUDY THE
SCIENTIST

ZERO WASTE

ITCS227 Introduction to Data Science

PROJECT GOAL

Misclassification of waste leads to poor recycling and environmental issues.

This project aims to build an image classification model that identifies garbage types and suggests the correct waste bin color.



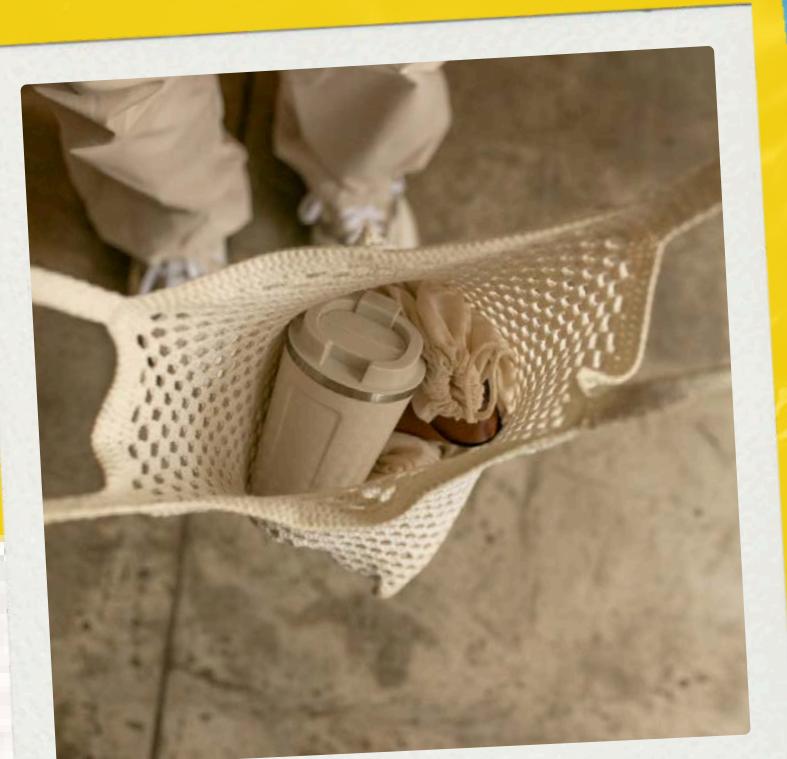
ZERO WASTE PRACTICE

The principles of zero-waste are straightforward actions to minimize waste, conserve resources, and safeguard the environment. By embracing these practices, we can make zero-waste living both simpler and more effective.

REDUCE

RECYCLE

REUSE





DATA SOURCE



- The data that we are using in this project is Garbage & Recyclable Image Dataset.
- The source that we use is from "**Recyclable and Household Waste Classification**" which is from the website "<https://www.kaggle.com>"
- The size of the data that we use is "19577" photo"
Train: 16427 images
Validate: 3150 images
- We get the data by searching for Public Datasets.

CLASSIFY CLASSES

CARDBOARD

CLOTHING

PLASTIC

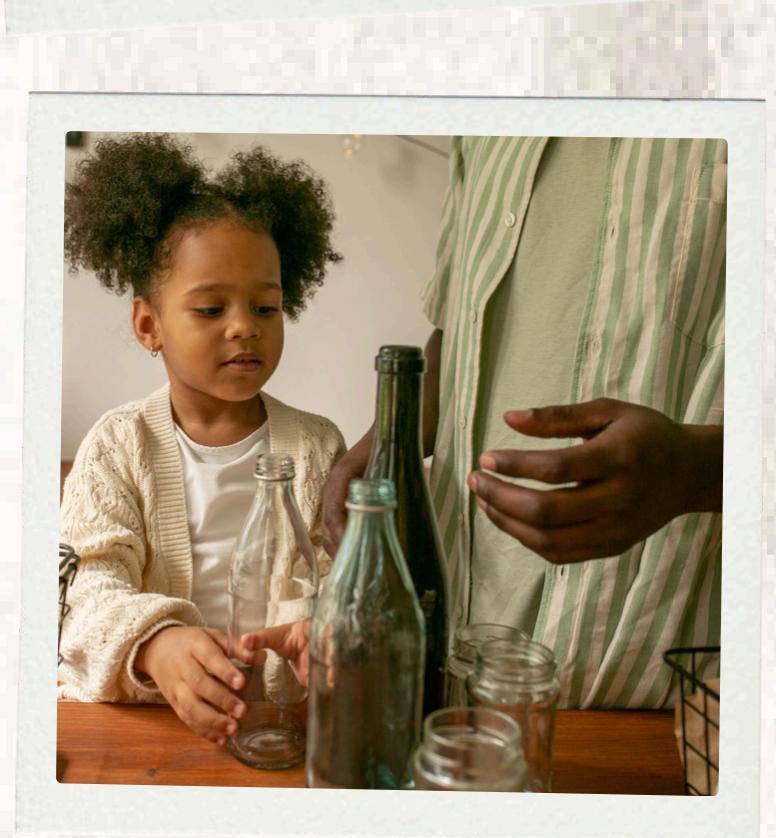
FOOD WASTE

GLASS

PAPER

METAL

STYROFOAM



PROPOSED METHODOLOGY

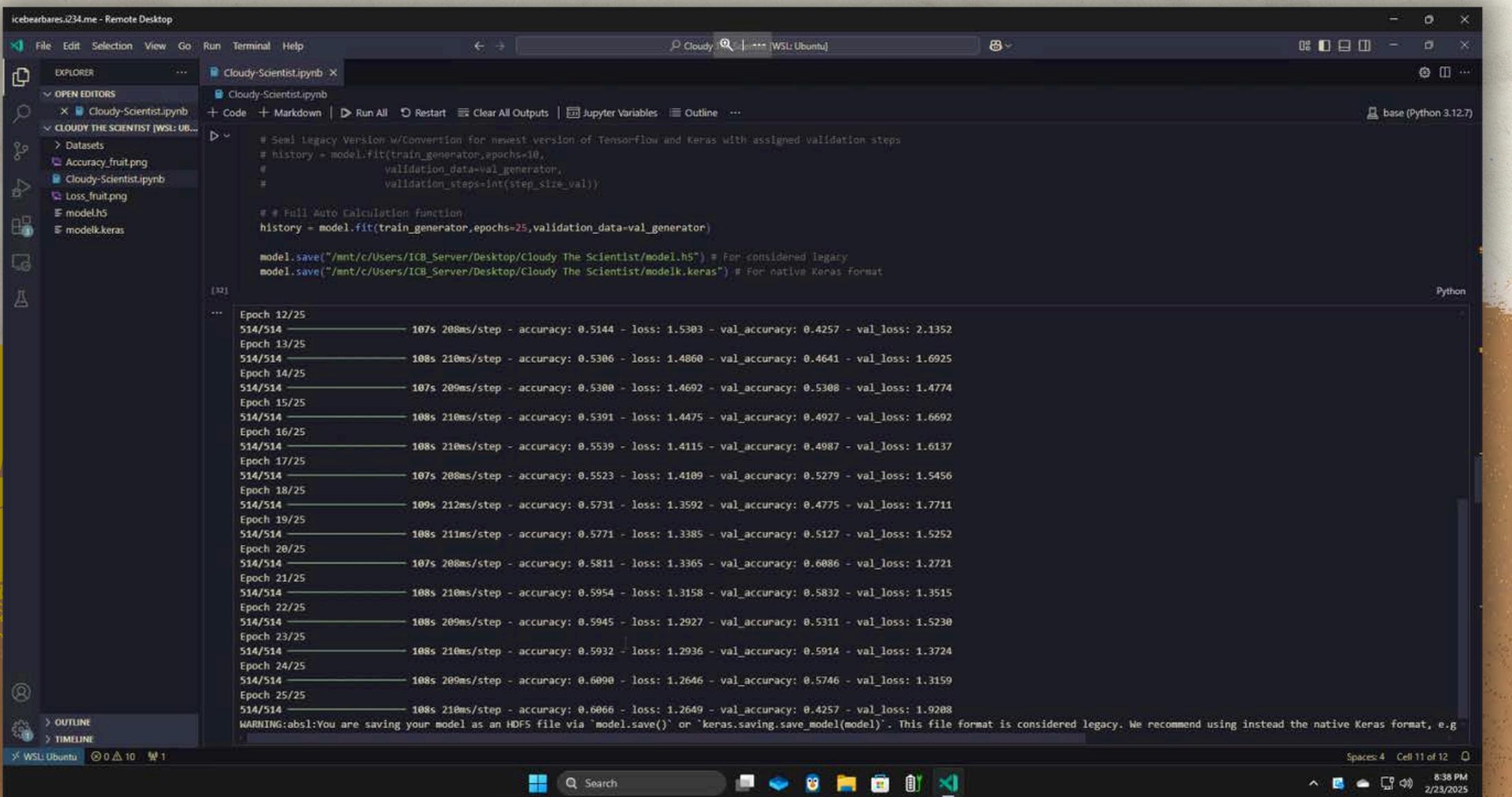
Tensorflow: Sequential model



A Sequential model is appropriate for a plain stack of layers where each layer has exactly one input tensor and one output tensor.

PROPOSED METHODOLOGY

Tensorflow: Sequential model



The screenshot shows a Jupyter Notebook interface running on a Windows desktop. The notebook is titled "Cloudy-Scientist.ipynb". The code cell contains Tensorflow code for training a sequential model. The output cell displays the training progress for 25 epochs, showing metrics like accuracy, loss, and validation accuracy over time.

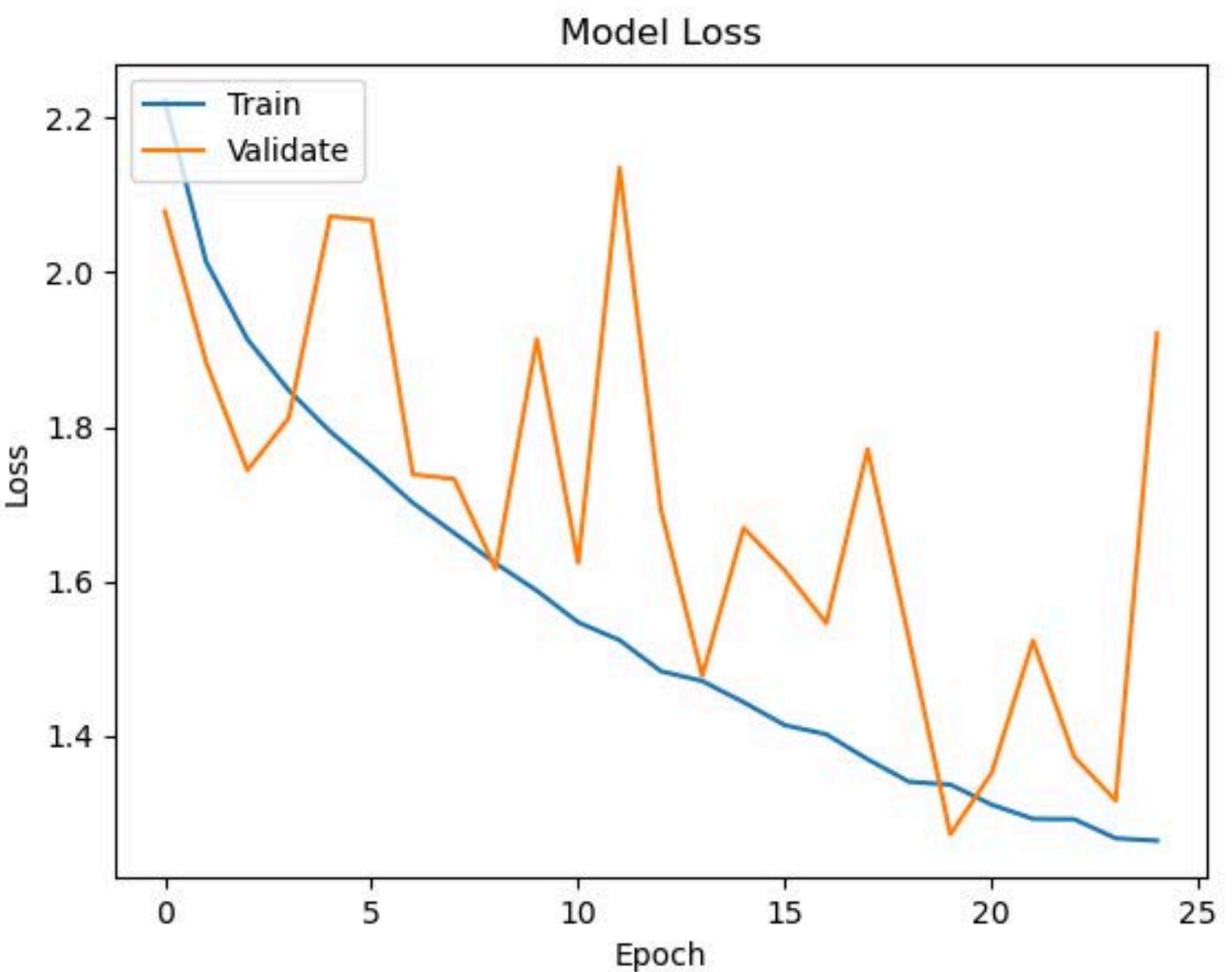
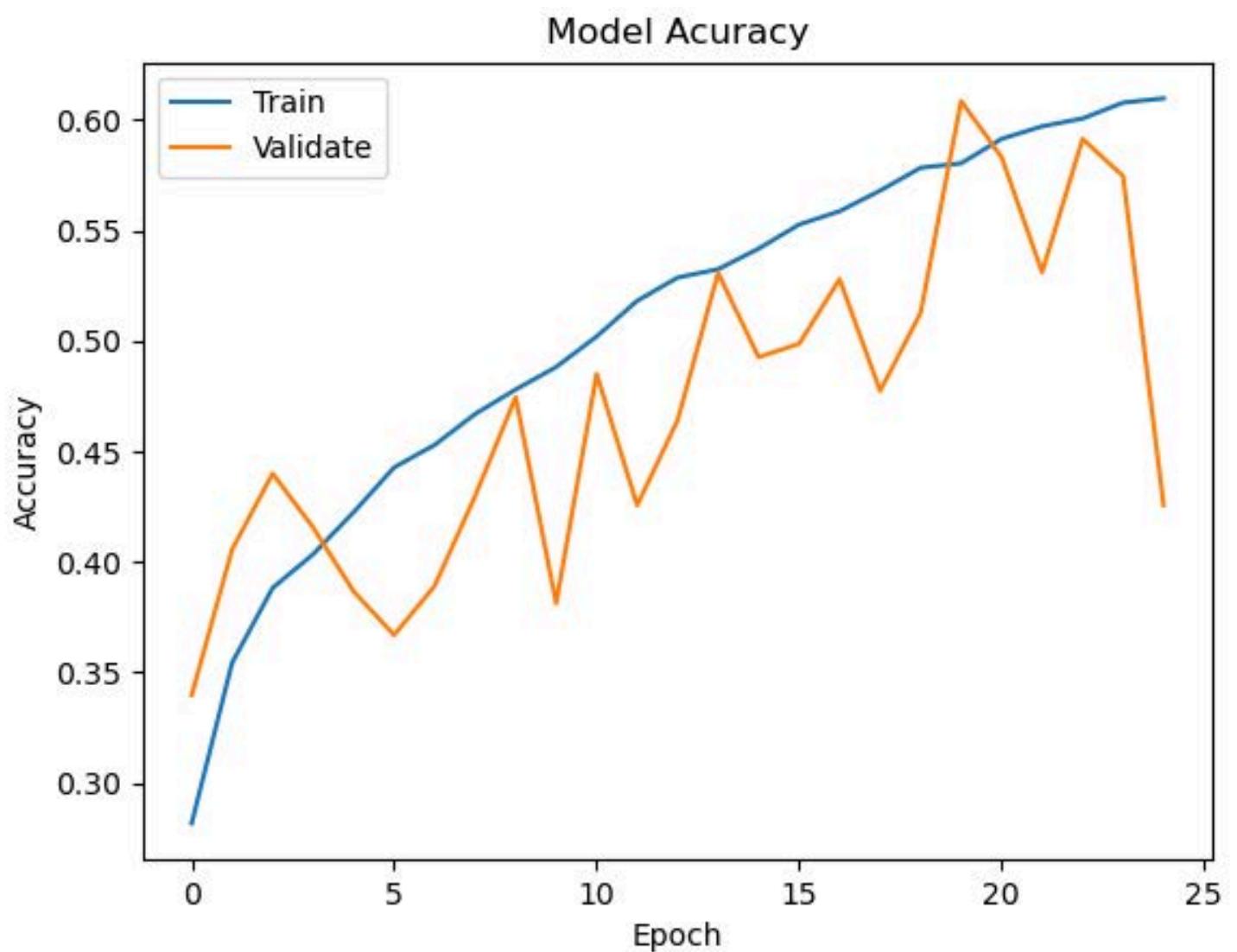
```
# Semi Legacy Version w/Conversion for newest version of Tensorflow and Keras with assigned validation_steps
# history = model.fit(train_generator,epochs=10,
#                      validation_data=val_generator,
#                      validation_steps=int(step_size_val))

# # Full Auto Calculation Function
history = model.fit(train_generator,epochs=25,validation_data=val_generator)

model.save("/mnt/c/Users/ICB_Server/Desktop/Cloudy The Scientist/model.h5") # For considered legacy
model.save("/mnt/c/Users/ICB_Server/Desktop/Cloudy The Scientist/model.keras") # For native Keras format

Epoch 12/25
514/514 107s 208ms/step - accuracy: 0.5144 - loss: 1.5303 - val_accuracy: 0.4257 - val_loss: 2.1352
Epoch 13/25
514/514 108s 210ms/step - accuracy: 0.5306 - loss: 1.4860 - val_accuracy: 0.4641 - val_loss: 1.6925
Epoch 14/25
514/514 107s 209ms/step - accuracy: 0.5300 - loss: 1.4692 - val_accuracy: 0.5308 - val_loss: 1.4774
Epoch 15/25
514/514 108s 210ms/step - accuracy: 0.5391 - loss: 1.4475 - val_accuracy: 0.4927 - val_loss: 1.6692
Epoch 16/25
514/514 108s 210ms/step - accuracy: 0.5539 - loss: 1.4115 - val_accuracy: 0.4987 - val_loss: 1.6137
Epoch 17/25
514/514 107s 208ms/step - accuracy: 0.5523 - loss: 1.4109 - val_accuracy: 0.5279 - val_loss: 1.5456
Epoch 18/25
514/514 109s 212ms/step - accuracy: 0.5731 - loss: 1.3592 - val_accuracy: 0.4775 - val_loss: 1.7711
Epoch 19/25
514/514 108s 211ms/step - accuracy: 0.5771 - loss: 1.3385 - val_accuracy: 0.5127 - val_loss: 1.5252
Epoch 20/25
514/514 107s 208ms/step - accuracy: 0.5811 - loss: 1.3365 - val_accuracy: 0.6086 - val_loss: 1.2721
Epoch 21/25
514/514 108s 210ms/step - accuracy: 0.5954 - loss: 1.3158 - val_accuracy: 0.5832 - val_loss: 1.3515
Epoch 22/25
514/514 108s 209ms/step - accuracy: 0.5945 - loss: 1.2927 - val_accuracy: 0.5311 - val_loss: 1.5230
Epoch 23/25
514/514 108s 210ms/step - accuracy: 0.5932 - loss: 1.2936 - val_accuracy: 0.5914 - val_loss: 1.3724
Epoch 24/25
514/514 108s 209ms/step - accuracy: 0.6090 - loss: 1.2646 - val_accuracy: 0.5746 - val_loss: 1.3159
Epoch 25/25
514/514 108s 210ms/step - accuracy: 0.6066 - loss: 1.2649 - val_accuracy: 0.4257 - val_loss: 1.9208
WARNING:absl:You are saving your model as an HDF5 file via `model.save()` or `keras.saving.save_model(model)`. This file format is considered legacy. We recommend using instead the native Keras format, e.g.
```

```
514/514 ----- 107s 208ms/step - accuracy: 0.5811 - loss: 1.3365 - val_accuracy: 0.6086 - val_loss: 1.2721
Epoch 21/25
514/514 ----- 108s 210ms/step - accuracy: 0.5954 - loss: 1.3158 - val_accuracy: 0.5832 - val_loss: 1.3515
Epoch 22/25
514/514 ----- 108s 209ms/step - accuracy: 0.5945 - loss: 1.2927 - val_accuracy: 0.5311 - val_loss: 1.5230
Epoch 23/25
514/514 ----- 108s 210ms/step - accuracy: 0.5932 - loss: 1.2936 - val_accuracy: 0.5914 - val_loss: 1.3724
Epoch 24/25
514/514 ----- 108s 209ms/step - accuracy: 0.6090 - loss: 1.2646 - val_accuracy: 0.5746 - val_loss: 1.3159
Epoch 25/25
514/514 ----- 108s 210ms/step - accuracy: 0.6066 - loss: 1.2649 - val_accuracy: 0.4257 - val_loss: 1.9208
WARNING:absl:You are saving your model as an HDF5 file via `model.save()` or `keras.saving.save_model(model)` . This file
```



PROPOSED METHODOLOGY (PLAN B)

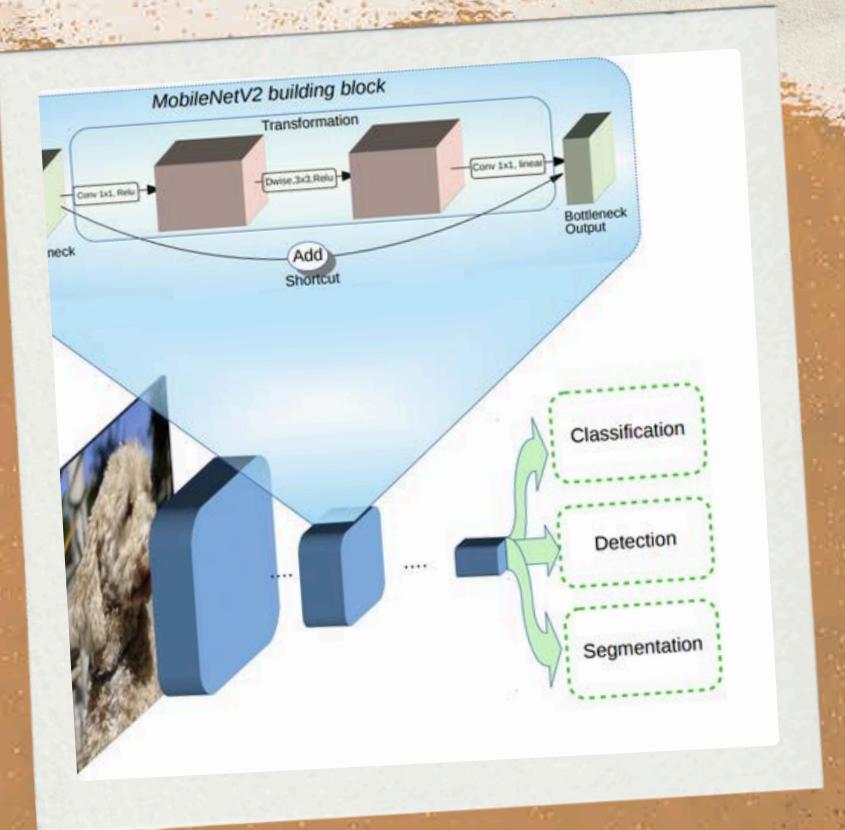
- USE EARLY STOPPING
- ADD MORE DATA AUGMENTATION
- LEARNING RATE ADJUSTMENT

PROPOSED METHODOLOGY <PLAN B>

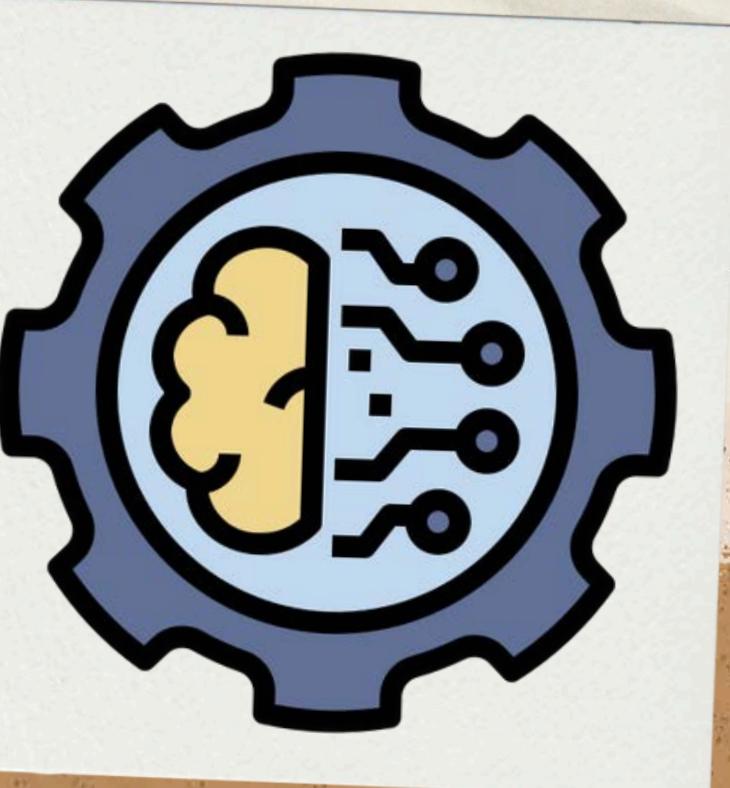
Tensorflow: Sequential model



Enhance with pre-trained
MobileNet-V2



Very accurate model





TIMELINE

Collect the data → Model Training → Model Testing/Optimization

Monitoring the result ← Deployment

EXPECTED OUTCOME



- **Effective waste categorization**

Efficient waste categorization results in the waste separation being as we want and it is in the correct categories.

- **Save budget on waste categorization**

Categorization with Machine Learning, we can save the budget on people to separate the waste manually.

- **Convenience for users**

A good and fast waste categorization system makes it more convenient for people to separate their waste.

THE IMPACT WE CAN MAKE



- Reduce waste and pollution
- Reuse and recycle to save energy
- Protect ecosystems from harmful waste

**THANK
YOU. FOR YOUR
ATTENTION!**