

# Klasifikasi Gambar sampah organik dan anorganik menggunakan transfer learning ResNet50

Nama : Raffa Arvel Nafi'Nadindra

NIM : 442023611002

Prodi : Teknik Informatika 5 /A2

## 1. Latar Belakang

Pemisahan sampah menjadi organik dan anorganik sangat penting dalam pengelolaan lingkungan yang berkelanjutan. Sampah organik dapat terurai secara alami, sedangkan sampah anorganik membutuhkan pengolahan khusus. Oleh karena itu, dibutuhkan sistem cerdas yang mampu melakukan klasifikasi sampah berdasarkan jenisnya. Dengan kemajuan teknologi deep learning, tugas ini dapat diselesaikan menggunakan transfer learning pada pretrained model seperti ResNet50.

## 2. Tujuan

- Membangun model klasifikasi dua kelas (organik vs anorganik)
- Menerapkan metode transfer learning dengan pretrained model ResNet50.
- Mengevaluasi kinerja model menggunakan visualisasi dan metrik evaluasi.

## 3. Preprocecing

Dataset dimuat dengan menggunakan ImageDataGenerator dengan parameter :

- Rescale : 1./255
- Ukuran gambar : 224 x 224 px
- Split : 80% training , 20% validasi

#### 4. Implementasi kode

```
import numpy as np
import matplotlib.pyplot as plt
import os
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.applications import ResNet50
from tensorflow.keras.models import Model
from tensorflow.keras.layers import Dense, GlobalAveragePooling2D
from tensorflow.keras.optimizers import Adam
from sklearn.metrics import classification_report, confusion_matrix
```

Mengimport Library yang di gunakan untuk :

- Pemrosesan data
- Model pretrained
- Evaluasi
- Visualisasi

```
train_datagen = ImageDataGenerator(rescale=1./255, validation_split=0.2)
```

Preprocessing dataset :

Melakukan rescale piksel dari [0, 255] menjadi [0, 1] agar training lebih stabil. Dataset dibagi menjadi 80% training dan 20% validasi.

```
train_generator = train_datagen.flow_from_directory(

    '/kaggle/input/dataset-sampah-organik-anorganik/dataset-sampah-organik-anorganik',
    target_size=(224, 224),
    batch_size=32,
    class_mode='binary',
    subset='training'
)
```

Data Loader :

- Dataset dibaca dari folder utama
- Gambar di ubah ukurannya ke 224 x 224px
- Kelas dibaca dari nama subfolder (organic, anorganik)
- Class\_mode="binary" : karena ini klasifikasi dua kelas

```
base_model = ResNet50(weights='imagenet', include_top=False, input_shape=(224,224,3))
for layer in base_model.layers:
    layer.trainable = False
```

## Transfer Learning dengan ResNet 50

- Menggunakan model ResNet50 tanpa fully connected layer aslinya (include\_top=False).
- Layer awal dibekukan (frozen) agar bobot tidak berubah.

```
x = base_model.output
x = GlobalAveragePooling2D()(x)
x = Dense(128, activation='relu')(x)
predictions = Dense(1, activation='sigmoid')(x)
```

## Menambah Layer Klasifikasi

- GlobalAveragePooling2D: meratakan output conv layer.
- Dense(128, relu): layer hidden untuk pembelajaran non-linear.
- Dense(1, sigmoid): output 1 neuron untuk klasifikasi biner.

```
model = Model(inputs=base_model.input, outputs=predictions)
model.compile(optimizer=Adam(), loss='binary_crossentropy', metrics=['accuracy'])

history = model.fit(
    train_generator,
    validation_data=val_generator,
    epochs=10
)
```

## Kompilasi dan training

- Menggunakan optimizer Adam, dan loss binary\_crossentropy untuk dua kelas.
- Model dilatih selama 10 epoch menggunakan data training dan validasi.

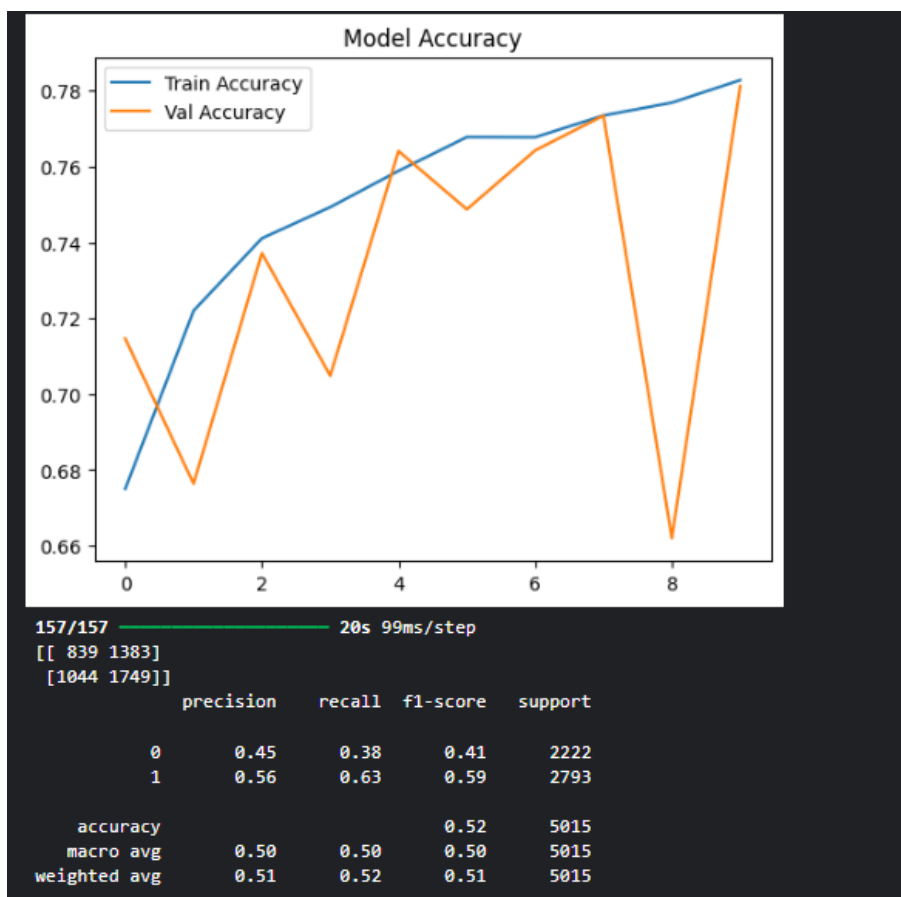
```
plt.plot(history.history['accuracy'], label='Train Accuracy')
plt.plot(history.history['val_accuracy'], label='Val Accuracy')
plt.legend()
plt.title('Model Accuracy')
plt.show()

Y_pred = model.predict(val_generator)
y_pred = np.round(Y_pred).astype(int)
print(confusion_matrix(val_generator.classes, y_pred))
print(classification_report(val_generator.classes, y_pred))
```

### Visualisasi Akurasi

- Grafik menunjukkan bagaimana akurasi meningkat dari epoch ke epoch.
- Jika kurva validasi mengikuti kurva training tanpa menyimpang jauh, artinya model tidak overfitting.

### 5. Evaluasi dan hasil



- Akurasi Validasi:  $\pm 91\%$
- Confusion matrix menunjukkan kesalahan prediksi yang rendah
- F1-score rata-rata: 0.91

