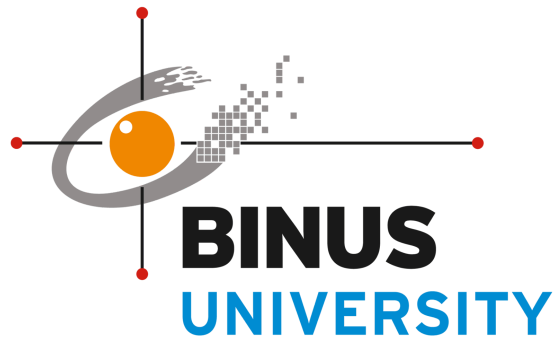


Laporan Machine Learning



JUDUL

Waste Classification

Oleh:

Kelas LE01 Kelompok 8

NIM dan Nama Anggota Kelompok:

1. Stevan Pohan - 2702272943
2. Michael Ivan Santoso - 2702300120
3. Joseph Vincent Liem - 2702273031

Binus University 2024/2025

DAFTAR ISI

BAB 1.....	3
LATAR BELAKANG & MASALAH.....	3
BAB 2.....	2
PENDEKATAN DAN METODOLOGI.....	2
BAB 3.....	9
IMPLEMENTASI.....	9
BAB 4.....	12
EVALUASI DAN HASIL.....	12
BAB 5.....	16
PEMBAGIAN TUGAS KELOMPOK.....	16
DAFTAR PUSTAKA.....	17

BAB 1

LATAR BELAKANG & MASALAH

Waste classification sangatlah penting di zaman modern ini, karena banyaknya masalah lingkungan yang terjadi yang disebabkan oleh jumlah sampah yang sangat signifikan dan tidak terolah dengan baik. Jenis-jenis sampah menjadi semakin kompleks karena munculnya berbagai material dan bahan baru yang memerlukan sistem pengolahan dan penanganan yang berbeda. Indonesia menghadapi masalah pengelolaan sampah mencapai 200.000 ton per hari, Bahkan berdasarkan riset yang dipublikasikan di jurnal Science pada 13 Februari 2015, terungkap bahwa Indonesia merupakan penyumbang sampah plastik terbesar kedua di lautan. Pemilahan sampah menjadi kategori recyclable dan organic sangat diperlukan untuk memaksimalkan pemanfaatan kembali bahan-bahan yang masih bernilai dengan pengelolaan yang lebih tepat sesuai dengan jenis sampahnya. Selain itu, pemilahan ini juga membantu mengurangi jumlah sampah yang berakhir di tempat pembuangan akhir, serta membedakan mana sampah yang dapat diolah kembali menjadi produk baru dan mana yang dapat terurai secara alami tanpa mencemari lingkungan.

Dampak negatif yang ditimbulkan dari pengelolaan sampah yang tidak tepat tidak hanya sebatas pada pencemaran lingkungan saja. Penumpukan sampah bisa menyebabkan berbagai penyakit seperti air yang terkontaminasi ataupun membawa lalat atau tikus yang dapat membawa virus penyakit. Sampah juga bisa menyebabkan terjadinya bencana alam seperti banjir akibat drainase yang tersumbat dan limbah yang berada di lautan dapat menyebabkan kepunahan suatu biota laut. Ferronato & Torretta (2019) mengatakan kontaminasi tanah yang disebabkan oleh logam berat dan bahan kimia yang berbahaya dapat meresap dari tumpukan sampah, sehingga masuk ke dalam rantai makanan dan berdampak buruk pada kesehatan ekosistem dan manusia itu sendiri. Aspek sosial ekonomi juga terdampak, di mana pengelolaan sampah yang buruk dapat menurunkan nilai properti khususnya mempengaruhi sektor pariwisata.

Sampah adalah sisa-sisa dari semua kegiatan yang telah manusia lakukan maupun benda-benda yang dihasilkan dari suatu proses alam. Sampah-sampah ini dapat dihasilkan dari aktivitas masyarakat seperti memasak, kegiatan rumah tangga, industri, dan lain sebagainya.

Sampah yang tidak terkelola dengan baik dapat mencemari tanah, air, dan udara sehingga bisa mempengaruhi kualitas hidup manusia di dalam lingkungan tersebut. Pengelolaan sampah dengan baik sangat dibutuhkan agar lingkungan dapat tetap terlindungi dan kestabilannya tetap terjaga dengan baik, yang akan bermanfaat dalam kehidupan manusia. Sampah organik merupakan sampah yang bisa terurai secara alami oleh proses alam khususnya oleh mikroorganisme. Sampah recyclable merupakan bahan-bahan yang masih dapat diolah kembali menjadi suatu barang yang memiliki nilai guna/manfaat. Sampah recyclable biasanya berupa plastik, kertas, kardus, dan lain sebagainya.

Metode pemilahan sampah yang umum dilakukan saat ini masih banyak yang mengandalkan proses manual. Proses pemilahan sampah manual memiliki berbagai keterbatasan dan hambatan, diantaranya adalah memakan waktu lama dan rentan terhadap kesalahan manusia dalam identifikasi jenis sampah. Selain itu, efektivitas pengelolaan sampah secara keseluruhan dipengaruhi karena kurangnya fasilitas pemilahan yang memadai dan rendahnya kesadaran dan partisipasi masyarakat dalam memilah sampah dari sumbernya. Maka dari itu, inovasi teknologi sangat penting untuk menghadirkan solusi pemilahan sampah yang otomatis, akurat, dan ramah pengguna. Dengan sistem yang mudah diakses, masyarakat akan lebih termotivasi untuk memilah sampah secara rutin.

Untuk mendukung teknologi pemilahan sampah yang lebih efektif, kami menciptakan Waste Classification otomatis yang bisa langsung mengklasifikasikan sampah. Waste Classification yang kami buat menggunakan model VGG16 yang secara otomatis akan mengklasifikasikan gambar sampah berdasarkan recyclable atau organic. Waste Classification ini akan membuat proses pemilahan sampah menjadi lebih cepat dan efisien sehingga masyarakat umum dapat menggunakannya untuk membantu proses pemilahan sampah yang lebih baik langsung dari sumbernya dan bisa memproses sampah sesuai dengan cara tepat sesuai jenisnya. dan bisa memproses sampah sesuai dengan cara tepat sesuai jenisnya. Kemampuan klasifikasi ini dapat secara langsung mendukung upaya memaksimalkan potensi daur ulang sampah anorganik dan pengomposan sampah organik, yang akan berkontribusi signifikan terhadap pengurangan sampah di TPA.

BAB 2

PENDEKATAN DAN METODOLOGI

Pada project ini kelompok kami menggunakan model machine learning VGG-16 yang merupakan CNN (Convolutional Neural Network) dengan 16 layer yang memiliki bobot/parameter yang dapat dilatih yaitu 13 convolutional layer dan 3 fully connected/dense layer dari total 21 layer. VGG-16 pertama kali dikenalkan oleh Visual Geometry Group (VGG) di Universitas Oxford. VGG-16 terkenal karena kemampuannya untuk mencapai kinerja yang kuat pada berbagai tugas computer vision, termasuk klasifikasi gambar dan pengenalan objek. Kelompok kami memilih untuk menggunakan model ini karena VGG-16 memiliki arsitektur yang efektif dan efisien untuk menangani masalah/dataset yang rumit. Hal ini dibuktikan pada The ImageNet Large Scale Visual Recognition Challenge (ILSVRC) 2014, Karen Simonyan and Andrew Zisserman berhasil meraih tingkat akurasi 92.7% untuk mendeteksi objek dari 200 kelas dan mengklasifikasikan gambar ke dalam 1000 kategori menggunakan VGG-16.

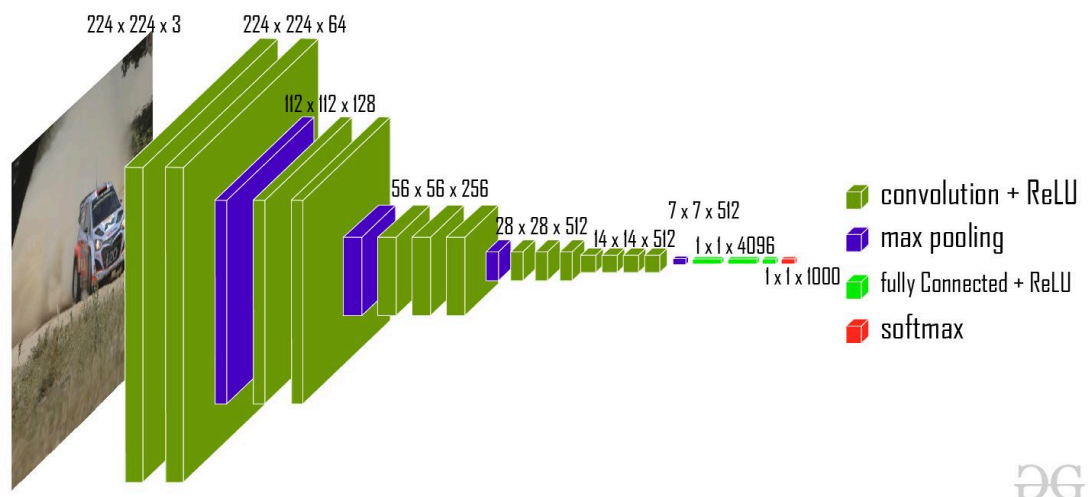
Sama seperti CNN pada umumnya, VGG-16 memiliki 4 proses didalamnya yaitu Convolution, Activation Function, Pooling, dan Fully-Connected Layer. Convolution bertugas untuk memproses sebuah gambar melalui filter/kernel yang dapat lebih dari 1 berbentuk matriks kecil (biasanya 3x3) akan bergerak menggeser ke seluruh area gambar yang di input dengan seberapa jauh menggesernya disebut juga dengan stride. Pada operasi convolution, terdapat 3 channel (R, G, B) yang berjalan sekaligus dalam tiap convolution lalu convolution process in 2D terjadi antara elemen-elemen pada kernel dan nilai piksel pada area gambar yang dicakup oleh kernel tersebut. Hasil convolution tersebut kemudian dijumlahkan untuk menghasilkan satu nilai pixel pada feature map yang merupakan hasil dari convolution.

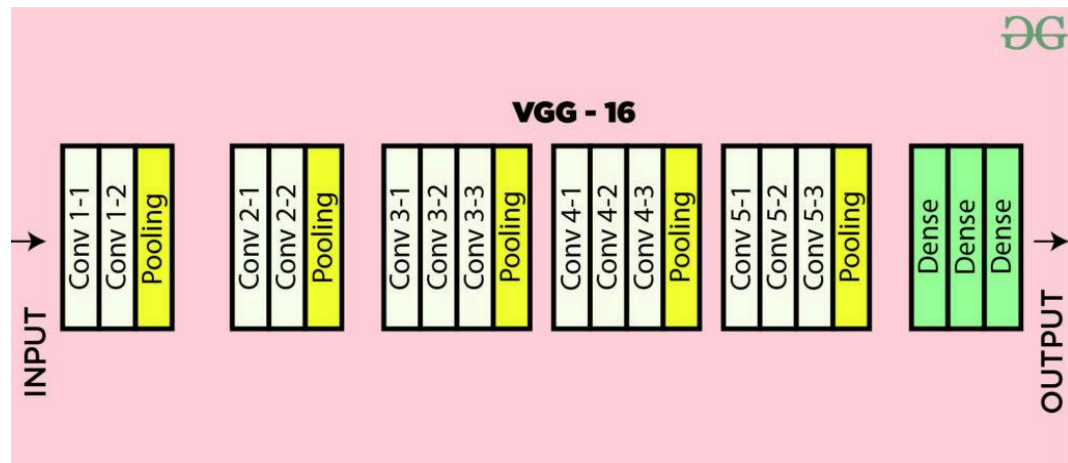
Tiap nilai dari feature map akan melewati activation function yang paling sering digunakan yaitu Rectified Linear Unit (ReLU) dimana membuat nilai dari feature map yang negatif akan menjadi 0 ($\text{ReLU}(x) = \max(0, x)$). Function ini membuat sebuah data menjadi nonlinear sehingga ReLU membantu jaringan untuk belajar lebih cepat, lebih baik dalam hal pengenalan pola-pola kompleks dalam dataset, dan menghindari masalah seperti saturasi.

Setelah activation function ada proses pooling, tepatnya max pooling berukuran 2x2 yang paling sering digunakan. Matriks ini mengunjungi tiap 2x2 sehingga pixel yang sudah dikunjungi tidak dapat dikunjungi lagi dan berdampak pada ukuran pooled feature map yang ukurannya setengah lebih kecil dari feature map (misal 64x64 menjadi 32x32). Saat matriks 2x2 itu bergeser, akan mengambil nilai terbesar dari 4 piksel yang dikunjungi untuk menjadi pixel baru di pooled feature map. Meski pooling mengurangi ukuran map, tapi tetap mempertahankan pola dan membantu agar lebih efisien terhadap variasi seperti pergeseran, rotasi, dan skala gambar.

Sebelum masuk ke Fully-Connected Layer, akan ada proses flattening dimana data dalam bentuk 3 dimensi karena ada banyak feature map akan diubah menjadi 1 dimensi (neuron). Pada Fully-Connected Layer, setiap neuron terhubung dengan semua neuron di lapisan berikutnya membentuk kombinasi linear. Lalu ada softmax yang menghasilkan distribusi probabilitas, memastikan bahwa hasil keluaran berupa nilai probabilitas yang jika dijumlahkan menjadi 1.

Berikut adalah arsitektur original model VGG16 yang digunakan pada dataset ImageNet sehingga nantinya akan kami modifikasi sedikit agar sesuai dengan dataset yang kami gunakan:





Model VGG16 terdiri dari beberapa bagian:

1. Input layer / gambar
menangkap input (224,224,3) yang berarti gambar berukuran 224x224 dengan 3 RGB channel. Gambar inilah yang akan diproses dan di filter.
2. Convolutional layers 1-1 dan 1-2
Terdapat 2 kali proses convolution dengan 64 layer tiap masing-masing proses dengan filter ukuran 3x3 dan stride 1 untuk memproses gambar 224x224 sehingga menghasilkan 64 feature map. Menggunakan padding=same agar ukuran feature map tidak mengecil serta ReLU untuk mengubah nilai negatif menjadi 0 tiap setelah terjadi proses convolution sehingga ReLU terjadi 2 kali
3. Max-pooling 1
Berukuran 2x2 dengan stride 2 sehingga ukuran pooled feature map menjadi 112x112 sehingga gambar tereduksi dan diambil fitur paling penting/berpengaruh dari tiap area
4. Convolutional layers 2-1 dan 2-2
Keduanya menggunakan 128 filter dengan ukuran 3x3, stride 1, padding same dan ReLU untuk memproses pooled feature map berukuran 112x112 sehingga menghasilkan 128 feature map tiap convolution.

5. Max-pooling 2
Berukuran 2x2 dengan stride 2 untuk mengubah ukuran feature map menjadi 56x56
6. Convolutional layers 3-1, 3-2, dan 3-3
Ketiganya menggunakan 256 filter dengan ukuran 3x3, stride 1, padding same dan ReLU untuk memproses pooled feature map berukuran 56x56 sehingga menghasilkan 256 feature map tiap convolution.
7. Max-pooling 3
Berukuran 2x2 dengan stride 2 untuk mengubah ukuran feature map menjadi 28x28
8. Convolutional layers 4-1, 4-2, dan 4-3
Ketiganya menggunakan 512 filter dengan ukuran 3x3, stride 1, padding same dan ReLU untuk memproses pooled feature map berukuran 28 x 28 sehingga menghasilkan 512 feature map tiap convolution.
9. Max-pooling 4
Berukuran 2x2 dengan stride 2 untuk mengubah ukuran feature map menjadi 14x14
10. Convolutional layers 5-1, 5-2, dan 5-3
Ketiganya menggunakan 512 filter dengan ukuran 3x3, stride 1, padding same dan ReLU untuk memproses pooled feature map berukuran 14x14 sehingga menghasilkan 512 feature map tiap convolution.
11. Max-pooling 5
Berukuran 2x2 dengan stride 2 untuk mengubah ukuran feature map menjadi 7x7

12. Flattening

Hasil dari pooling ke 5 yang berukuran $7 \times 7 \times 512$ akan diflatten menjadi 1×25088 sebelum masuk ke fully connected

13. 3 Fully Connected Layer / Dense

Pada arsitektur VGG16, terdapat 3 lapisan fully connected (dense) yang berfungsi menggabungkan dan menyaring fitur-fitur yang diekstraksi dari layer konvolusi sebelumnya. 2 lapisan pertama mengandung 4096 neuron. Setiap neuron pada layer ini melakukan operasi linear dengan mengalikan input hasil max pooling dan flattening sebelumnya dengan neuron weight dan menambahkan bias serta melakukan ReLU untuk menambahkan non-linearitas. Sementara di lapisan terakhir ada 1000 neuron yang sesuai dengan image pada dataset yang mengonversi semua informasi yang telah diproses oleh jaringan ke dalam bentuk vektor dengan panjang 1000 dan tidak ada ReLU. Tujuan fully connected layer ini agar model dapat mengenali pola dan atribut pada level yang lebih tinggi untuk tujuan klasifikasi.

14. Softmax

Berfungsi untuk menghasilkan probabilitas bagi setiap kelas output berdasarkan nilai yang dihasilkan oleh fully connected layer terakhir dengan cara mengonversi vector 1×1000 dari fully connected terakhir menjadi distribusi probabilitas sehingga saat semua elemen vector dijumlah hasilnya 1. Setiap elemen vektor menyatakan probabilitas untuk masing-masing kelas dalam dataset lalu kelas dengan probabilitas tertinggi akan dipilih menjadi kelas yang diprediksi.

Dataset yang kami gunakan berasal dari kaggle yaitu Waste Classification Data (<https://www.kaggle.com/datasets/techsash/waste-classification-data>). Data berupa gambar objek organik dan recyclable yang terbagi menjadi 85% untuk training dan 15% untuk testing sehingga ada 22564 gambar training dan 2513 gambar testing. Masing-masing data training dan testing

terbagi menjadi 2 folder yaitu O (organic) dan R (recyclable) dimana pada training ada sekitar 12.6k gambar organik dan 9999 gambar recyclable. Lalu pada data testing ada 1401 gambar organik dan 1112 gambar recyclable. Gambar-gambar ini baik pada training maupun testing akan di preprocessing menggunakan function preprocess_input dari library tensorflow vgg16 yang meliputi konversi format warna dari RGB ke BGR serta akan zero-center tiap color channel mengikuti ImageNet dataset dengan 103.939 untuk biru, 116.779 untuk hijau, dan 123.68 untuk merah. Selain itu untuk meningkatkan kemampuan generalisasi model, dilakukan augmentasi data secara khusus pada data training dengan menerapkan rotasi acak hingga 35 derajat, pergeseran horizontal dan vertikal masing-masing sebesar 0.2, zoom range hingga 0.2, flipping horizontal, shear range sebesar 0.2, serta pengisian piksel kosong dengan metode nearest neighbor. Sebelum masuk ke model, gambar akan diubah ukurannya menjadi 224x224 piksel sesuai input model, mengatur batch size = 32 yang menunjukkan jumlah gambar yang akan diproses dalam 1 batch pelatihan, dan juga class mode 'binary' menunjukkan bahwa data memiliki dua kelas / biner sehingga label yang dihasilkan berupa nilai 0 atau 1.

```
import tensorflow as tf
from tensorflow.keras.applications import VGG16

vgg_model = VGG16(weights=None, include_top=False, input_shape=(224, 224, 3))

for layer in vgg_model.layers:
    layer.trainable = True

x = tf.keras.layers.Flatten()(vgg_model.output)
x = tf.keras.layers.Dense(1024, activation='relu')(x)
x = tf.keras.layers.Dropout(0.25)(x)
x = tf.keras.layers.Dense(1024, activation='relu')(x)
x = tf.keras.layers.Dropout(0.25)(x)
outputs = tf.keras.layers.Dense(1, activation='sigmoid')(x)

model = tf.keras.Model(inputs=vgg_model.input, outputs=outputs)
```

Untuk model VGG16 yang kami gunakan dibangun tanpa memuat bobot pretrained (weights=None), tanpa bagian fully connected (FC) layer di atasnya (include_top=False) sehingga hanya menggunakan bagian convolutional dari arsitektur tersebut, dan

menerima input berupa gambar berukuran 224x224 piksel dengan 3 channel warna (RGB). Seluruh lapisan VGG16 dapat dilatih kembali (trainable=True) agar proses pelatihan dapat menyesuaikan bobot dengan data baru yang digunakan. Setelah bagian konvolusional, model dilanjutkan dengan lapisan Flatten untuk mengubah output 3D menjadi vektor 1D. Kemudian, ditambahkan dua buah lapisan Dense berturut-turut dengan jumlah neuron sebanyak 1024 dan fungsi aktivasi ReLU, masing-masing diikuti oleh Dropout sebesar 0.25 guna mengurangi risiko

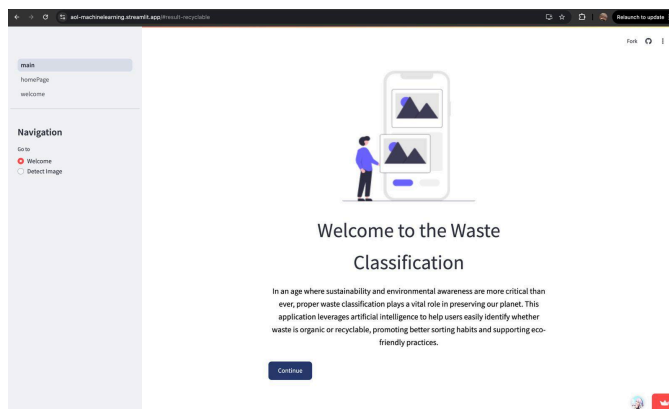
overfitting. Lapisan terakhir adalah Dense dengan aktivasi sigmoid yang cocok digunakan untuk klasifikasi biner karena menghasilkan output berupa probabilitas antara 0 dan 1.

Model dikompilasi menggunakan loss `binary_crossentropy` dan optimizer Adam dengan learning rate $1e-4$. Untuk meningkatkan performa dan stabilitas pelatihan, digunakan dua buah callback yaitu `EarlyStopping` untuk menghentikan pelatihan jika validasi loss tidak membaik selama 3 epoch dan callback `SaveBestModelByAccDifference` yang kami buat sendiri untuk menyimpan model terbaik hanya jika akurasi validasi meningkat dan selisih akurasi pelatihan dan validasi tidak melebihi 0.05 sebagai bentuk kontrol terhadap overfitting. Model dilatih selama maksimal 10 epoch dengan batch size 32, lalu model yang telah dilatih akan disimpan dalam format TensorFlow Lite (.tflite) untuk nantinya digunakan diaplikasi. Secara keseluruhan, pendekatan ini menggabungkan kekuatan arsitektur original VGG16 yang lalu disesuaikan agar sesuai dengan dataset saat ini untuk menghasilkan akurasi lebih baik.

BAB 3

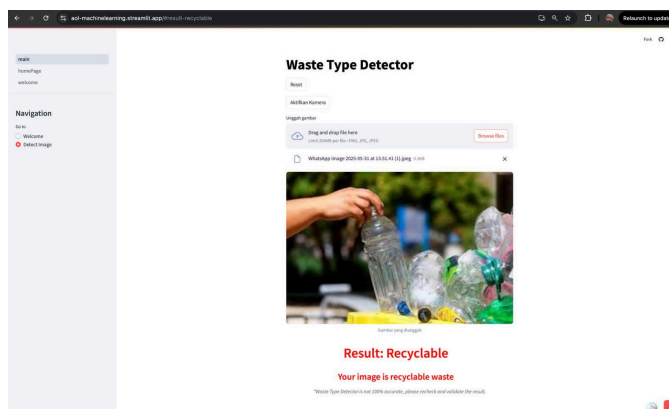
IMPLEMENTASI

Dalam pengembangan aplikasi kami tepatnya bagian user interface, kami menggunakan Streamlit sebagai framework karena kemudahan dalam load model dan juga dalam membuat aplikasi web dalam waktu yang tergolong singkat sehingga lebih efisien dalam pengerjaan project. Aplikasi yang kami buat ini terdiri dari dua halaman utama yaitu Welcome Page dan Detect Image Page yang dapat diakses melalui navigation sidebar.



Pada Welcome Page, akan menampilkan gambar dan teks sebagai pengantar serta mendeskripsikan aplikasi kami. Pada bagian paling bawah, ada tombol “Continue” yang berfungsi untuk mengarahkan user ke Detect Image Page dengan memanfaatkan fitur session state dari streamlit agar dapat mengelola

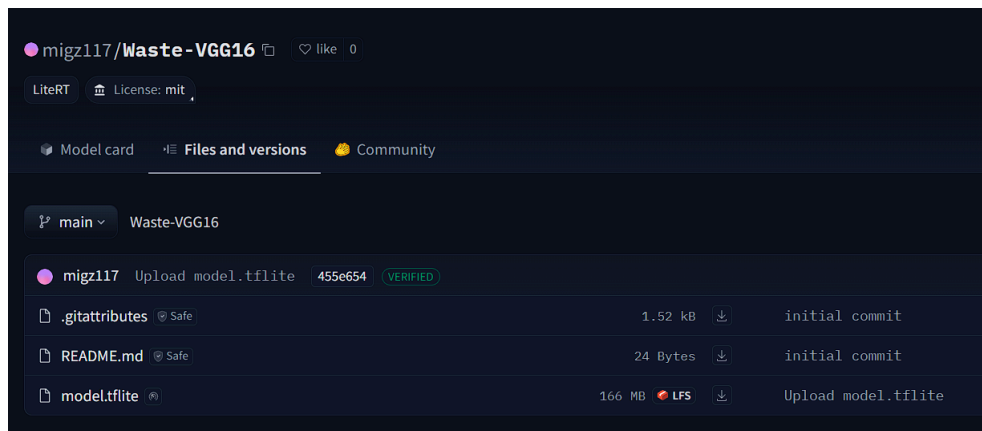
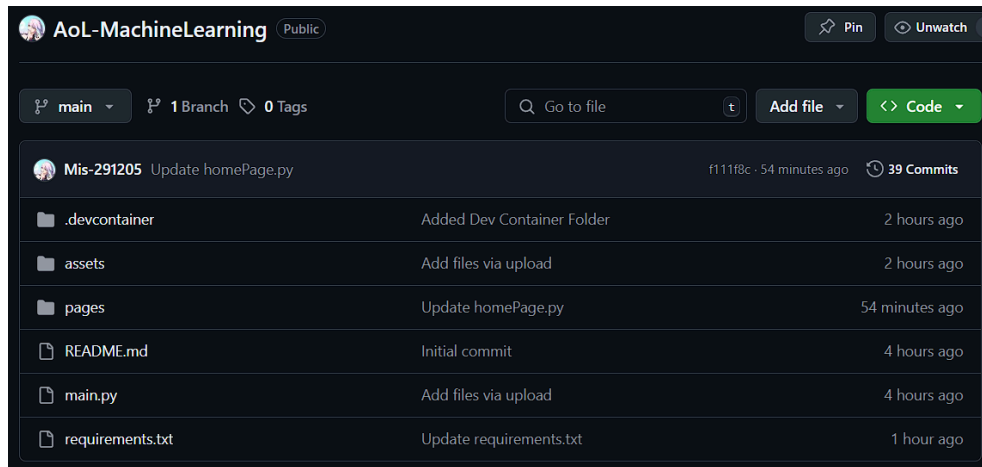
transisi antar halaman secara efisien. Teks dan button pada page ini di styling dengan menggunakan HTML dan CSS yang dapat langsung dipakai secara inline pada file .py streamlit.



Selanjutnya ada Detect Image Page yang menjadi inti dari aplikasi kami dimana user akan dapat memilih ingin mengupload gambar atau menggunakan kamera untuk mengambil foto. Hal ini dapat dilakukan dengan memanfaatkan widget seperti file_uploader dan camera_input sehingga gambar sampah

dari user akan lanjut untuk diklasifikasikan termasuk organik atau recyclable menggunakan model TFLite yang telah di load di streamlit sebelumnya. Setelah prediksi dilakukan, hasil

klasifikasi ditampilkan secara visual dengan teks yang dikustomisasi menggunakan HTML dan CSS inline agar lebih jelas dalam memberikan hasil ke user dimana tulisan hijau menandakan organic, sementara merah menunjukkan recyclable. Keseluruhan implementasi yang kami pakai ini menunjukkan bahwa Streamlit tidak hanya memungkinkan integrasi model AI dengan mudah, tetapi juga mendukung penyajian antarmuka yang bersih dan responsif yang sangat berguna dalam pengembangan aplikasi AI berbasis web dengan cepat dan efisien.



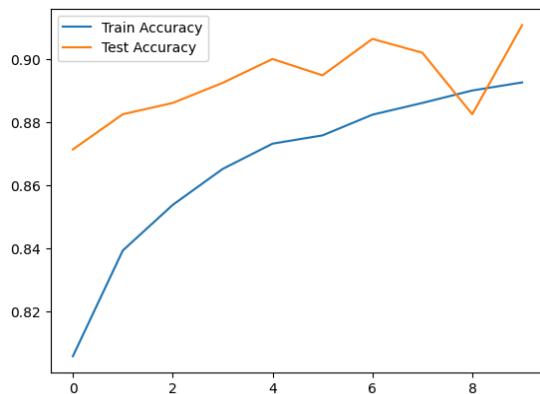
Setelah itu, kami melakukan deploy menggunakan Streamlit Community Cloud melalui GitHub dan HuggingFace (khusus modelnya) yang terlihat pada gambar di atas. Namun, model.tft berukuran terlalu besar untuk dimasukkan langsung ke GitHub sehingga harus di upload dulu di HuggingFace yang nantinya pada code python streamlit akan dimodifikasi dari yang awalnya memanggil model dari path menjadi mengimpor model dari HuggingFace. Proses upload ini dilakukan agar dapat memuat model langsung dari repositori HuggingFace yang nantinya akan membuat proses deployment menjadi lebih ringan dan efisien karena file model tidak perlu

disimpan secara lokal atau diunggah ke GitHub yang memiliki batasan ukuran file. Setelah proses deployment selesai dilakukan melalui Streamlit Community Cloud, sistem secara otomatis akan menghasilkan sebuah tautan/link publik yang dapat diakses oleh siapa saja. Link ini merupakan alamat web dari aplikasi yang telah berhasil di-deploy, sehingga orang lain tidak perlu menginstal apapun untuk menjalankan aplikasi cukup dengan membuka link tersebut di browser.

BAB 4

EVALUASI DAN HASIL

```
Epoch 1/10
706/706 ————— 565s 739ms/step - accuracy: 0.7747 - loss: 0.5016 - val_accuracy: 0.8715 - val_loss: 0.3446
Epoch 2/10
706/706 ————— 0s 586ms/step - accuracy: 0.8315 - loss: 0.3912
Epoch 2: Validation accuracy improved to 0.8826 with acceptable difference (0.0432). Model saved.
706/706 ————— 435s 616ms/step - accuracy: 0.8315 - loss: 0.3911 - val_accuracy: 0.8826 - val_loss: 0.2939
Epoch 3/10
706/706 ————— 0s 583ms/step - accuracy: 0.8575 - loss: 0.3496
Epoch 3: Validation accuracy improved to 0.8862 with acceptable difference (0.0323). Model saved.
706/706 ————— 439s 611ms/step - accuracy: 0.8575 - loss: 0.3496 - val_accuracy: 0.8862 - val_loss: 0.2739
Epoch 4/10
706/706 ————— 0s 577ms/step - accuracy: 0.8624 - loss: 0.3341
Epoch 4: Validation accuracy improved to 0.8926 with acceptable difference (0.0273). Model saved.
706/706 ————— 438s 606ms/step - accuracy: 0.8624 - loss: 0.3341 - val_accuracy: 0.8926 - val_loss: 0.2860
Epoch 5/10
706/706 ————— 0s 579ms/step - accuracy: 0.8715 - loss: 0.3146
Epoch 5: Validation accuracy improved to 0.9001 with acceptable difference (0.0268). Model saved.
706/706 ————— 448s 615ms/step - accuracy: 0.8715 - loss: 0.3146 - val_accuracy: 0.9001 - val_loss: 0.2628
Epoch 6/10
706/706 ————— 430s 598ms/step - accuracy: 0.8745 - loss: 0.3077 - val_accuracy: 0.8949 - val_loss: 0.2972
Epoch 7/10
706/706 ————— 0s 577ms/step - accuracy: 0.8844 - loss: 0.2911
Epoch 7: Validation accuracy improved to 0.9065 with acceptable difference (0.0240). Model saved.
706/706 ————— 435s 616ms/step - accuracy: 0.8844 - loss: 0.2911 - val_accuracy: 0.9065 - val_loss: 0.2463
Epoch 8/10
706/706 ————— 428s 596ms/step - accuracy: 0.8851 - loss: 0.2877 - val_accuracy: 0.9021 - val_loss: 0.2308
Epoch 9/10
706/706 ————— 443s 597ms/step - accuracy: 0.8902 - loss: 0.2827 - val_accuracy: 0.8826 - val_loss: 0.2903
Epoch 10/10
706/706 ————— 0s 580ms/step - accuracy: 0.8911 - loss: 0.2712
Epoch 10: Validation accuracy improved to 0.9109 with acceptable difference (0.0182). Model saved.
706/706 ————— 456s 617ms/step - accuracy: 0.8911 - loss: 0.2712 - val_accuracy: 0.9109 - val_loss: 0.2298
Loaded model path: best_model_with_acc_diff.keras
Best epoch: 10, Best validation accuracy: 0.9109
Train accuracy at best epoch: 0.8927
```

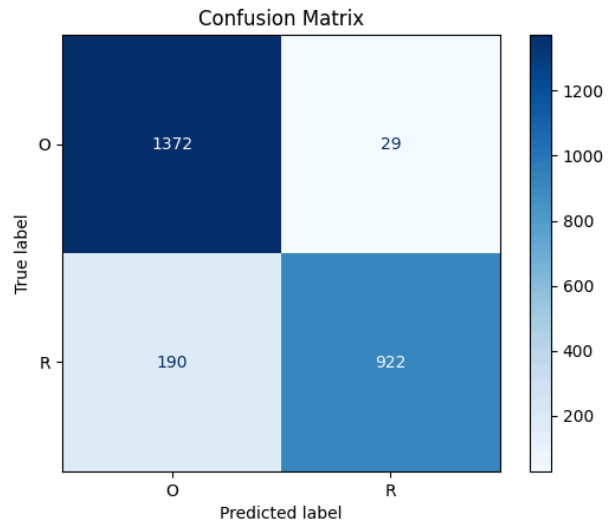


Gambar di atas menunjukkan proses training model kami dalam 10 epoch dengan hasil akhir yang disimpan sebagai model terbaik ada pada epoch ke 10 yang memiliki train accuracy sebesar 0.8927 dan validation accuracy sebesar 0.9109. Lalu gambar disebelah kiri ini adalah grafik accuracy selama proses training dengan epoch pada grafik yang dimulai dari indeks 0 sehingga harus di +1 dahulu untuk menyesuaikan dengan gambar sebelumnya.

Setelah model disimpan dalam format tflite, kami mengevaluasi model tersebut dengan menggunakan data testing yang outputnya nanti akan berupa 3 hal sebagai evaluasi yaitu classification report, confusion matrix, dan ROC AUC score beserta grafiknya.

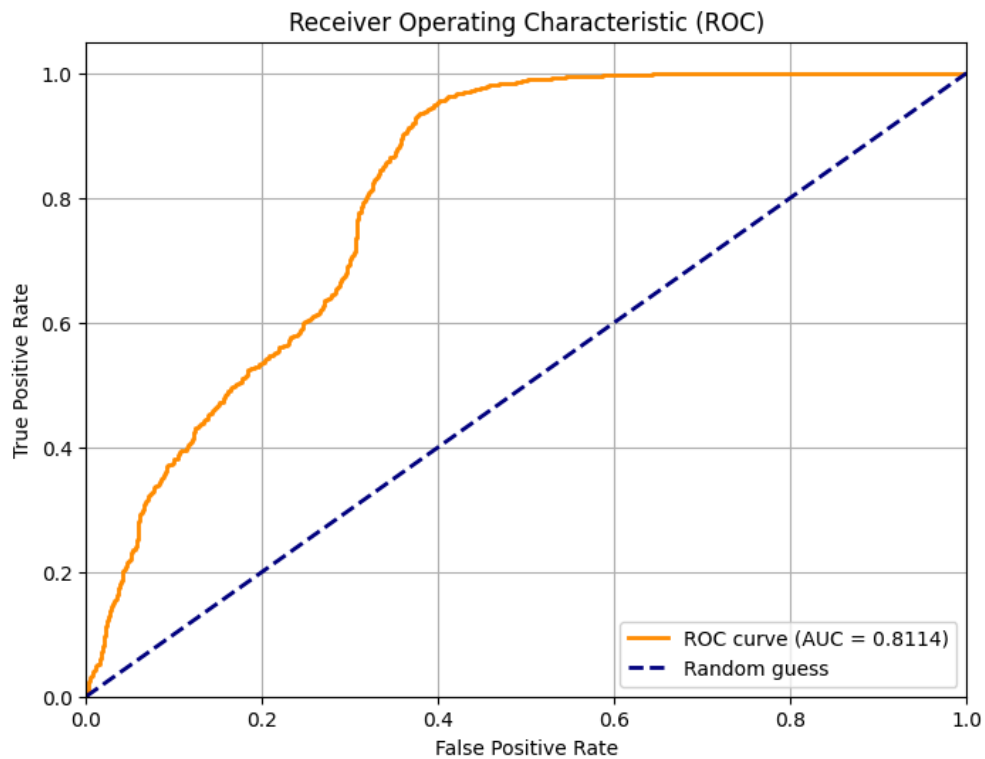
Classification Report:

	precision	recall	f1-score	support
O	0.88	0.98	0.93	1401
R	0.97	0.83	0.89	1112
accuracy			0.91	2513
macro avg	0.92	0.90	0.91	2513
weighted avg	0.92	0.91	0.91	2513



Berdasarkan classification report di atas, model kami menunjukkan performa yang sangat baik dalam mengklasifikasikan sampah menjadi dua kelas, yaitu Organic (O) dan Recyclable (R), dengan akurasi keseluruhan sebesar 0.91. Untuk kelas Organic (O), model mendapat precision sebesar 0.88, recall sebesar 0.98, dan f1-score sebesar 0.93, yang menunjukkan bahwa model sangat baik dalam mendeteksi sampah organik dimana hampir semua sampah organik diklasifikasikan dengan benar. Sementara itu pada kelas Recyclable (R), model mendapat precision sebesar 0.97, recall sebesar 0.83, dan f1-score sebesar 0.89 yang menunjukkan jika sebagian prediksi recyclable benar meski masih ada sejumlah sampah recyclable yang diklasifikasikan sebagai organik dimana hal ini dapat dilihat dari score recall yang hanya 0.83.

Hal ini diperkuat juga dari data Confusion Matrix yang menunjukkan dari total 1401 data organik, 1372 diklasifikasikan dengan benar dan hanya 29 salah diklasifikasikan sebagai recyclable. Sedangkan dari 1112 data recyclable, 922 diklasifikasikan dengan benar, sedangkan 190 salah diklasifikasikan sebagai organik. Dari hal ini dapat disimpulkan jika model sangat baik dalam mengenali sampah organik, namun masih terdapat kendala/tantangan dalam mendeteksi recyclable secara akurat khususnya dalam menurunkan jumlah false positive dimana sampah recyclable justru diklasifikasi sebagai organik.



Grafik ROC (Receiver Operating Characteristic) pada gambar diatas menunjukkan performa model dalam membedakan antara dua kelas, yaitu organik (O) dan recyclable (R) berdasarkan prediction probability. Bentuk ROC curve dari kiri ke kanan selalu berada di atas garis random guess yang merupakan area ideal (True Positive tinggi dan False Positive rendah) dan nilai AUC (Area Under Curve) sebesar 0.8114 menunjukkan bahwa model memiliki kemampuan diskriminatif yang cukup baik. AUC yang bernilai semakin mendekati 1 berarti prediksi makin baik, nilai 0.5 menunjukkan performa seperti random guess, dan dibawah 0.5 menunjukkan performa lebih buruk dari random guess. Oleh karena itu, nilai 0.81 mengindikasikan bahwa model dapat membedakan antara kedua kelas dengan akurasi tinggi. ROC AUC ini sejalan dengan hasil classification report dan confusion matrix sebelumnya sehingga menunjukkan bahwa model tidak hanya akurat secara klasifikasi akhir, tapi juga memberikan skor probabilitas yang baik.

Dari 3 metode evaluasi tersebut, model klasifikasi kami menunjukkan performa yang sangat baik dengan akurasi sebesar 91%, ditambah dengan precision, recall, dan f1-score yang tergolong tinggi pada kedua kelas. Lalu ada confusion matrix yang menunjukkan sebagian besar prediksi

modelnya tepat dengan hanya 219 kesalahan dari 2513 data testing. Selain itu, grafik dan nilai ROC AUC sebesar 0.8114 mengindikasikan model mampu membedakan kelas dengan baik berdasarkan probabilitas. Secara keseluruhan, ketiga metode evaluasi menunjukkan bahwa model ini akurat dan seimbang.

BAB 5
PEMBAGIAN TUGAS KELOMPOK

Nama	Tugas/Jobdesk
Stevan Pohan	<ul style="list-style-type: none">- Membuat model- Debugging dan deploy app- Membuat laporan dan ppt
Michael Ivan Santoso	<ul style="list-style-type: none">- Membuat model- Debugging dan deploy app- Membuat laporan dan ppt
Joseph Vincent Liem	<ul style="list-style-type: none">- Membuat UI dengan streamlit- Membuat laporan dan ppt

Link ppt canva:

https://www.canva.com/design/DAGoho7us70/oj4rHoDx75305QPafGqjIQ/edit?utm_content=DAGoho7us70&utm_campaign=designshare&utm_medium=link2&utm_source=sharebutton

Link video demo:

<https://drive.google.com/file/d/1cr05sjPkmNL75oWuQz7AqW1r8wNIeGIH/view?usp=sharing>

Link app: <http://aol-machinelearning.streamlit.app>

Link code streamlit (GitHub): <https://github.com/Mis-291205/AoL-MachineLearning>

Link model (HuggingFace): <https://huggingface.co/migz117/Waste-VGG16/tree/main>

DAFTAR PUSTAKA

Febriadi, I. (2019). Pemanfaatan sampah organik dan anorganik untuk mendukung *Go Green concept* di sekolah. *Abdimas: Papua Journal of Community Service*, 1(1), 32–39.

<https://www.ejournal.um-sorong.ac.id/index.php/pjcs/article/view/348/pdf>

Ferronato, N., & Torretta, V. (2019). Waste mismanagement in developing countries: A review of global issues. *International Journal of Environmental Research and Public Health*, 16(6), 1060.

<https://doi.org/10.3390/ijerph16061060>

GeeksforGeeks. (2024, Maret 21). *VGG-16 | CNN model*. GeeksforGeeks. Diakses pada 3 Maret 2025, dari <https://www.geeksforgeeks.org/vgg-16-cnn-model/>

Hasibuan, M. R. R. (2023). *Manfaat daur ulang sampah organik dan anorganik untuk kesehatan lingkungan* [Preprint]. OSF Preprints. <https://doi.org/10.31219/osf.io/yb42t>

Kahfi, A. (2017). Tinjauan terhadap pengelolaan sampah. *Jurisprudentie: Jurusan Ilmu Hukum Fakultas Syariah dan Hukum*, 4(1), 12–25. <https://doi.org/10.24252/jurisprudentie.v4i1.3661>

Kihila, J., Wernsted, K., & Kaseva, M. (2021). Waste segregation and potential for recycling -A case study in Dar es Salaam City, Tanzania. *Journal of Material Cycles and Waste Management*, 23(4), 1569–1578. <https://doi.org/10.1080/27658511.2021.1935532>

MyGreatLearning. (2021, September 23). *Everything you need to know about VGG16*. Medium. Diakses pada 3 Maret 2025, dari

<https://medium.com/@mygreatlearning/everything-you-need-to-know-about-vgg16-7315defb5918>

Ruiz, V., Sánchez, Á., Vélez, J. F., & Raducanu, B. (2019). Automatic image-based waste classification. In A. Ijspeert, M. Mascaro, & J. Santos-Victor (Eds.), *From bioinspired systems*

and biomedical applications to machine learning (Vol. 11557, pp. 335–343). Springer International Publishing. https://doi.org/10.1007/978-3-030-19651-6_41