Laporan Tugas 4
Interpretasi dan Pengolahan Citra

Disusun oleh:

13519008 Ronggur Mahendra

PROGRAM STUDI TEKNIK INFORMATIKA

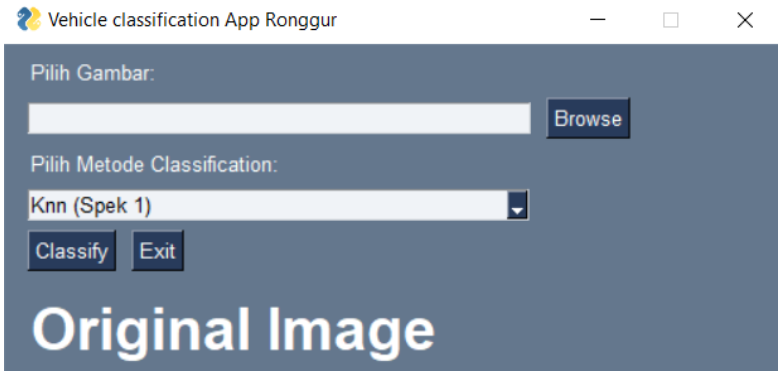
SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA

INSTITUT TEKNOLOGI BANDUNG

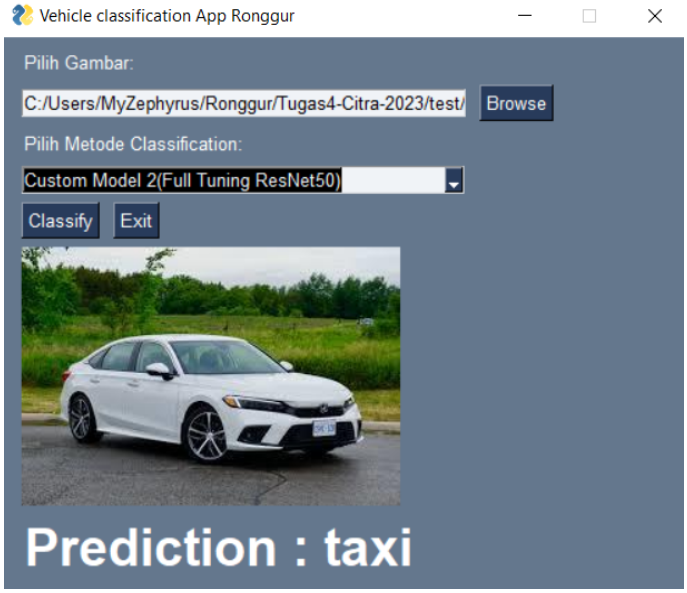
2023

1. Screenshot GUI

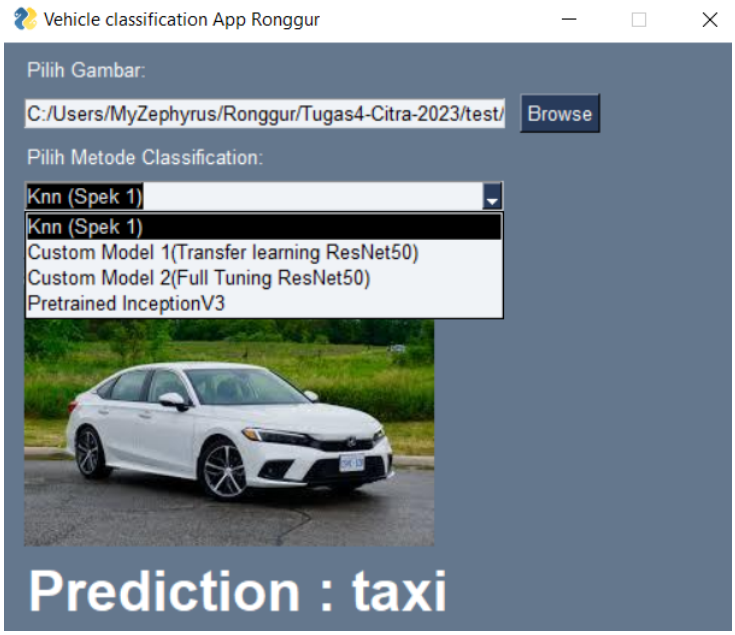
GUI Default



GUI Setelah prediksi



GUI Dropdown bar dengan semua opsi classifier



2. Program

a. Knn (Spek 1)

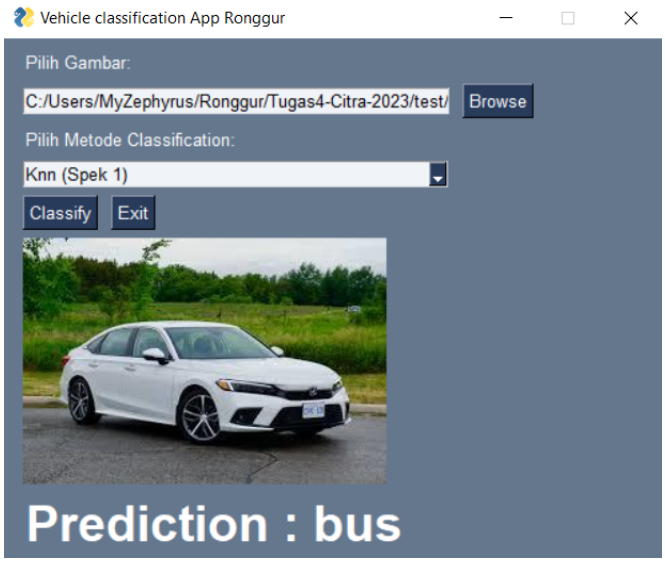
i. Kode Program

Pada Repository yang terlampir terdapat pada file Tugas4_1_KNN.py

```
import os
import numpy as np
from skimage import io
from skimage.transform import resize
from sklearn.neighbors import KNeighborsClassifier
import matplotlib.pyplot as plt

def load_images_from_folder(folder):
    images = []
    labels = []
    for filename in os.listdir(folder):
        img_path = os.path.join(folder, filename)
        img = io.imread(img_path)
        img = resize(img, (50, 50))  # resize ke 50 50
        images.append(img)  # Keep the image as a 2D array, not
flattened
        labels.append(folder.split("/")[-1])  # Use the folder name as
the label
    return np.array(images), np.array(labels)

def load_test_images_from_folder(folder):
    images = []
    for filename in os.listdir(folder):
        img_path = os.path.join(folder, filename)
        img = io.imread(img_path)
        img = resize(img, (50, 50))  # Resize the image to a common size
        images.append(img)  # Keep the image as a 2D array, not
flattened
    return np.array(images)
```

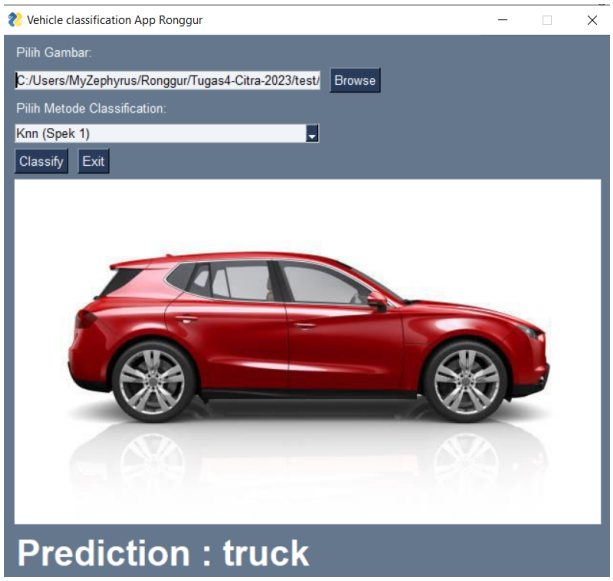
```
# Load images and labels
car_images, car_labels = load_images_from_folder('./train/car')
bus_images, bus_labels = load_images_from_folder('./train/bus')
truck_images, truck_labels = load_images_from_folder('./train/truck')

# Combine data dari smeua kelas
X = np.concatenate([car_images, bus_images, truck_images], axis=0)
y = np.concatenate([car_labels, bus_labels, truck_labels])

#  KNN classifier
knn_classifier = KNeighborsClassifier(n_neighbors=3)

# Train
knn_classifier.fit(X.reshape(X.shape[0], -1), y)  # Flatten the images
for KNN

# Load  images
test_images = load_test_images_from_folder('./test')

# predictions  test set
test_predictions =
knn_classifier.predict(test_images.reshape(test_images.shape[0], -1))

# predictions dan display  images
for i, (image, prediction) in enumerate(zip(test_images,
test_predictions)):
    # Display the image
    plt.imshow(image)
    plt.title(f"Image {i + 1}: {prediction}")
    plt.axis('off')
    plt.show()
```

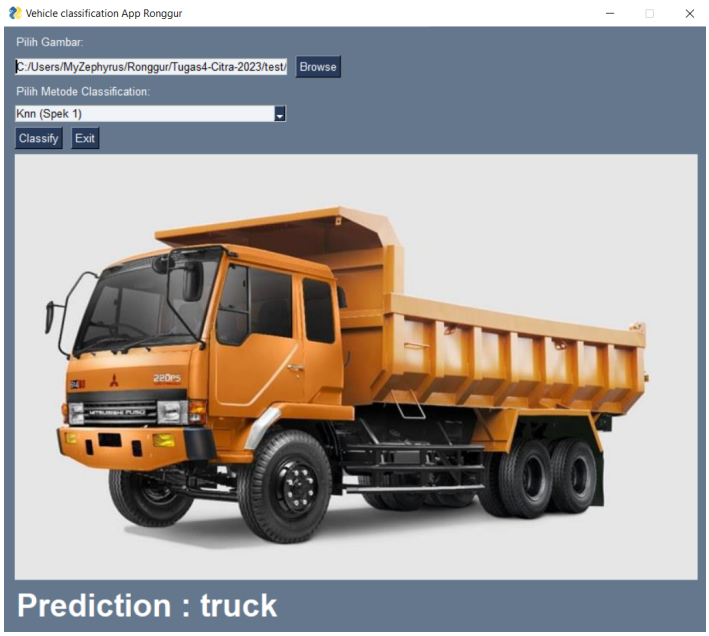
ii. Contoh hasil eksekusi

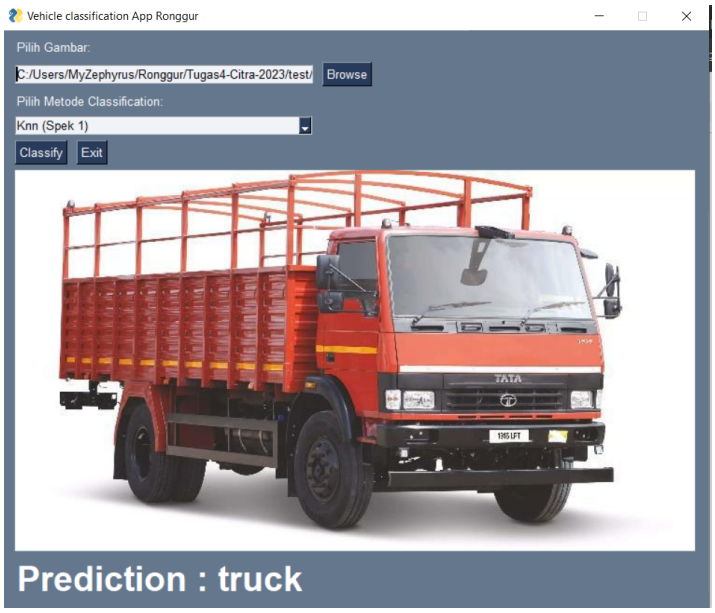

Vehicle classification App Ronggur
Pilih Gambar:
C:/Users/MyZephyrus/Ronggur/Tugas4-Citra-2023/test/
Browse
Pilih Metode Classification:
Knn (Spek 1)
Classify
Exit
Prediction : bus


Vehicle classification App Ronggur
Pilih Gambar:
C:/Users/MyZephyrus/Ronggur/Tugas4-Citra-2023/test/
Browse
Pilih Metode Classification:
Knn (Spek 1)
Classify
Exit
BLUE STAR
Prediction : bus


Vehicle classification App Ronggur
Pilih Gambar:
C:/Users/MyZephyrus/Ronggur/Tugas4-Citra-2023/test/
Browse
Pilih Metode Classification:
Knn (Spek 1)
Classify
Exit
Prediction : bus


Vehicle classification App Ronggur
Pilih Gambar:
C:/Users/MyZephyrus/Ronggur/Tugas4-Citra-2023/test/
Browse
Pilih Metode Classification:
Knn (Spek 1)
Classify
Exit
Prediction : truck





iii. Analisis

Model KNN dengan 3 label, Pada test image yang disediakan 4 dari 6 gambar test diprediksi dengan benar. Model KNN di fit kan pada data training dataset. Dan meninfer dataset baru dan memprediksi maka k neighbor terdekat dan menjadikan neighbor terdekat itu menjadi prediksinya. Salah satu keuntungan utama dari algoritma k-NN adalah bahwa itu mudah diimplementasikan

dan dipahami, serta tidak memperlukan learning. Namun k-NN memiliki masalah dimana k-NN lebih cocok untuk ruang fitur berdimensi rendah (dimana gambar bukan dimensi rendah). Juga model k-NN bisa menjadi masalah ketika ukuran data sangat besar karena memerlukan berbagi semua data.

b. Custom Model 1(Transfer learning ResNet50)

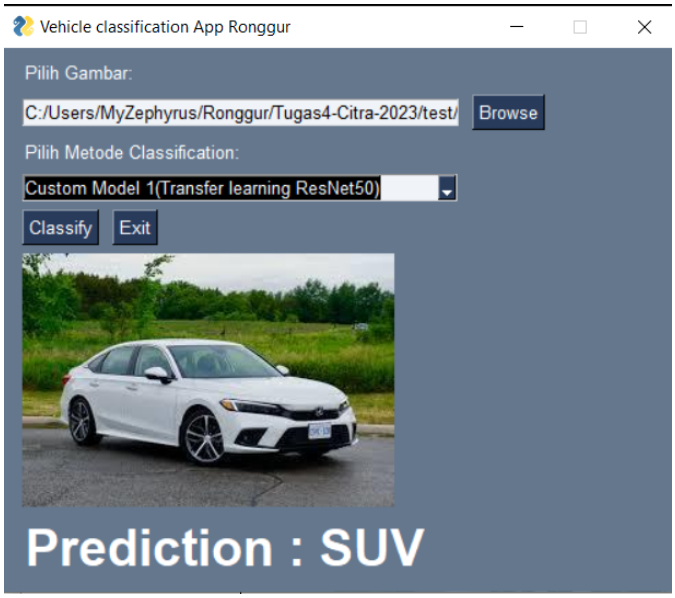
i. Kode Program

Pada Repository yang terlampir terdapat pada file Tugas4_2_Resnet.ipynb dan Tugas4_2_Resnet.pdf

Karena panjangnya kode tidak akan ditunjukan seluruh kode program namun definisi model adalah sebagai berikut

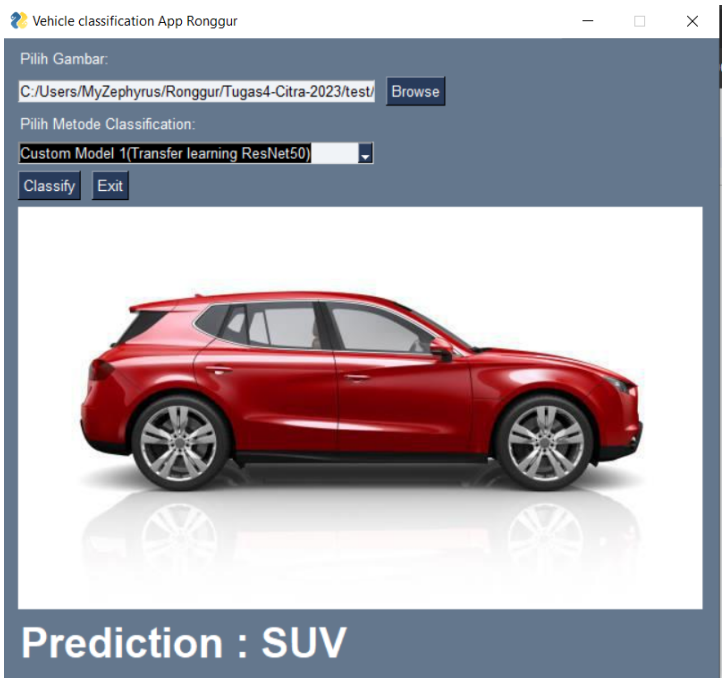
```
conv_base = ResNet50(
    include_top=False,
    weights='imagenet')
# freeze untuk transfer learning
for layer in conv_base.layers:
    layer.trainable = False

x = conv_base.output
x = layers.GlobalAveragePooling2D()(x)
x = layers.Dense(1024, activation='relu')(x)
x = layers.Dense(512, activation='relu')(x)
x = layers.Dense(128, activation='relu')(x)
predictions = layers.Dense(10, activation='softmax')(x)
model = Model(conv_base.input, predictions)
```

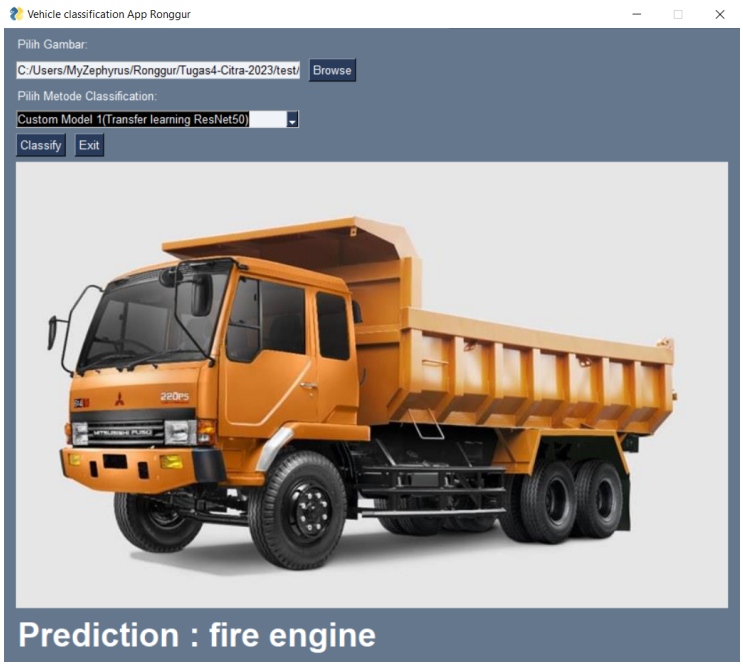
ii. Contoh hasil eksekusi

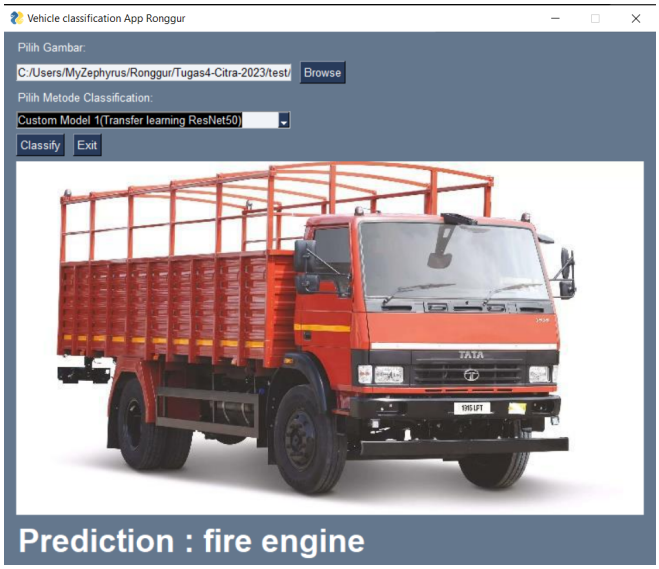

Vehicle classification App Ronggur
Pilih Gambar:
C:/Users/MyZephyrus/Ronggur/Tugas4-Citra-2023/test/
Browse
Pilih Metode Classification:
Custom Model 1(Transfer learning ResNet50)
Classify
Exit
Prediction : SUV


Vehicle classification App Ronggur
Pilih Gambar:
C:/Users/MyZephyrus/Ronggur/Tugas4-Citra-2023/test/
Browse
Pilih Metode Classification:
Custom Model 1(Transfer learning ResNet50)
Classify
Exit
BLUE STAR
BLUE STAR
Prediction : racing car


Vehicle classification App Ronggur
Pilih Gambar:
C:/Users/MyZephyrus/Ronggur/Tugas4-Citra-2023/test/
Browse
Pilih Metode Classification:
Custom Model 1(Transfer learning ResNet50)
Classify
Exit
G20
Prediction : fire engine


Vehicle classification App Ronggur
Pilih Gambar:
C:/Users/MyZephyrus/Ronggur/Tugas4-Citra-2023/test/
Browse
Pilih Metode Classification:
Custom Model 1(Transfer learning ResNet50)
Classify
Exit
Prediction : SUV

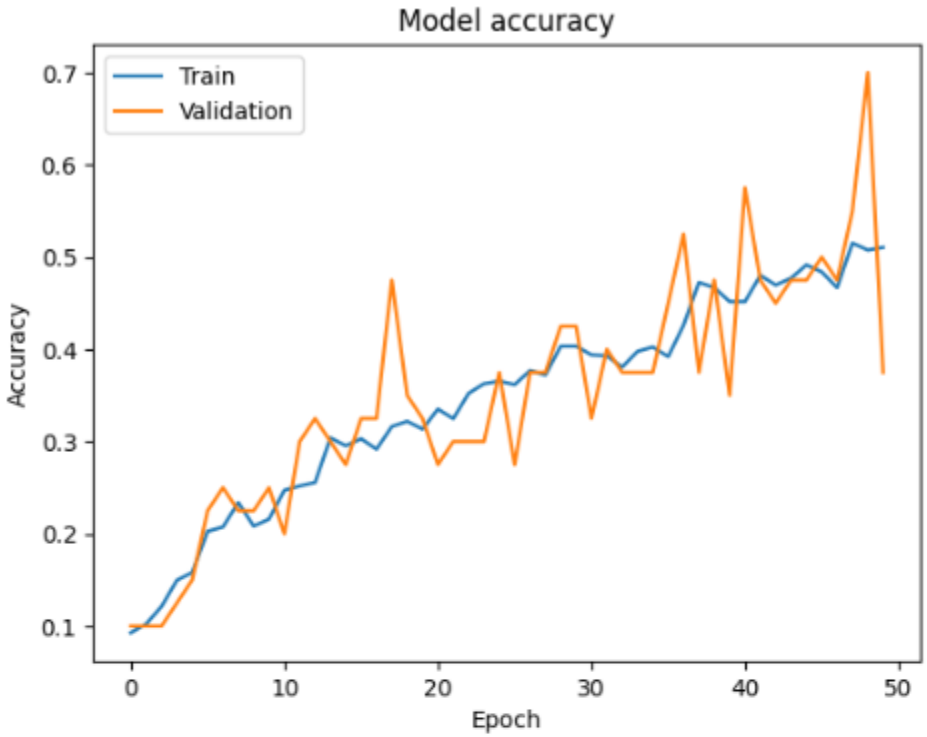


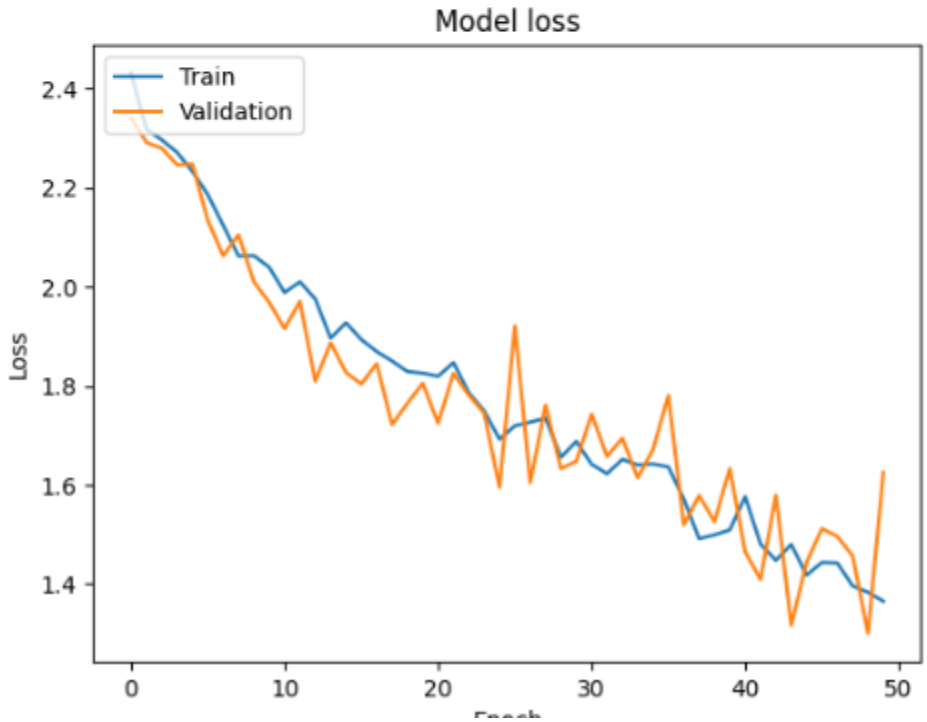


iii. Analisis

Custom Model 1 dengan 10 label, Pada test image yang disediakan 2 dari 6 gambar test diprediksi dengan benar. Custom model 1 (Transfer learning ResNet50) pertama mendownload basemodel ResNet50 lalu menfreeze weight - weight dari basemodel50 tersebut. Setelah itu menambahkan beberapa layer FCNN untuk tugas klasifikasi.

Model kemudian di train pada dataset dengan history seperti dibawah ini.





Setelah training model dievaluasi memiliki akurasi 51.04%

c. Custom Model 2(Full Tuning ResNet50)

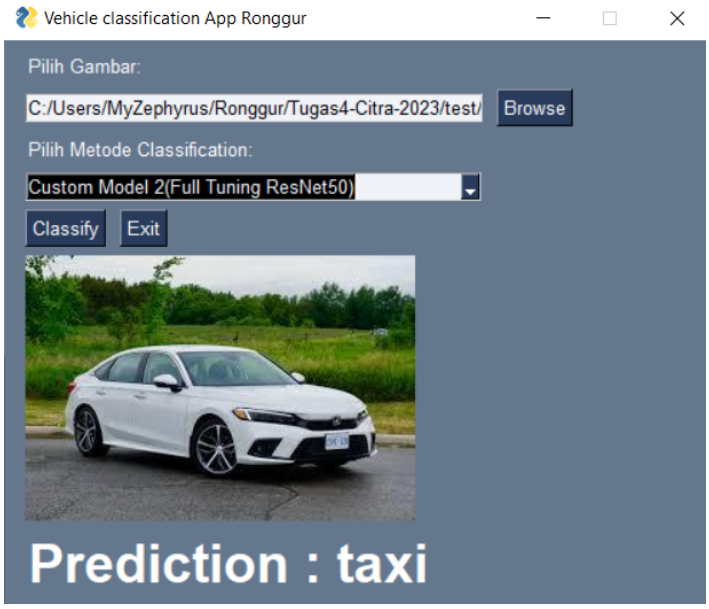
i. Kode Program

Pada Repository yang terlampir terdapat pada file Tugas4_2_Resnet.ipynb dan Tugas4_2_Resnet.pdf

Karena panjangnya kode tidak akan ditunjukan seluruh kode program namun definisi model adalah sebagai berikut

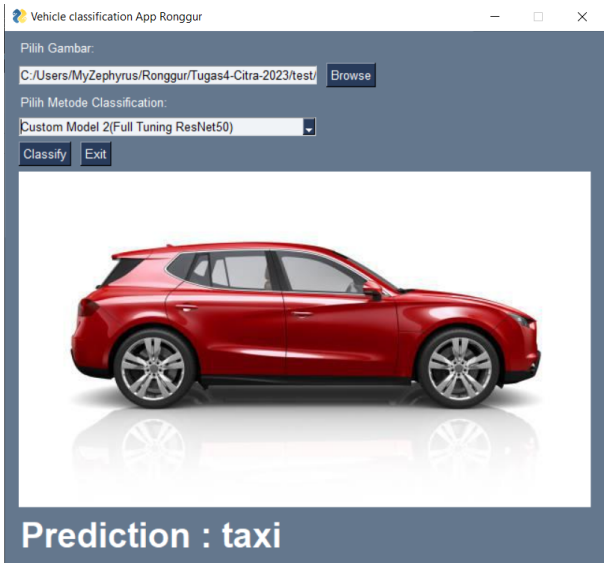
```
conv_base = ResNet50(
    include_top=False,
    weights='imagenet')

x = conv_base.output
x = layers.GlobalAveragePooling2D()(x)
x = layers.Dense(1024, activation='relu')(x)
x = layers.Dense(512, activation='relu')(x)
x = layers.Dense(128, activation='relu')(x)
predictions = layers.Dense(10, activation='softmax')(x)
model_2 = Model(conv_base.input, predictions)

optimizer = keras.optimizers.Adam()
model_2.compile(loss='categorical_crossentropy',
              optimizer=optimizer,
              metrics=['accuracy'])
```

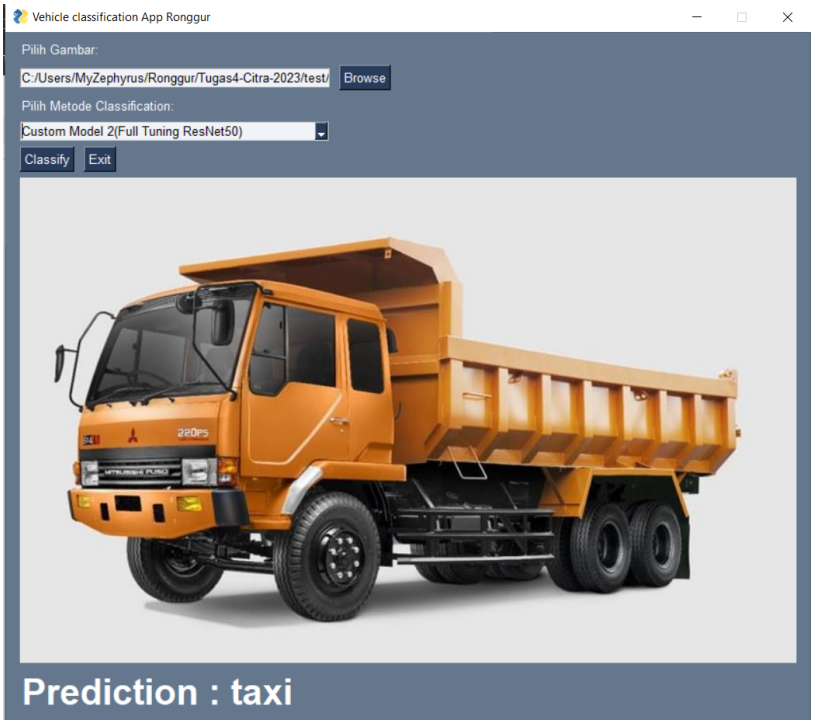
ii. Contoh hasil eksekusi

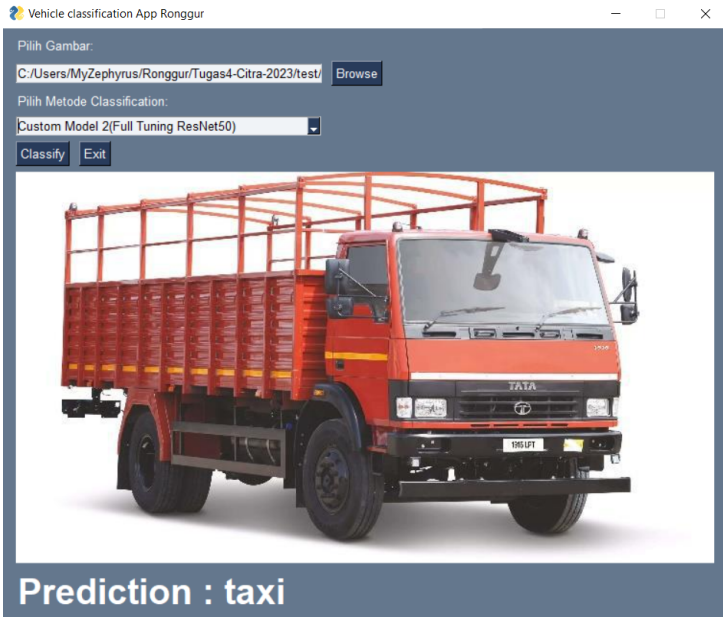

Vehicle classification App Ronggur
Pilih Gambar:
C:/Users/MyZephyrus/Ronggur/Tugas4-Citra-2023/test/
Browse
Pilih Metode Classification:
Custom Model 2(Full Tuning ResNet50)
Classify
Exit
Prediction : taxi


Vehicle classification App Ronggur
Pilih Gambar:
C:/Users/MyZephyrus/Ronggur/Tugas4-Citra-2023/test/
Browse
Pilih Metode Classification:
Custom Model 2(Full Tuning ResNet50)
Classify
Exit
BLUE STAR
BLUE STAR
Prediction : taxi


Vehicle classification App Ronggur
Pilih Gambar:
C:/Users/MyZephyrus/Ronggur/Tugas4-Citra-2023/test/
Browse
Pilih Metode Classification:
Custom Model 2(Full Tuning ResNet50)
Classify
Exit
G20
Prediction : taxi


Vehicle classification App Ronggur
Pilih Gambar:
C:/Users/MyZephyrus/Ronggur/Tugas4-Citra-2023/test/
Browse
Pilih Metode Classification:
Custom Model 2(Full Tuning ResNet50)
Classify
Exit
Prediction : taxi



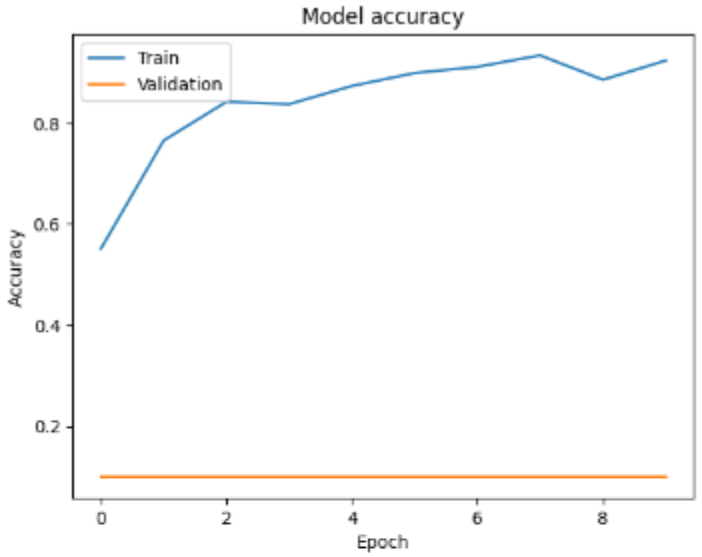


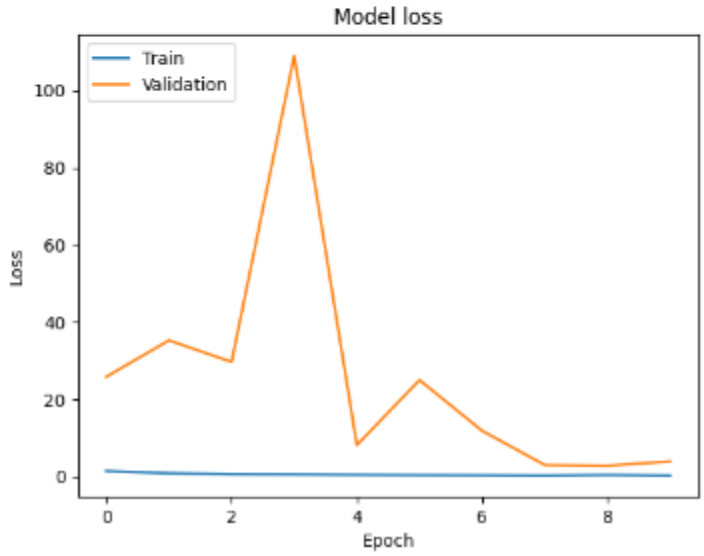
iii. Analisis

Custom Model 2 dengan 10 label, Pada test image yang disediakan 2 dari 6 gambar test diprediksi dengan benar. Custom model 2 (Full finetune ResNet50) pertama mendownload basemodel ResNet50 namun tidak seperti transfer learning tidak

menfreeze weight - weight dari basemodel50 tersebut. Setelah itu menambahkan beberapa layer FCNN untuk tugas klasifikasi.

Model kemudian di train (dengan layer - layer dari resnet50) pada dataset dengan history seperti dibawah ini.





Dari history tersebut penulis simpulkan bahwa pada training terjadi overfit.

Setelah training model dievaluasi memiliki akurasi 12.5%

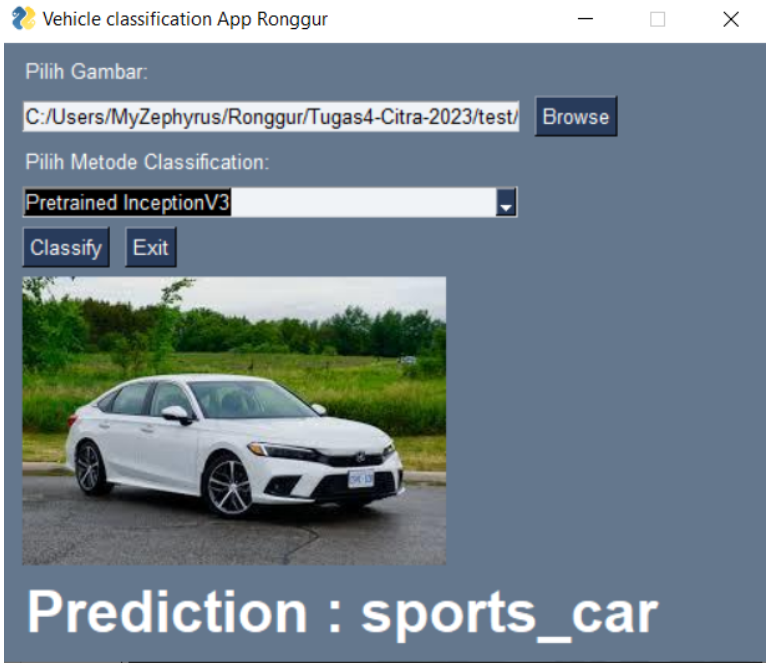
d. Pretrained InceptionV3

i. Kode Program

Pada Repository yang terlampir terdapat pada file Tugas4_2_InceptionV3.py

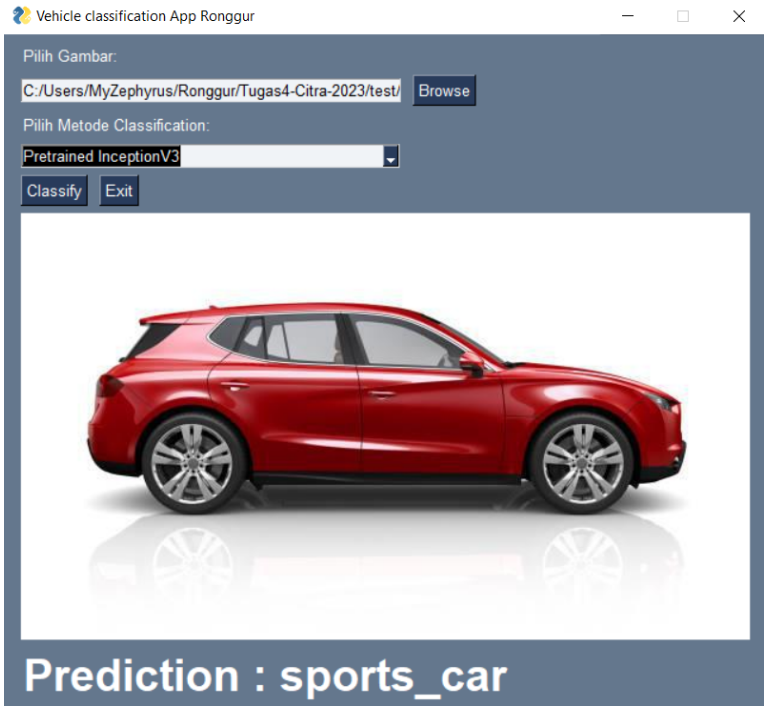
```
import tensorflow as tf
from tensorflow.keras.preprocessing import image
from tensorflow.keras.applications.inception_v3 import InceptionV3
from tensorflow.keras.applications.inception_v3 import preprocess_input,
decode_predictions
import numpy as np

# Load pre-trained InceptionV3 model
model = InceptionV3(weights='imagenet')

def classify_vehicle(image_path):
    # Load and preprocess the image
    img = image.load_img(image_path, target_size=(299, 299))
    img_array = image.img_to_array(img)
    img_array = np.expand_dims(img_array, axis=0)
    img_array = preprocess_input(img_array)

    #  predictions
    predictions = model.predict(img_array)

    # Decode and print predicted classes
    decoded_predictions = decode_predictions(predictions, top=3)[0]
    print("Predictions:")
    for i, (imagenet_id, label, score) in
enumerate(decoded_predictions):
        print(f"{i + 1}: {label} ({score:.2f})")
    return decoded_predictions

# Example usage
for i in range(6):
    image_path = f'./data/{i+1}.jpg'
    classify_vehicle(image_path)
```

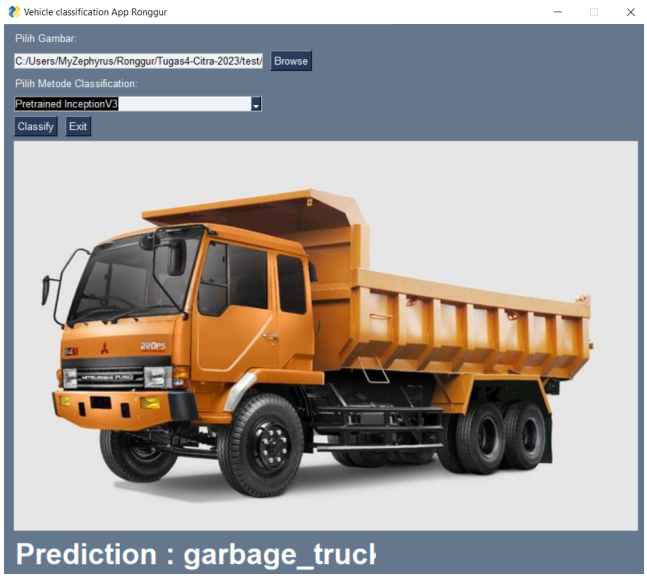
ii. Contoh hasil eksekusi

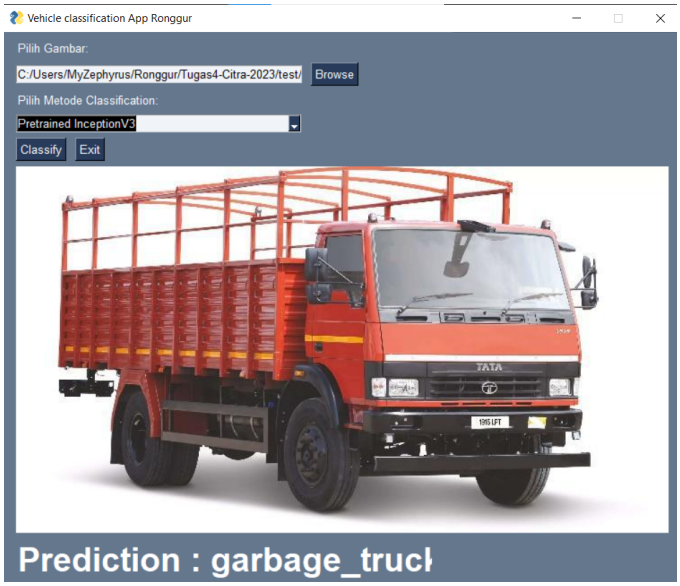



Vehicle classification App Ronggur
Pilih Gambar:
C:/Users/MyZephyrus/Ronggur/Tugas4-Citra-2023/test/
Browse
Pilih Metode Classification:
Pretrained InceptionV3
Classify
Exit
BLUE STAR
Prediction : passenger_ca


Vehicle classification App Ronggur
Pilih Gambar:
C:/Users/MyZephyrus/Ronggur/Tugas4-Citra-2023/test/
Browse
Pilih Metode Classification:
Pretrained InceptionV3
Classify
Exit
G20
Prediction : moving_van


Vehicle classification App Ronggur
Pilih Gambar:
C:/Users/MyZephyrus/Ronggur/Tugas4-Citra-2023/test/
Browse
Pilih Metode Classification:
Pretrained InceptionV3
Classify
Exit
Prediction : sports_car


Vehicle classification App Ronggur
Pilih Gambar:
C:/Users/MyZephyrus/Ronggur/Tugas4-Citra-2023/test/
Browse
Pilih Metode Classification:
Pretrained InceptionV3
Classify
Exit
Prediction : garbage_truck



iii. Analisis

Pretrained InceptionV3, Pada test image yang disediakan 2 dari 6 gambar test diprediksi dengan benar. Model InceptionV3 adalah arsitektur NN yang telah dilatih pada dataset ImageNet untuk tugas klasifikasi gambar. InceptionV3 memiliki 1000 label. Pada task ini model InceptionV3 memiiliki performa qualitatif terbaik dibandingkan model - model lainya.

e. GUI

i. Kode Program

```
import PySimpleGUI as sg
import cv2
import numpy as np
import io
from PIL import Image
import os
from skimage import io
from skimage.transform import resize
from sklearn.neighbors import KNeighborsClassifier
import matplotlib.pyplot as plt


import tensorflow as tf
```

```
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image
from tensorflow.keras.applications.inception_v3 import InceptionV3
from tensorflow.keras.applications.inception_v3 import preprocess_input,
decode_predictions

def load_images_from_folder(folder):
    images = []
    labels = []
    for filename in os.listdir(folder):
        img_path = os.path.join(folder, filename)
        img = io.imread(img_path)
        img = resize(img, (50, 50))  # resize ke 50 50
        images.append(img)
        labels.append(folder.split("/")[-1])
    return np.array(images), np.array(labels)


def main():
    # Make the KNN model
    # Load images and labels knn
    print("  Making KNN Model ...")
    car_images, car_labels = load_images_from_folder('./train/car')
    bus_images, bus_labels = load_images_from_folder('./train/bus')
    truck_images, truck_labels =
load_images_from_folder('./train/truck')

    # Combine data from different classes
    X = np.concatenate([car_images, bus_images, truck_images], axis=0)
    y = np.concatenate([car_labels, bus_labels, truck_labels])

    # Create a KNN classifier
    knn_classifier = KNeighborsClassifier(n_neighbors=3)

    # Train the model
    knn_classifier.fit(X.reshape(X.shape[0], -1), y)  # Flatten the
images for KNN

    print(" LOADING vehicle_classifier_1.h5 ...")
```

```
    # Load Custom Model 1(Transfer learning ResNet50)
    custom_model_1 = load_model('./model/vehicle_classifier_1.h5')

    print(" LOADING vehicle_classifier_2.h5 ...")
    # Load Custom Model 2(Full Tuning ResNet50)
    custom_model_2 = load_model('./model/vehicle_classifier_2.h5')


    print(" DOWNLOADING InceptionV3 ...")
    # load Pretrained model InceptionV3
    inceptionV3_model = InceptionV3(weights='imagenet')

    print(" Done Ready to use ...")

    # UI Stuff
    # Define the layout
    layout = [
        [sg.Text("Pilih Gambar:")],
        [sg.Input(key="-FILE-", enable_events=True), sg.FileBrowse()],
        [sg.Text("Pilih Metode Classification:")],
        [sg.Drop(values=('Knn (Spek 1)', 'Custom Model 1(Transfer
learning ResNet50)', 'Custom Model 2(Full Tuning ResNet50)', 'Pretrained
InceptionV3'), default_value='Knn (Spek 1)', key="-OPTION-")],
        [sg.Button("Classify"), sg.Button("Exit")],
        [sg.Image(key="-ORIGINAL-")],
        [sg.Text("Original Image", key="-LABEL-ORIGINAL-", size=(20, 1),
font=('Helvetica', 28, 'bold'))]
    ]

    # Create the window
    window = sg.Window("Vehicle classification App Ronggur", layout)

    while True:
        event, values = window.read()

        if event == sg.WINDOW_CLOSED or event == "Exit":
            break

        if event == "Classify":
```

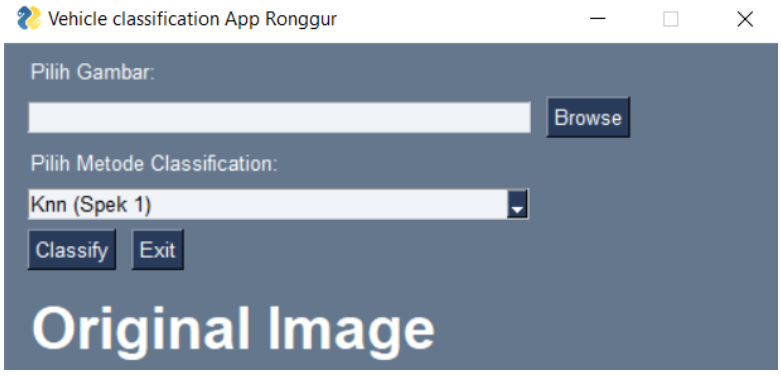
```
            image_path = values["-FILE-"]
            clasiffy_option = values["-OPTION-"]

            if image_path:
                # display the iamge
                original_image = cv2.imread(image_path)

                # Convert the images to a format that PySimpleGUI can
display
                original_imgbytes = cv2.imencode(".png",
original_image)[1].tobytes()

                # Update the images on the window
                window["-ORIGINAL-"].update(data=original_imgbytes)

                # predict based off of the option selected
                stringlabel = ''
                stringlabel += "Prediction : "

                print("clasiffy_option : ",clasiffy_option)
                # KNN Prediction
                if clasiffy_option == 'Knn (Spek 1)':

                    img = io.imread(image_path)
                    img = resize(img, (50, 50))  # Resize the image to a
common size
                    image_flattened = img.reshape(1, -1)
                    prediction = knn_classifier.predict(image_flattened)
                    print("KNN Predicts : ", prediction)
                    stringlabel += str(prediction[0])

                # custom model 1 prediction
                elif clasiffy_option == 'Custom Model 1(Transfer
learning ResNet50)':

                    img_size = (224, 224)
                    img = image.load_img(image_path,
target_size=img_size)
                    img_array = image.img_to_array(img)
```

```
                    img_array = np.expand_dims(img_array, axis=0)
                    img_array /= 255.0  # Normalize pixel values to
between 0 and 1

                    prediction = custom_model_1.predict(img_array)
                    class_index = np.argmax(prediction)

                    class_labels = ['family sedan', 'heavy truck',
'SUV', 'minibus', 'fire engine', 'bus', 'racing car', 'truck', 'jeep',
'taxi']
                    true_label_name = class_labels[class_index]
                    stringlabel += true_label_name

                    print("Custom Model 1 predicts : ", true_label_name)
                # custom model 2 prediction
                elif clasiffy_option == 'Custom Model 2(Full Tuning
ResNet50)':

                    img_size = (224, 224)
                    img = image.load_img(image_path,
target_size=img_size)
                    img_array = image.img_to_array(img)
                    img_array = np.expand_dims(img_array, axis=0)
                    img_array /= 255.0  # Normalize pixel values to
between 0 and 1

                    prediction = custom_model_2.predict(img_array)
                    class_index = np.argmax(prediction)

                    class_labels = ['family sedan', 'heavy truck',
'SUV', 'minibus', 'fire engine', 'bus', 'racing car', 'truck', 'jeep',
'taxi']
                    true_label_name = class_labels[class_index]
                    stringlabel += true_label_name

                    print("Custom Model 2 predicts : ", true_label_name)
                # InceptionV3 prediction
                elif clasiffy_option == 'Pretrained InceptionV3':
```

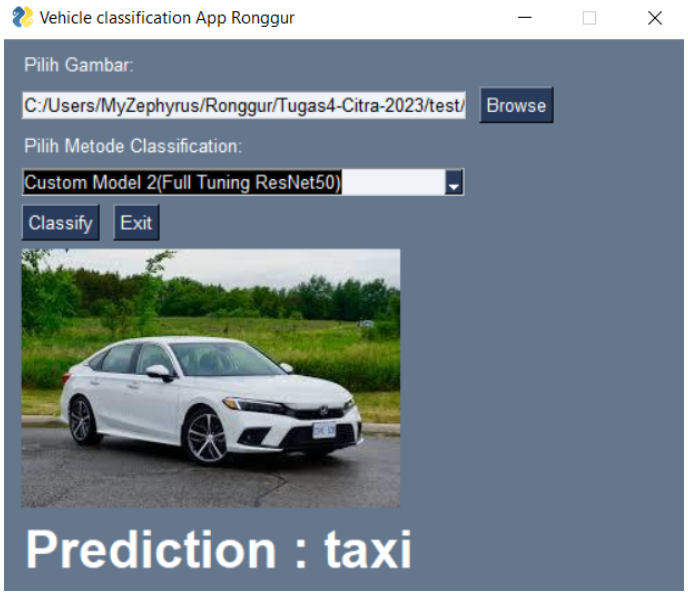
```
                    img = image.load_img(image_path, target_size=(299,
299))
                    img_array = image.img_to_array(img)
                    img_array = np.expand_dims(img_array, axis=0)
                    img_array = preprocess_input(img_array)

                    # Make predictions
                    predictions = inceptionV3_model.predict(img_array)

                    # Decode and print the top-3 predicted classes
                    decoded_predictions =
decode_predictions(predictions, top=1)[0]
                    print("Pretrained InceptionV3 Predictions:")
                    for i, (imagenet_id, label, score) in
enumerate(decoded_predictions):
                        print(f"{i + 1}: {label} ({score:.2f})")
                        stringlabel += label

                # update label
                window["-LABEL-ORIGINAL-"].update(stringlabel)
    window.close()

if __name__ == "__main__":
    main()
```

ii. Contoh hasil eksekusi

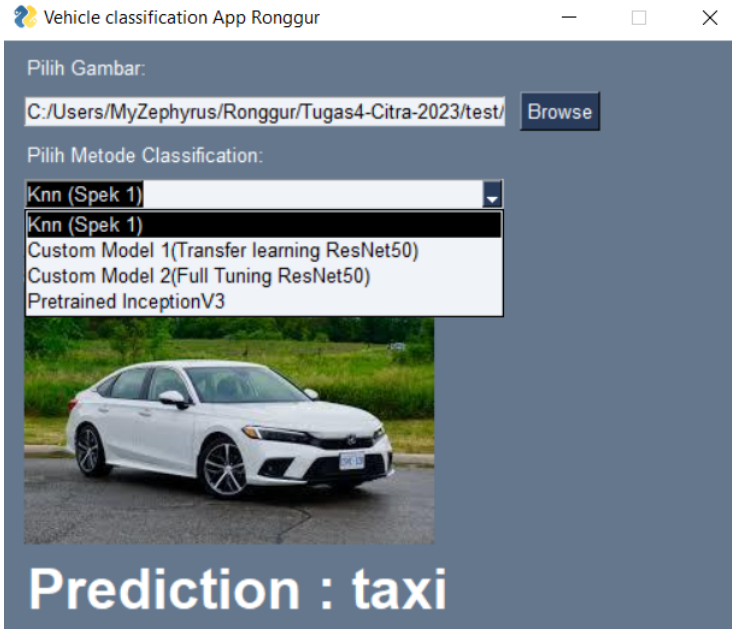
GUI Default



GUI Setelah prediksi



GUI Dropdown bar dengan semua opsi classifier



iii. Analisis

GUI diimplementasi menggunakan library PySimpleGUI dan berhasil memenuhi semua requirement orek tugas besar.

3. Kesimpulan

Berdasarkan berbagai jenis model yang digunakan disimpulkan bahwa beberapa model memiliki kelebihan masing - masing. Secara umum untuk tugas klasifikasi Pretrained InceptionV3 memiliki performa paling bagus, namun Pretrained InceptionV3 memiliki kekurangan dimana model sangat besar sehingga tidak dapat di finetune (error cuda out of memory), dan tidak fleksibel.

Custom model memiliki fleksibilitas yang Pretrained InceptionV3 tidak miliki namun untuk men finetune / transfer learning membutuhkan waktu yang lama dan computer resource yang intensif dengan performa yang bervariasi tergantung dengan hyper parameter tuning dan kualitas dataset.

KNN walaupun memiliki performa yang terbutuk sangat lah mudah untuk diimplementasi dan tidak dibutuhkan “Learning”, namun karena karateristik KNN yang kesulitan dalam data dengan dimensi tinggi (seperti Citra) kurang cocok untuk task image classification ini, dan juga model KNN pada dasarnya memiliki size yang sangat besar(karena perlu menyimpan seluruh dataset).

4. **Komentar dan refleksi anda terhadap tugas ini**

Banyak yang dipelajari pada tugas besar ini dan topik yang menarik walaupun computer resource dan waktu yang dibutuhkan lama dan banyak.

5. **Alamat Github**

https://github.com/ronggurmahendra/Tugas4-Citra-2023.git

https://colab.research.google.com/drive/1TWQ8yWiHDpgCAxo1xjlfjI_3Db8vq_uW?usp=sharing