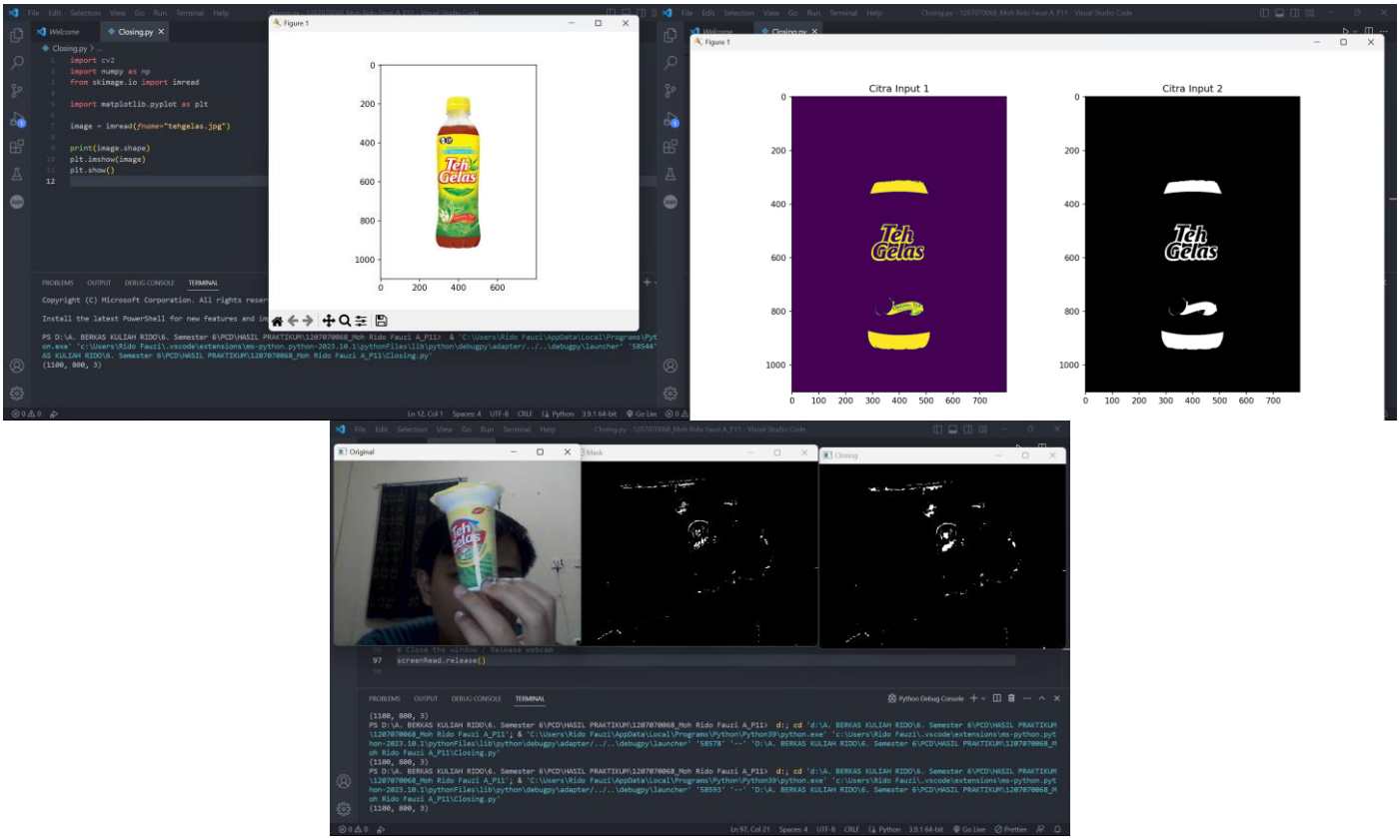
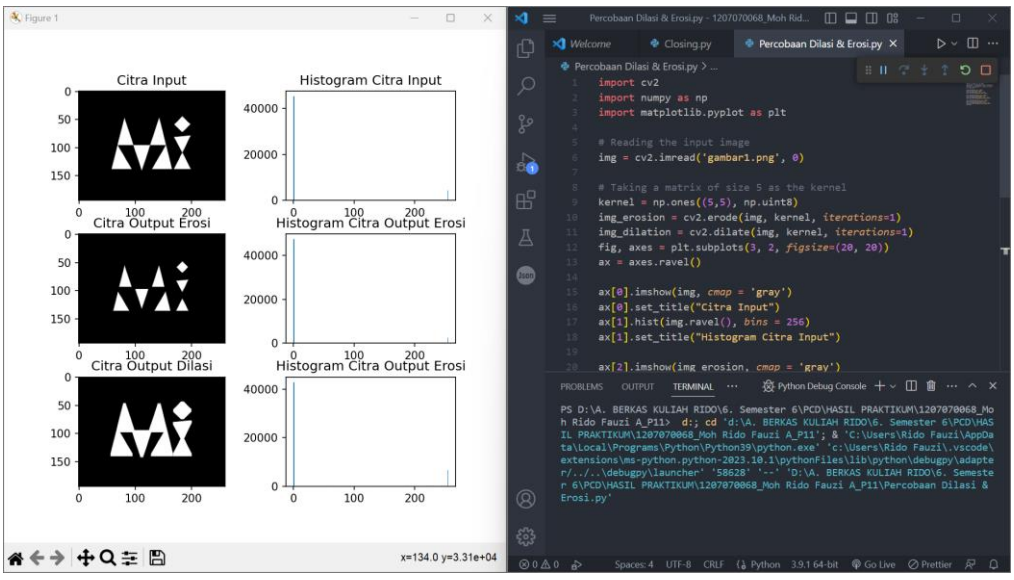
Nama : Moh Rido Fauzi Arbiansyah
NIM : 1207070068
Kelas : TKK https://github.com/ridofauzi/1207070068_MohRidoFauziA_P11.git

1. PERCOBAAN CLOSING



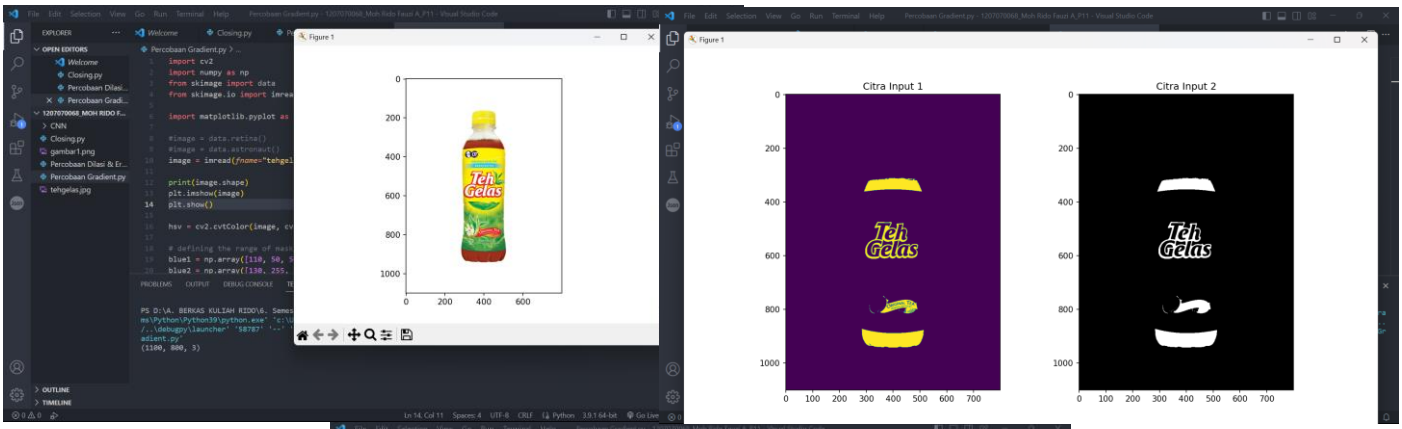
Program ini menggunakan OpenCV untuk melakukan operasi pemrosesan gambar, terutama operasi closing morfologi, dalam percobaan Closing ini. Dua array numpy, blue1 dan blue2, mendefinisikan batas bawah dan atas rentang warna biru yang diinginkan untuk dimask. Selanjutnya, dengan menggunakan cv2.morphologyEx, rentang warna ini digunakan untuk membuat masker closing biner morfologi pada gambar. Operasi closing ini menggabungkan operasi dilasi (dilation) dan erosi (erosion) untuk menutup lubang kecil dan menghaluskan kontur objek pada gambar. Program memproses video dari webcam dengan menggunakan loop while. Setelah frame webcam dibaca, ruang warna HSV diubah, setiap frame ditutup dan dimasker, dan cv2.imshow digunakan untuk menampilkan hasilnya.

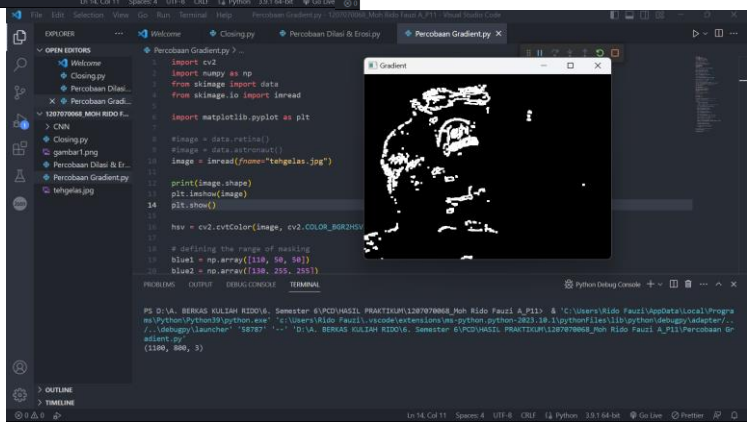
2. Percobaan Dilasi & Erosi



Dengan menggunakan cv2.erode, operasi erosi dilakukan pada gambar masukan, yang menghasilkan hasil dalam variabel img_erosion, yang menghilangkan piksel di tepi objek dan menurunkan ukurannya. Kemudian, operasi dilasi dilakukan pada gambar masukan, yang menghasilkan hasil dalam variabel img_dilation, yang menambah area objek dengan menambahkan lebih banyak piksel di sekitar tepinya. Kemudian gunakan plt.subplots dan ax[i].imshow untuk menampilkan gambar input, hasil erosi, dan dilasi dalam subplot. Subplot pertama menampilkan gambar input, subplot kedua menampilkan histogram gambar input, subplot ketiga menampilkan gambar hasil erosi, subplot keempat menampilkan histogram gambar hasil erosi, dan subplot keenam menampilkan histogram gambar hasil dilasi. Program ini memberikan visualisasi efek erosi dan perluasan pada gambar. Erosi mengecilkan objek dan menghilangkan piksel dari tepinya, sementara dilatasi memperluas objek dengan menambahkan piksel di tepinya. Histogram juga memberikan informasi tentang sebaran intensitas piksel pada citra masukan dan hasil erosi/dilasi.

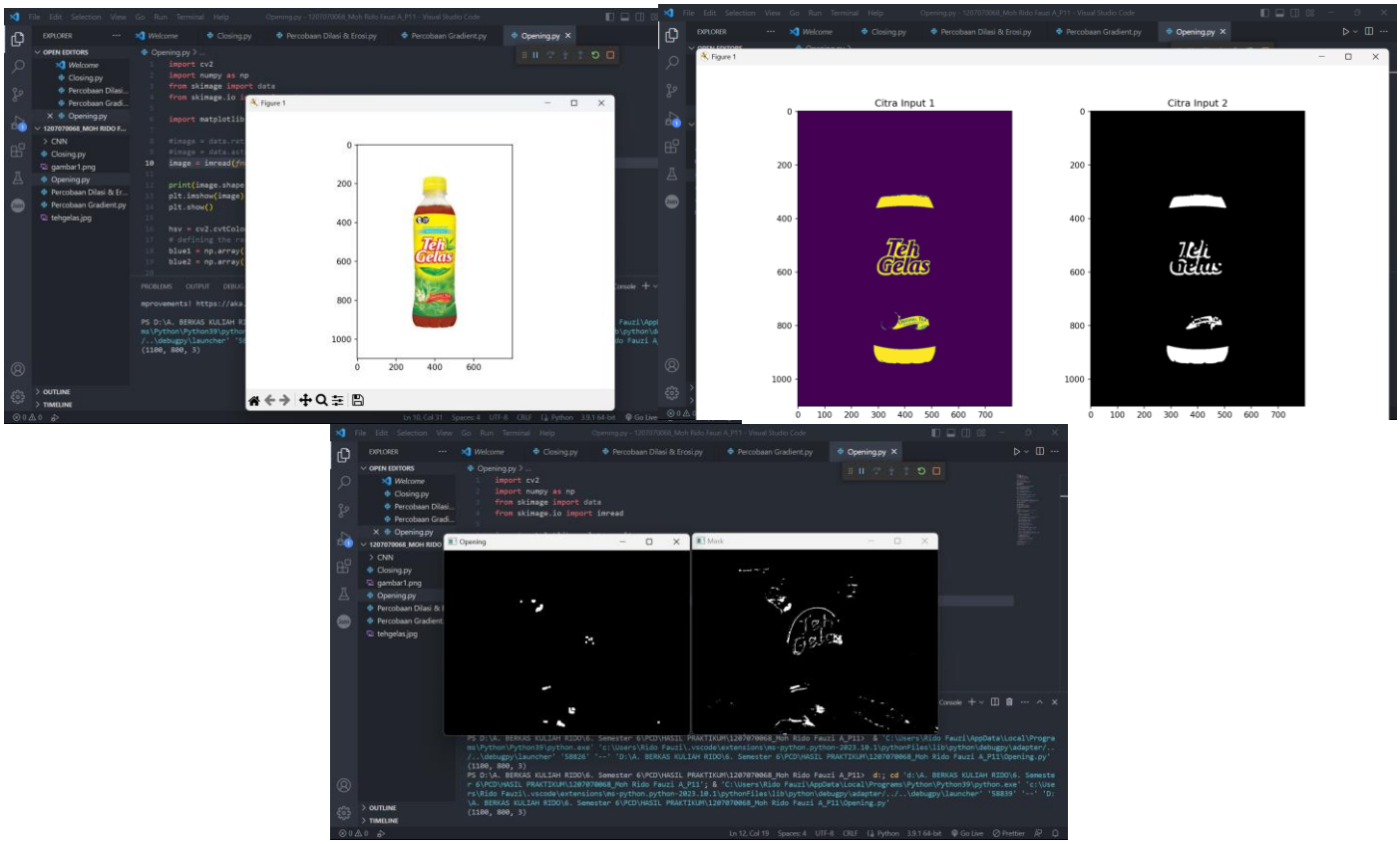
3. Percobaan Gradient





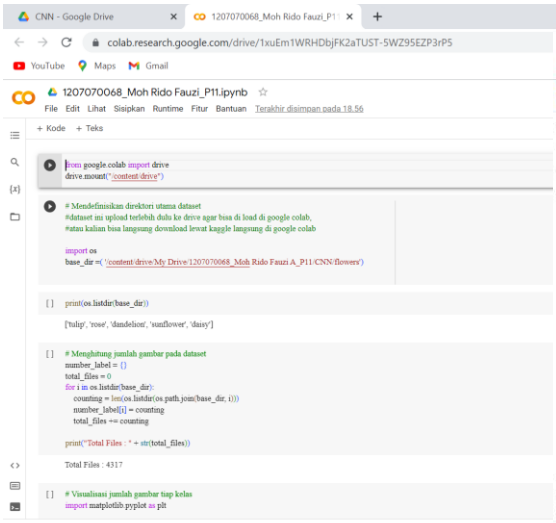
Pada percobaan ini program memberikan visualisasi dari efek gradient morfologi pada gambar dan video dari webcam. Operasi gradient morfologi menyoroti tepi objek dalam gambar dengan menampilkan perbedaan intensitas piksel pada tepi objek. Dilakukan operasi gradient morfologi pada gambar hasil segmentasi warna menggunakan cv2.morphologyEx dengan parameter cv2.MORPH_GRADIENT dan kernel yang telah dibuat sebelumnya. Operasi gradient menghasilkan gambar yang menunjukkan perbedaan intensitas piksel pada tepi objek.

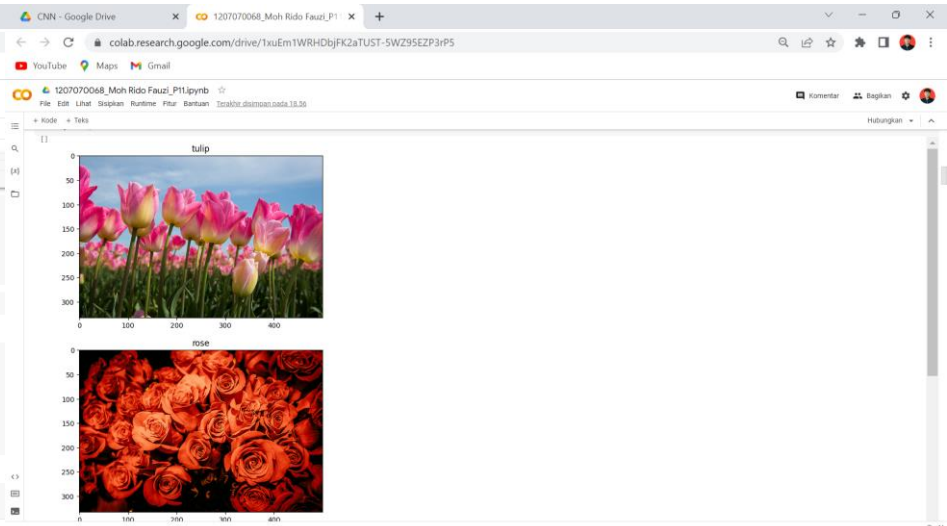
4. Percobaan Opening

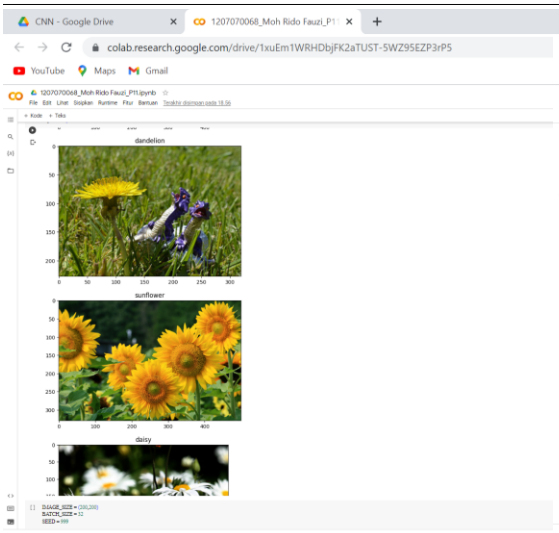


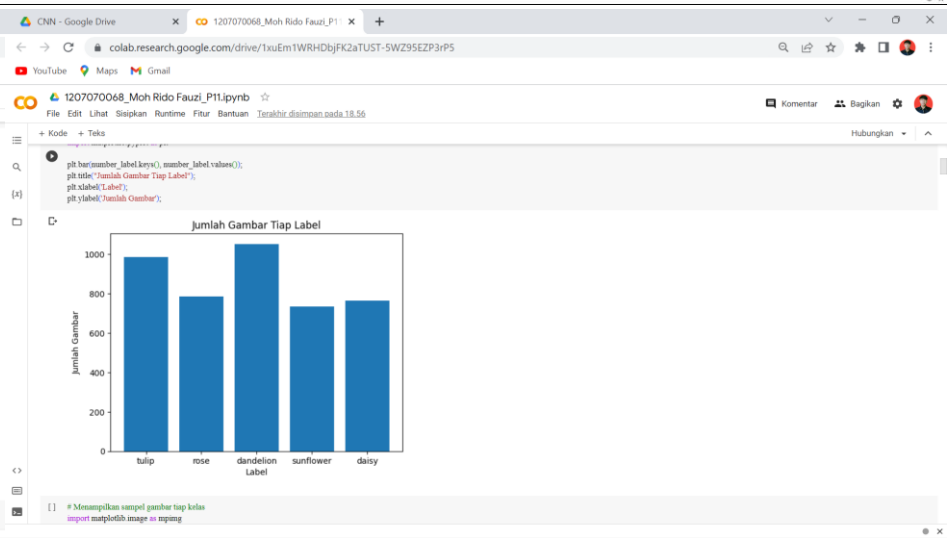
Hasil percobaan ini hampir sama seperti pada percobaan percobaan sebelumnya dimana Dilakukan operasi opening pada gambar hasil segmentasi warna menggunakan cv2.morphologyEx dengan parameter cv2.MORPH_OPEN dan kernel yang telah dibuat sebelumnya. Operasi opening bertujuan untuk menghilangkan noise kecil dan mencerahkan tepi objek. Percobaan ini memberikan visualisasi dari efek opening morfologi pada gambar dan video dari webcam. Operasi opening morfologi menghilangkan noise kecil dan mencerahkan tepi objek dalam gambar.

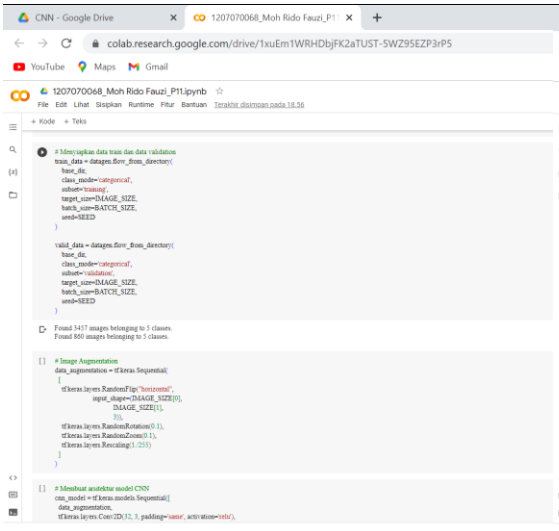
5. CNN

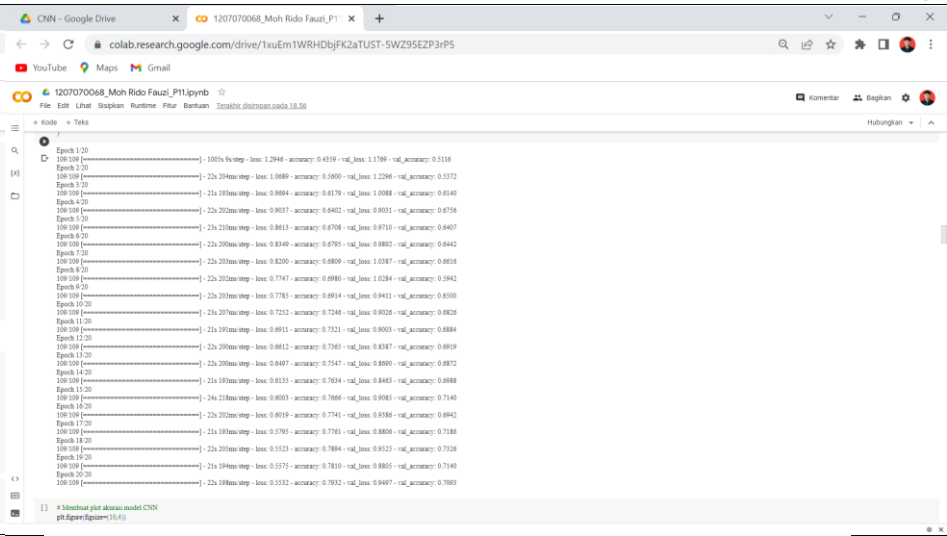

CNN - Google Drive
1207070068_Moh Rido Fauzi_P1
colab.research.google.com/drive/1xuEm1WRHDbjFK2aTUST-5WZ95EZP3rP5
YouTube
Maps
Gmail
1207070068_Moh Rido Fauzi_P11.ipynb
File Edit Lihat Sisipkan Runtime Fitur Bantuan Terakhir disimpan pada 18.56
+ Kode + Teks
from google.colab import drive
drive.mount("/content/drive")
# Mendefinisikan direktori utama dataset
#dataset ini upload terlebih dulu ke drive agar bisa di load di google colab,
#atau kalian bisa langsung download lewat kaggle langsung di google colab
import os
base_dir =('/content/drive/My Drive/1207070068_Moh Rido Fauzi A_P11/CNN/flowers')
print(os.listdir(base_dir))
['tulip', 'rose', 'dandelion', 'sunflower', 'daisy']
# Menghitung jumlah gambar pada dataset
number_label = {}
total_files = 0
for i in os.listdir(base_dir):
counting = len(os.listdir(os.path.join(base_dir, i)))
number_label[i] = counting
total_files += counting
print("Total Files : " + str(total_files))
Total Files : 4317
# Visualisasi jumlah gambar tiap kelas
import matplotlib.pyplot as plt

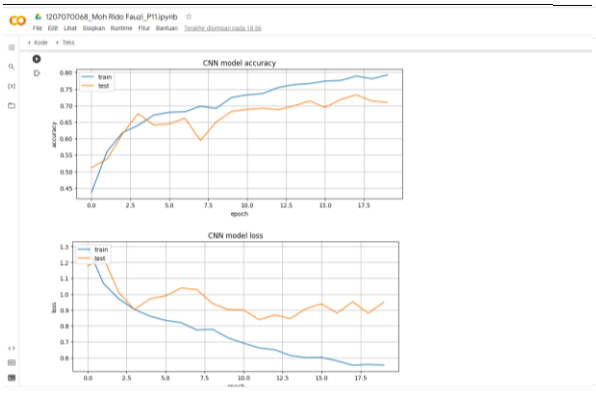

CNN - Google Drive
1207070068_Moh Rido Fauzi_P1
colab.research.google.com/drive/1xuEm1WRHDbjFK2aTUST-5WZ95EZP3rP5
YouTube
Maps
Gmail
1207070068_Moh Rido Fauzi_P11.ipynb
File Edit Lihat Sisipkan Runtime Fitur Bantuan Terakhir disimpan pada 18.56
Komentar
Bagikan
+ Kode + Teks
Hubungkan
tulip
rose


CNN - Google Drive
1207070068_Moh Rido Fauzi_P1
colab.research.google.com/drive/1xuEm1WRHDbjFK2aTUST-5WZ95EZP3rP5
YouTube
Maps
Gmail
dandelion
sunflower
daisy
IMAGE_SIZE = [200,200]
BATCH_SIZE = 32
SEED = 999

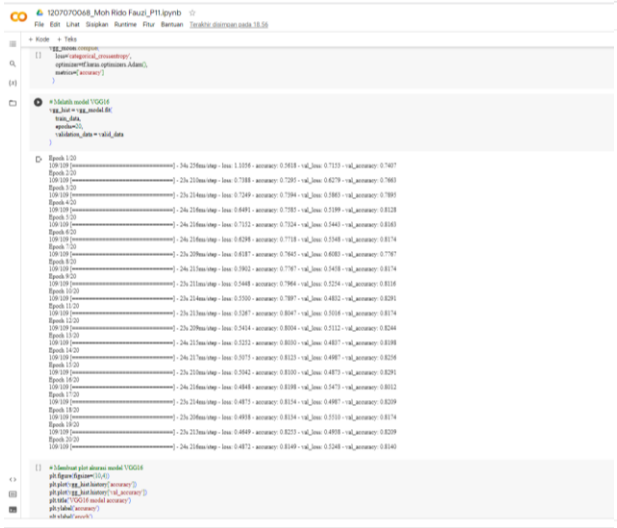

CNN - Google Drive
1207070068_Moh Rido Fauzi_P1
colab.research.google.com/drive/1xuEm1WRHDbjFK2aTUST-5WZ95EZP3rP5
YouTube
Maps
Gmail
1207070068_Moh Rido Fauzi_P11.ipynb
File Edit Lihat Sisipkan Runtime Fitur Bantuan Terakhir disimpan pada 18.56
Komentar
Bagikan
+ Kode + Teks
Hubungkan
plt.bar(number_label.keys(), number_label.values());
plt.title('Jumlah Gambar Tiap Label');
plt.xlabel('Label');
plt.ylabel('Jumlah Gambar');
Jumlah Gambar Tiap Label
Jumlah Gambar
Label
tulip
rose
dandelion
sunflower
daisy
# Menampilkan sampel gambar tiap kelas
import matplotlib.image as mpimg


CNN - Google Drive
1207070068_Moh Rido Fauzi_P1
colab.research.google.com/drive/1xuEm1WRHDbjFK2aTUST-5WZ95EZP3rP5
YouTube
Maps
Gmail
1207070068_Moh Rido Fauzi_P11.ipynb
File Edit Lihat Sisipkan Runtime Fitur Bantuan Terakhir disimpan pada 18.56
+ Kode + Teks
# Menyiapkan data train dan data validation
train_data = datagen.flow_from_directory(
base_dir,
class_mode='categorical',
subset='training',
target_size=IMAGE_SIZE,
batch_size=BATCH_SIZE,
seed=SEED
)
valid_data = datagen.flow_from_directory(
base_dir,
class_mode='categorical',
subset='validation',
target_size=IMAGE_SIZE,
batch_size=BATCH_SIZE,
seed=SEED
)
Found 3457 images belonging to 5 classes.
Found 860 images belonging to 5 classes.
# Image Augmentation
data_augmentation = tf.keras.Sequential(
[
tf.keras.layers.RandomFlip("horizontal",
input_shape=(IMAGE_SIZE[0],
IMAGE_SIZE[1],
3)),
tf.keras.layers.RandomRotation(0.1),
tf.keras.layers.RandomZoom(0.1),
tf.keras.layers.Rescaling(1./255)
]
)
# Membuat arsitektur model CNN
cnn_model = tf.keras.models.Sequential([
data_augmentation,
tf.keras.layers.Conv2D(32, 3, padding='same', activation='relu'),

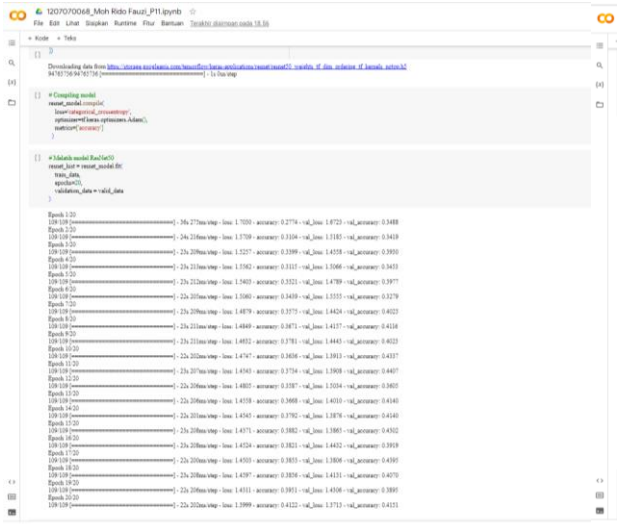

CNN - Google Drive
1207070068_Moh Rido Fauzi_P1
colab.research.google.com/drive/1xuEm1WRHDbjFK2aTUST-5WZ95EZP3rP5
YouTube
Maps
Gmail
1207070068_Moh Rido Fauzi_P11.ipynb
File Edit Lihat Sisipkan Runtime Fitur Bantuan Terakhir disimpan pada 18.56
Komentar
Bagikan
+ Kode + Teks
Hubungkan
Epoch 1/20
109/109 [==========] - 100s 9s/step - loss: 1.2946 - accuracy: 0.4359 - val_loss: 1.1789 - val_accuracy: 0.5116
Epoch 2/20
109/109 [==========] - 22s 204ms/step - loss: 1.0689 - accuracy: 0.5600 - val_loss: 1.2296 - val_accuracy: 0.5372
Epoch 3/20
109/109 [==========] - 21s 193ms/step - loss: 0.9694 - accuracy: 0.6179 - val_loss: 1.0088 - val_accuracy: 0.6140
Epoch 4/20
109/109 [==========] - 22s 202ms/step - loss: 0.9037 - accuracy: 0.6402 - val_loss: 0.9031 - val_accuracy: 0.6756
Epoch 5/20
109/109 [==========] - 23s 210ms/step - loss: 0.8613 - accuracy: 0.6708 - val_loss: 0.9710 - val_accuracy: 0.6407
Epoch 6/20
109/109 [==========] - 22s 200ms/step - loss: 0.8349 - accuracy: 0.6795 - val_loss: 0.9892 - val_accuracy: 0.6442
Epoch 7/20
109/109 [==========] - 22s 203ms/step - loss: 0.8200 - accuracy: 0.6809 - val_loss: 1.0387 - val_accuracy: 0.6616
Epoch 8/20
109/109 [==========] - 22s 202ms/step - loss: 0.7747 - accuracy: 0.6980 - val_loss: 1.0284 - val_accuracy: 0.5942
Epoch 9/20
109/109 [==========] - 22s 203ms/step - loss: 0.7785 - accuracy: 0.6914 - val_loss: 0.9411 - val_accuracy: 0.6500
Epoch 10/20
109/109 [==========] - 23s 207ms/step - loss: 0.7252 - accuracy: 0.7246 - val_loss: 0.9026 - val_accuracy: 0.6826
Epoch 11/20
109/109 [==========] - 21s 191ms/step - loss: 0.6911 - accuracy: 0.7321 - val_loss: 0.9003 - val_accuracy: 0.6884
Epoch 12/20
109/109 [==========] - 22s 200ms/step - loss: 0.6612 - accuracy: 0.7365 - val_loss: 0.8387 - val_accuracy: 0.6919
Epoch 13/20
109/109 [==========] - 22s 200ms/step - loss: 0.6497 - accuracy: 0.7547 - val_loss: 0.8690 - val_accuracy: 0.6872
Epoch 14/20
109/109 [==========] - 21s 193ms/step - loss: 0.6135 - accuracy: 0.7634 - val_loss: 0.8463 - val_accuracy: 0.6988
Epoch 15/20
109/109 [==========] - 24s 218ms/step - loss: 0.6003 - accuracy: 0.7666 - val_loss: 0.9085 - val_accuracy: 0.7140
Epoch 16/20
109/109 [==========] - 22s 202ms/step - loss: 0.6019 - accuracy: 0.7741 - val_loss: 0.9386 - val_accuracy: 0.6942
Epoch 17/20
109/109 [==========] - 21s 193ms/step - loss: 0.5795 - accuracy: 0.7761 - val_loss: 0.8806 - val_accuracy: 0.7186
Epoch 18/20
109/109 [==========] - 22s 201ms/step - loss: 0.5523 - accuracy: 0.7894 - val_loss: 0.9525 - val_accuracy: 0.7326
Epoch 19/20
109/109 [==========] - 21s 194ms/step - loss: 0.5575 - accuracy: 0.7810 - val_loss: 0.8805 - val_accuracy: 0.7140
Epoch 20/20
109/109 [==========] - 22s 198ms/step - loss: 0.5532 - accuracy: 0.7932 - val_loss: 0.9497 - val_accuracy: 0.7093
# Membuat plot akurasi model CNN
plt.figure(figsize=(10,4))


1207070068_Moh Rido Fauzi_P11.ipynb
File Edit Lihat Sisipkan Runtime Fitur Bantuan Terakhir disimpan pada 18.56
+ Kode + Teks
CNN model accuracy
train
test
accuracy
epoch
CNN model loss
train
test
loss
epoch

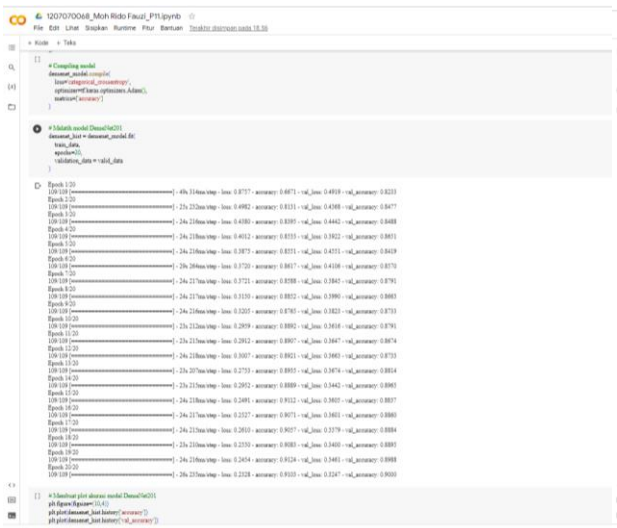

1207070068_Moh Rido Fauzi_P11.ipynb
File Edit Lihat Sisipkan Runtime Fitur Bantuan Terakhir disimpan pada 18.56
Komentar
Bagikan
+ Kode + Teks
Hubungkan
epoch
import tensorflow as tf
from tensorflow.keras.applications.vgg16 import VGG16
Loading VGG16 model
base_vgg_model = VGG16(weights="imagenet", include_top=False, input_shape=(IMAGE_SIZE[0], IMAGE_SIZE[1], 3))
base_vgg_model.trainable = False
# Preprocessing Input
vgg_preprocess = tf.keras.applications.vgg16.preprocess_input
train_data_preprocessing_function = vgg_preprocess
Downloading data from https://storage.googleapis.com/tensorflow/keras-applications/vgg16/vgg16_weights_tf_dim_ordering_tf_kernels_notop.h5
58889256/58889256 [==========] - 0s 0us/step
# Transfer learning dengan VGG16
vgg_model = tf.keras.models.Sequential([
data_augmentation,
base_vgg_model,
tf.keras.layers.Dropout(0.7),
tf.keras.layers.Flatten(),
tf.keras.layers.Dense(64, activation='relu'),
tf.keras.layers.Dense(64, activation='relu'),
tf.keras.layers.Dense(5, activation='softmax')
])
Compiling model
vgg_model.compile(
loss='categorical_crossentropy',
optimizer=tf.keras.optimizers.Adam(),
metrics=['accuracy']
)

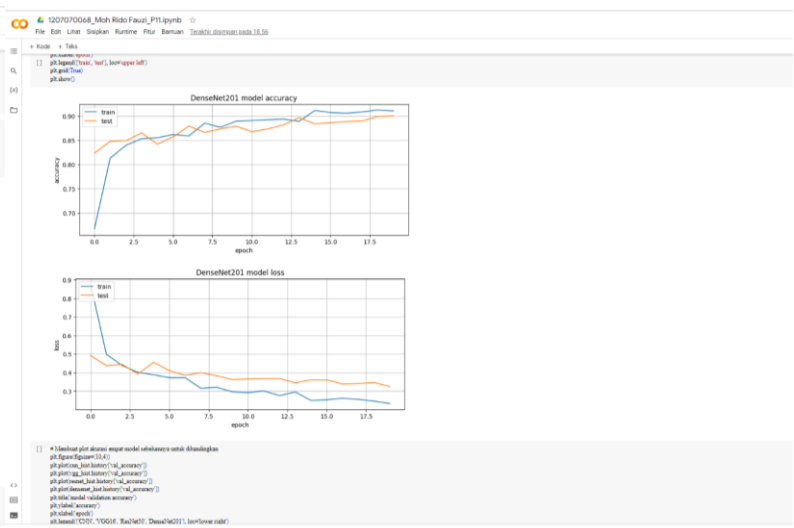


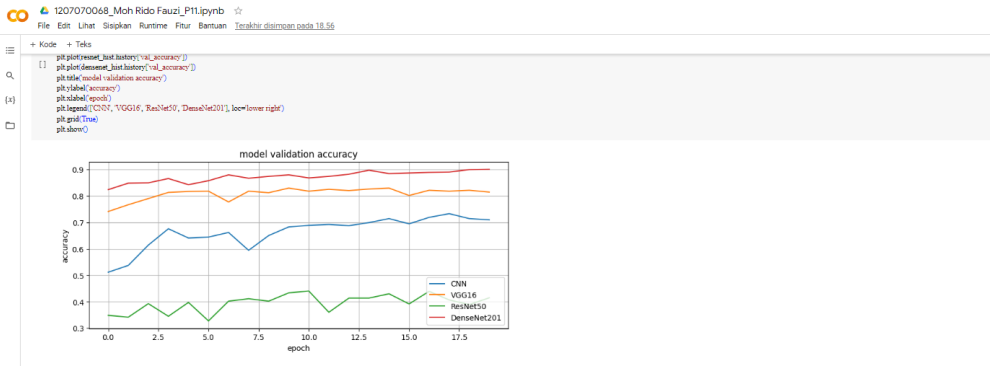

VGG16 model accuracy
train
test
accuracy
epoch
VGG16 model loss
train
test
loss
epoch

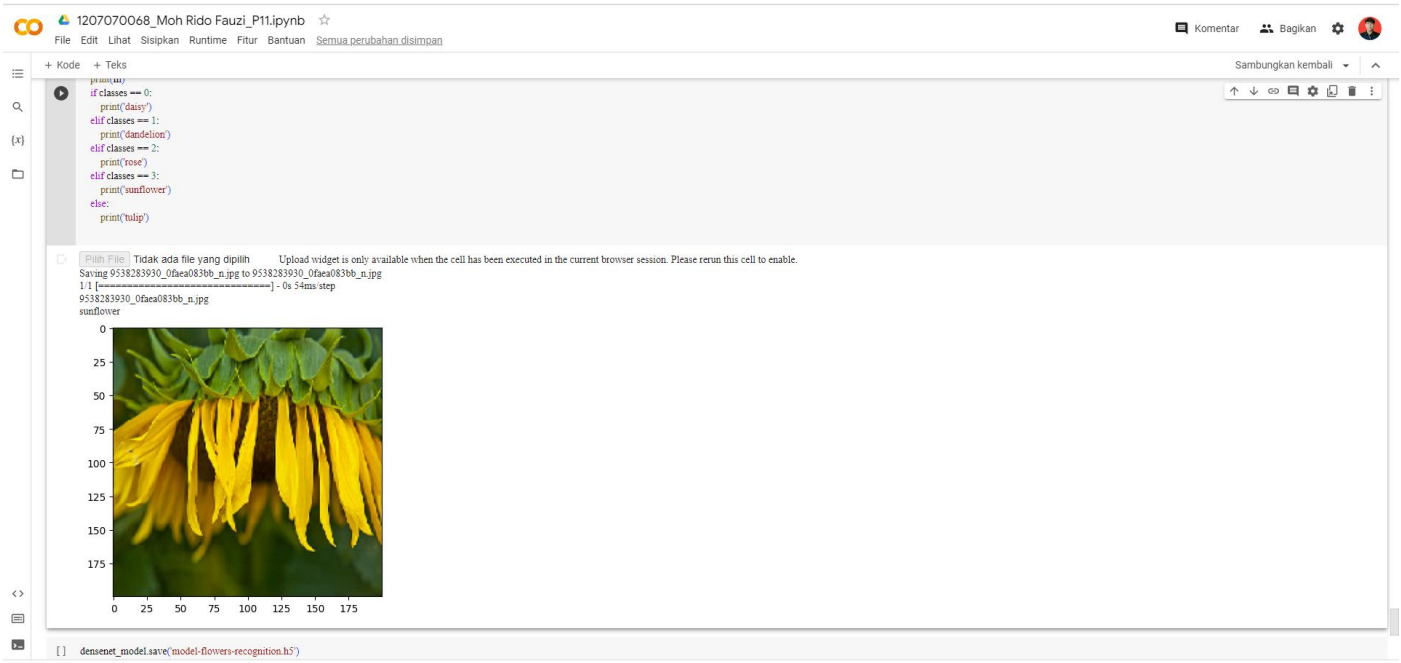



ResNet50 model accuracy
train
test
accuracy
epoch
ResNet50 model loss
train
test
loss
epoch




DenseNet201 model accuracy
train
test
accuracy
epoch
DenseNet201 model loss
train
test
loss
epoch


1207070068_Moh Rido Fauzi_P11.ipynb
File Edit Lihat Sisipkan Runtime Fitur Bantuan Terakhir disimpan pada 18.56
+ Kode + Teks
plt.plot(resnet_hist.history['val_accuracy'])
plt.plot(densenet_hist.history['val_accuracy'])
plt.title('model validation accuracy')
plt.ylabel('accuracy')
plt.xlabel('epoch')
plt.legend(['CNN', 'VGG16', 'ResNet50', 'DenseNet201'], loc='lower right')
plt.grid(True)
plt.show()
model validation accuracy
accuracy
epoch
CNN
VGG16
ResNet50
DenseNet201
Instal TensorflowJS
!pip install tensorflowjs
Menampilkan daftar kelas atau label gambar
train_data.class_indices



Pada Percobaan Cnn ini dimulai dengan mendefinisikan direktori utama dataset yang berisi gambar-gambar bunga. Direktori ini diakses menggunakan os.listdir dan disimpan dalam variabel base_dir. Kemudian menghitung menghitung jumlah gambar pada setiap kelas dalam dataset. Jumlah gambar tiap kelas disimpan dalam number_label, dan total jumlah gambar dihitung dengan variabel total_files. tf.keras.preprocessing.image.ImageDataGenerator untuk melakukan preprocessing pada gambar, dengan membagi data menjadi data train dan data validation. Model CNN dalam program ini menggunakan lapisan tf.keras.Sequential. Terdapat beberapa lapisan konvolusi dan pooling untuk mengekstraksi fitur dari gambar, serta lapisan dropout untuk mengurangi overfitting. Akhirnya, terdapat lapisan densly connected (fully connected) dengan fungsi aktivasi softmax untuk klasifikasi. Yang pada intinya percobaan ini mengimplementasikan model CNN untuk mengenali kelas-kelas gambar dalam dataset bunga. Dilakukan preprocessing pada gambar, pelatihan model CNN, dan visualisasi hasil pelatihan dalam bentuk grafik akurasi dan loss. Percobaan ini juga melakukan transfer learning dengan menggunakan model-model terkenal dan melatihnya pada dataset gambar bunga. Model-model tersebut dikompilasi, dilatih, dan hasilnya divisualisasikan dalam bentuk grafik akurasi dan loss. Grafik akurasi dari keempat model juga dibandingkan untuk melihat performa relatif mereka.

Kemudian pada bagian akhir percobaan ini terdapat Program yang merupakan bagian bertanggung jawab untuk memprediksi kelas (jenis) dari gambar bunga yang diunggah oleh pengguna.pada Program ini memungkinkan pengguna untuk mengunggah file gambar bunga dan melakukan prediksi kelas (jenis) dari gambar tersebut menggunakan model DenseNet201 yang telah dilatih sebelumnya. Hasil prediksi kemudian dicetak dalam bentuk nama kelas yang sesuai.