

Laporan Proyek Machine Learning



Yehezkiel Kristanto	C14190007
Leonardo Susanto	C14190010
Anderson Montella	C14190014

**Fakultas Teknologi Industri
Program Studi Teknik Informatika
Universitas Kristen Petra
Surabaya**

ANGGOTA KELOMPOK DAN PEMBAGIAN KERJA

- Yehezkiel K. / C14190007 : penerapan program (Epoch), merancang laporan, edit video
- Leonando L. S. / C14190010 : Mencari dataset, Membuat video promosi
- Anderson M. K. / C14190014 : Memodifikasi layer dari model, Merancang laporan

MODEL YANG DIUJIKAN

Di pengujian kelompok kami fokus menguji apa ketika layer dari model kita modifikasi seperti activation method yang digunakan.

Untuk ini percobaan untuk Epoch 20:

```
▶ history = model.fit(x=train_dataset,  
                      epochs= 20,  
                      validation_data=val_dataset)
```

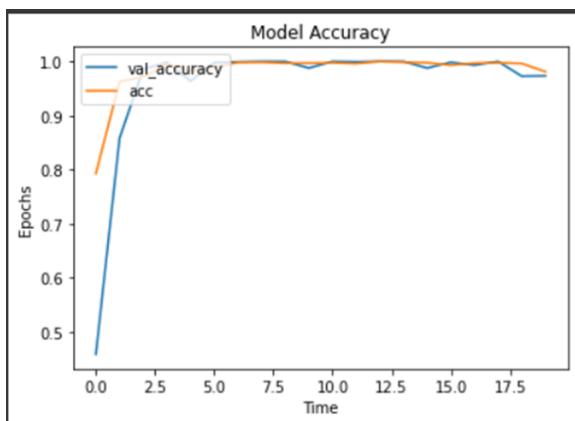
```
class_names = train_dataset.class_names  
print(len(class_names))  
print(class_names)
```

Untuk memasukan nama dari dataset class name mengambil dari label

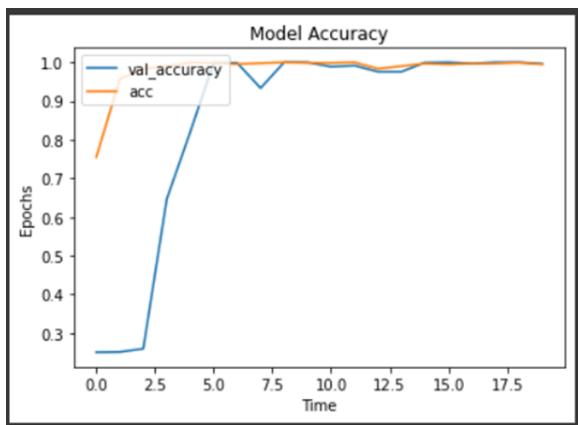
- ▶ circle
- ▶ square
- ▶ star
- ▶ triangle

Dari label folder ini class name diambil untuk mengambil label datasetnya

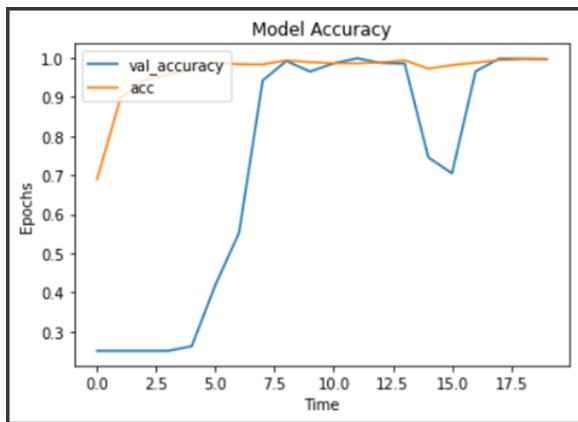
Grafik *Accuracy* dari training dataset 1 menggunakan *activation method* ‘relu’



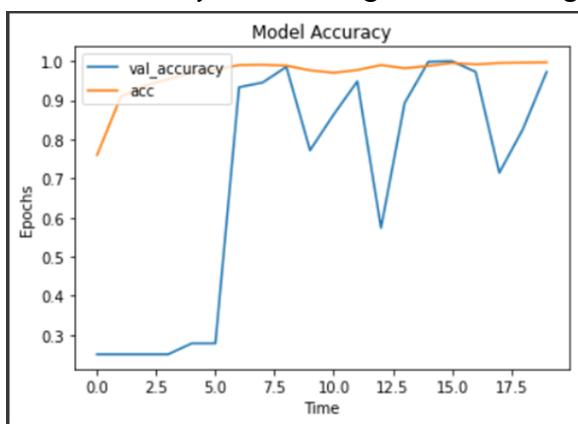
Grafik *Accuracy* dari training dataset 1 menggunakan *activation method* ‘*relu relu sigmoid*’



Grafik *Accuracy* dari training dataset 1 menggunakan *activation method* ‘*sigmoid*’

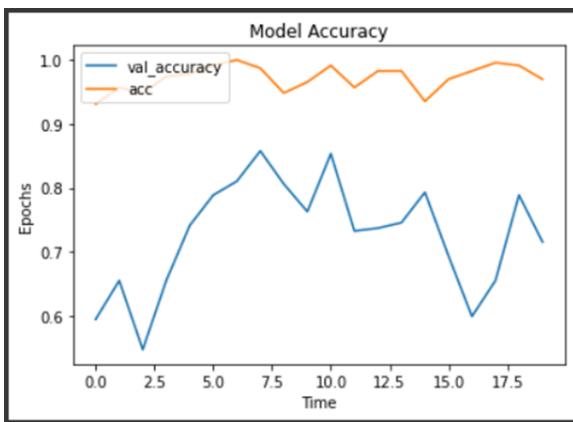


Grafik *Accuracy* dari training dataset 1 menggunakan *activation method* ‘*sigmoid sigmoid relu*’

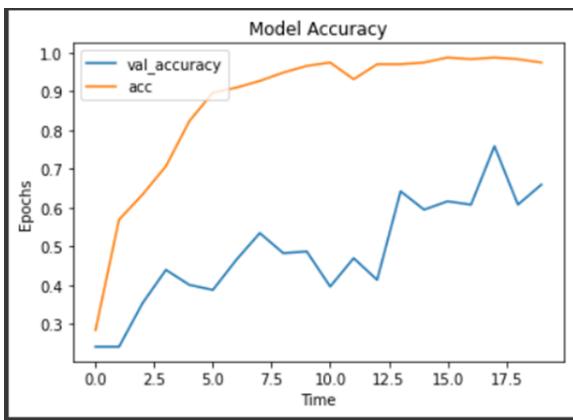


Dari grafik diatas perubahan paling stabil di dataset 1 ketika menggunakan *activation method* ‘*relu*’

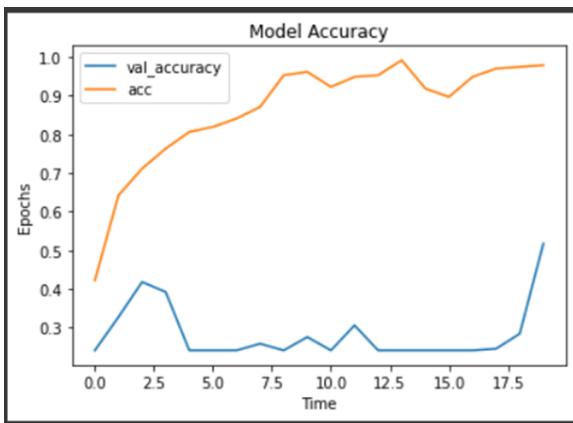
Grafik *Accuracy* dari training dataset 2 menggunakan *activation method* ‘relu’



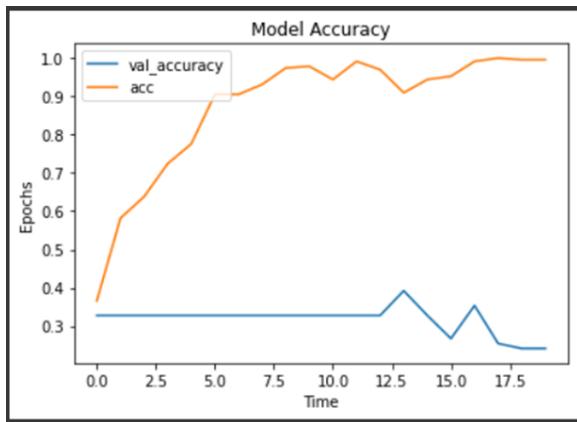
Grafik *Accuracy* dari training dataset 2 menggunakan *activation method* ‘relu relu sigmoid’



Grafik *Accuracy* dari training dataset 2 menggunakan *activation method* ‘sigmoid sigmoid relu’



Grafik *Accuracy* dari training dataset 2 menggunakan *activation method* ‘sigmoid’



Dari grafik diatas perubahan paling stabil di dataset 2 ketika menggunakan *activation method* ‘sigmoid’

```
model = Sequential()
model.add(Conv2D(32, kernel_size = (3, 3), activation='relu', input_shape=(100,100,3)))
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(BatchNormalization())
model.add(Conv2D(64, kernel_size=(3,3), activation='relu'))
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(BatchNormalization())
model.add(Conv2D(64, kernel_size=(3,3), activation='relu'))
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(BatchNormalization())
model.add(Conv2D(96, kernel_size=(3,3), activation='relu'))
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(BatchNormalization())
model.add(Dropout(0.2))
model.add(Flatten())
model.add(Dense(128, activation='relu'))
#model.add(Dropout(0.3))
model.add(Dense(len(class_names),activation='softmax'))
```

epoch	dataset	loss	Accuracy	Val-loss	Val-accuracy
10	1	0.00036	1	0.00016	1
20	1	0.00013	1	0.000053	1
50	1	0.000014	1	0.0000039	1
10	2	0.0377	0.9871	1.123	0.5905
20	2	0.0075	1	0.2071	0.9440

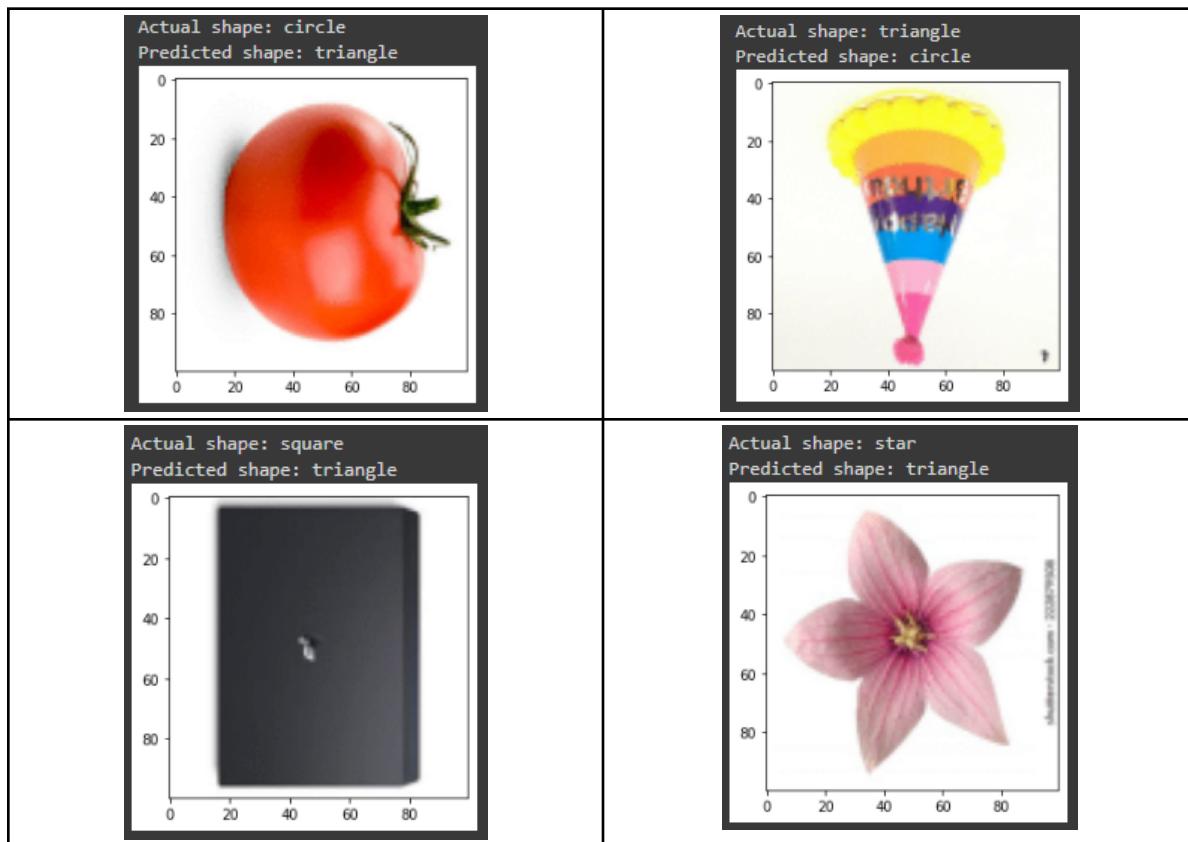
50	2	0.0143	1	09138	0.8147
----	---	--------	---	-------	--------

Dataset 1 merupakan gabungan dari apa yang kami buat dengan kaggle

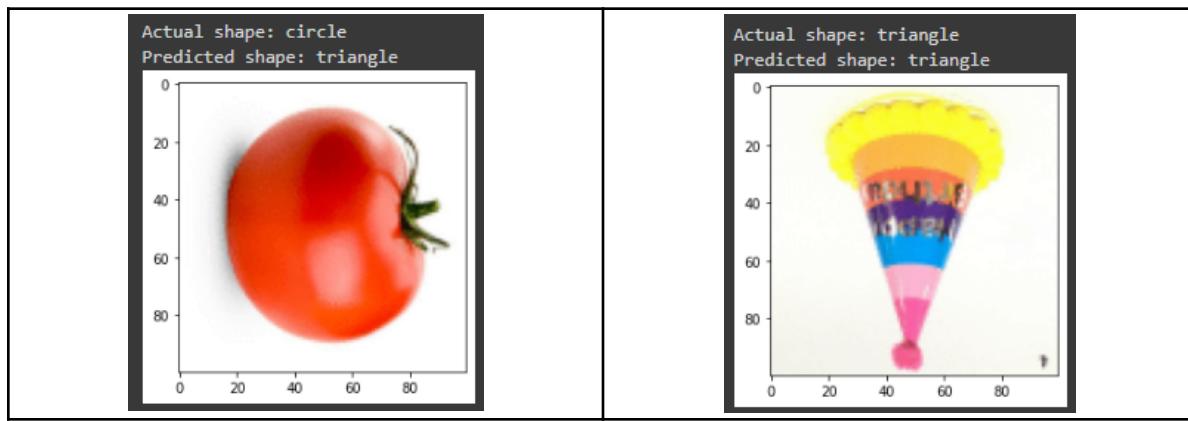
Dataset 2 merupakan foto foto dari barang barang di dunia nyata

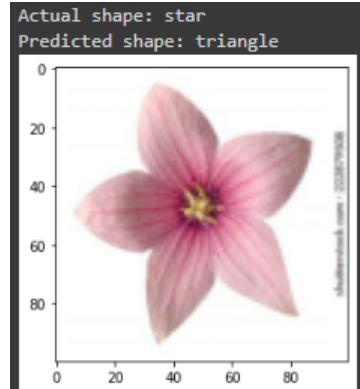
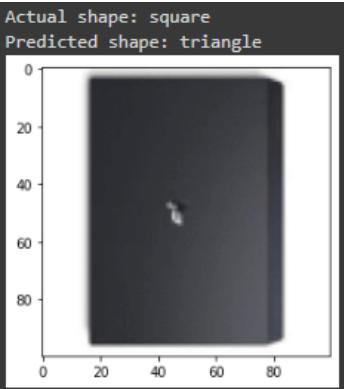
Percobaan

Epoch 10 dataset 1

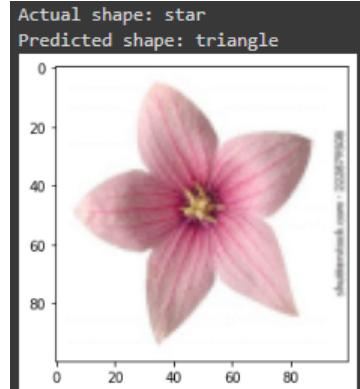
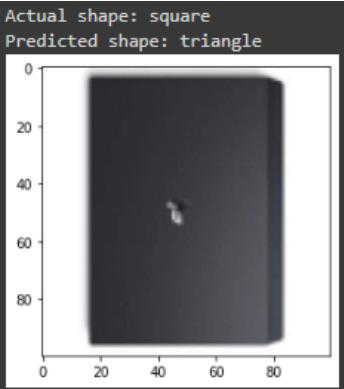
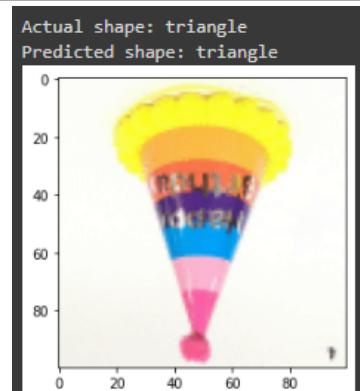
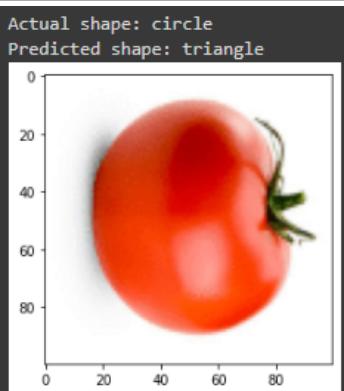


Epoch 20 dataset 1

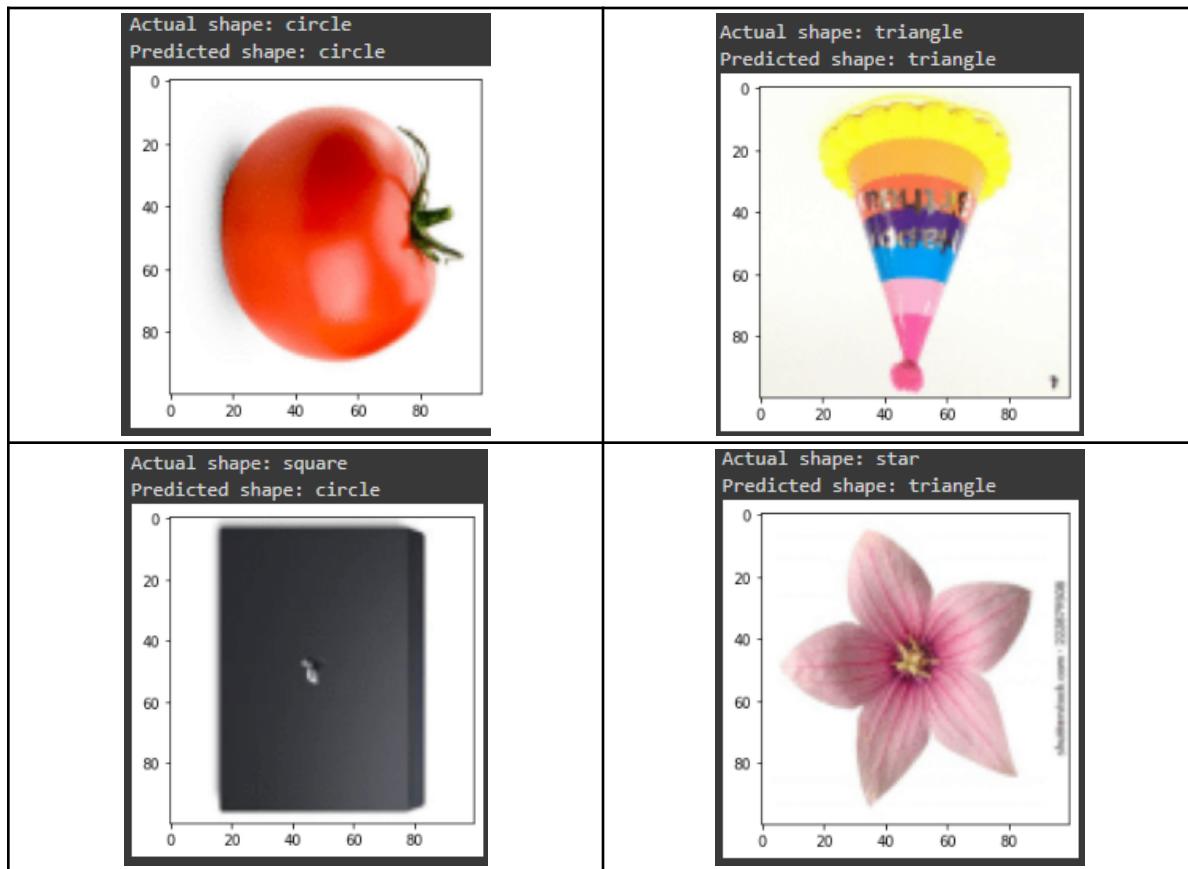




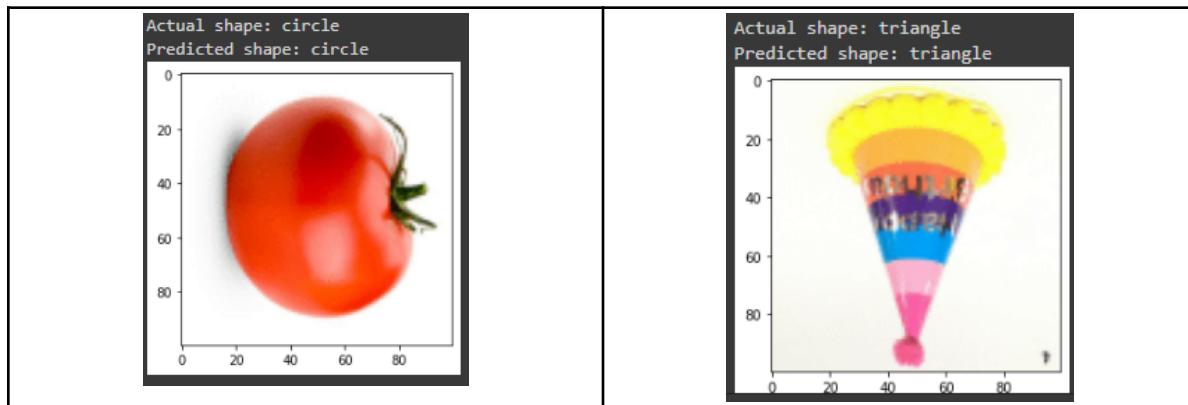
Epoch 50 dataset 1

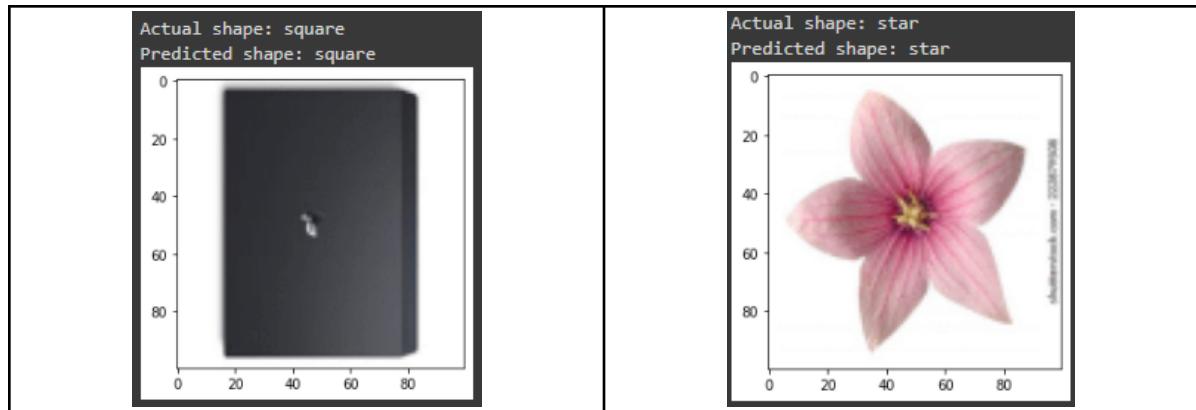


Epoch 10 dataset 2

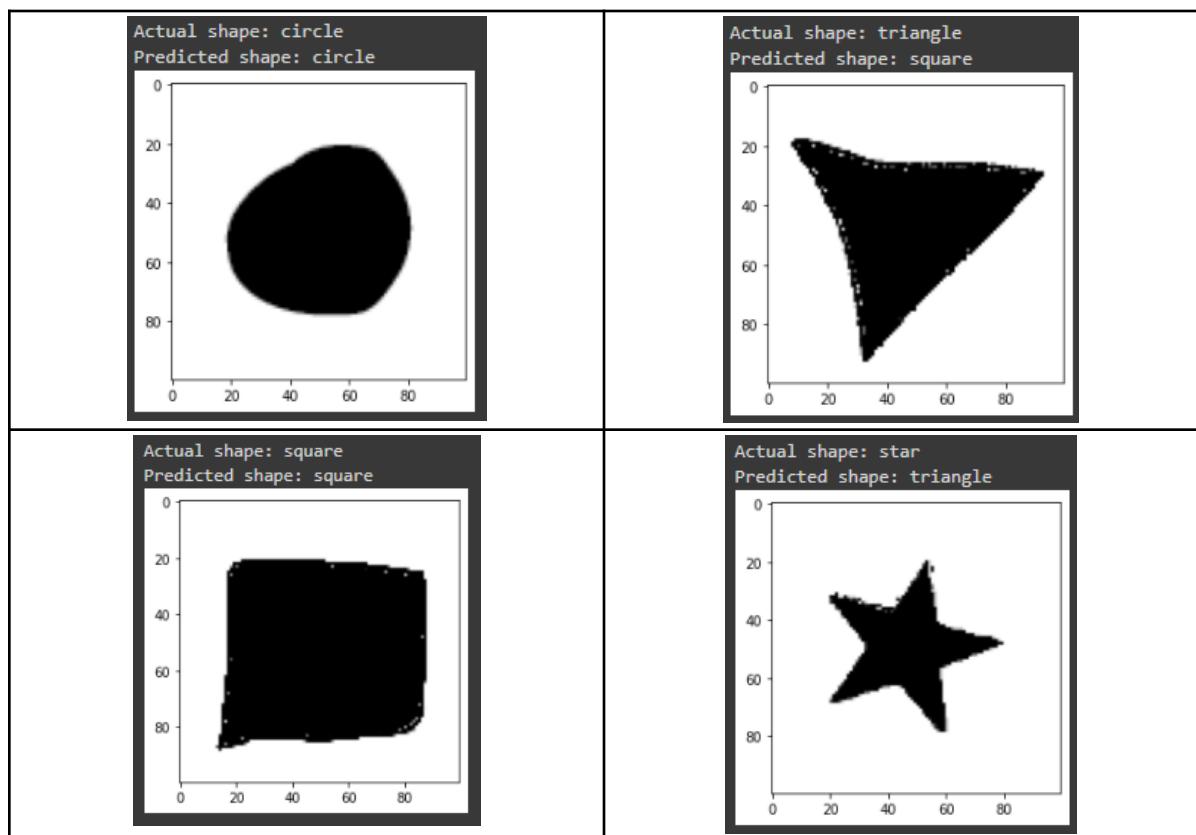


Epoch 20 dataset 2

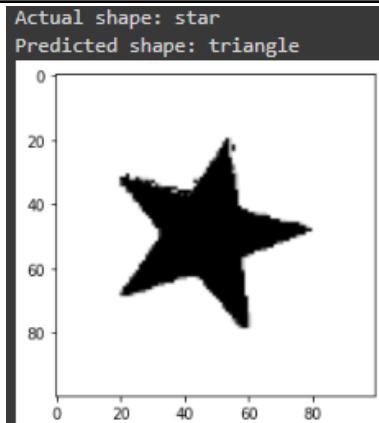
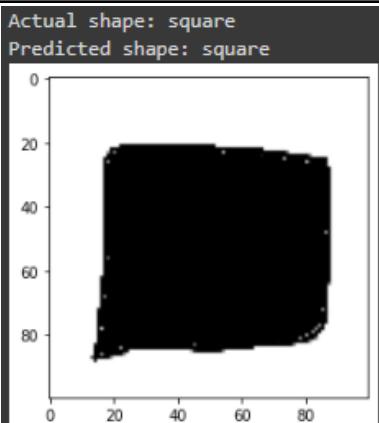
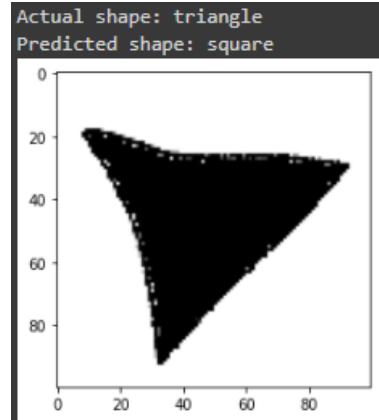
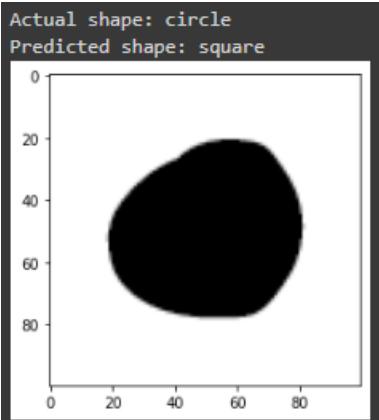




Epoch 20 dataset 2



Epoch 50 dataset 2



```

model = Sequential()
model.add(Conv2D(32, kernel_size = (3, 3), activation='relu', input_shape=(100,100,3)))
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(BatchNormalization())
model.add(Conv2D(64, kernel_size=(3,3), activation='relu'))
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(BatchNormalization())
model.add(Conv2D(64, kernel_size=(3,3), activation='relu'))
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(BatchNormalization())
model.add(Dropout(0.2))
model.add(Flatten())
model.add(Dense(128, activation='relu'))
#model.add(Dropout(0.3))
model.add(Dense(len(class_names),activation='softmax'))

```

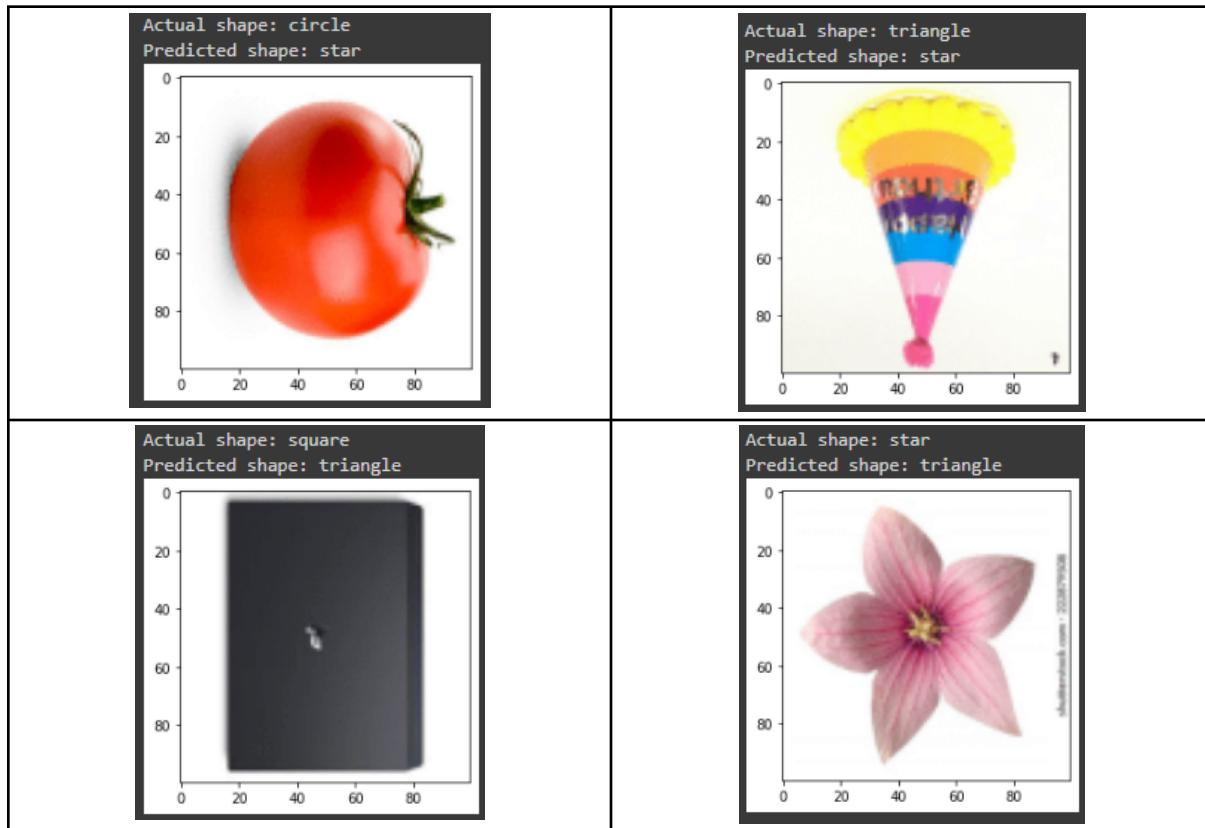
epoch	dataset	loss	Accuracy	Val-loss	Val-accuracy
10	1	0.00032	1	0.00019	1
20	1	0.00010	1	0.000051	1
50	1	0.0000195	1	0.00000889	1
10	2	0.0177	0.9914	2.6136	0.6681
20	2	0.00047	1	0.3651	0.9181
50	2	0.1767	0.9483	14.4160	0.5474

Dataset 1 merupakan gabungan dari apa yang kami buat dengan kaggle

Dataset 2 merupakan foto foto dari barang barang di dunia nyata

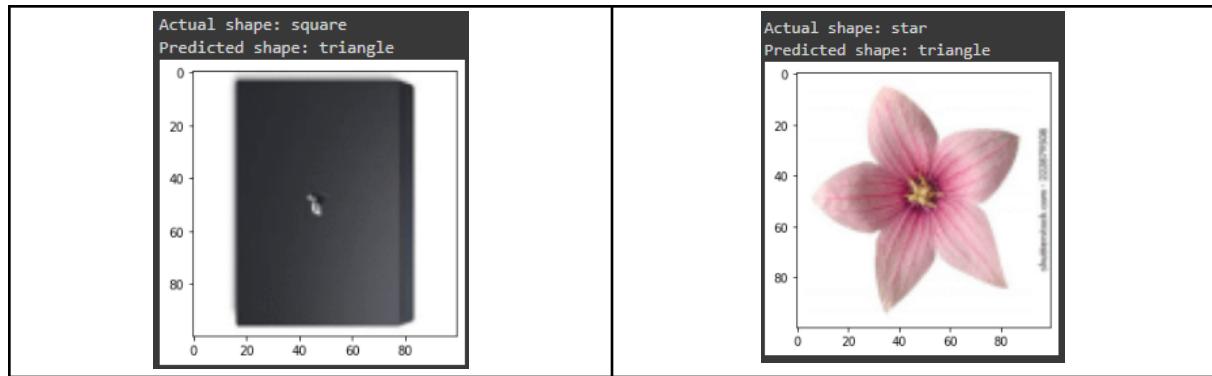
Percobaan

Epoch 10 dataset 1

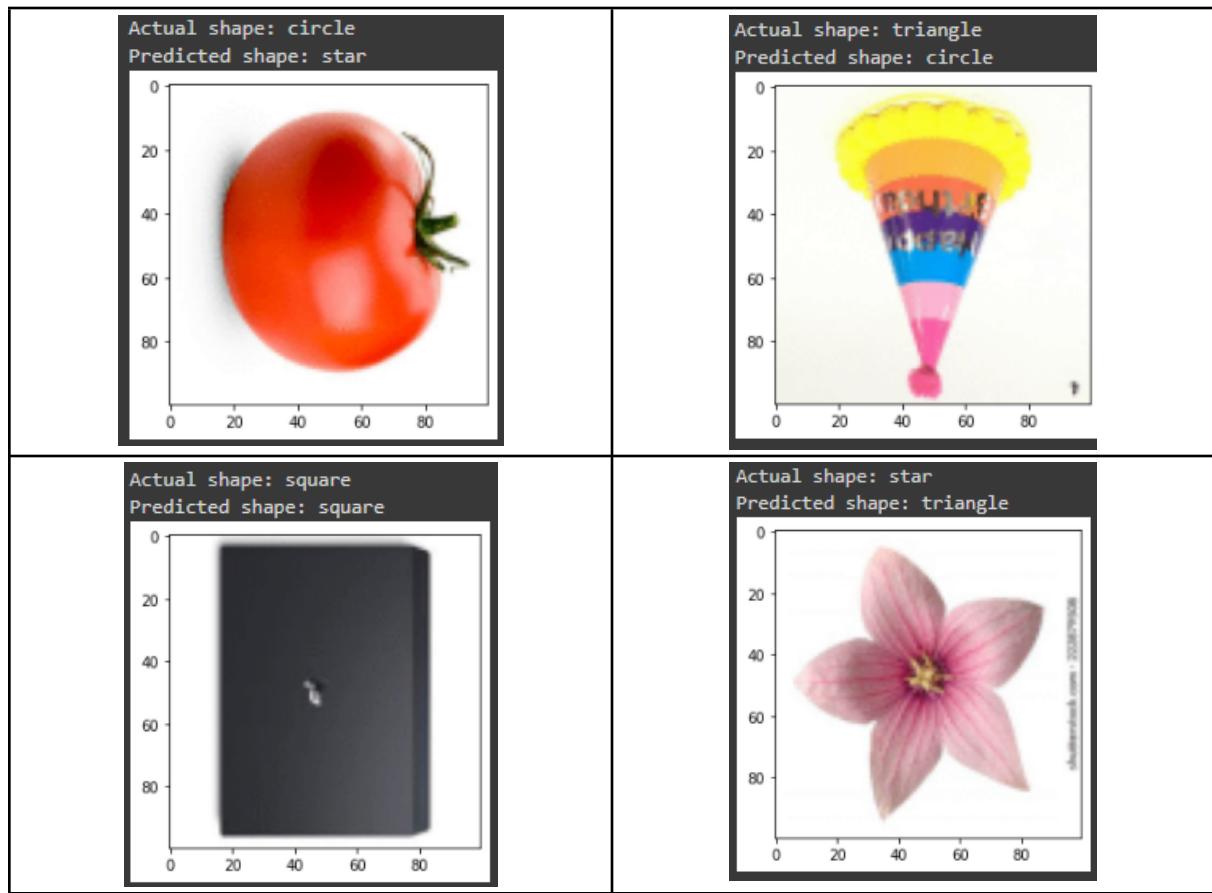


Epoch 20 dataset 1

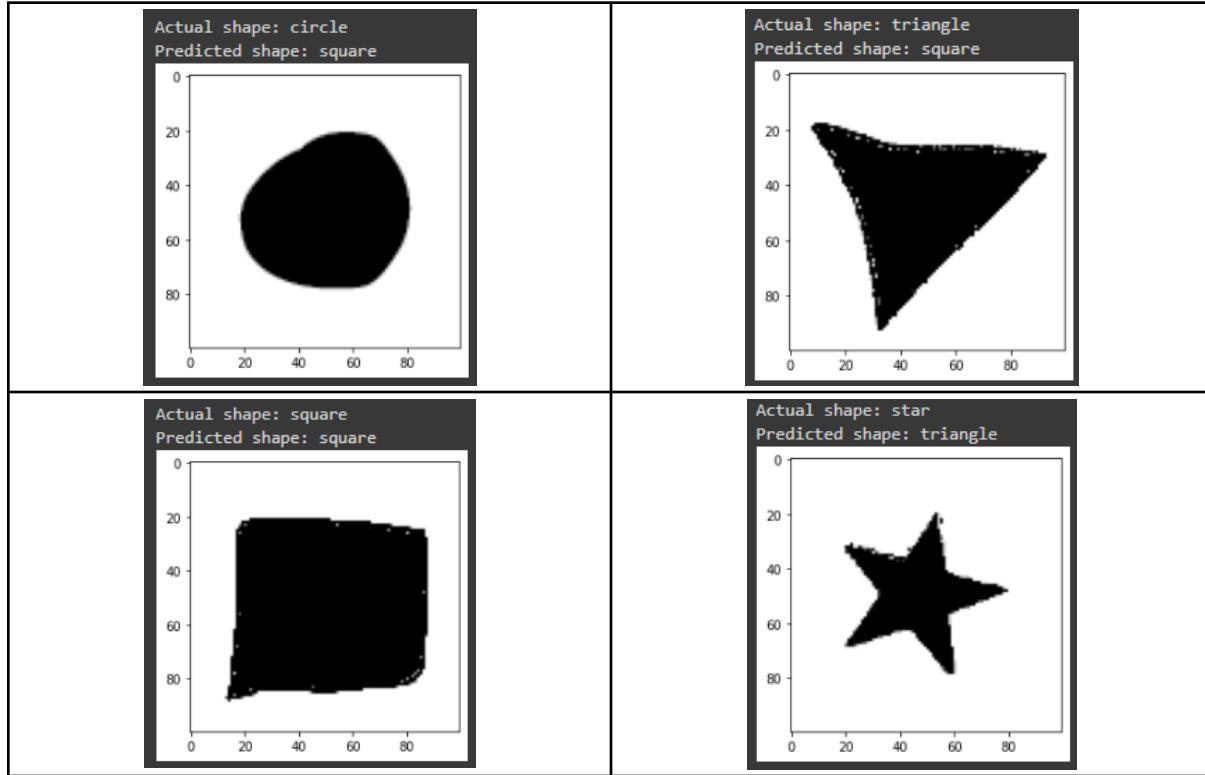




Epoch 50 dataset 1

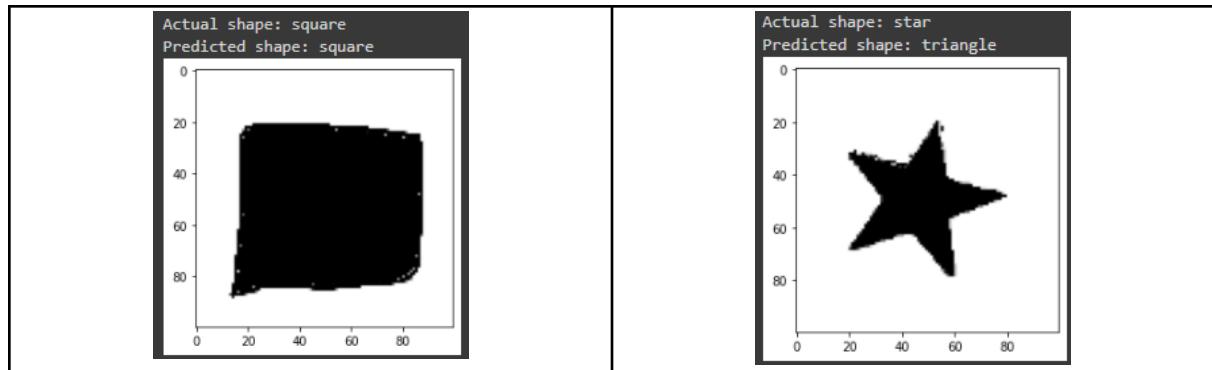


Epoch 10 dataset 2

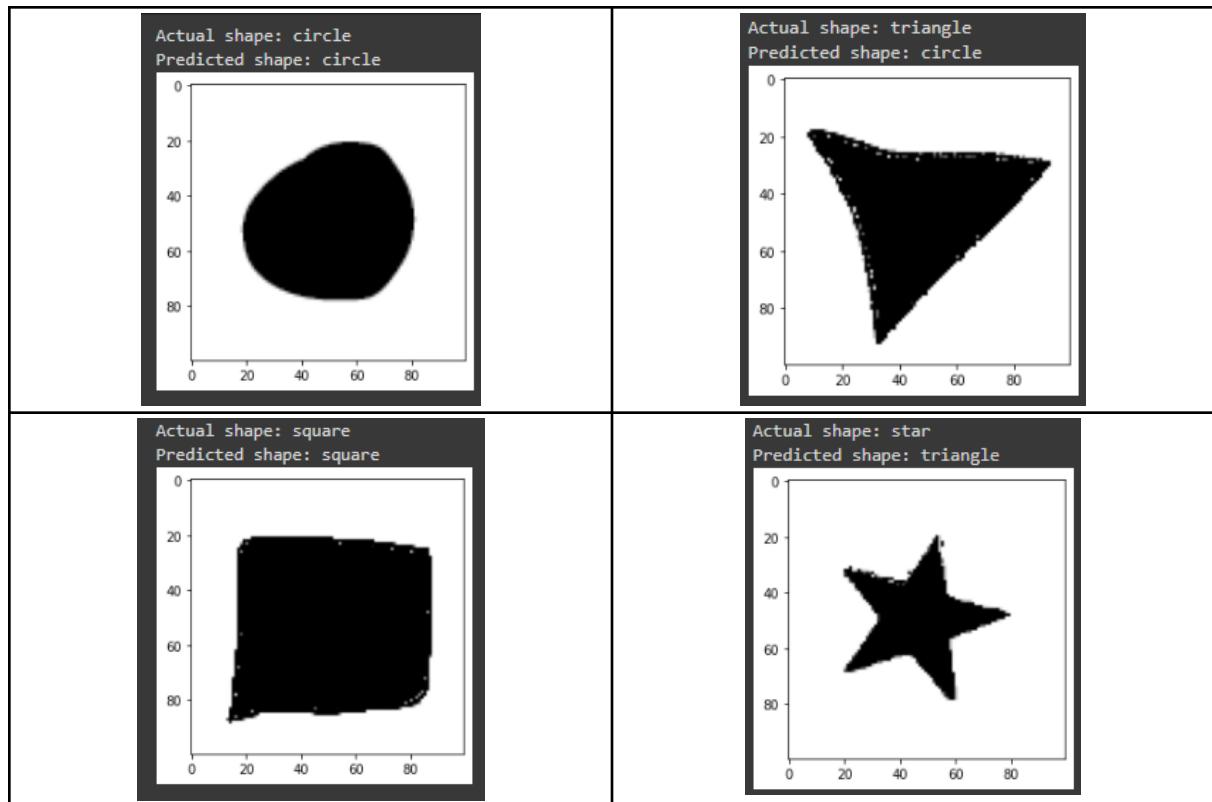


Epoch 20 dataset 2





Epoch 50 dataset 2



//////////

```
model = Sequential()
model.add(Conv2D(32, kernel_size = (3, 3), activation='relu', input_shape=(100,100,3)))
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(BatchNormalization())
model.add(Conv2D(64, kernel_size=(3,3), activation='relu'))
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(BatchNormalization())
model.add(Conv2D(128, kernel_size=(3,3), activation='relu'))
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(BatchNormalization())
model.add(Conv2D(256, kernel_size=(3,3), activation='relu'))
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(BatchNormalization())
model.add(Conv2D(512, kernel_size=(3,3), activation='relu'))
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(BatchNormalization())
model.add(Dropout(0.2))
model.add(Flatten())
model.add(Dense(128, activation='relu'))
#model.add(Dropout(0.3))
model.add(Dense(len(class_names),activation='softmax'))
```

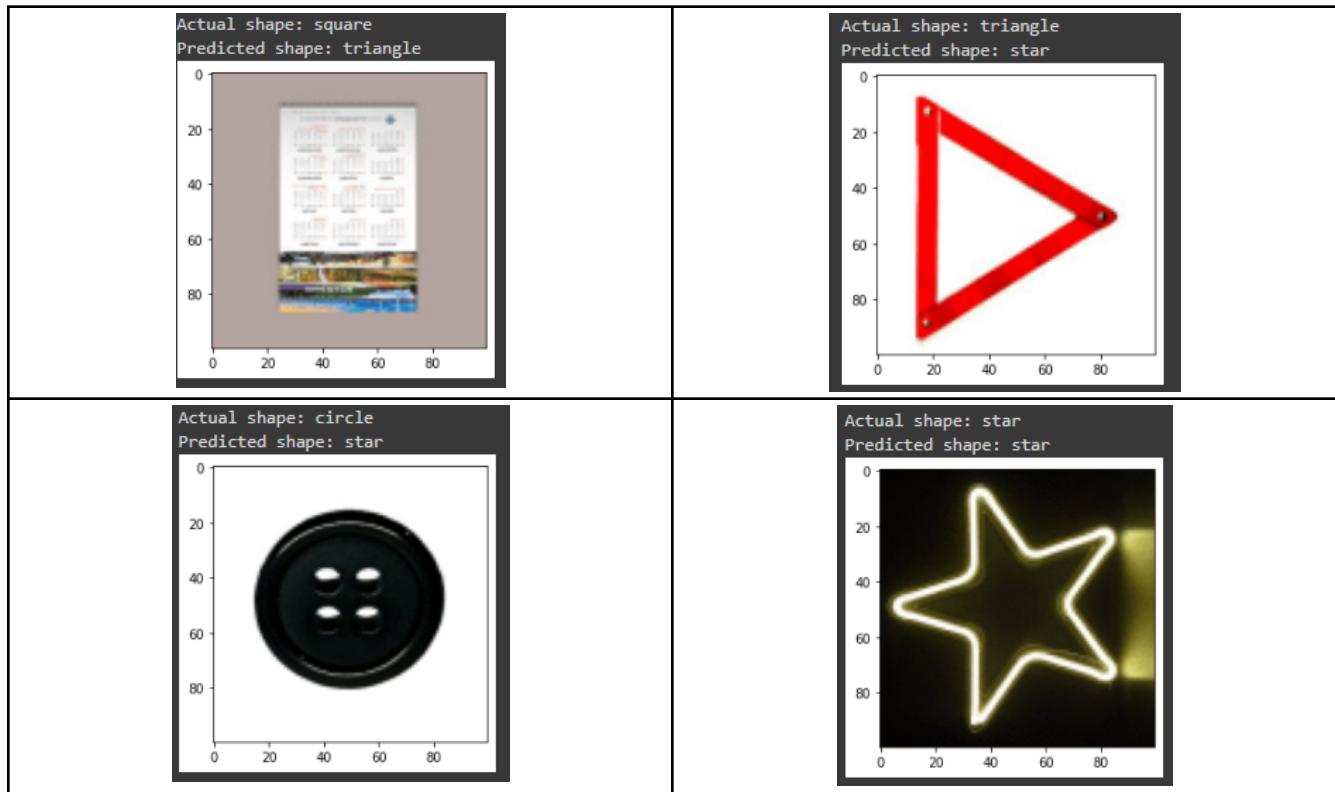
epoch	dataset	loss	Accuracy	Val-loss	Val-accuracy
10	1	0.097	0.9970	0.230	0.9932
20	1	0.0060	0.9977	0.0071	0.9977
50	1	0.0025	0.9985	0.00021	1
10	2	0.2861	0.8879	1.7788	0.5862
20	2	0.0512	0.9828	0.2299	0.9009
50	2	0.0099	1	0.3198	0.8966

Dataset 1 merupakan gabungan dari apa yang kami buat dengan kaggle

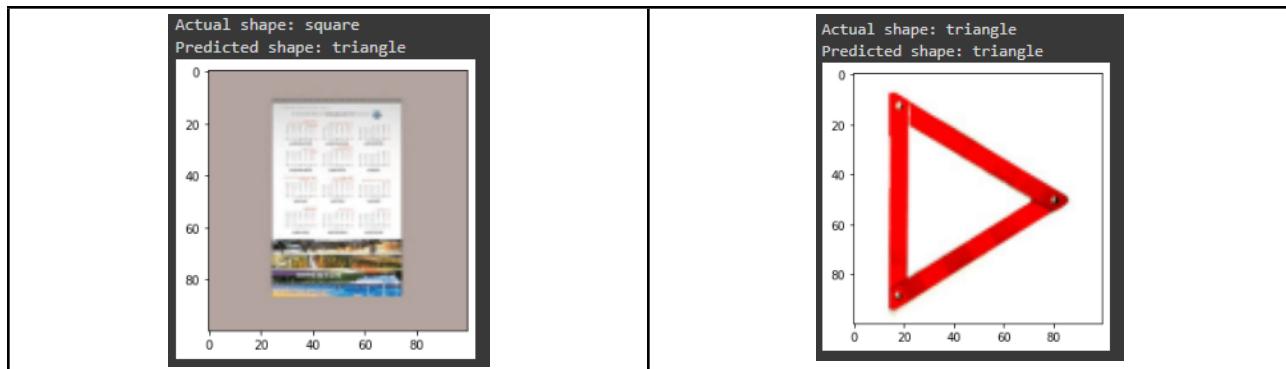
Dataset 2 merupakan foto foto dari barang barang di dunia nyata

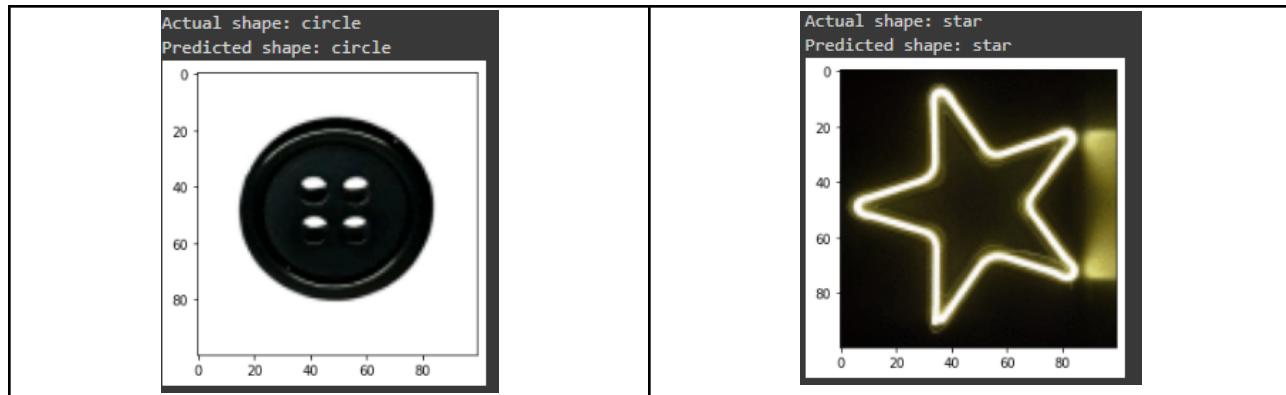
Percobaan

Epoch 10 dataset 1

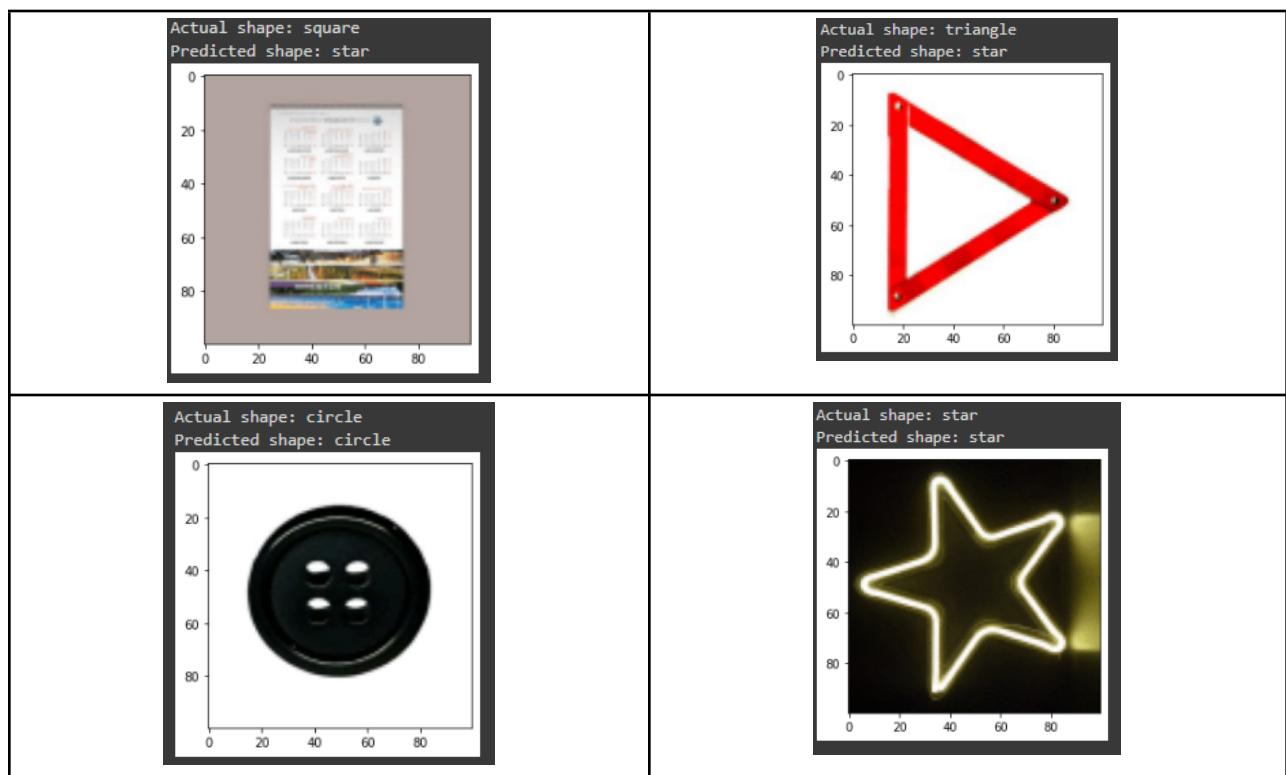


Epoch 20 dataset 1

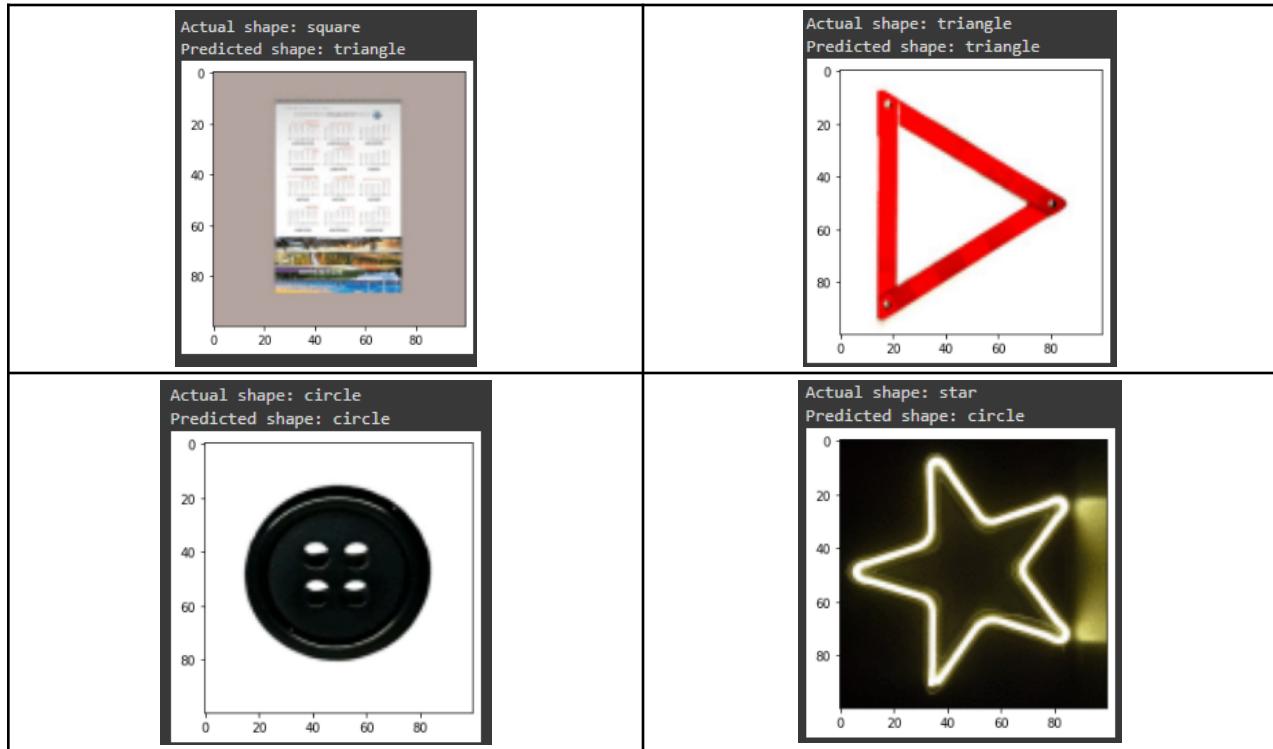




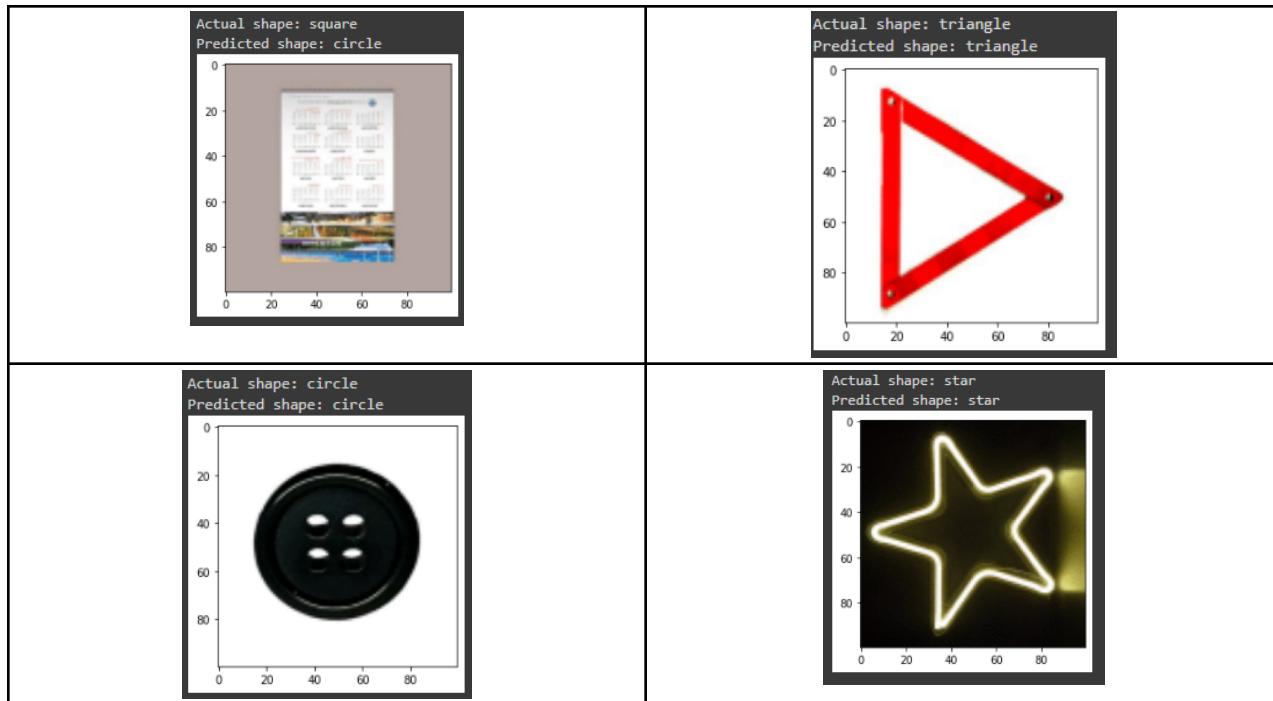
Epoch 50 dataset 1



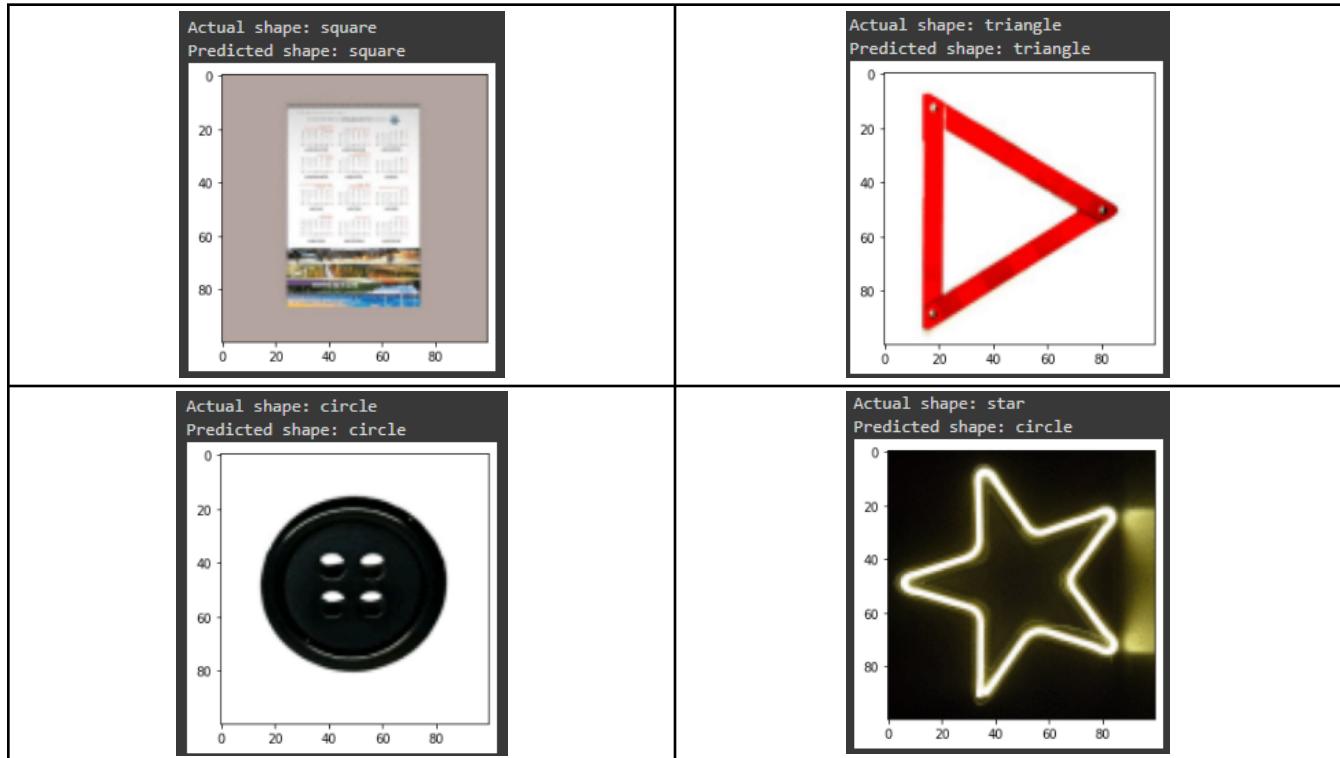
Epoch 10 dataset 2



Epoch 20 dataset 2



Epoch 50 dataset 2



//////////

```
model = Sequential()
model.add(Conv2D(32, kernel_size = (3, 3), activation='sigmoid', input_shape=(100,100,3)))
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(BatchNormalization())
model.add(Conv2D(64, kernel_size=(3,3), activation='sigmoid'))
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(BatchNormalization())
model.add(Conv2D(128, kernel_size=(3,3), activation='sigmoid'))
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(BatchNormalization())
model.add(Conv2D(256, kernel_size=(3,3), activation='sigmoid'))
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(BatchNormalization())
model.add(Conv2D(512, kernel_size=(3,3), activation='sigmoid'))
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(BatchNormalization())
model.add(Dropout(0.2))
model.add(Flatten())
model.add(Dense(128, activation='sigmoid'))
#model.add(Dropout(0.3))sigmoid
model.add(Dense(len(class_names),activation='softmax'))
```

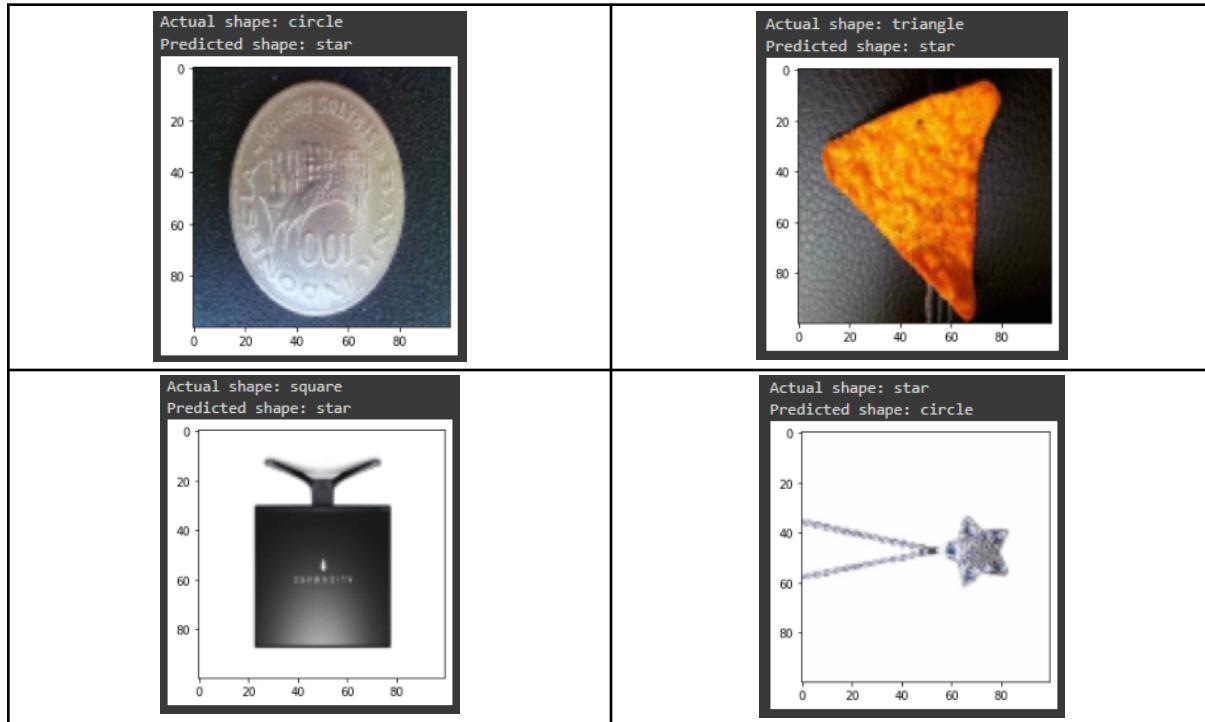
epoch	dataset	loss	Accuracy	Val-loss	Val-accuracy
10	1	0.0309	0.9909	0.1034	0.9621
20	1	0.0290	0.9894	0.9860	0.8000
50	1	0.0277	0.9909	0.0486	0.9833
10	2	0.1565	0.9612	2.0019	0.3276
20	2	0.0605	0.9914	2.6644	0.2414
50	2	0.1227	0.9655	2.3955	0.4526

Dataset 1 merupakan gabungan dari apa yang kami buat dengan kaggle

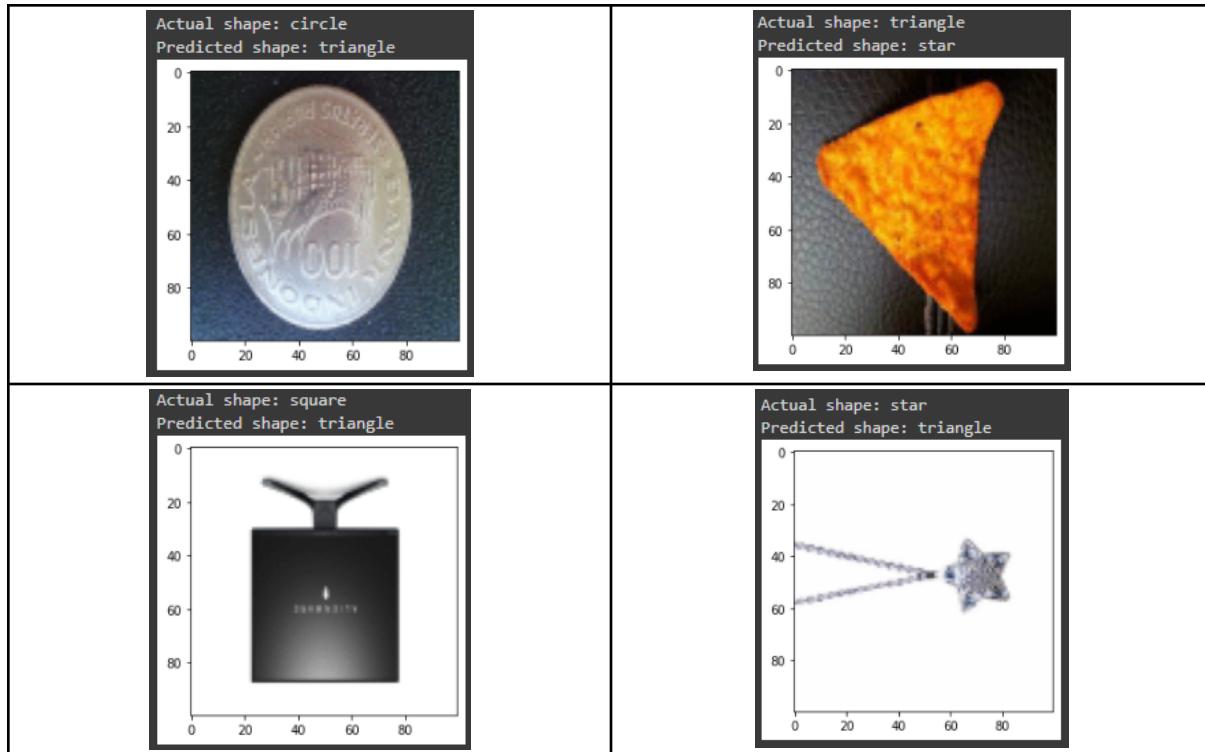
Dataset 2 merupakan foto foto dari barang barang di dunia nyata

Percobaan

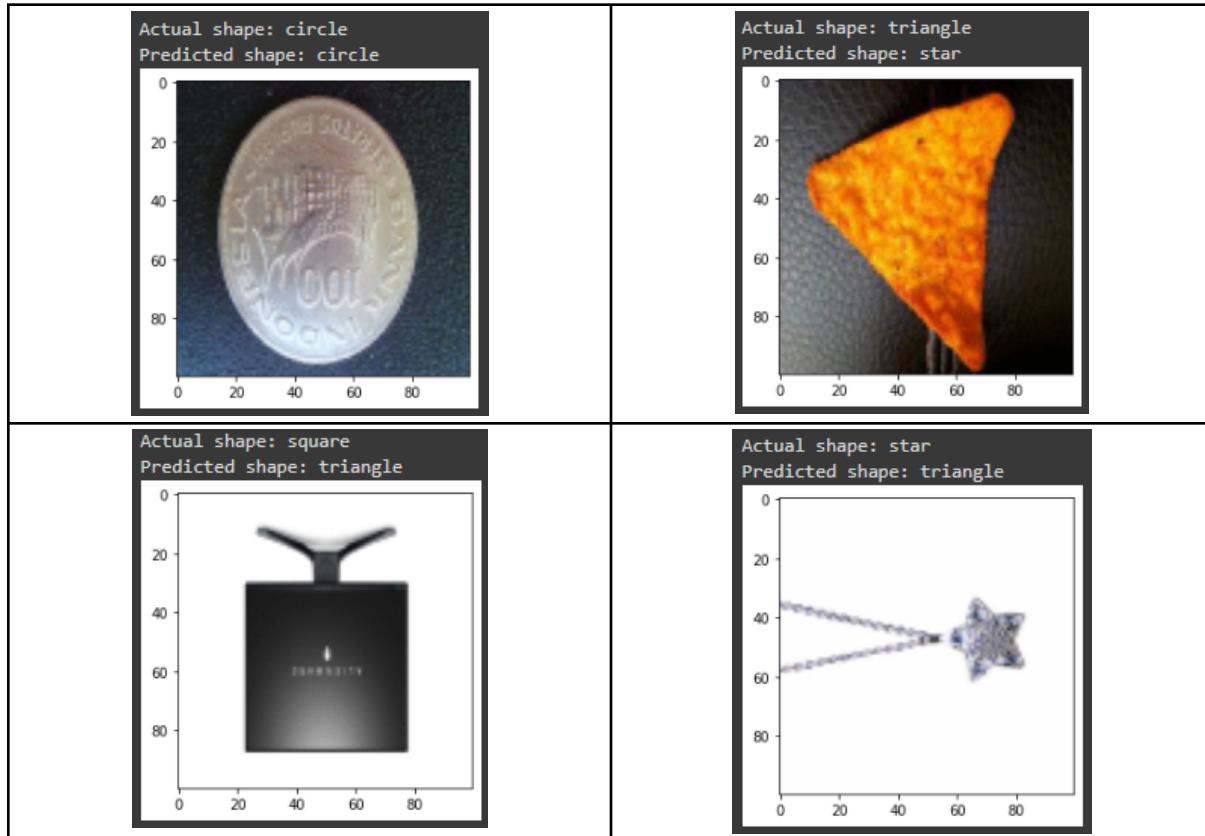
Epoch 10 dataset 1



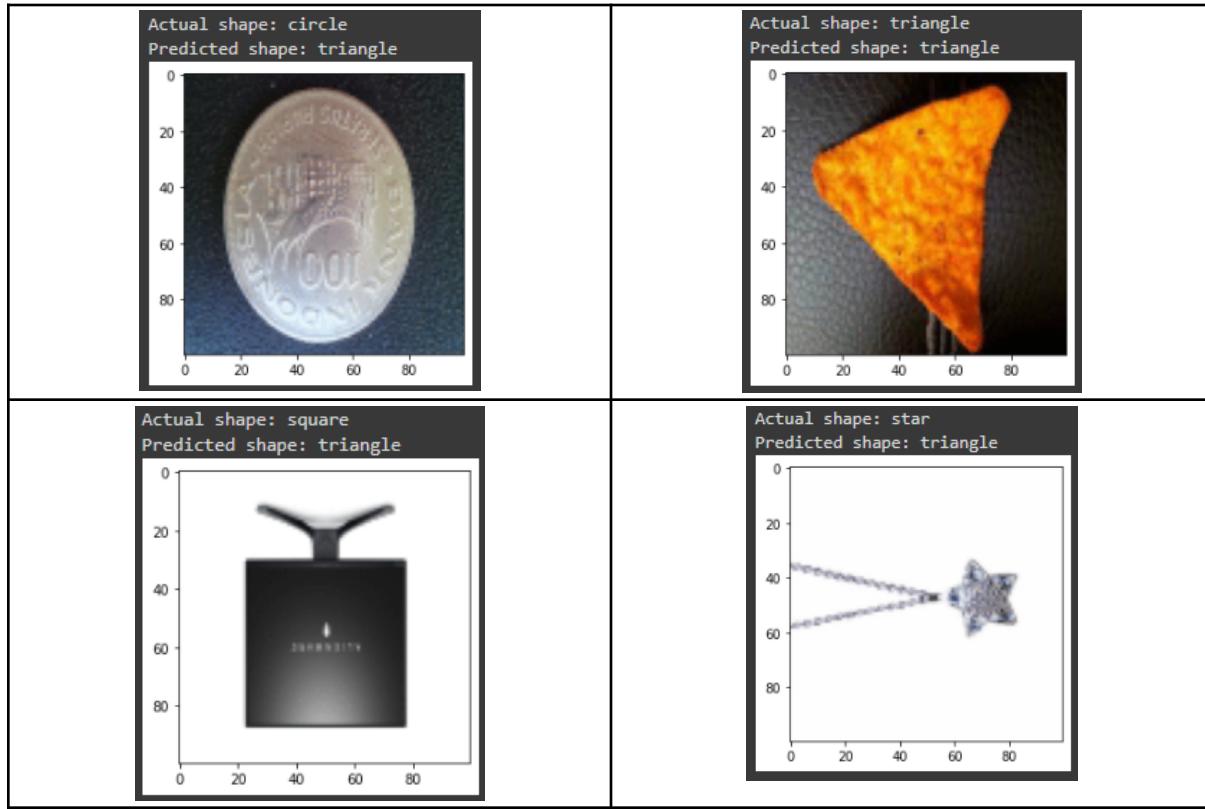
Epoch 20 dataset 1



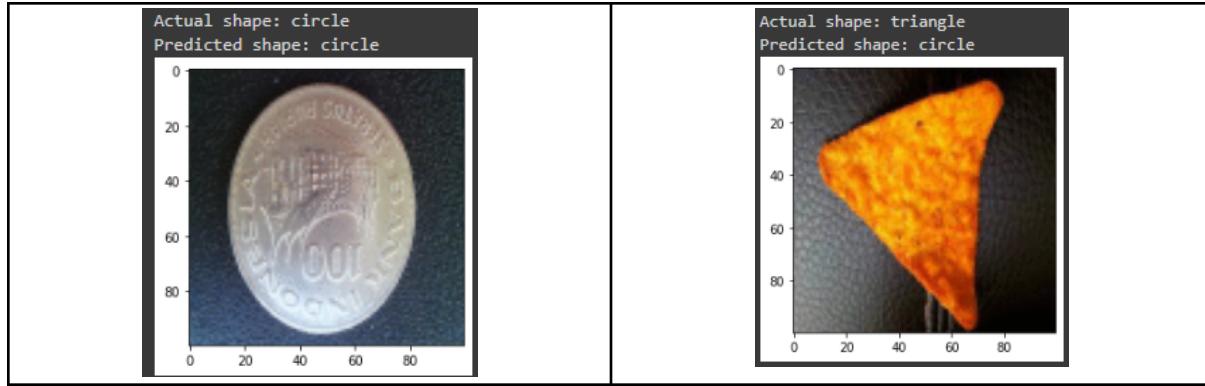
Epoch 50 dataset 1

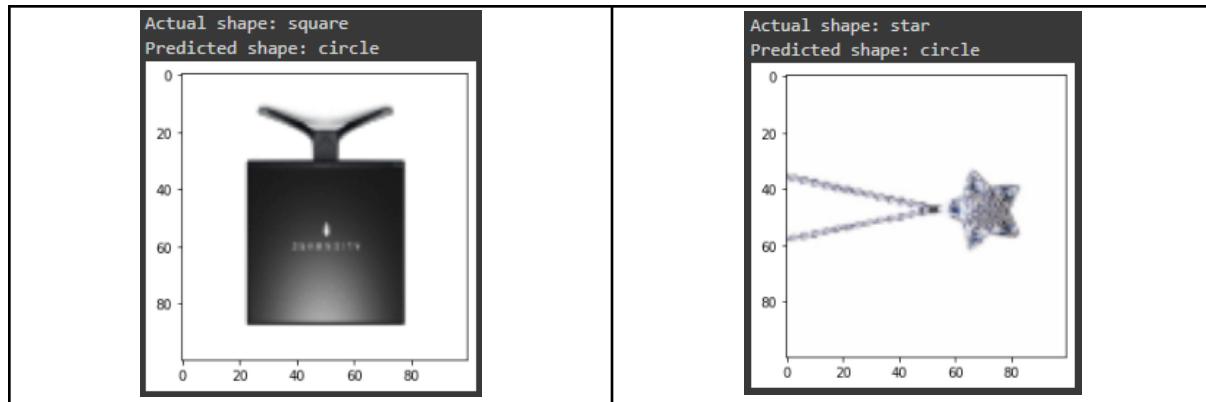


Epoch 10 dataset 2

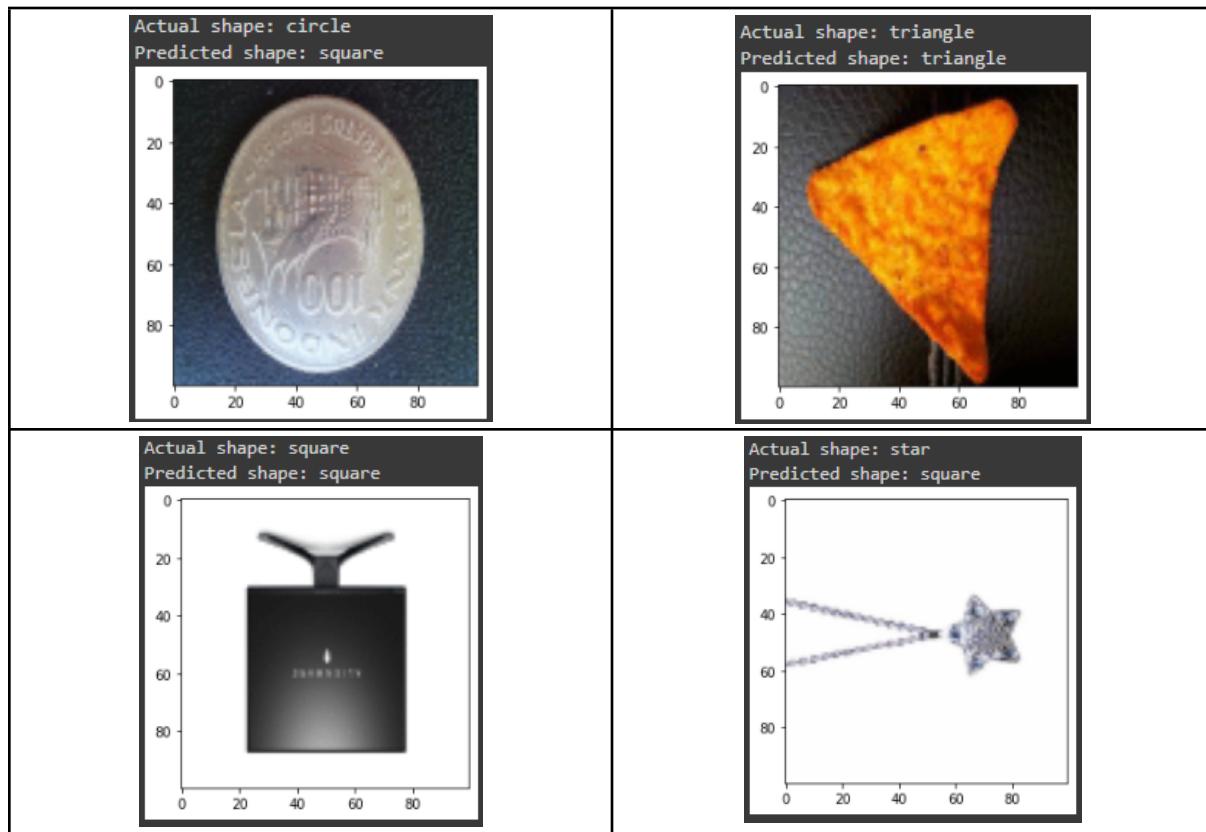


Epoch 20 dataset 2





Epoch 50 dataset 2



PENJELASAN DATASET

Dataset kami adalah gambar - gambar digital yang dibuat menggunakan software paint. Kita menggambar menggunakan brush yang ada di paint sehingga gambar yang dihasilkan tidak sempurna. Selain menggambar di paint kelompok kami juga menambahkan gambar - gambar foto dari yang nyata atau riil. Bentuk yang nyata ini kita ambil dari mengambil foto menggunakan smartphone atau mencari di google image. Kami juga mencoba berbagai batch size yang digunakan untuk membaca data dari dataset kami. Dataset ini juga diambil dengan kolaborasi dari kelompok Shape Recognition yang lain.

Berdasarkan hasil pengujian yang dilakukan kelompok kami, berkisar dari mengganti activation method yang digunakan, menambah dan mengurangi epoch, mengubah jumlah layer, semua ini mempunyai efek ke hasil akhir model yang dibuat. Dataset yang digunakan juga mempunyai peranan penting, dataset yang disediakan harus sesuai dengan tujuan mesin yang dibuat.

Pada awal kelompok menggunakan dataset yang berisi gambar berwarna hitam di background yang putih. Tetapi program akan diminta untuk melakukan prediksi pada gambar nyata yang mempunyai background, dan warna nya bukan hanya 1 warna saja. Karena itu saat pertama kelompok melakukan testing dari prediksi program cukup banyak prediksi yang salah.

HASIL PENGUJIAN DAN ANALISA

Kelompok kami memilih topik shape recognition yang bertujuan agar komputer bisa mengenali bentuk yang ditunjukkan.

Kelompok memberi dataset yang digunakan untuk bahan latihan yang akan digunakan oleh program untuk membuat sebuah model yang akan digunakan untuk melakukan sebuah prediksi yang akurat.

```
import os
import numpy as np
import tensorflow as tf
import keras
from keras.utils import np_utils
from keras.models import Sequential
from keras.layers import Conv2D,MaxPooling2D,Dense,Flatten,Dropout
from tensorflow.keras.callbacks import EarlyStopping
import matplotlib.pyplot as plt
from tensorflow.keras.layers import BatchNormalization
from tensorflow.keras.preprocessing.image import ImageDataGenerator
```

Kita mengimport library yang kita gunakan, disini kita menggunakan Numpy, Tensorflow, Keras, Matplotlib,

Kita menggunakan metode CNN untuk menentukan image nya, karena CNN adalah salah satu jenis neural network yang biasa digunakan pada data image. CNN bisa digunakan untuk mendeteksi dan mengenali objek pada sebuah image.

Jumlah Epoch yang kami gunakan adalah 20 epoch, 50 epoch, dan 10 epoch

Dari hasil percobaan menambah jumlah epoch yang kami, tidak terlihat perubahan yang signifikan di accuracy model kami. Dan masih ada beberapa image yang tidak bisa terdeteksi.

Pertama kita membaca file image nya

```
val_path="/content/shapes/shapes_real"
train_path="/content/shapes/shapes_real"

train_dataset =
tf.keras.preprocessing.image_dataset_from_directory(train_path,
seed=2509,

image_size=(100, 100),

batch_size=32)
val_dataset =
tf.keras.preprocessing.image_dataset_from_directory(val_path,
                                                    seed=2509,
image_size=(100, 100),
shuffle=False,
batch_size=32)
```

Setiap file yang di read di masukan kedalam image size 100x100, untuk image size nya ini bisa dirubah tetapi karena data set kami ukuran nya juga 100x100 jadi kami menggunakan 100x100.

KESIMPULAN

Kesimpulan yang ditarik dari analisa dan percobaan yang dilakukan kelompok adalah ada perlu nya kesesuaian dari data yang dikumpulkan dengan tujuan prediksi yang akan dilakukan. Ketika kelompok mencoba melakukan *training* dengan dataset gambar hitam putih dan diminta untuk menebak gambar riil tebakan program sering salah. Begitu juga dengan activation method yang digunakan, karena metode *relu* adalah metode yang sering dipakai bukan berarti itu adalah metode yang sesuai untuk semua situasi. Ketika dataset 2 di train menggunakan metode *relu* perubahan Accuracy yang terjadi tidak stabil Mesin tidak selalu tepat tetapi bisa dilatih untuk mendekatkan hingga akurasinya sempurna

ketentuan penggeraan proyek ML:

1. mempelajari dan mengerti teori dan code yang digunakan (setiap barisnya)
2. melakukan percobaan dengan mengganti jumlah epoch (tambah/kurang) (Epoch dikit gimana, Epoch banyak gimana)
3. melakukan percobaan seperti yang disepakati waktu pertemuan terakhir. (Layer ditambah kurangi (tambah bisa error), efek yang terlihat tidak besar), activation diganti ke sigmoid semua ngarah ke satu class (perlu sc" an dari uji coba e)
4. membuat laporan yang isinya
 - anggota kelompok dan pembagian kerja
 - teori termasuk model yang diujikan, penjelasan dataset
 - hasil pengujian + analisa
 - kesimpulan
5. membuat video demo yang menarik (untuk promosi)