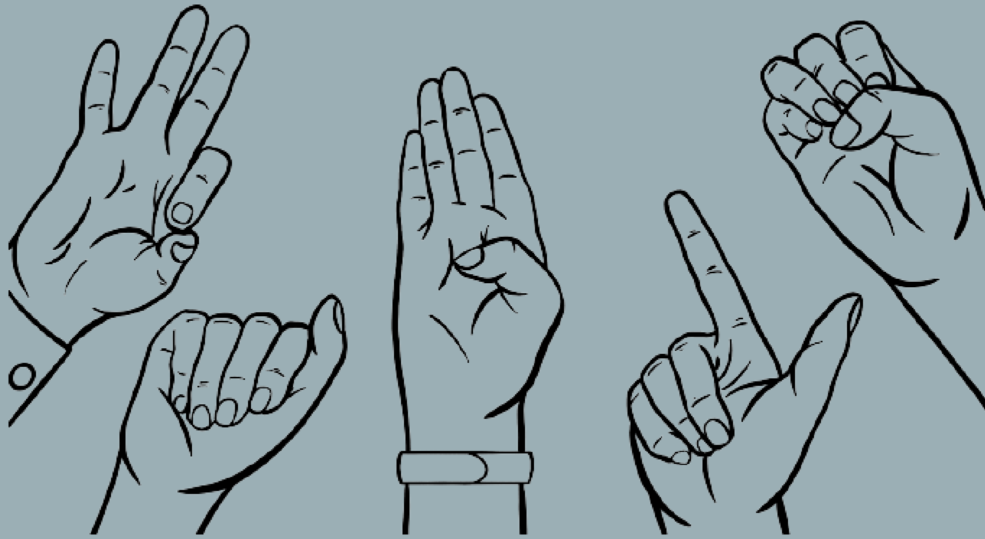
İŞARET DİLİ İŞLEME

Mustafa Uğur GÜL 203405031

Gökay HELVACI 203405033

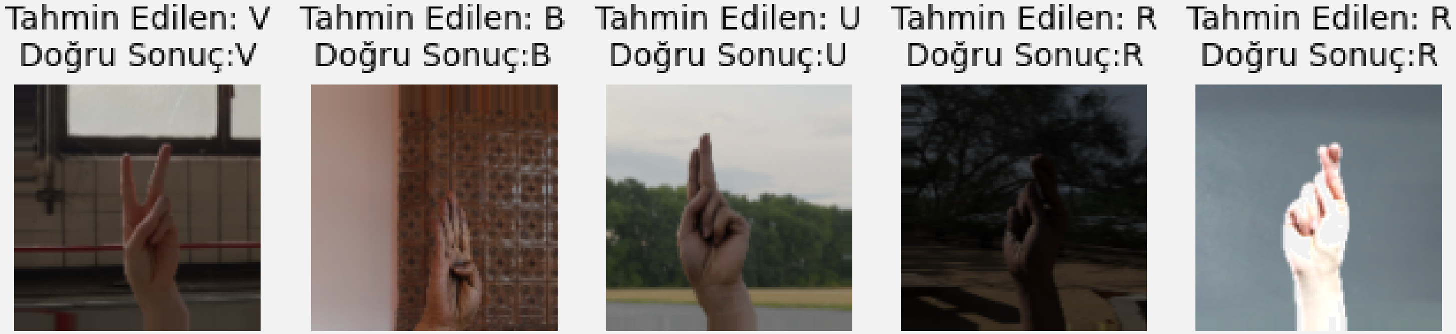


PROJENIN AMACI

- Projemizin temel amacı training klasöründeki işaret dili fotoğraflarla modelimizi eğitip test ederek modelimizi geliştirmek ve işaret dilini yazıya çevirerek işitme engelli bireylere yardımcı olmak.

PROJE ADIMLARI

- pandas, numpy, matplotlib, tenserflow, sklearn, time ve keras kütüphaneleri eklendi
- Modeli eğitmek için veri ekledik ve çoğalttık.
- Ardından elde bulunan işaret dili fotoğraflarıyla modeli eğittik
- Daha sonra eğittiğimiz modelin tutarlılığını test ettik.



Tahmin Edilen: V
Doğru Sonuç:V

Tahmin Edilen: B
Doğru Sonuç:B

Tahmin Edilen: U
Doğru Sonuç:U

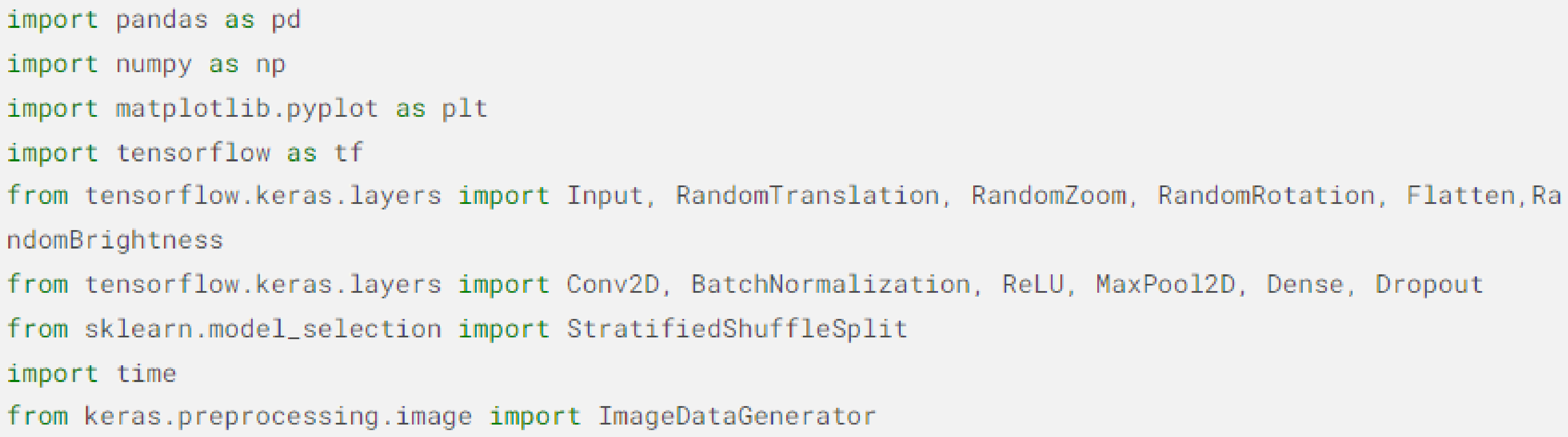
Tahmin Edilen: R
Doğru Sonuç:R

Tahmin Edilen: R
Doğru Sonuç:R

PROJEDE YARARLANILAN ALGORITMALAR

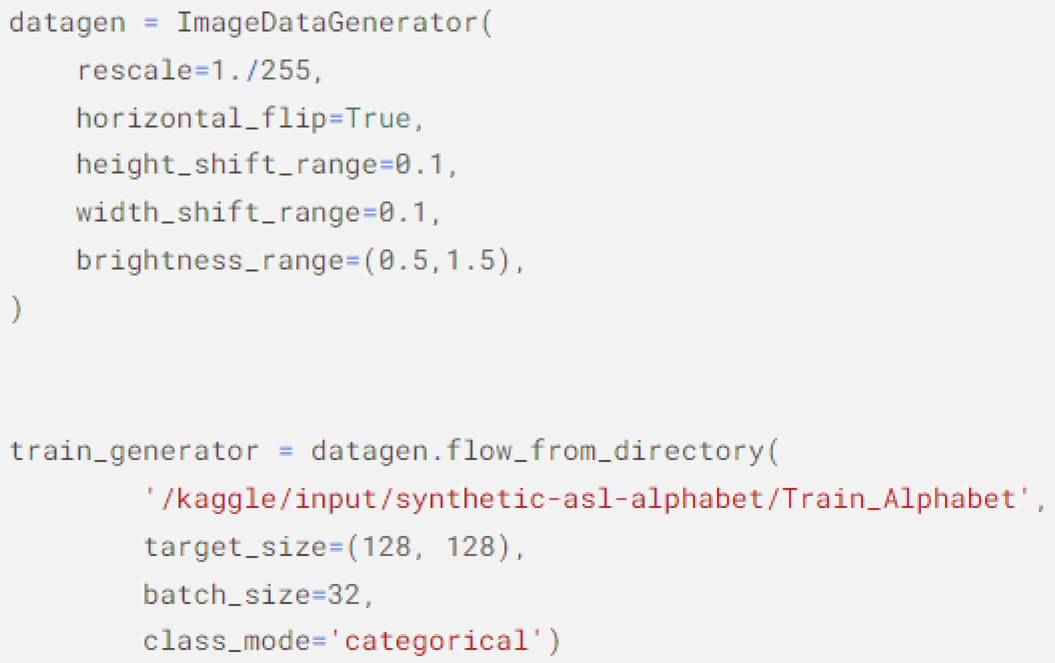
- Projede sklearn ve tenserflow kütüphaneleri başta olmak üzere pandas, numpy, matplotlib vb. kütüphaneler
- Veri ekleme ve çoğaltmada; "ImageDataGenerator" ve "atagen.flow_from_directory"
- Modelde "image_shape", "BatchNormalization()", "ReLU()", "BatchNormalization()"… fonksiyonları
- Train kısmında "Model()", "model.compile", "time.time()" fonksiyonları
- Test kısmında "datagen.flow_from_directory", "test_generator.next()", "np.argmax" fonksiyonları
- Modeli kaydetmede "model.save"
- Modeli yüklemede; "tf.keras.models.load_model"
- Yüklenen modeli test etmede "datagen.flow_from_directory", "plt.subplots" fonksiyonları kullanılmıştır.

KÜTÜPHANELER



```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import tensorflow as tf
from tensorflow.keras.layers import Input, RandomTranslation, RandomZoom, RandomRotation, Flatten,RandomBrightness
from tensorflow.keras.layers import Conv2D, BatchNormalization, ReLU, MaxPool2D, Dense, Dropout
from sklearn.model_selection import StratifiedShuffleSplit
import time
from keras.preprocessing.image import ImageDataGenerator
```

VERİ EKLEME VE ÇOĞALTMA



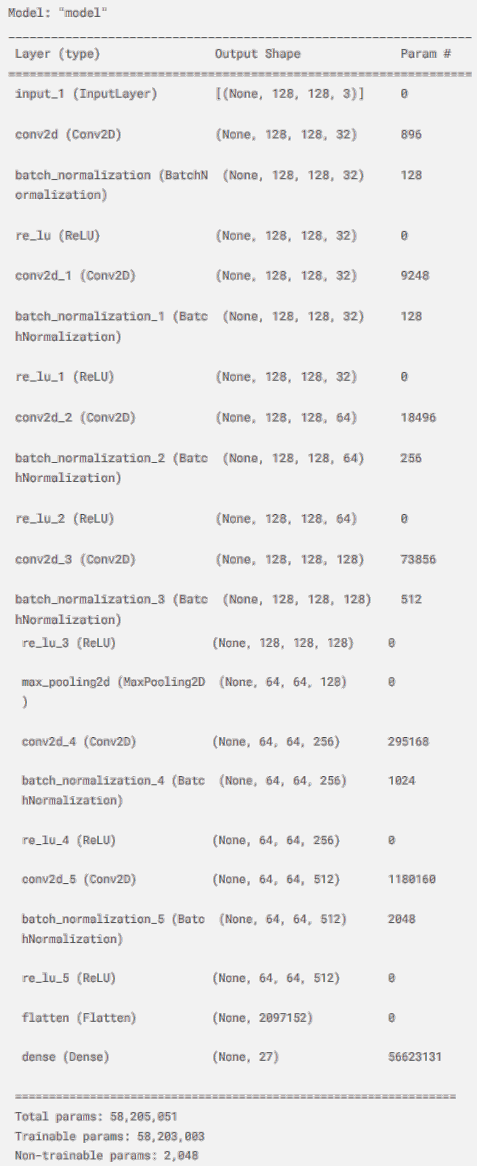
```
datagen = ImageDataGenerator(
    rescale=1./255,
    horizontal_flip=True,
    height_shift_range=0.1,
    width_shift_range=0.1,
    brightness_range=(0.5,1.5),
)


train_generator = datagen.flow_from_directory(
        '/kaggle/input/synthetic-asl-alphabet/Train_Alphabet',
        target_size=(128, 128),
        batch_size=32,
        class_mode='categorical')
```

MODEL

```python
def Model():
    image_shape = (128,128,3)

    input_tensor = Input(shape=image_shape)
    x = Conv2D(32, kernel_size=(3,3), strides=1, padding="same")(input_tensor)
    x = BatchNormalization()(x)
    x = ReLU()(x)
    x = Conv2D(32, kernel_size=(3,3), strides=1, padding="same")(x)
    x = BatchNormalization()(x)
    x = ReLU()(x)
    x = Conv2D(64, kernel_size=(3,3), strides=1, padding="same")(x)
    x = BatchNormalization()(x)
    x = ReLU()(x)
    x = Conv2D(128, kernel_size=(3,3), strides=1, padding="same")(x)
    x = BatchNormalization()(x)
    x = ReLU()(x)
    x = MaxPool2D(pool_size=2, strides=2, padding="valid")(x)
    x = Conv2D(256, kernel_size=(3,3), strides=1, padding="same")(x)
    x = BatchNormalization()(x)
    x = ReLU()(x)
    x = Conv2D(512, kernel_size=(3,3), strides=1, padding="same")(x)
    x = BatchNormalization()(x)
    x = ReLU()(x)
    x = Flatten()(x)

    out = Dense(27, activation='softmax')(x)

    return tf.keras.Model(inputs=input_tensor,outputs=out)
```

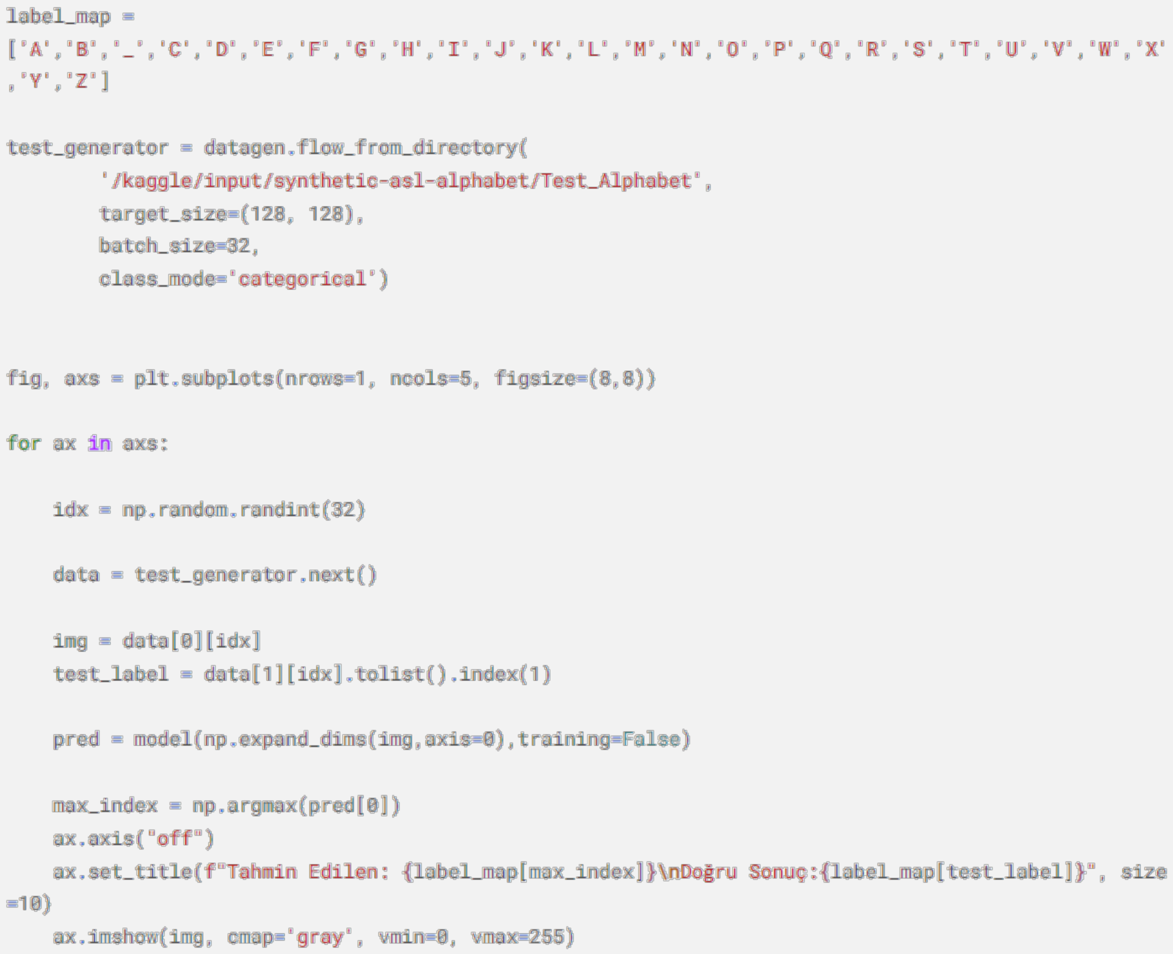
TRAİN

```
model = Model()

model.summary()

model.compile(loss="categorical_crossentropy", optimizer=tf.keras.optimizers.Adam(0.0005), metrics=
["accuracy"])

start_train = time.time()
history = model.fit(train_generator, epochs=15,batch_size=32)
end_train = time.time()
print(end_train - start_train)
```



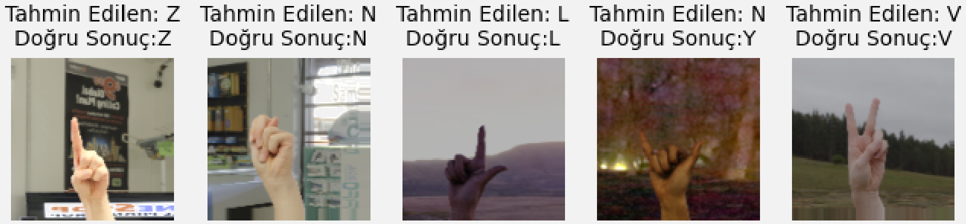
```
Model: "model"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input_1 (InputLayer)        [(None, 128, 128, 3)]     0

 conv2d (Conv2D)             (None, 128, 128, 32)      896

 batch_normalization (BatchN  (None, 128, 128, 32)     128
 ormalization)

 re_lu (ReLU)                (None, 128, 128, 32)      0

 conv2d_1 (Conv2D)           (None, 128, 128, 32)      9248

 batch_normalization_1 (Batc  (None, 128, 128, 32)     128
 hNormalization)

 re_lu_1 (ReLU)              (None, 128, 128, 32)      0

 conv2d_2 (Conv2D)           (None, 128, 128, 64)      18496

 batch_normalization_2 (Batc  (None, 128, 128, 64)     256
 hNormalization)

 re_lu_2 (ReLU)              (None, 128, 128, 64)      0

 conv2d_3 (Conv2D)           (None, 128, 128, 128)     73856

 batch_normalization_3 (Batc  (None, 128, 128, 128)    512
 hNormalization)
  re_lu_3 (ReLU)             (None, 128, 128, 128)    0

  max_pooling2d (MaxPooling2D  (None, 64, 64, 128)    0
  )

  conv2d_4 (Conv2D)          (None, 64, 64, 256)      295168

  batch_normalization_4 (Batc  (None, 64, 64, 256)    1024
  hNormalization)

  re_lu_4 (ReLU)             (None, 64, 64, 256)      0

  conv2d_5 (Conv2D)          (None, 64, 64, 512)      1180160

  batch_normalization_5 (Batc  (None, 64, 64, 512)    2048
  hNormalization)

  re_lu_5 (ReLU)             (None, 64, 64, 512)      0

  flatten (Flatten)          (None, 2097152)          0

  dense (Dense)              (None, 27)               56623131

 =================================================================
 Total params: 58,205,051
 Trainable params: 58,203,003
 Non-trainable params: 2,048
```

TEST

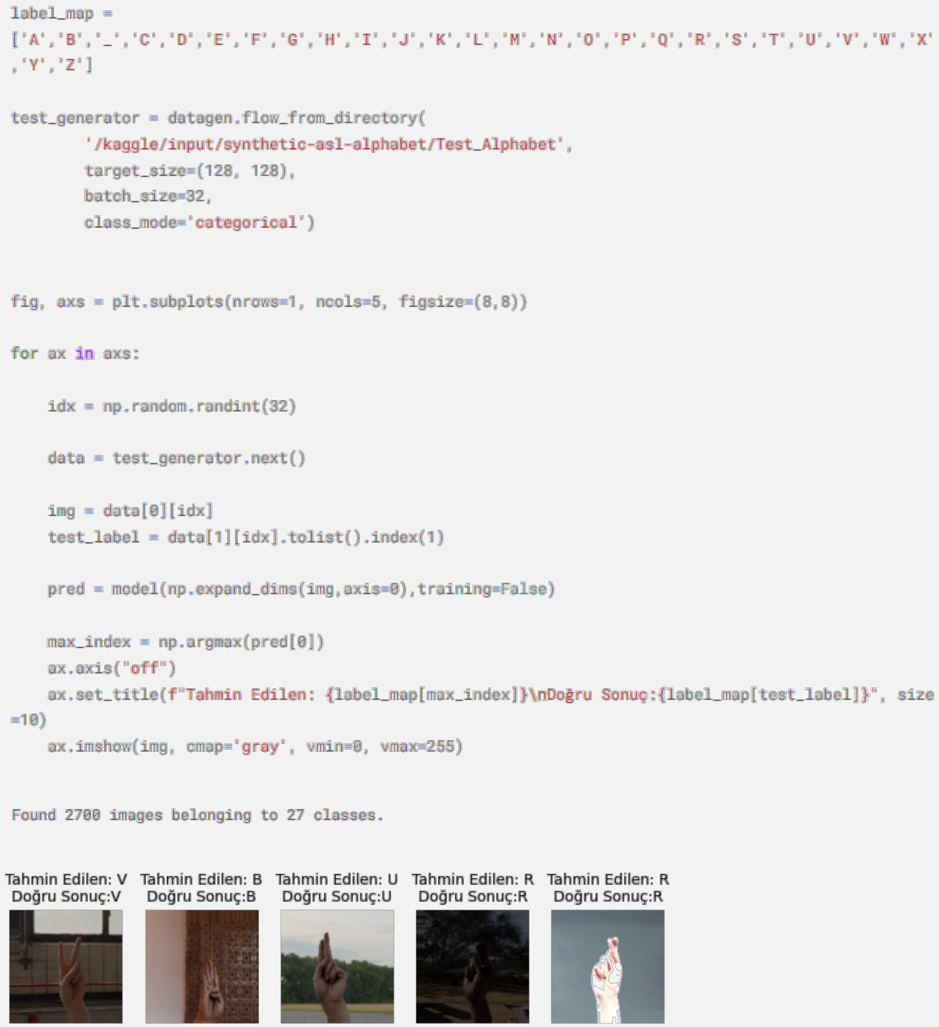


```
label_map =
['A','B','_','C','D','E','F','G','H','I','J','K','L','M','N','O','P','Q','R','S','T','U','V','W','X'
,'Y','Z']

test_generator = datagen.flow_from_directory(
        '/kaggle/input/synthetic-asl-alphabet/Test_Alphabet',
        target_size=(128, 128),
        batch_size=32,
        class_mode='categorical')


fig, axs = plt.subplots(nrows=1, ncols=5, figsize=(8,8))

for ax in axs:

    idx = np.random.randint(32)

    data = test_generator.next()

    img = data[0][idx]
    test_label = data[1][idx].tolist().index(1)

    pred = model(np.expand_dims(img,axis=0),training=False)

    max_index = np.argmax(pred[0])
    ax.axis("off")
    ax.set_title(f"Tahmin Edilen: {label_map[max_index]}\nDoğru Sonuç:{label_map[test_label]}", size
=10)
    ax.imshow(img, cmap='gray', vmin=0, vmax=255)
```

```
Found 2700 images belonging to 27 classes.
```



MODELİ KAYDETME, YÜKLEME VE TEST ETME



```
label_map =
['A','B','_','C','D','E','F','G','H','I','J','K','L','M','N','O','P','Q','R','S','T','U','V','W','X'
,'Y','Z']

test_generator = datagen.flow_from_directory(
        '/kaggle/input/synthetic-asl-alphabet/Test_Alphabet',
        target_size=(128, 128),
        batch_size=32,
        class_mode='categorical')


fig, axs = plt.subplots(nrows=1, ncols=5, figsize=(8,8))

for ax in axs:

    idx = np.random.randint(32)

    data = test_generator.next()

    img = data[0][idx]
    test_label = data[1][idx].tolist().index(1)

    pred = model(np.expand_dims(img,axis=0),training=False)

    max_index = np.argmax(pred[0])
    ax.axis("off")
    ax.set_title(f"Tahmin Edilen: {label_map[max_index]}\nDoğru Sonuç:{label_map[test_label]}", size
=10)
    ax.imshow(img, cmap='gray', vmin=0, vmax=255)


Found 2700 images belonging to 27 classes.
```

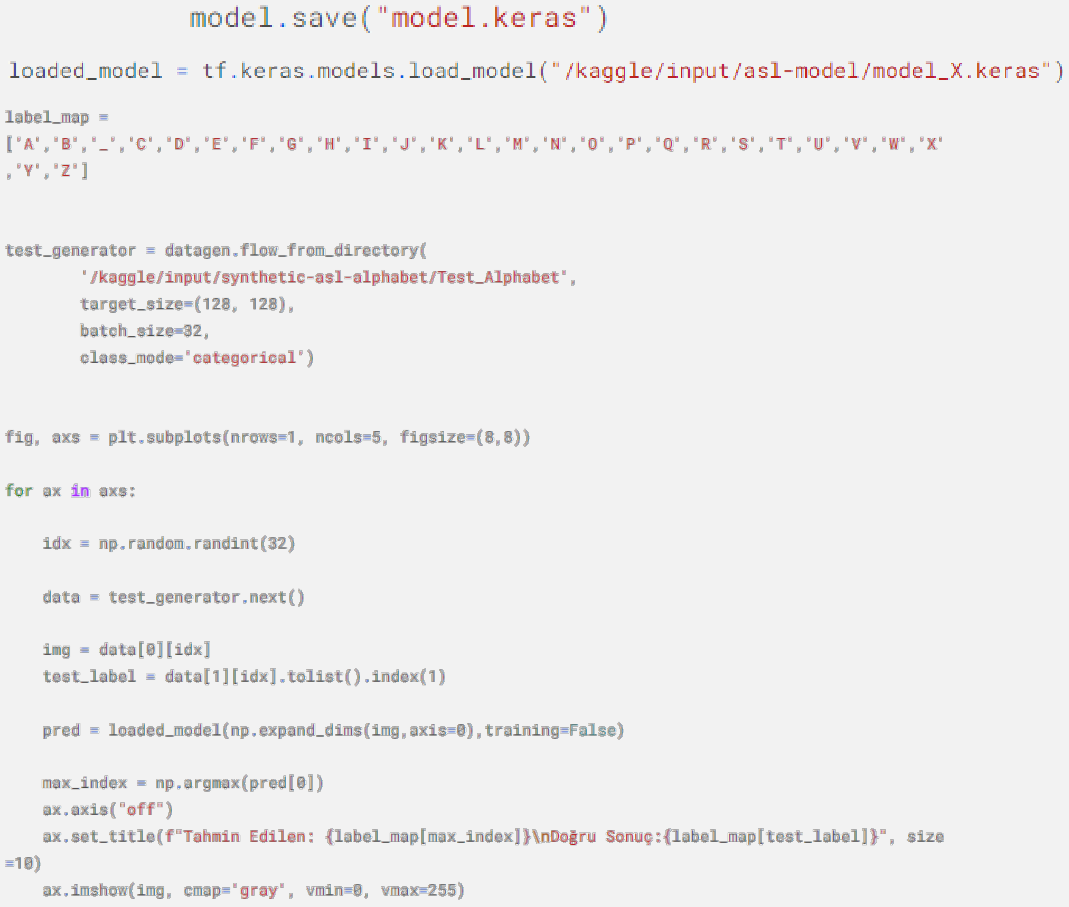
Tahmin Edilen: V
Doğru Sonuç:V

Tahmin Edilen: B
Doğru Sonuç:B

Tahmin Edilen: U
Doğru Sonuç:U

Tahmin Edilen: R
Doğru Sonuç:R

Tahmin Edilen: R
Doğru Sonuç:R



```
model.save("model.keras")

loaded_model = tf.keras.models.load_model("/kaggle/input/asl-model/model_X.keras")
```

```
label_map =
['A','B','_','C','D','E','F','G','H','I','J','K','L','M','N','O','P','Q','R','S','T','U','V','W','X'
,'Y','Z']


test_generator = datagen.flow_from_directory(
        '/kaggle/input/synthetic-asl-alphabet/Test_Alphabet',
        target_size=(128, 128),
        batch_size=32,
        class_mode='categorical')


fig, axs = plt.subplots(nrows=1, ncols=5, figsize=(8,8))

for ax in axs:

    idx = np.random.randint(32)

    data = test_generator.next()

    img = data[0][idx]
    test_label = data[1][idx].tolist().index(1)

    pred = loaded_model(np.expand_dims(img,axis=0),training=False)

    max_index = np.argmax(pred[0])
    ax.axis("off")
    ax.set_title(f"Tahmin Edilen: {label_map[max_index]}\nDoğru Sonuç:{label_map[test_label]}", size
=10)
    ax.imshow(img, cmap='gray', vmin=0, vmax=255)
```

BAĞLANTILAR

- https://www.kaggle.com/code/mustafagull/sign-language-classification/
- https://www.kaggle.com/datasets/lexset/synthetic-asl-alphabet
- https://github.com/GokayHELVACI/IsaretDili