Assignment:

Project: American Sign language detection, Dataset Link : https://www.kaggle.com/grassknoted/asl-alphabet

About the dataset- The data set is a collection of images of alphabets from the American Sign Language, separated in 29 folders which represent the various classes.

The training data set contains 87,000 images which are 200x200 pixels. There are 29 classes, of which 26 are for the letters A-Z and 3 classes for SPACE, DELETE and NOTHING. These 3 classes are very helpful in real-time applications, and classification. The test data set contains a mere 29 images, to encourage the use of real-world test images.

```
1 from google.colab import files
2 files.upload()
```

Choose Files kaggle.json

- **kaggle.json**(application/json) - 63 bytes, last modified: 9/2/2022 - 100% done

```
Saving kaggle.json to kaggle.json
{'kaggle.json': b'{"username":"munch17","key":"3483d3b44d34c3711d8fb0220a82d2bf"}'}
```

```
1 !ls -lha kaggle.json
2
```

```
-rw-r--r-- 1 root root 63 Sep 12 13:44 kaggle.json
```

```
1 !pip install -q kaggle                          # installing the kaggle package
```

```
1 !mkdir -p ~/.kaggle                             # creating .kaggle folder where the k
```

```
1 !cp kaggle.json ~/.kaggle/                      # move the key to the folder
```

```
1 !pwd                                           # checking the present working directo
```

```
/content
```

```
1 !chmod 600 ~/.kaggle/kaggle.json
```

```
1 !kaggle datasets download -d grassknoted/asl-alphabet
```

```
Downloading asl-alphabet.zip to /content
100% 1.02G/1.03G [00:04<00:00, 231MB/s]
100% 1.03G/1.03G [00:04<00:00, 220MB/s]
```

```
!unzip asl-alphabet.zip
```

```
  inflating: asl_alphabet_train/asl_alphabet_train/space/space947.jpg
  inflating: asl_alphabet_train/asl_alphabet_train/space/space948.jpg
  inflating: asl_alphabet_train/asl_alphabet_train/space/space949.jpg
  inflating: asl_alphabet_train/asl_alphabet_train/space/space95.jpg
  inflating: asl_alphabet_train/asl_alphabet_train/space/space950.jpg
  inflating: asl_alphabet_train/asl_alphabet_train/space/space951.jpg
  inflating: asl_alphabet_train/asl_alphabet_train/space/space952.jpg
  inflating: asl_alphabet_train/asl_alphabet_train/space/space953.jpg
  inflating: asl_alphabet_train/asl_alphabet_train/space/space954.jpg
  inflating: asl_alphabet_train/asl_alphabet_train/space/space955.jpg
  inflating: asl_alphabet_train/asl_alphabet_train/space/space956.jpg
  inflating: asl_alphabet_train/asl_alphabet_train/space/space957.jpg
  inflating: asl_alphabet_train/asl_alphabet_train/space/space958.jpg
  inflating: asl_alphabet_train/asl_alphabet_train/space/space959.jpg
  inflating: asl_alphabet_train/asl_alphabet_train/space/space96.jpg
  inflating: asl_alphabet_train/asl_alphabet_train/space/space960.jpg
  inflating: asl_alphabet_train/asl_alphabet_train/space/space961.jpg
  inflating: asl_alphabet_train/asl_alphabet_train/space/space962.jpg
  inflating: asl_alphabet_train/asl_alphabet_train/space/space963.jpg
  inflating: asl_alphabet_train/asl_alphabet_train/space/space964.jpg
  inflating: asl_alphabet_train/asl_alphabet_train/space/space965.jpg
  inflating: asl_alphabet_train/asl_alphabet_train/space/space966.jpg
  inflating: asl_alphabet_train/asl_alphabet_train/space/space967.jpg
  inflating: asl_alphabet_train/asl_alphabet_train/space/space968.jpg
  inflating: asl_alphabet_train/asl_alphabet_train/space/space969.jpg
  inflating: asl_alphabet_train/asl_alphabet_train/space/space97.jpg
  inflating: asl_alphabet_train/asl_alphabet_train/space/space970.jpg
  inflating: asl_alphabet_train/asl_alphabet_train/space/space971.jpg
  inflating: asl_alphabet_train/asl_alphabet_train/space/space972.jpg
  inflating: asl_alphabet_train/asl_alphabet_train/space/space973.jpg
  inflating: asl_alphabet_train/asl_alphabet_train/space/space974.jpg
  inflating: asl_alphabet_train/asl_alphabet_train/space/space975.jpg
  inflating: asl_alphabet_train/asl_alphabet_train/space/space976.jpg
  inflating: asl_alphabet_train/asl_alphabet_train/space/space977.jpg
  inflating: asl_alphabet_train/asl_alphabet_train/space/space978.jpg
  inflating: asl_alphabet_train/asl_alphabet_train/space/space979.jpg
  inflating: asl_alphabet_train/asl_alphabet_train/space/space98.jpg
  inflating: asl_alphabet_train/asl_alphabet_train/space/space980.jpg
  inflating: asl_alphabet_train/asl_alphabet_train/space/space981.jpg
  inflating: asl_alphabet_train/asl_alphabet_train/space/space982.jpg
  inflating: asl_alphabet_train/asl_alphabet_train/space/space983.jpg
  inflating: asl_alphabet_train/asl_alphabet_train/space/space984.jpg
  inflating: asl_alphabet_train/asl_alphabet_train/space/space985.jpg
  inflating: asl_alphabet_train/asl_alphabet_train/space/space986.jpg
  inflating: asl_alphabet_train/asl_alphabet_train/space/space987.jpg
  inflating: asl_alphabet_train/asl_alphabet_train/space/space988.jpg
  inflating: asl_alphabet_train/asl_alphabet_train/space/space989.jpg
  inflating: asl_alphabet_train/asl_alphabet_train/space/space99.jpg
  inflating: asl_alphabet_train/asl_alphabet_train/space/space990.jpg
  inflating: asl_alphabet_train/asl_alphabet_train/space/space991.jpg
  inflating: asl_alphabet_train/asl_alphabet_train/space/space992.jpg
  inflating: asl_alphabet_train/asl_alphabet_train/space/space993.jpg
  inflating: asl_alphabet_train/asl_alphabet_train/space/space994.jpg
  inflating: asl_alphabet_train/asl_alphabet_train/space/space995.jpg
  inflating: asl_alphabet_train/asl_alphabet_train/space/space996.jpg
  inflating: asl_alphabet_train/asl_alphabet_train/space/space997.jpg
  inflating: asl_alphabet_train/asl_alphabet_train/space/space998.jpg
  inflating: asl_alphabet_train/asl_alphabet_train/space/space999.jpg
```

data is ready. lets start with some preprocessing

```
# Import the libraries
import os
import numpy as np
import cv2
import matplotlib.pyplot as plt
import tensorflow as tf

```

getting the trainig and testing data from the train dir

```
# assigning the batch size and image size
batch_size= 32
img_ht= 224
img_wd= 224

# getting the training data
train_ds= tf.keras.utils.image_dataset_from_directory('../content/asl_alphabet_train/as
    validation_split= 0.2,
    subset= 'training',
    seed= 123,
    image_size= (img_ht, img_wd),
    batch_size= batch_size)

```

```
Found 87000 files belonging to 29 classes.
Using 69600 files for training.
```

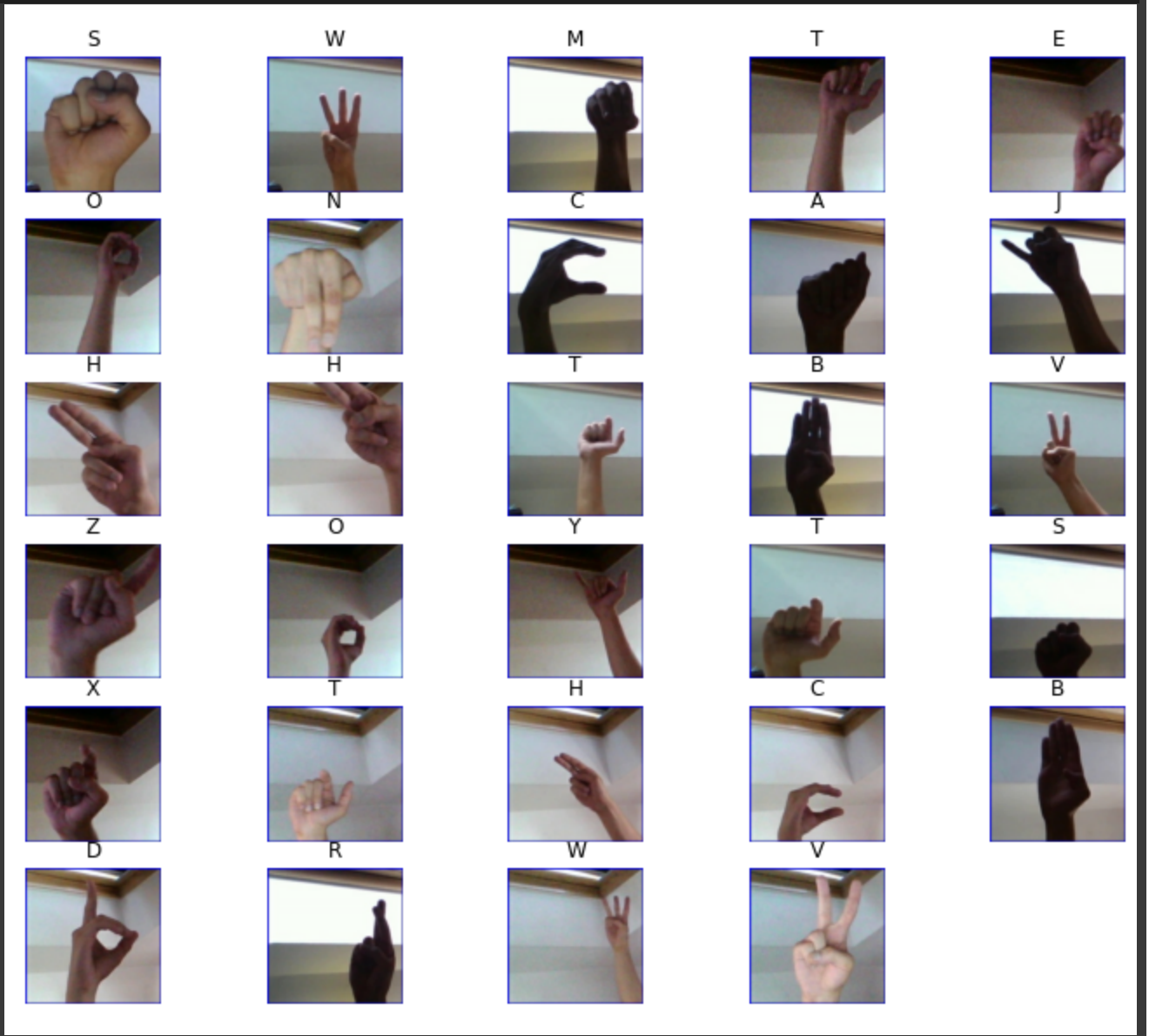
```
# getting the testing data

test_ds= tf.keras.utils.image_dataset_from_directory('../content/asl_alphabet_train/asl
           validation_split= 0.2,
           subset= 'validation',
           seed= 123,
           image_size= (img_ht, img_wd),
           batch_size= batch_size)

```

```
Found 87000 files belonging to 29 classes.
Using 17400 files for validation.
```

```
class_names= train_ds.class_names
print('Class names: ', class_names)
print('Total classes: ', len(class_names))

```

```
Class names:  ['A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J', 'K', 'L', 'M', 'N',
Total classes:  29
```

```
# lets display random samples from training set
plt.figure(figsize=(12,10))
for images, labels in train_ds.take(1):
  for i in range(29):
    ax= plt.subplot(6,5, i+1)
    plt.imshow(images[i].numpy().astype("uint8"))
    plt.title(class_names[labels[i]])
    plt.axis("off")                                  # after displaying each i image ax


```




```
# x= []
# y= []

# for images, labels in train_ds:
#   x.append(images)
#   y.append(labels)
# # want to extract x and y from train_ds but colab is crashing again and again as teh
```

Modeling

```
from tensorflow.keras import Sequential, layers
from tensorflow.keras.layers import Conv2D, MaxPooling2D, Flatten, Dense

model = Sequential([
  layers.Rescaling(1./255, input_shape=(img_ht, img_wd, 3)),            # rescaling x(i
  layers.Conv2D(16, 3, padding='same', activation='relu'),
  layers.MaxPooling2D(),
  layers.Conv2D(32, 3, padding='same', activation='relu'),
  layers.MaxPooling2D(),
  layers.Conv2D(64, 3, padding='same', activation='relu'),
  layers.MaxPooling2D(),
  layers.Flatten(),
  layers.Dense(128, activation='relu'),
  layers.Dense(29,activation='softmax')
])
# model.summary()

```

```
# compiling the model
model.compile(loss='sparse_categorical_crossentropy',optimizer='adam',metrics=['accurac

```

```
# Will save the model behaviour in history so that we can later see the accuracy and lo

history = model.fit(train_ds, batch_size=32,validation_batch_size=32, validation_data=

# Stopping the epoch as the data is too much and is taking time.
```

```
======] - 125s 52ms/step - loss: 0.8590 - accuracy: 0.7340 - val_loss: 0.2425 - val_a

======] - 115s 53ms/step - loss: 0.1076 - accuracy: 0.9637 - val_loss: 0.1043 - val_a
```

```
#  lets get actual labels and predicted labels is the right format

actual = []
pred = []
for images, labels in test_ds:
    for i in range(0, len(images)):
        image = images[i]
        image = np.expand_dims(image, axis=0)
        result = model.predict(image)
        pred.append(class_names[np.argmax(result)])
        actual.append(class_names[labels[i].numpy()])
```

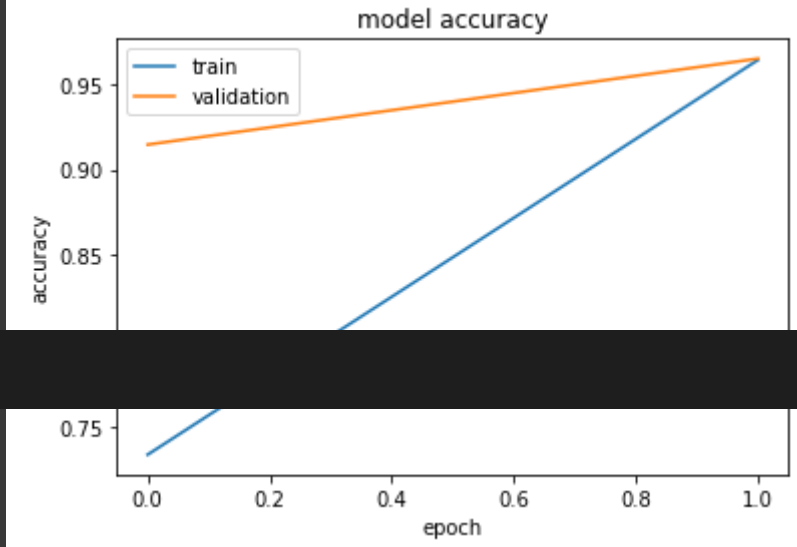
```
actual[:5], pred[:5]                                    # comparing test and predict
```

```
(['A', 'space', 'H', 'P', 'S'], ['A', 'space', 'H', 'P', 'S'])
```

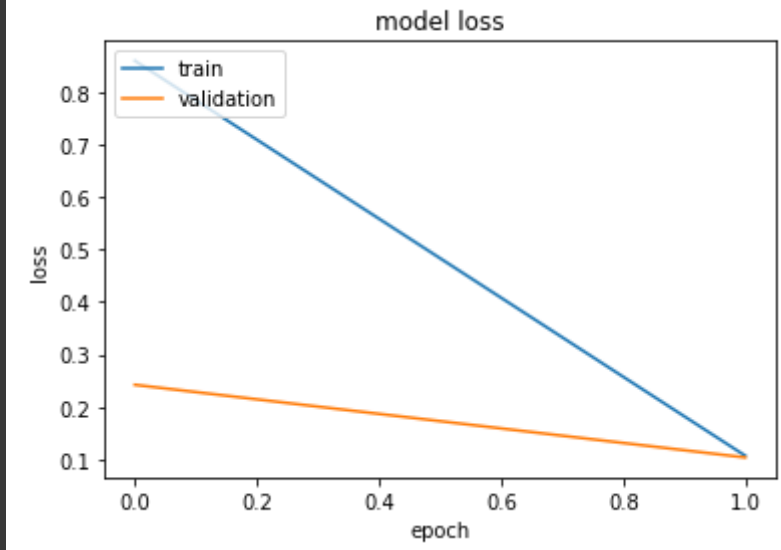
```
from sklearn.metrics import f1_score, accuracy_score

accurcay= f1_score(actual, pred, average= 'weighted')
print('F1_score accuaracy: ', accurcay)

print("Test accuracy=",accuracy_score(actual, pred))
```

```
F1_score accuaracy:  0.9645236806084252
Test accuracy= 0.9644827586206897
```

```
# Plotting the accuracy and loss for both training and validation(tesiting) data

plt.plot(history.history['accuracy'])
plt.plot(history.history['val_accuracy'])
plt.title('model accuracy')
plt.ylabel('accuracy')
plt.xlabel('epoch')
plt.legend(['train', 'validation'], loc='upper left')
plt.show()


plt.plot(history.history['loss'])
plt.plot(history.history['val_loss'])
plt.title('model loss')
plt.ylabel('loss')
plt.xlabel('epoch')
plt.legend(['train', 'validation'], loc='upper left')
plt.show()
```




```
1
```



Colab paid products - Cancel contracts here