

```
1 # TensorFlow and tf.keras
2 import tensorflow as tf
3
4 # Helper libraries
5 import numpy as np
6 import matplotlib.pyplot as plt
7
8 print(tf.__version__)
```

2.12.0

```
1 !unzip /content/GurNum.zip
```

```
Archive: /content/GurNum.zip
  inflating: GurNum/val/5/14.tiff
  inflating: GurNum/val/5/13.tiff
  inflating: GurNum/val/5/9.tiff
  inflating: GurNum/val/5/18.tiff
  inflating: GurNum/val/5/20.tiff
  inflating: GurNum/val/5/12.tiff
  inflating: GurNum/val/5/7.tiff
  inflating: GurNum/val/5/17.tiff
  inflating: GurNum/val/1/9.tiff
  inflating: GurNum/val/1/6.tiff
  inflating: GurNum/val/1/5.tiff
  inflating: GurNum/val/5/11.tiff
  inflating: GurNum/val/5/5.tiff
  inflating: GurNum/val/1/16.tiff
  inflating: GurNum/val/1/12.tiff
  inflating: GurNum/val/1/8.tiff
  inflating: GurNum/val/5/4.tiff
  inflating: GurNum/val/5/6.tiff
  inflating: GurNum/val/5/8.tiff
  inflating: GurNum/val/5/19.tiff
  inflating: GurNum/val/5/2.tiff
  inflating: GurNum/val/1/11.tiff
  inflating: GurNum/val/1/2.tiff
  inflating: GurNum/val/1/18.tiff
  inflating: GurNum/val/1/13.tiff
  inflating: GurNum/val/5/16.tiff
  inflating: GurNum/val/5/10.tiff
  inflating: GurNum/val/5/1.tiff
  inflating: GurNum/val/1/1.tiff
  inflating: GurNum/val/1/10.tiff
  inflating: GurNum/val/8/15.tiff
  inflating: GurNum/val/8/3.tiff
  inflating: GurNum/val/1/19.tiff
  inflating: GurNum/val/1/3.tiff
  inflating: GurNum/val/1/17.tiff
  inflating: GurNum/val/1/7.tiff
  inflating: GurNum/val/8/7.tiff
  inflating: GurNum/val/8/19.tiff
  inflating: GurNum/val/8/6.tiff
  inflating: GurNum/val/8/2.tiff
  inflating: GurNum/val/8/16.tiff
  inflating: GurNum/val/8/13.tiff
```

✓ 0s completed at 12:02 AM



```
inflating: GurNum/val/8/1.tiff
inflating: GurNum/val/8/4.tiff
inflating: GurNum/val/8/20.tiff
inflating: GurNum/val/8/18.tiff
inflating: GurNum/val/8/8.tiff
inflating: GurNum/val/8/17.tiff
inflating: GurNum/val/8/5.tiff
inflating: GurNum/val/8/12.tiff
inflating: GurNum/val/8/14.tiff
inflating: GurNum/val/4/17.tiff
inflating: GurNum/val/4/14.tiff
inflating: GurNum/val/4/12.tiff
inflating: GurNum/val/4/5.tiff
inflating: GurNum/val/4/15.tiff
```

```
1 from PIL import Image
2 import os
3
4 path = "GurNum/train/1/"
5 images= os.listdir(path)
6 for img in images:
7     Image.open('GurNum/train/1/'+img).save(os.path.join(path+ str(img).replace(".bmp
8     os.remove(os.path.join(path + img))
9
10 path = "GurNum/train/2/"
11 images= os.listdir(path)
12 for img in images:
13     Image.open('GurNum/train/2/'+img).save(os.path.join(path+ str(img).replace(".bmp
14     os.remove(os.path.join(path + img))
15 path = "GurNum/train/3/"
16 images= os.listdir(path)
17 for img in images:
18     Image.open('GurNum/train/3/'+img).save(os.path.join(path+ str(img).replace(".bmp
19     os.remove(os.path.join(path + img))
20 path = "GurNum/train/4/"
21 images= os.listdir(path)
22 for img in images:
23     Image.open('GurNum/train/4/'+img).save(os.path.join(path+ str(img).replace(".bmp
24     os.remove(os.path.join(path + img))
25 path = "GurNum/train/5/"
26 images= os.listdir(path)
27 for img in images:
28     Image.open('GurNum/train/5/'+img).save(os.path.join(path+ str(img).replace(".bmp
29     os.remove(os.path.join(path + img))
30
31 path = "GurNum/train/6/"
32 images= os.listdir(path)
33 for img in images:
34     Image.open('GurNum/train/6/'+img).save(os.path.join(path+ str(img).replace(".bmp
35     os.remove(os.path.join(path + img))
36 path = "GurNum/train/7/"
37 images= os.listdir(path)
38 for img in images:
39     Image.open('GurNum/train/7/'+img).save(os.path.join(path+ str(img).replace(".bmp
```

```
40     os.remove(os.path.join(path + img))
41 path = "GurNum/train/8/"
42 images= os.listdir(path)
43 for img in images:
44     Image.open('GurNum/train/8/'+img).save(os.path.join(path+ str(img).replace(".bmp
45     os.remove(os.path.join(path + img))
46
47 path = "GurNum/train/9/"
48 images= os.listdir(path)
49 for img in images:
50     Image.open('GurNum/train/9/'+img).save(os.path.join(path+ str(img).replace(".bmp
51     os.remove(os.path.join(path + img))
52
53 path = "GurNum/train/0/"
54 images= os.listdir(path)
55 for img in images:
56     Image.open('GurNum/train/0/'+img).save(os.path.join(path+ str(img).replace(".bmp
57     os.remove(os.path.join(path + img))
```

```
1 import pathlib
2 data_dir = pathlib.Path('GurNum/train').with_suffix('')
```

```
1 batch_size = 32
2 img_height = 32
3 img_width = 32
```

```
1 train_ds = tf.keras.utils.image_dataset_from_directory(
2     data_dir,
3     color_mode='grayscale',
4     seed=123,
5     image_size=(img_height, img_width),
6     batch_size=batch_size,
7     validation_split=0.2,
8     subset="training")
```

Found 1000 files belonging to 10 classes.
Using 800 files for training.

```
1 val_ds = tf.keras.utils.image_dataset_from_directory(
2     data_dir,
3     validation_split=0.2,
4     subset="validation",
5     seed=123,
6     image_size=(img_height, img_width),
7     batch_size=batch_size,
8     color_mode='grayscale',
9 )
```

Found 1000 files belonging to 10 classes.
Using 200 files for validation.

```
1 class_names = train_ds.class_names
2 print(class_names)

['0', '1', '2', '3', '4', '5', '6', '7', '8', '9']
```

```
1 normalization_layer = tf.keras.layers.Rescaling(1./255)
```

```
1 for image_batch, labels_batch in train_ds:
2     print(image_batch.shape)
3     print(labels_batch.shape)
4     break
```

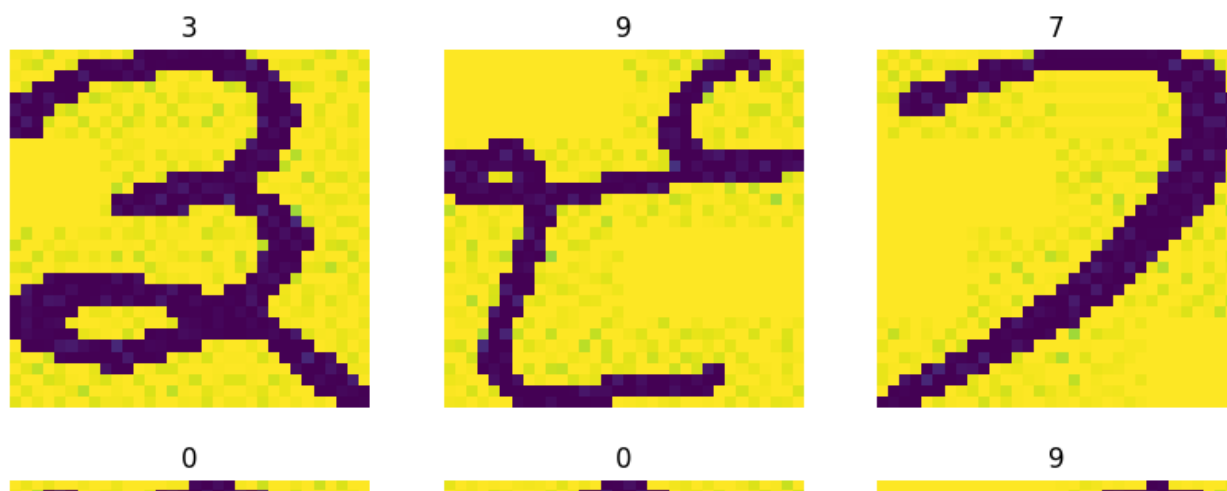
```
(32, 32, 32, 1)
(32,)
```

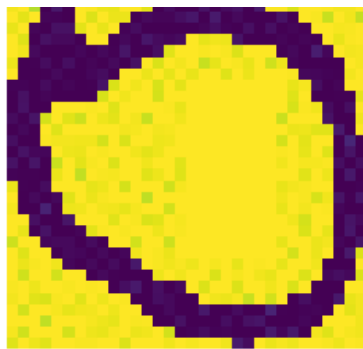
```
1 normalized_ds = train_ds.map(lambda x, y: (normalization_layer(x), y))
2 image_batch, labels_batch = next(iter(normalized_ds))
3 first_image = image_batch[0]
4 # Notice the pixel values are now in `[0,1]`.
5 print(np.min(first_image), np.max(first_image))
```

```
0.0 1.0
```

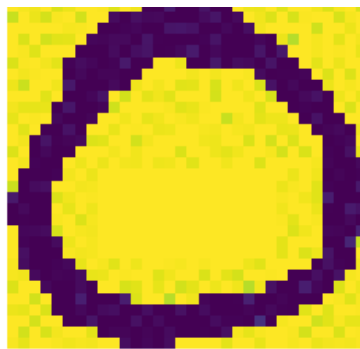
```
1 AUTOTUNE = tf.data.AUTOTUNE
2 train_ds = train_ds.cache().prefetch(buffer_size=AUTOTUNE)
3
4 val_ds = val_ds.cache().prefetch(buffer_size=AUTOTUNE)
```

```
1 import matplotlib.pyplot as plt
2
3 plt.figure(figsize=(10, 10))
4 for images, labels in train_ds.take(1):
5     for i in range(9):
6         ax = plt.subplot(3, 3, i + 1)
7         plt.imshow(images[i].numpy().astype("uint8"))
8         plt.title(class_names[labels[i]])
9         plt.axis("off")
```

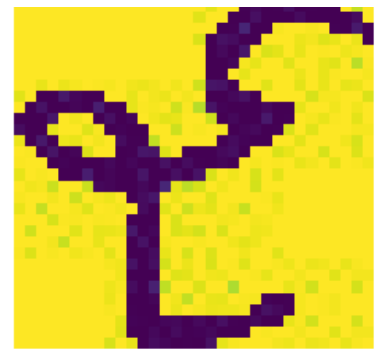




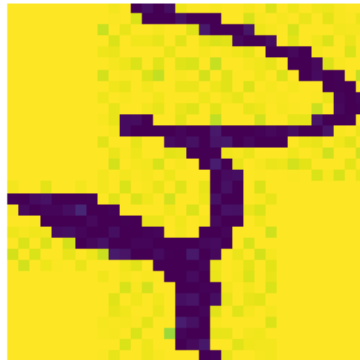
2



3



1



```

1 model = tf.keras.Sequential([
2     tf.keras.layers.Flatten(input_shape=(32, 32)),
3     tf.keras.layers.Dense(128, activation='relu'),
4     tf.keras.layers.Dense(128, activation='relu'),
5     tf.keras.layers.Dense(10)
6 ])

1 model.compile(
2     optimizer='adam',
3     loss=tf.keras.losses.SparseCategoricalCrossentropy(from_logits=True),
4     metrics=['accuracy'])
5
6 model.fit(
7     train_ds,
8     validation_data=val_ds,
9     epochs=30
10 )

```

Epoch 1/30

25/25 [=====] - 2s 18ms/step - loss: 58.6488 - accuracy:

Epoch 2/30

25/25 [=====] - 0s 6ms/step - loss: 11.9595 - accuracy: 6

Epoch 3/30

```
epoch 3/30
25/25 [=====] - 0s 6ms/step - loss: 7.4929 - accuracy: 0.
Epoch 4/30
25/25 [=====] - 0s 7ms/step - loss: 7.0640 - accuracy: 0.
Epoch 5/30
25/25 [=====] - 0s 7ms/step - loss: 3.7112 - accuracy: 0.
Epoch 6/30
25/25 [=====] - 0s 7ms/step - loss: 1.7334 - accuracy: 0.
Epoch 7/30
25/25 [=====] - 0s 6ms/step - loss: 0.6250 - accuracy: 0.
Epoch 8/30
25/25 [=====] - 0s 6ms/step - loss: 0.4168 - accuracy: 0.
Epoch 9/30
25/25 [=====] - 0s 6ms/step - loss: 0.7215 - accuracy: 0.
Epoch 10/30
25/25 [=====] - 0s 6ms/step - loss: 0.5088 - accuracy: 0.
Epoch 11/30
25/25 [=====] - 0s 6ms/step - loss: 0.3059 - accuracy: 0.
Epoch 12/30
25/25 [=====] - 0s 7ms/step - loss: 0.2466 - accuracy: 0.
Epoch 13/30
25/25 [=====] - 0s 7ms/step - loss: 0.8865 - accuracy: 0.
Epoch 14/30
25/25 [=====] - 0s 7ms/step - loss: 1.2925 - accuracy: 0.
Epoch 15/30
25/25 [=====] - 0s 10ms/step - loss: 1.3899 - accuracy: 0.
Epoch 16/30
25/25 [=====] - 0s 12ms/step - loss: 1.0421 - accuracy: 0.
Epoch 17/30
25/25 [=====] - 0s 10ms/step - loss: 0.8203 - accuracy: 0.
Epoch 18/30
25/25 [=====] - 0s 10ms/step - loss: 0.2121 - accuracy: 0.
Epoch 19/30
25/25 [=====] - 0s 13ms/step - loss: 0.0886 - accuracy: 0.
Epoch 20/30
25/25 [=====] - 0s 6ms/step - loss: 0.1593 - accuracy: 0.
Epoch 21/30
25/25 [=====] - 0s 6ms/step - loss: 0.2798 - accuracy: 0.
Epoch 22/30
25/25 [=====] - 0s 6ms/step - loss: 0.3346 - accuracy: 0.
Epoch 23/30
25/25 [=====] - 0s 6ms/step - loss: 0.3019 - accuracy: 0.
Epoch 24/30
25/25 [=====] - 0s 6ms/step - loss: 0.4906 - accuracy: 0.
Epoch 25/30
25/25 [=====] - 0s 5ms/step - loss: 0.3932 - accuracy: 0.
Epoch 26/30
25/25 [=====] - 0s 6ms/step - loss: 0.7125 - accuracy: 0.
Epoch 27/30
25/25 [=====] - 0s 7ms/step - loss: 0.8579 - accuracy: 0.
Epoch 28/30
25/25 [=====] - 0s 7ms/step - loss: 1.0038 - accuracy: 0.
Epoch 29/30
25/25 [=====] - 0s 7ms/step - loss: 1.8652 - accuracy: 0.
```

1

1 from PIL import Image

```
2 import os
3
4 path = "GurNum/val/1/"
5 images= os.listdir(path)
6 for img in images:
7     Image.open('GurNum/val/1/'+img).save(os.path.join(path+ str(img).replace(".bmp",
8     os.remove(os.path.join(path + img))
9
10 path = "GurNum/val/2/"
11 images= os.listdir(path)
12 for img in images:
13     Image.open('GurNum/val/2/'+img).save(os.path.join(path+ str(img).replace(".bmp",
14     os.remove(os.path.join(path + img))
15 path = "GurNum/val/3/"
16 images= os.listdir(path)
17 for img in images:
18     Image.open('GurNum/val/3/'+img).save(os.path.join(path+ str(img).replace(".bmp",
19     os.remove(os.path.join(path + img))
20 path = "GurNum/val/4/"
21 images= os.listdir(path)
22 for img in images:
23     Image.open('GurNum/val/4/'+img).save(os.path.join(path+ str(img).replace(".bmp",
24     os.remove(os.path.join(path + img))
25 path = "GurNum/val/5/"
26 images= os.listdir(path)
27 for img in images:
28     Image.open('GurNum/val/5/'+img).save(os.path.join(path+ str(img).replace(".bmp",
29     os.remove(os.path.join(path + img))
30
31 path = "GurNum/val/6/"
32 images= os.listdir(path)
33 for img in images:
34     Image.open('GurNum/val/6/'+img).save(os.path.join(path+ str(img).replace(".bmp",
35     os.remove(os.path.join(path + img))
36 path = "GurNum/val/7/"
37 images= os.listdir(path)
38 for img in images:
39     Image.open('GurNum/val/7/'+img).save(os.path.join(path+ str(img).replace(".bmp",
40     os.remove(os.path.join(path + img))
41 path = "GurNum/val/8/"
42 images= os.listdir(path)
43 for img in images:
44     Image.open('GurNum/val/8/'+img).save(os.path.join(path+ str(img).replace(".bmp",
45     os.remove(os.path.join(path + img))
46
47 path = "GurNum/val/9/"
48 images= os.listdir(path)
49 for img in images:
50     Image.open('GurNum/val/9/'+img).save(os.path.join(path+ str(img).replace(".bmp",
51     os.remove(os.path.join(path + img))
52
53 path = "GurNum/val/0/"
54 images= os.listdir(path)
55 for img in images:
```

```
56 Image.open('GurNum/val/0/'+img).save(os.path.join(path+ str(img).replace(".bmp",
57 os.remove(os.path.join(path + img))
```

```
1 import pathlib
2 data_dir = pathlib.Path('GurNum/val').with_suffix('')
```

```
1 test_ds = tf.keras.utils.image_dataset_from_directory(
2     data_dir,
3     color_mode='grayscale',
4     seed=123,
5     image_size=(img_height, img_width),
6     batch_size=batch_size,
7 )
```

Found 178 files belonging to 10 classes.

```
1 test_loss, test_acc = model.evaluate(test_ds, verbose=2)
2
3 print('\nTest accuracy:', test_acc)
```

6/6 - 0s - loss: 7.7909 - accuracy: 0.8876 - 61ms/epoch - 10ms/step

Test accuracy: 0.8876404762268066

```
1 # Referance:
2 # https://www.tensorflow.org/guide/data#decoding\_image\_data\_and\_resizing\_it
3 # https://stackoverflow.com/questions/71047609/what-is-the-best-way-of-converting-a-
4 # https://colab.research.google.com/github/tensorflow/docs/blob/master/site/en/tutor
```




This document was created with the Win2PDF "Print to PDF" printer available at

<https://www.win2pdf.com>

This version of Win2PDF 10 is for evaluation and non-commercial use only.

Visit <https://www.win2pdf.com/trial/> for a 30 day trial license.

This page will not be added after purchasing Win2PDF.

<https://www.win2pdf.com/purchase/>