

# Worksheet-4 Output

```
Requirement already satisfied: gdown in /usr/local/lib/python3.11/dist-packages (5.2.0)
Requirement already satisfied: beautifulsoup4 in /usr/local/lib/python3.11/dist-packages (from gdown) (4.13.3)
Requirement already satisfied: filelock in /usr/local/lib/python3.11/dist-packages (from gdown) (3.18.0)
Requirement already satisfied: requests[socks] in /usr/local/lib/python3.11/dist-packages (from gdown) (2.32.3)
Requirement already satisfied: tqdm in /usr/local/lib/python3.11/dist-packages (from gdown) (4.67.1)
Requirement already satisfied: soupsieve>1.2 in /usr/local/lib/python3.11/dist-packages (from beautifulsoup4->gdown) (2.6)
Requirement already satisfied: typing-extensions>=4.0.0 in /usr/local/lib/python3.11/dist-packages (from beautifulsoup4->gdown) (4.12.2)
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.11/dist-packages (from requests[socks]->gdown) (3.4.1)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.11/dist-packages (from requests[socks]->gdown) (3.10)
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.11/dist-packages (from requests[socks]->gdown) (2.3.0)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.11/dist-packages (from requests[socks]->gdown) (2025.1.31)
Requirement already satisfied: PySocks!=1.5.7,>=1.5.6 in /usr/local/lib/python3.11/dist-packages (from requests[socks]->gdown) (1.7.1)
```

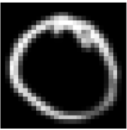
```
import tensorflow as tf
print(tf.keras.__version__)
from PIL import Image
import glob
import numpy as np
import matplotlib.pyplot as plt
import os
from tensorflow.keras.utils import to_categorical
from tensorflow.keras import layers, models

3.8.0
```

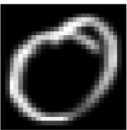
```
plt.title('Label: {np.argmax(y_hat[1])}')
plt.axis("off")
plt.show()
```

Training set: (17000, 28, 28, 1), Labels: (17000, 10)  
Testing set: (3000, 28, 28, 1), Labels: (3000, 10)

Label: 0

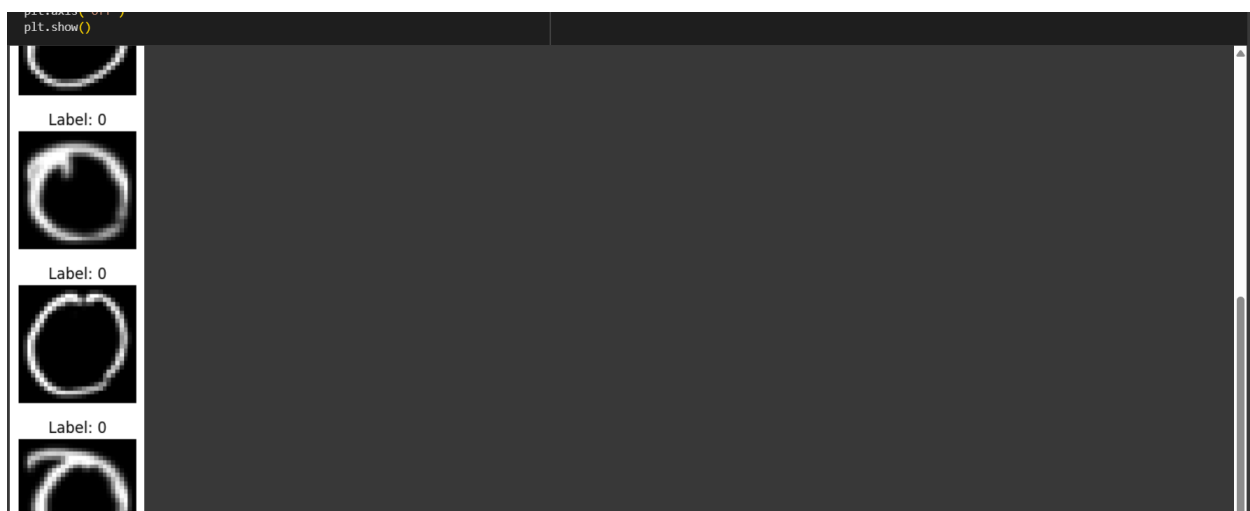


Label: 0



Label: 0





```
model.summary()
```

Model: "sequential\_10"

Layer (type)	Output Shape	Param #
flatten_10 (Flatten)	(None, 784)	0
dense_40 (Dense)	(None, 10)	12,560
dense_41 (Dense)	(None, 32)	344
dense_42 (Dense)	(None, 32)	1,056
dense_43 (Dense)	(None, 10)	330

Total params: 14,490 (56.60 KB)  
Trainable params: 14,490 (56.60 KB)  
Non-trainable params: 0 (0.00 B)

```
batch_size = 128
epochs = 20

callbacks = [
    keras.callbacks.EarlyStopping(monitor="val_loss", patience = 4)
]

model.fit(
    x = x_train,
    y = y_train,
    batch_size = batch_size,
    epochs = epochs,
    validation_split = 0.2,
    callbacks = callbacks
)
```

Epoch 1/20  
107/107 — 2s 6ms/step - accuracy: 0.2046 - loss: 2.3074 - val\_accuracy: 0.0000e+00 - val\_loss: 3.9464  
Epoch 2/20  
107/107 — 0s 4ms/step - accuracy: 0.4878 - loss: 1.9105 - val\_accuracy: 0.0000e+00 - val\_loss: 4.6360  
Epoch 3/20  
107/107 — 1s 4ms/step - accuracy: 0.6297 - loss: 1.4534 - val\_accuracy: 0.0000e+00 - val\_loss: 5.1182  
Epoch 4/20  
107/107 — 1s 4ms/step - accuracy: 0.7542 - loss: 1.0365 - val\_accuracy: 0.0000e+00 - val\_loss: 5.5507  
Epoch 5/20  
107/107 — 0s 4ms/step - accuracy: 0.8376 - loss: 0.7891 - val\_accuracy: 0.0000e+00 - val\_loss: 5.9151  
<keras.src.callbacks.history.History at 0x7ce37d4da1d0>

```

    batch_size=batch_size,
    verbose=2,
    sample_weight=None,
    steps=None,
    callbacks=callbacks
)

print(f"Testing accuracy: {test_acc:.4f} The accuracy value computed based on the test dataset. It shows the proportion of correct predictions made by the model.")
print(f"Testing loss: {test_loss:.4f} The loss value computed based on the test dataset. This gives an indication of how well the model is performing on unseen data.")

24/24 - 0s - 4ms/step - accuracy: 0.6920 - loss: 1.7144
Testing accuracy: 0.6920 The accuracy value computed based on the test dataset. It shows the proportion of correct predictions made by the model.
Testing loss: 1.7144 The loss value computed based on the test dataset. This gives an indication of how well the model is performing on unseen data.

```

```

model.save("devanagari_digit_model.keras")

loaded_model = tf.keras.models.load_model("devanagari_digit_model.keras")

loaded_model.compile(
    optimizer="adam",
    loss="categorical_crossentropy",
    metrics=["accuracy"]
)

test_loss, test_acc = loaded_model.evaluate(
    x=x_test,
    y=y_test,
    batch_size=batch_size,
    verbose=2
)

print(f"Testing accuracy: {test_acc:.4f}")
print(f"Testing loss: {test_loss:.4f}")

24/24 - 0s - 16ms/step - accuracy: 0.6920 - loss: 1.7144
Testing accuracy: 0.6920
Testing loss: 1.7144

```

```

predictions = model.predict(x_test)

prediction_labels = np.argmax(predictions, axis=1)
actual_labels = np.argmax(y_test, axis=1)

print(f"Predicted labels: {prediction_labels[:10]}")
print(f"Actual labels: {actual_labels[:10]}")

94/94 ----- 0s 2ms/step
Predicted labels: [0 0 0 0 0 0 0 0 0 0]
Actual labels: [0 0 0 0 0 0 0 0 0 0]

history = model.fit(
    x=x_train,
    y=y_train,
    batch_size=batch_size,
    epochs=epochs,
    validation_split=0.2,
    callbacks=callbacks
)

Epoch 1/20 ----- 0s 4ms/step - accuracy: 0.8805 - loss: 0.6245 - val_accuracy: 0.0000e+00 - val_loss: 6.2283
107/107
Epoch 2/20 ----- 1s 4ms/step - accuracy: 0.9001 - loss: 0.5043 - val_accuracy: 0.0000e+00 - val_loss: 6.5030
107/107
Epoch 3/20 ----- 1s 4ms/step - accuracy: 0.9274 - loss: 0.4099 - val_accuracy: 0.0000e+00 - val_loss: 6.7833
107/107

```

```
plt.show()
```

