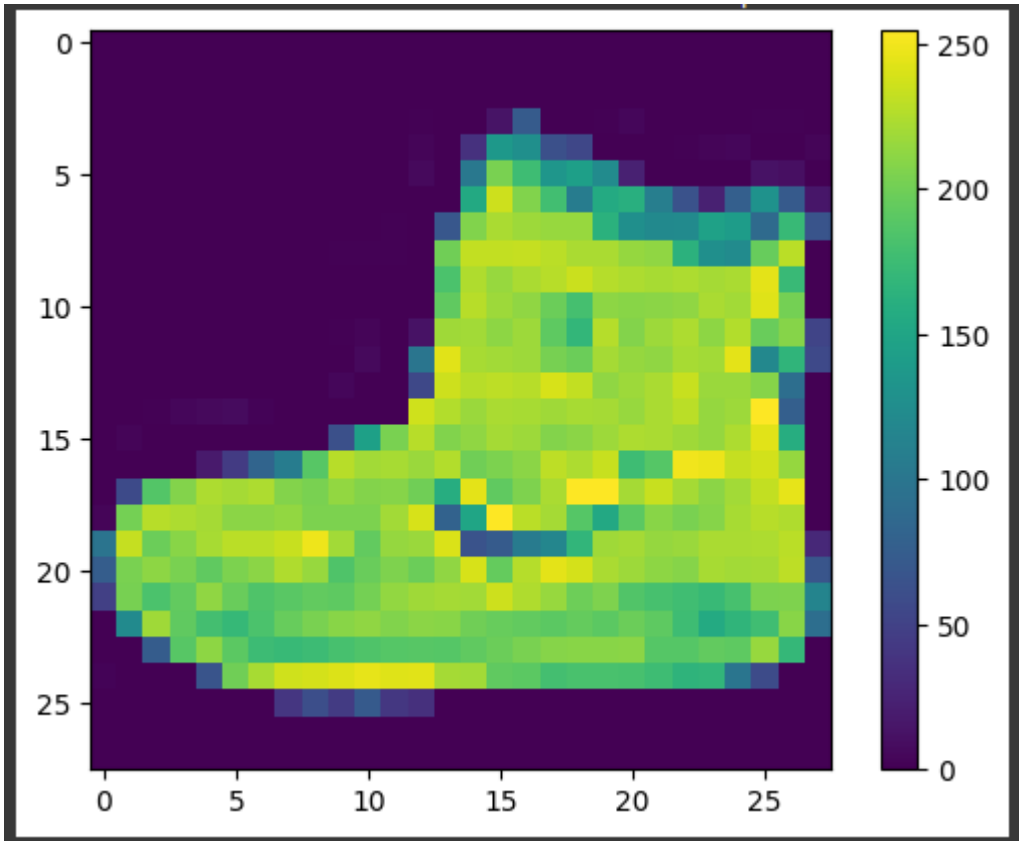
CIS 515: COMPUTER GRAPHICS
LAB – 9
UNIVERSITY OF MICHIGAN – DEARBORN
FALL 2024

By,
VISHVENDRA REDDY BHOOMIDI
bhoomidi@umich.edu

TASK 1: MNIST Dataset

```
import tensorflow as tf
from tensorflow import keras

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

print(tf.__version__)

mnist = keras.datasets.fashion_mnist

(x_train, y_train), (x_test, y_test) = mnist.load_data()

class_names = ['top', 'trouser', 'pullover', 'dress', 'coat', \
    'sandal', 'shirt', 'sneaker', 'bag', 'ankle_boot']

plt.figure()
plt.imshow(x_train[0])
plt.colorbar()
plt.show()

# image normalization
x_train = x_train/255.0
x_test = x_test/255.0

from tensorflow.keras import Sequential
from tensorflow.keras.layers import Flatten, Dense

# Create a simple Neural Network learning model
model = Sequential()
model.add(Flatten(input_shape = (28, 28)))
```

```
model.add(Dense(128, activation = 'relu'))
model.add(Dense(10, activation = 'softmax'))

print(model.summary())

# compile the machine learning model
model.compile(optimizer='adam', loss = 'sparse_categorical_crossentropy', \
    metrics = ['accuracy'])

# Machine learning process
model.fit(x_train, y_train, epochs = 2)

# Evaluation of machine learning result
test_loss, test_acc = model.evaluate(x_test, y_test)
print("Evaluation accuracy = ", 100*test_acc, "%")

# Predict on test data
predictions = model.predict(x_test[:5])

# Display sample predictions
for i, prediction in enumerate(predictions):
    print(f"Sample {i + 1} - Predicted Label: {prediction.argmax()}, True Label: {y_test[i]}")

# Save weights to a HDF5 file
# The Hierarchical Data Format version 5 (HDF5), is an open source file format
# that supports large, complex, heterogeneous data.
model.save("model2.h5")
print("Saved model to disk")
```

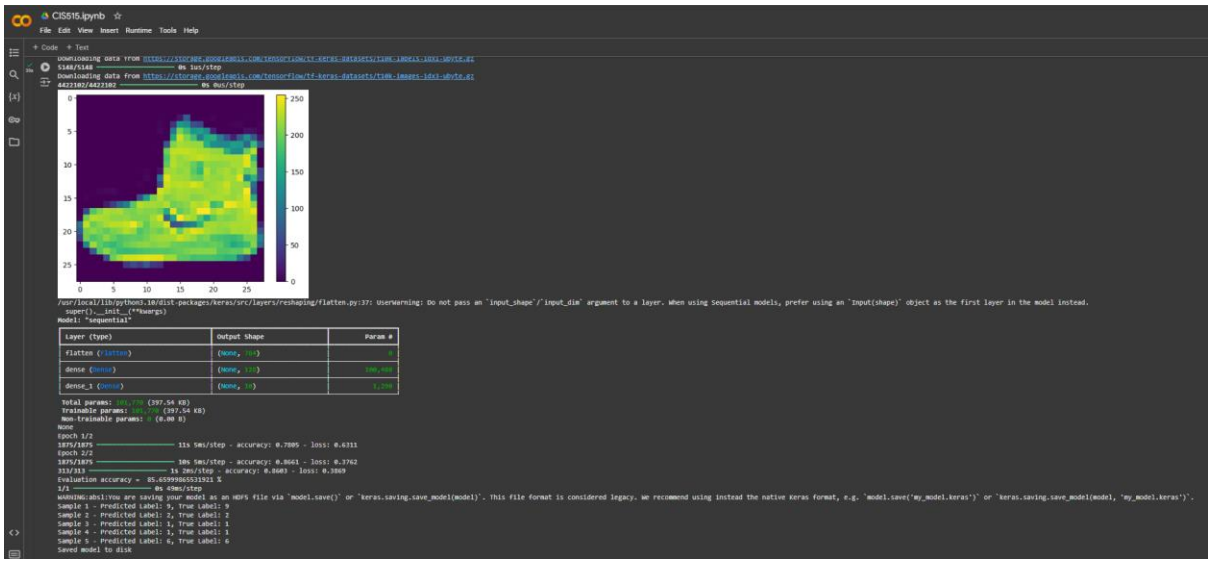



```
Model: "sequential"
```

Layer (type)	Output Shape	Param #
flatten (Flatten)	(None, 784)	0
dense (Dense)	(None, 128)	100,480
dense_1 (Dense)	(None, 10)	1,290

```
 Total params: 101,770 (397.54 KB)
 Trainable params: 101,770 (397.54 KB)
 Non-trainable params: 0 (0.00 B)
None
Epoch 1/2
1875/1875 ━━━━━━━━━━━━━━━━━━━━ 11s 5ms/step - accuracy: 0.7805 - loss: 0.6311
Epoch 2/2
1875/1875 ━━━━━━━━━━━━━━━━━━━━ 10s 5ms/step - accuracy: 0.8661 - loss: 0.3762
313/313 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - accuracy: 0.8603 - loss: 0.3869
Evaluation accuracy =  85.65999865531921 %
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 49ms/step
WARNING:absl:You are saving your model as an HDF5 file via `model.save()` or `keras.savin
Sample 1 - Predicted Label: 9, True Label: 9
Sample 2 - Predicted Label: 2, True Label: 2
Sample 3 - Predicted Label: 1, True Label: 1
Sample 4 - Predicted Label: 1, True Label: 1
Sample 5 - Predicted Label: 6, True Label: 6
Saved model to disk
```

TASK 2: Google Collab



TASK 3: Testing of your own Image

```
import cv2

model = load_model("model2.h5")

image_path = "/content/sandal.png"

image = cv2.imread(image_path, cv2.IMREAD_GRAYSCALE)
imageResize = cv2.resize(image, (28, 28))
imageNormalize = imageResize / 255.0
imageNormalize = np.expand_dims(imageNormalize, axis = 0)

predictions = model.predict(imageNormalize)
predictedLabel = predictions.argmax()

class_names = ['top', 'trouser', 'pullover', 'dress', 'coat',
```

'sandal', 'shirt', 'sneaker', 'bag', 'ankle_boot']

```
plt.figure()
plt.imshow(imageResize, cmap='gray')
plt.title(f"Predicted Label: {class_names[predictedLabel]}")
plt.colorbar()
plt.show()
```