**COMSATS University   
Park Road, Chak Shahzad, Islamabad Pakistan**

Assignment # 01

**Handling Data**

***By***

**Waleed Butt CU/SP18-BCS-170/ISB**

***Course Instructor*Mr. Usman Yaseen**

Class/Section: BCS-6B

Submission Date: 10/16/2020

*Bachelor of Science in Computer Science (2018-2022)*

# Problem

In the images folder, you will see 12 folders which are from 12 human donors at different stages of type 1 diabetes. Each folder contains a number of images in tiff format. The images folder also contains an excel file with the name ‘Metadata’. The first column contains the name of the image, second column tells the folder name and the last column tells the stage of each image. So the stage column is the desired output column and this makes this problem a supervised learning problem.

Write a python code in Colab by using keras and other supporting libraries to load the dataset and apply any machine learning model of your choice such as neural network by using the libraries. Train the model on Colab’s GPU and perform classification. At this stage there is no need to dig deep in the algorithm, just implement with the help of libraries.

# Code

import numpy as np

import pandas as pd

Metadata = pd.read\_csv('./train/Metadata.csv')

from tensorflow import keras

from keras.preprocessing.image import load\_img

from keras.preprocessing.image import img\_to\_array

import glob

x = 150

y = 150

train\_data = np.stack(

[img\_to\_array(

load\_img(glob.glob("./train\\\*\\"+i+"\*.tiff")[0], target\_size=(x, y, 3))) for i in Metadata['image'][:600]])

#Reshaping

train\_data = train\_data.reshape( len(train\_data), x\*y\*3 )

#Normalizing

train\_data = train\_data / 255.0

train\_labels = np.array(Metadata['stage'][:600])

test\_data = np.stack(

[img\_to\_array(

load\_img(glob.glob("./train\\\*\\"+i+"\*.tiff")[0], target\_size=(x, y, 3))) for i in Metadata['image'][600:]])

#Reshaping

test\_data = test\_data.reshape( len(test\_data), x\*y\*3 )

#Normalizing

test\_data = test\_data / 255.0

test\_labels = np.array(Metadata['stage'][600:])

from keras import models

from keras import layers

network = models.Sequential()

network.add(layers.Dense(512, activation='relu', input\_shape=(x \* y \* 3,)))

network.add(layers.Dense(256, activation='relu'))

network.add(layers.Dense(64, activation='relu'))

network.add(layers.Dense(3, activation='softmax'))

temp = []

for i in range(len(train\_labels)):

if train\_labels[i] == "Long-duration":

temp.append(0)

elif train\_labels[i] == "Non-diabetic":

temp.append(1)

else:

temp.append(2)

train\_labels = to\_categorical(np.array(temp))

temp = []

for i in range(len(test\_labels)):

if test\_labels[i] == "Long-duration":

temp.append(0)

elif test\_labels[i] == "Non-diabetic":

temp.append(1)

else:

temp.append(2)

test\_labels = to\_categorical(np.array(temp))

network.compile(optimizer='rmsprop',loss='categorical\_crossentropy',metrics=['accuracy'])

network.fit(train\_data, train\_labels, epochs=5, batch\_size=128)

# Output of this cell

Epoch 1/5

5/5 [==============================] - 4s 779ms/step - loss: 8.2973 - accuracy: 0.3433

Epoch 2/5

5/5 [==============================] - 3s 591ms/step - loss: 3.2754 - accuracy: 0.3950

Epoch 3/5

5/5 [==============================] - 3s 600ms/step - loss: 1.9129 - accuracy: 0.3533

Epoch 4/5

5/5 [==============================] - 3s 576ms/step - loss: 1.3953 - accuracy: 0.4333

Epoch 5/5

5/5 [==============================] - 3s 625ms/step - loss: 1.5396 - accuracy: 0.4017

Out[105]:

test\_loss, test\_acc = network.evaluate(test\_data, test\_labels)

print('test\_acc:', test\_acc)

print('test\_loss:', test\_loss)

Output of this cell

8/8 [==============================] - 0s 46ms/step - loss: 1.2941 - accuracy: 0.4122

test\_acc: 0.41224488615989685

test\_loss: 1.2940518856048584

# Screenshot

