Group - 95

Cotton Disease Prediction

Agenda

Introduction

Gantt Chart

Deep Learning Using Resnet50

Website

Future Goal

Introduction

Cotton is also called “White Gold” and “The King of Fibers. ” For growers, processors, exporters, and producing countries, cotton is the earnest point of supply. This work presents cotton plant disease detection using image processing technique for automated vision system used at agricultural field. In agriculture research of automatic plant disease detection is essential one in monitoring large fields of crops and thus automatically detects symptoms of disease as soon as they appear on plant leaves. It is very difficult for a farmer to identified various disease in plants. The estimated annual crop losses due to plant disease at the worldwide is $60 Billions. The traditional tools and techniques are not very useful since it takes lots of time and manual work.

Gantt Chart

Work Flow Chart

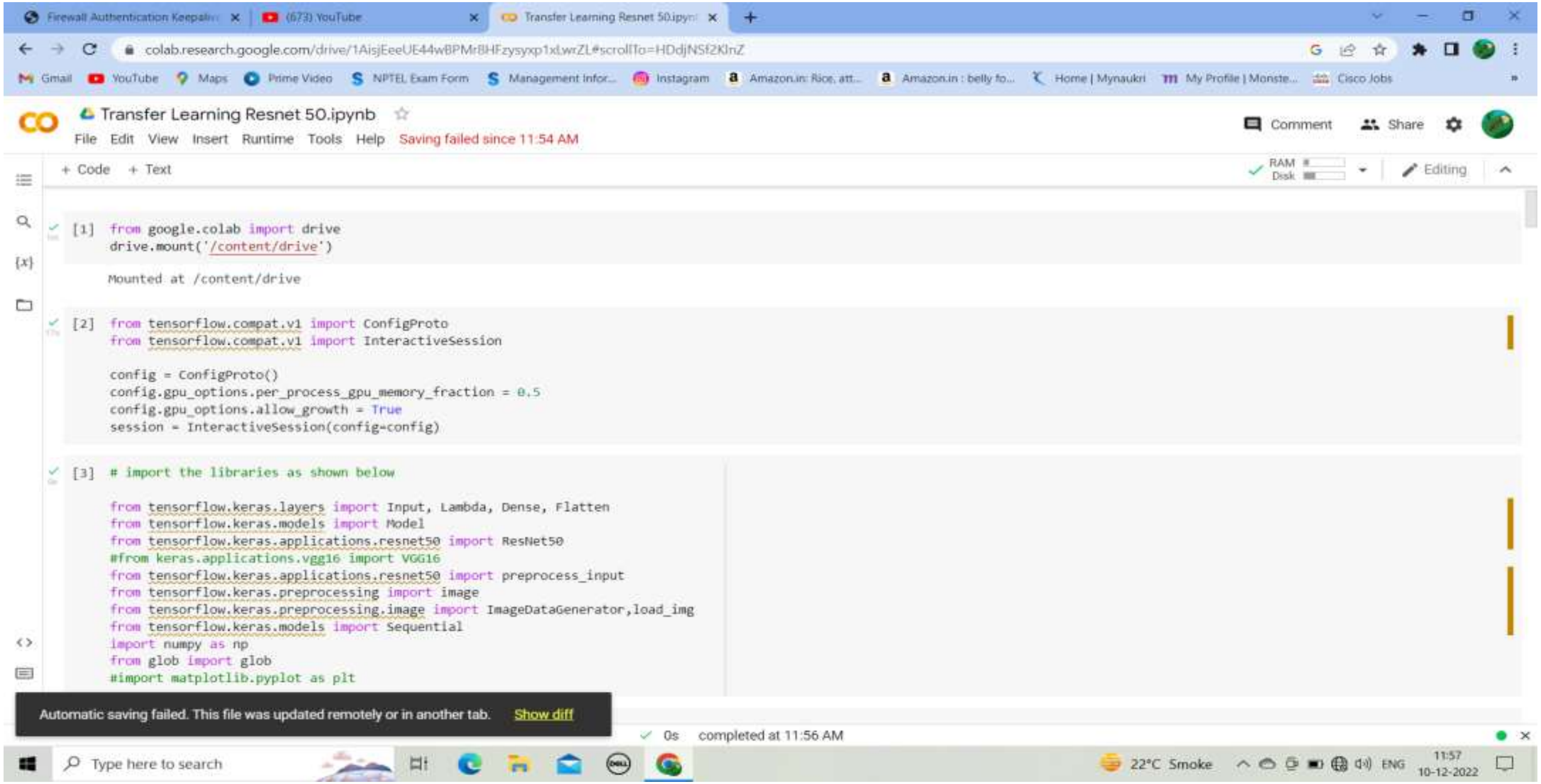
Work Flow

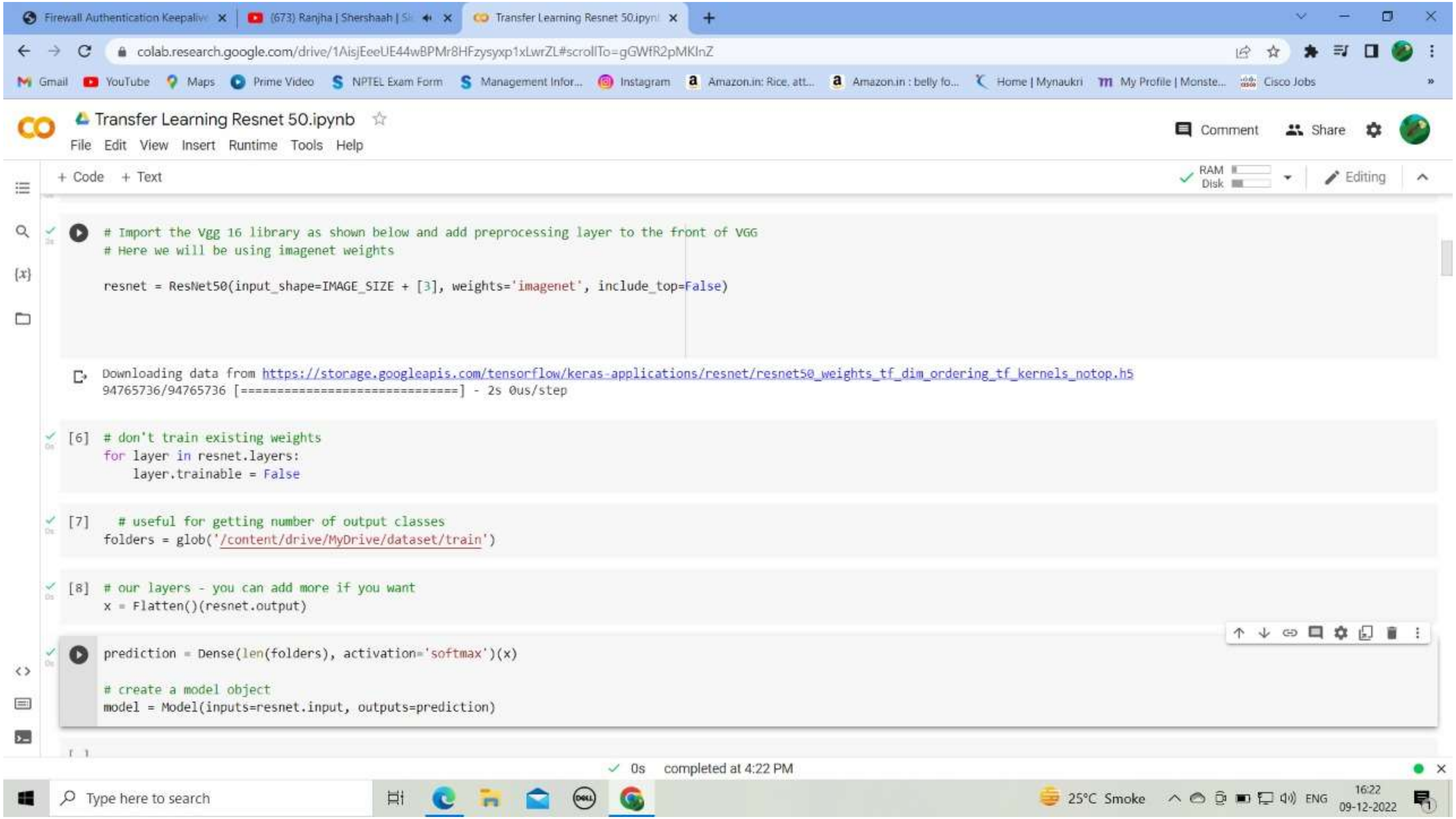
Project Design	September	October	November	December	January	February	March
Goal Identification							
Project Approval							
Get Dataset For project							
Frontend Development							
Backend Development							
Testing							
Deployment							

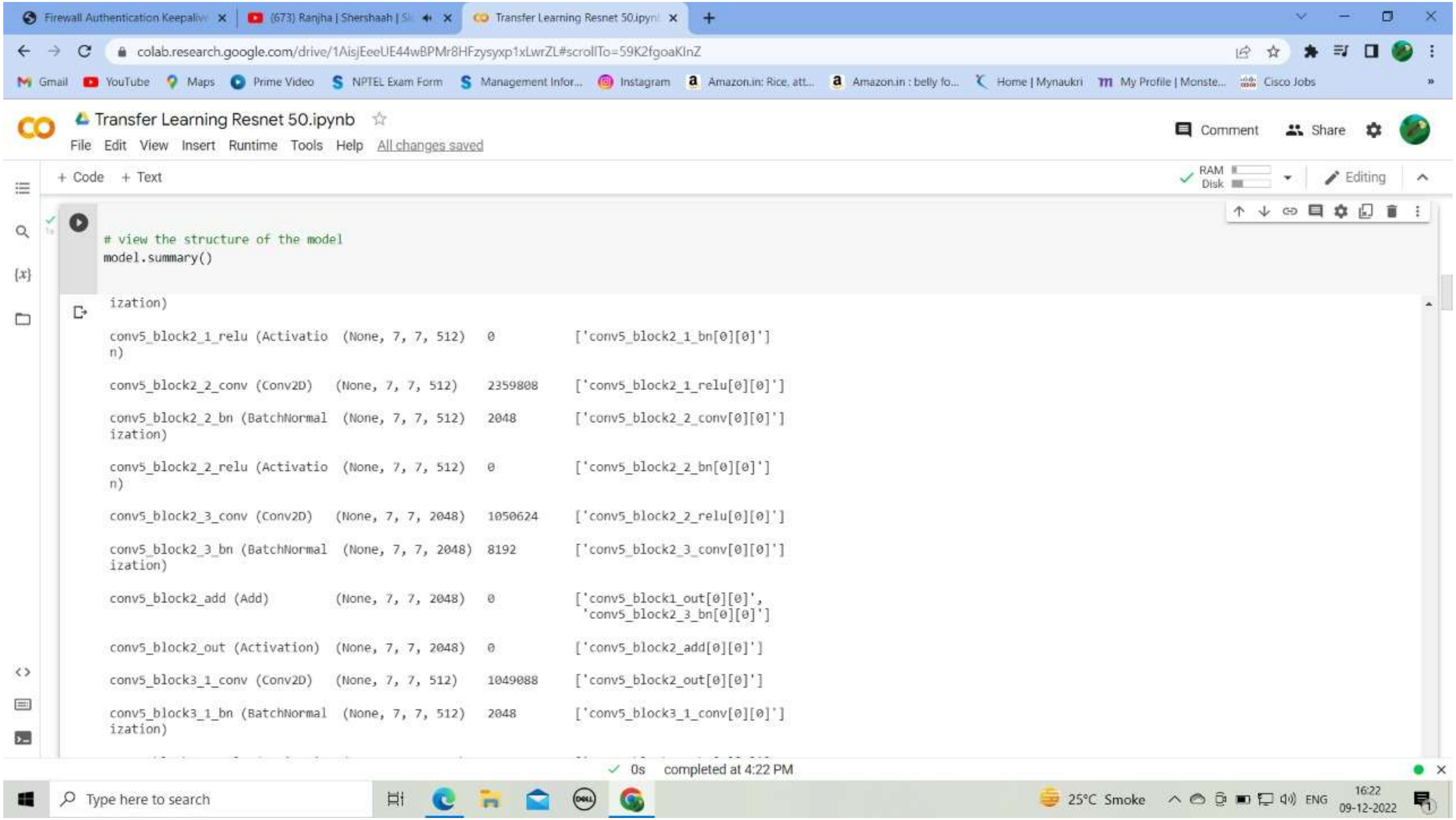
Deep Learning Using Resnet50

Backend Work

Work Flow


colab.research.google.com/drive/1AisjEeeUE44wBPMrBHFzysyxp1xLwrZL#scrollTo=HDdjNSf2KlnZ
Transfer Learning Resnet 50.ipynb
File Edit View Insert Runtime Tools Help Saving failed since 11:54 AM
Comment Share
+ Code + Text
Editing
[1] from google.colab import drive
drive.mount('/content/drive')
Mounted at /content/drive
[2] from tensorflow.compat.v1 import ConfigProto
from tensorflow.compat.v1 import InteractiveSession
config = ConfigProto()
config.gpu_options.per_process_gpu_memory_fraction = 0.5
config.gpu_options.allow_growth = True
session = InteractiveSession(config=config)
[3] # import the libraries as shown below
from tensorflow.keras.layers import Input, Lambda, Dense, Flatten
from tensorflow.keras.models import Model
from tensorflow.keras.applications.resnet50 import ResNet50
#from keras.applications.vgg16 import VGG16
from tensorflow.keras.applications.resnet50 import preprocess_input
from tensorflow.keras.preprocessing import image
from tensorflow.keras.preprocessing.image import ImageDataGenerator,load_img
from tensorflow.keras.models import Sequential
import numpy as np
from glob import glob
#import matplotlib.pyplot as plt
Automatic saving failed. This file was updated remotely or in another tab. Show diff
0s completed at 11:56 AM
Type here to search
22°C Smoke
ENG
11:57
10-12-2022


Transfer Learning Resnet 50.ipynb
File Edit View Insert Runtime Tools Help
# Import the Vgg 16 library as shown below and add preprocessing layer to the front of VGG
# Here we will be using imagenet weights
resnet = ResNet50(input_shape=IMAGE_SIZE + [3], weights='imagenet', include_top=False)
Downloading data from https://storage.googleapis.com/tensorflow/keras-applications/resnet/resnet50_weights_tf_dim_ordering_tf_kernels_notop.h5
94765736/94765736 [==============================] - 2s 0us/step
[6] # don't train existing weights
for layer in resnet.layers:
    layer.trainable = False
[7] # useful for getting number of output classes
folders = glob('/content/drive/MyDrive/dataset/train')
[8] # our layers - you can add more if you want
x = Flatten()(resnet.output)
prediction = Dense(len(folders), activation='softmax')(x)
# create a model object
model = Model(inputs=resnet.input, outputs=prediction)
completed at 4:22 PM

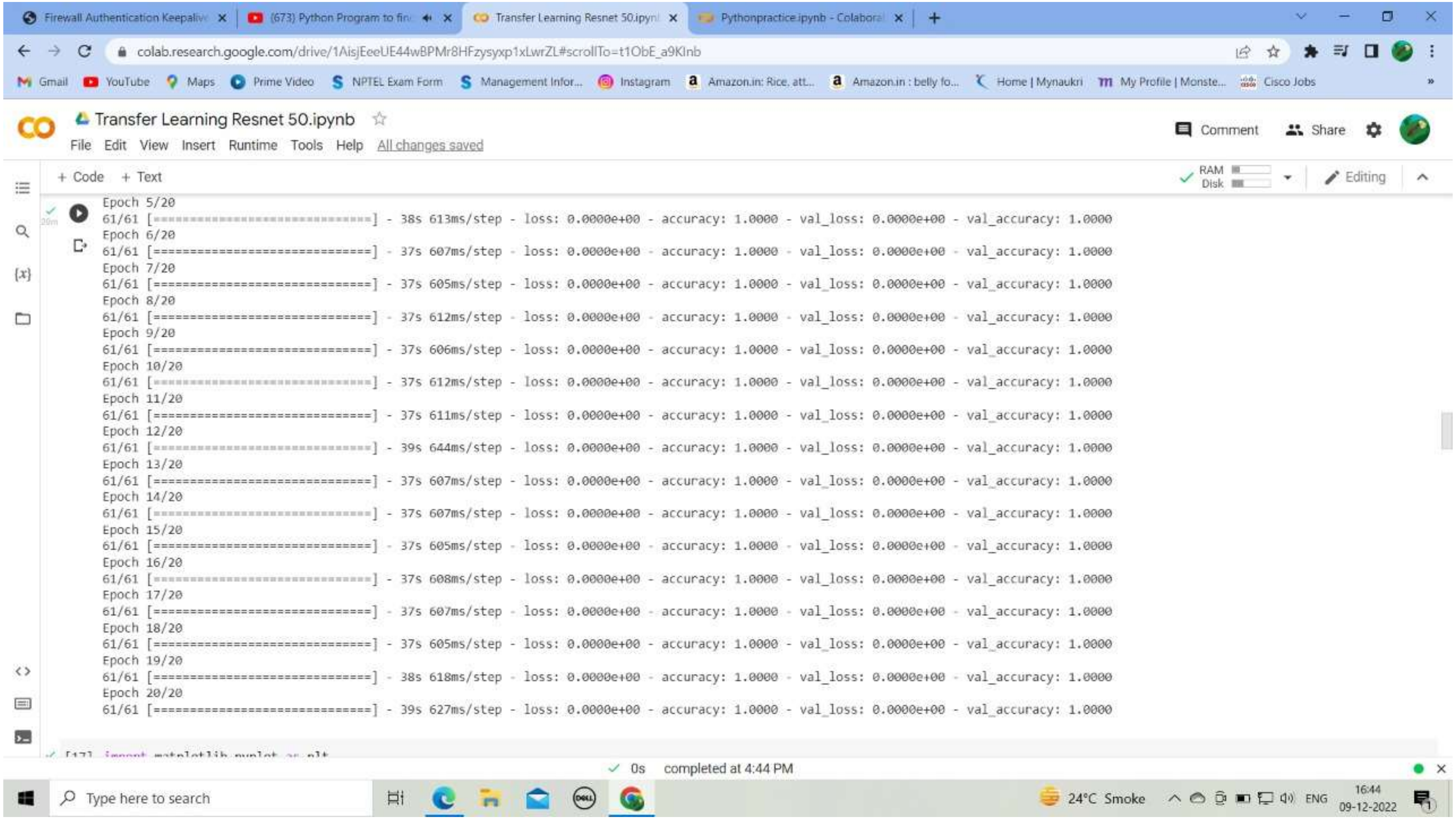

Transfer Learning Resnet 50.ipynb
# view the structure of the model
model.summary()
ization)
conv5_block2_1_relu (Activation) (None, 7, 7, 512) 0 ['conv5_block2_1_bn[0][0]']
conv5_block2_2_conv (Conv2D) (None, 7, 7, 512) 2359808 ['conv5_block2_1_relu[0][0]']
conv5_block2_2_bn (BatchNormalization) (None, 7, 7, 512) 2048 ['conv5_block2_2_conv[0][0]']
conv5_block2_2_relu (Activation) (None, 7, 7, 512) 0 ['conv5_block2_2_bn[0][0]']
conv5_block2_3_conv (Conv2D) (None, 7, 7, 2048) 1050624 ['conv5_block2_2_relu[0][0]']
conv5_block2_3_bn (BatchNormalization) (None, 7, 7, 2048) 8192 ['conv5_block2_3_conv[0][0]']
conv5_block2_add (Add) (None, 7, 7, 2048) 0 ['conv5_block1_out[0][0]', 'conv5_block2_3_bn[0][0]']
conv5_block2_out (Activation) (None, 7, 7, 2048) 0 ['conv5_block2_add[0][0]']
conv5_block3_1_conv (Conv2D) (None, 7, 7, 512) 1049088 ['conv5_block2_out[0][0]']
conv5_block3_1_bn (BatchNormalization) (None, 7, 7, 512) 2048 ['conv5_block3_1_conv[0][0]']
0s completed at 4:22 PM

```
[19] # save it as a h5 file

     from tensorflow.keras.models import load_model

     model.save('model_resnet50.h5')
```

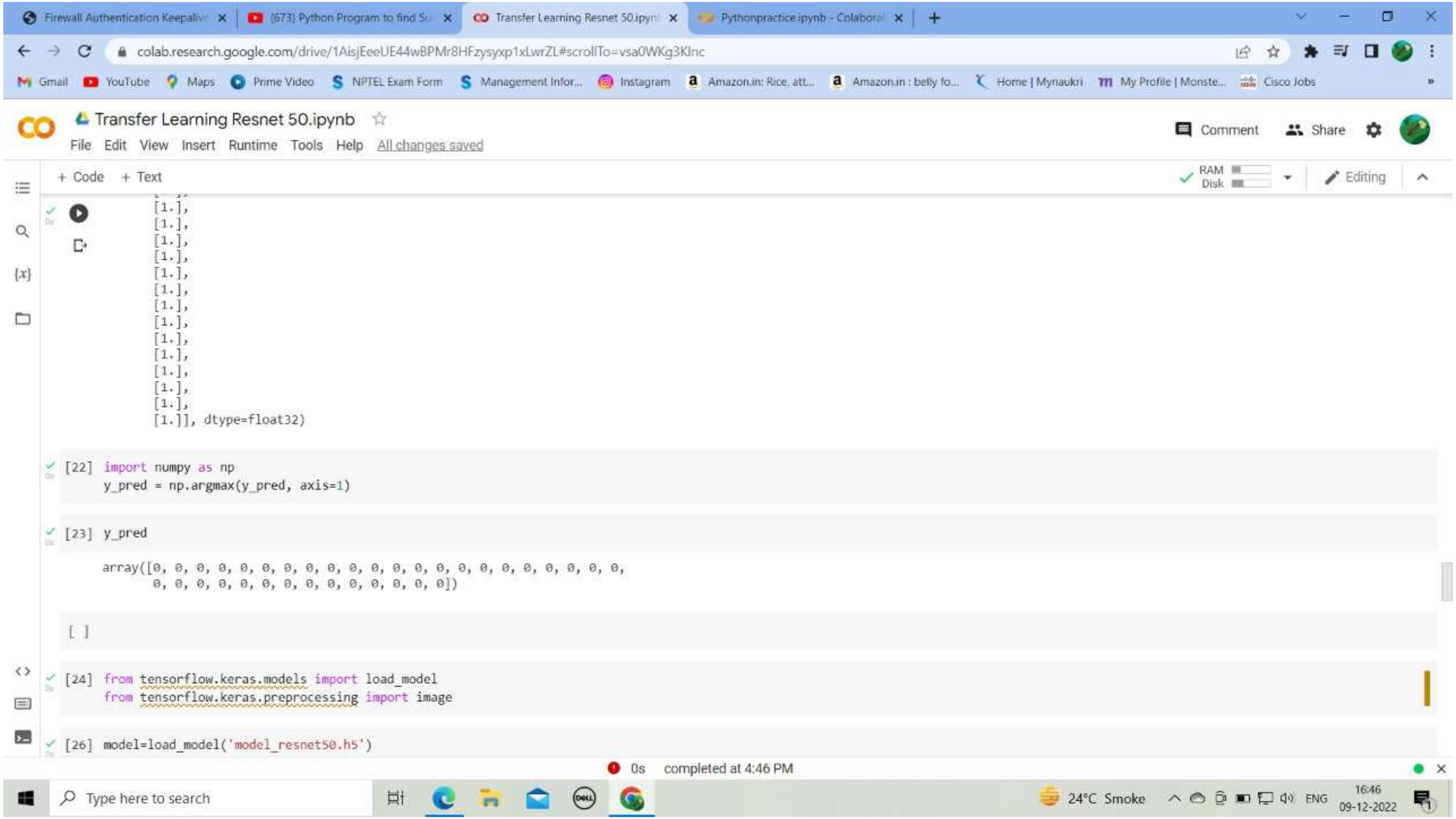
```
[ ]
```

```
[20]
     y_pred = model.predict(test_set)

     2/2 [==============================] - 4s 125ms/step
```

```
y_pred

array([[1.],
       [1.],
       [1.],
       [1.],
       [1.],
       [1.],
       [1.],
       [1.],
       [1.],
       [1.],
       [1.],
       [1.],
```



```
Epoch 5/20
61/61 [==============================] - 38s 613ms/step - loss: 0.0000e+00 - accuracy: 1.0000 - val_loss: 0.0000e+00 - val_accuracy: 1.0000
Epoch 6/20
61/61 [==============================] - 37s 607ms/step - loss: 0.0000e+00 - accuracy: 1.0000 - val_loss: 0.0000e+00 - val_accuracy: 1.0000
Epoch 7/20
61/61 [==============================] - 37s 605ms/step - loss: 0.0000e+00 - accuracy: 1.0000 - val_loss: 0.0000e+00 - val_accuracy: 1.0000
Epoch 8/20
61/61 [==============================] - 37s 612ms/step - loss: 0.0000e+00 - accuracy: 1.0000 - val_loss: 0.0000e+00 - val_accuracy: 1.0000
Epoch 9/20
61/61 [==============================] - 37s 606ms/step - loss: 0.0000e+00 - accuracy: 1.0000 - val_loss: 0.0000e+00 - val_accuracy: 1.0000
Epoch 10/20
61/61 [==============================] - 37s 612ms/step - loss: 0.0000e+00 - accuracy: 1.0000 - val_loss: 0.0000e+00 - val_accuracy: 1.0000
Epoch 11/20
61/61 [==============================] - 37s 611ms/step - loss: 0.0000e+00 - accuracy: 1.0000 - val_loss: 0.0000e+00 - val_accuracy: 1.0000
Epoch 12/20
61/61 [==============================] - 39s 644ms/step - loss: 0.0000e+00 - accuracy: 1.0000 - val_loss: 0.0000e+00 - val_accuracy: 1.0000
Epoch 13/20
61/61 [==============================] - 37s 607ms/step - loss: 0.0000e+00 - accuracy: 1.0000 - val_loss: 0.0000e+00 - val_accuracy: 1.0000
Epoch 14/20
61/61 [==============================] - 37s 607ms/step - loss: 0.0000e+00 - accuracy: 1.0000 - val_loss: 0.0000e+00 - val_accuracy: 1.0000
Epoch 15/20
61/61 [==============================] - 37s 605ms/step - loss: 0.0000e+00 - accuracy: 1.0000 - val_loss: 0.0000e+00 - val_accuracy: 1.0000
Epoch 16/20
61/61 [==============================] - 37s 608ms/step - loss: 0.0000e+00 - accuracy: 1.0000 - val_loss: 0.0000e+00 - val_accuracy: 1.0000
Epoch 17/20
61/61 [==============================] - 37s 607ms/step - loss: 0.0000e+00 - accuracy: 1.0000 - val_loss: 0.0000e+00 - val_accuracy: 1.0000
Epoch 18/20
61/61 [==============================] - 37s 605ms/step - loss: 0.0000e+00 - accuracy: 1.0000 - val_loss: 0.0000e+00 - val_accuracy: 1.0000
Epoch 19/20
61/61 [==============================] - 38s 618ms/step - loss: 0.0000e+00 - accuracy: 1.0000 - val_loss: 0.0000e+00 - val_accuracy: 1.0000
Epoch 20/20
61/61 [==============================] - 39s 627ms/step - loss: 0.0000e+00 - accuracy: 1.0000 - val_loss: 0.0000e+00 - val_accuracy: 1.0000
```

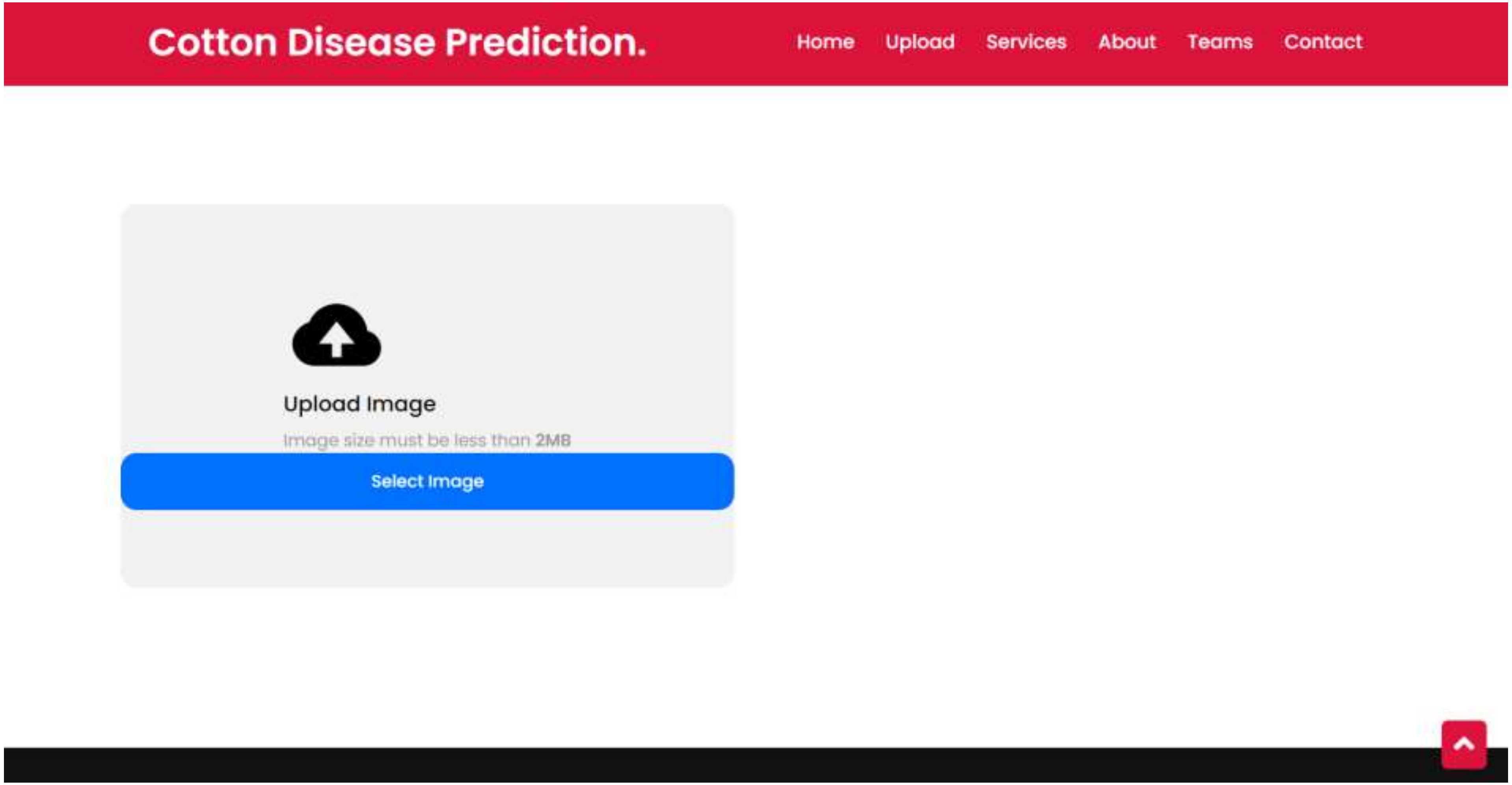

Transfer Learning Resnet 50.ipynb
File Edit View Insert Runtime Tools Help All changes saved
[1.],
[1.],
[1.],
[1.],
[1.],
[1.],
[1.],
[1.],
[1.],
[1.],
[1.],
[1.],
[1.],
[1.]], dtype=float32)
[22] import numpy as np
y_pred = np.argmax(y_pred, axis=1)
[23] y_pred
array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0])
[24] from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image
[26] model=load_model('model_resnet50.h5')
0s completed at 4:46 PM

Website

First Page

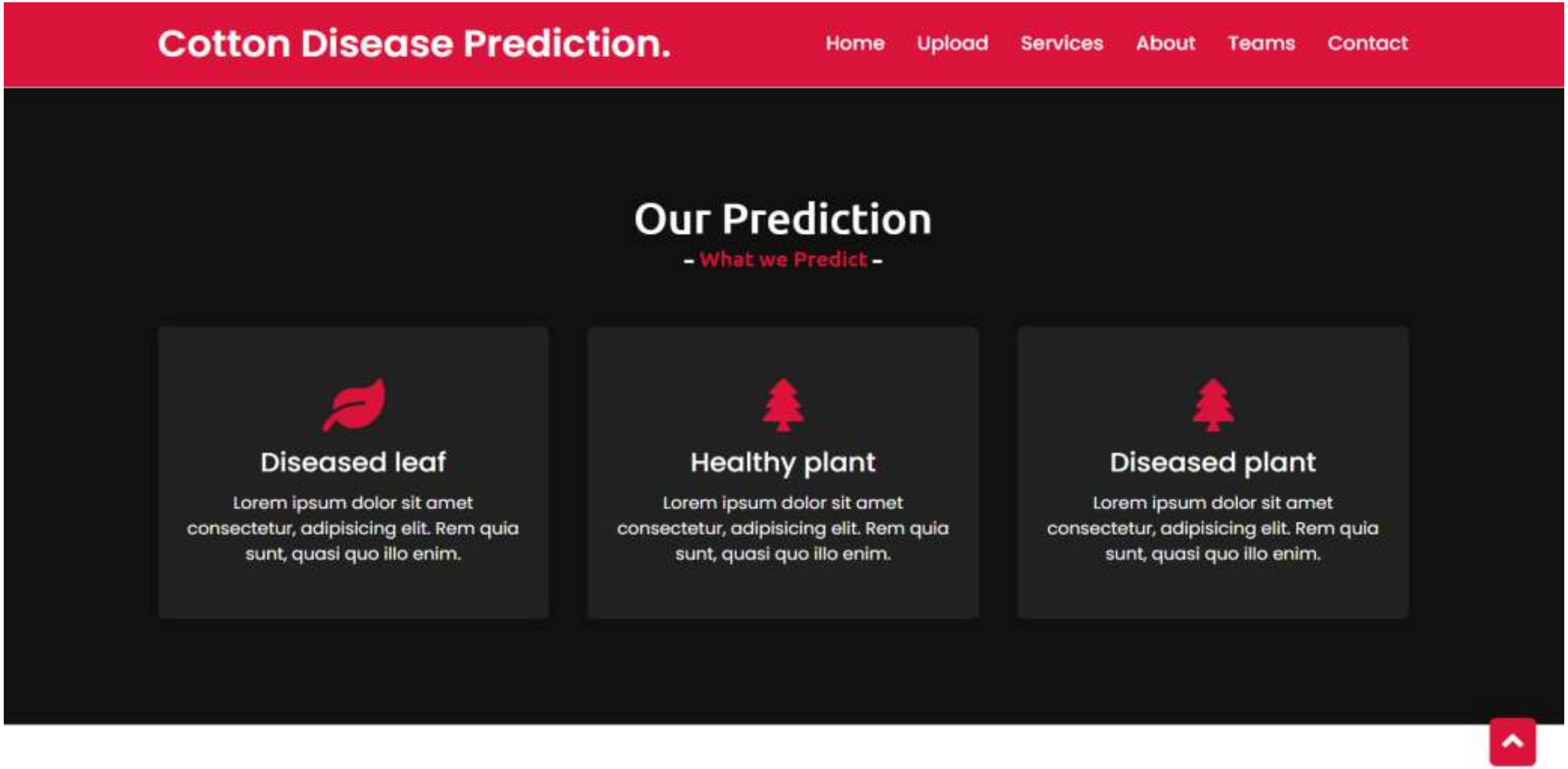

Cotton Disease Prediction.
Home
Upload
Services
About
Teams
Contact
Welcome to the
Cotton Disease Prediction

Upload Page



Cotton Disease Prediction.
Home
Upload
Services
About
Teams
Contact
Upload Image
Image size must be less than 2MB
Select Image

Prediction Page



About Page

Cotton Disease Prediction.

Home | Upload | Services | About | Teams | Contact

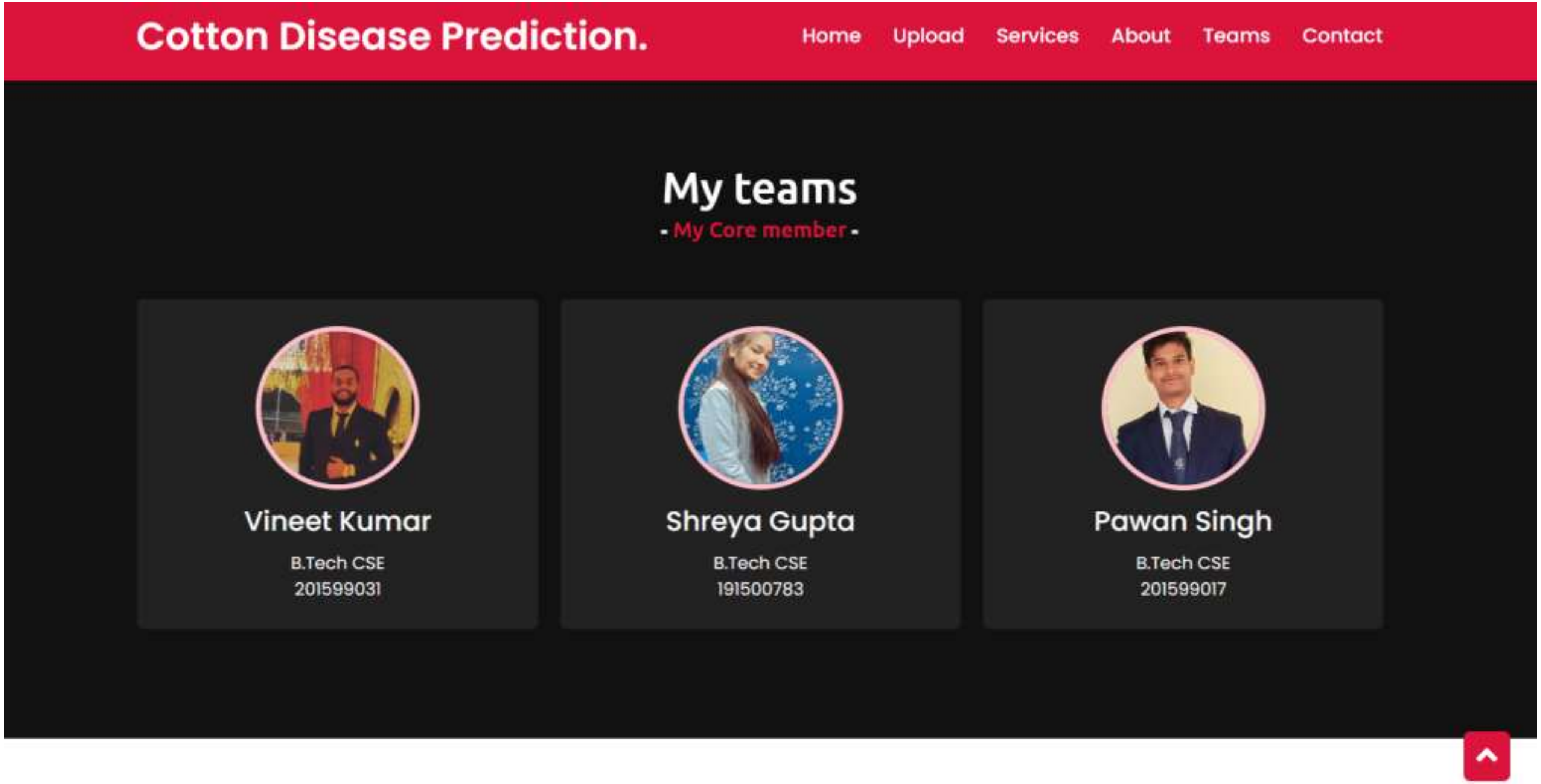
About The Project

what it is

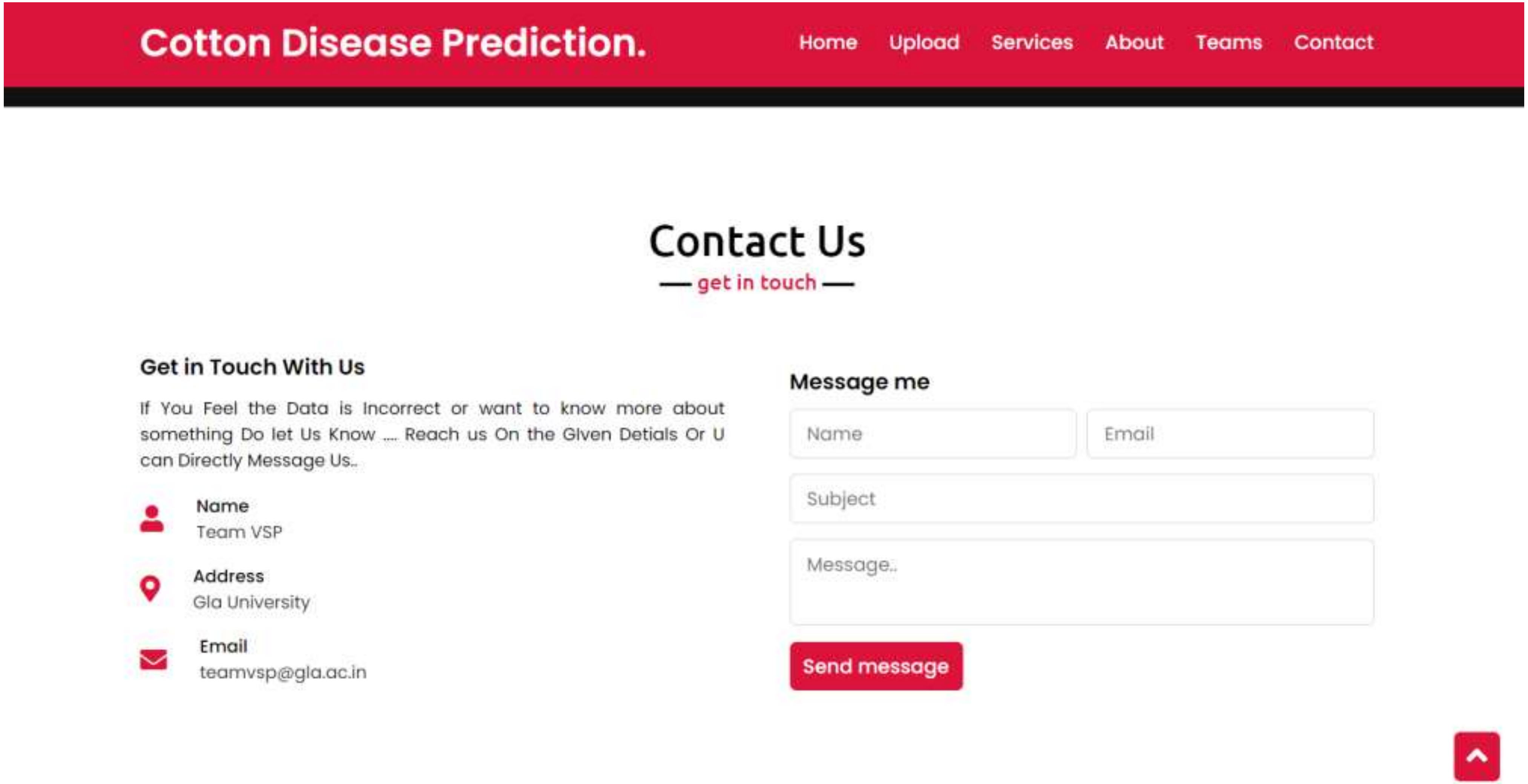
Cotton Disease Prediction

Cotton is one of the economically significant agricultural products in Ethiopia, but it is exposed to different constraints in the leaf area. Mostly, these constraints are identified as diseases and pests that are hard to detect with bare eyes. This study focused to develop a model to boost the detection of cotton leaf disease and pests using the deep learning technique, CNN. To do so, the researchers have used common cotton leaf disease and pests such as bacterial blight, spider mite, and leaf miner. K-fold cross-validation strategy was worn to dataset splitting and boosted generalization of the CNN model. For this research, nearly 2400 specimens (600 images in each class) were accessed for training purposes. This developed model is implemented using python version 3.7.3 and the model is equipped on the deep learning package called Keras, TensorFlow backed, and Jupyter which are used as the developmental environment. This model achieved an accuracy of 96.4% for identifying classes of leaf disease and pests in cotton plants. This revealed the feasibility of its usage in real-time applications and the potential need for IT-based solutions to support traditional or manual disease and pest's identification.

Team Members



Contact Us


Cotton Disease Prediction.
Home
Upload
Services
About
Teams
Contact
Contact Us
get in touch
Get in Touch With Us
If You Feel the Data is Incorrect or want to know more about something Do let Us Know .... Reach us On the Given Detials Or U can Directly Message Us..
Name
Team VSP
Address
Gla University
Email
teamvsp@gla.ac.in
Message me
Name
Email
Subject
Message..
Send message

Future Goal

This area of research appears to have great potential in terms of increased accuracy. It will implemented along with several visualization techniques to detect and classify the symptoms of plant diseases. It will provides a comprehensive explanation of DL models used to visualize various plant diseases. In addition, some research gaps will identify from which to obtain greater transparency for detecting diseases in plants, even before their symptoms appear clearly

Thank you

Presented by G-95

Do you have any questions for us?