In [ ]:

```
'''This code was written by Danny Joel Devarapalli, Dheeraj Mavilla, Prashanth Karri, Harshit Gorijavolu'''
```

In [ ]:

```
from google.colab import drive
drive.mount('/content/drive')
```

```
Go to this URL in a browser: https://accounts.google.com/o/oauth2/auth?cli
ent_id=947318989803-6bn6qk8qdgf4n4g3pfee6491hc0brc4i.apps.googleuserconten
t.com&redirect_uri=urn%3aietf%3awg%3aoauth%3a2.0%3aoob&response_type=code&
scope=email%20https%3a%2f%2fwww.googleapis.com%2fauth%2fdocs.test%20https%
3a%2f%2fwww.googleapis.com%2fauth%2fdrive%20https%3a%2f%2fwww.googleapis.c
om%2fauth%2fdrive.photos.readonly%20https%3a%2f%2fwww.googleapis.com%2faut
h%2fpeopleapi.readonly

Enter your authorization code:
··········
Mounted at /content/drive
```

In [ ]:

```
!pip install -U -q PyDrive
from pydrive.auth import GoogleAuth
from pydrive.drive import GoogleDrive
import google.colab
from oauth2client.client import GoogleCredentials
import glob, os

folder_id = '/content/drive/My Drive/Colab Backup'
dir_to_backup = '/content/drive/My Drive/Colab Backup/Copy Big data'
mycreds_file_contents = '{"access_token": "ya29.a0AfH6SMAfrsGMg6lBqN6LEZW484rDvGid96t
pvYEFLT6pAWJpXNc6WyWDLaW5hKAjju46KDLPUkTD9PMn81Zm9wQN5rTrq3AGQEHZ2GLSwu0gIqs3k3L6oBHU
n2ttGiJFEXYS1FBrAEypIjq-nOb5axCpobIf05ooSfRUiT5NHIhX", "client_id": "32555940559.app
s.googleusercontent.com", "client_secret": "ZmssLNjJy2998hD4CTg2ejr2", "refresh_toke
n": "1//05nkqfGkJ9maPCgYIARAAGAUSNwF-L9IrGmqCGJIBvoEXeyvUaLOBD9teJk_IQ2HFUPk6fhGTqCDh
DW4WKzwi1erXV3c_EsJmMok", "token_expiry": "2020-05-14T17:28:32Z", "token_uri": "http
s://oauth2.googleapis.com/token", "user_agent": "Python client library", "revoke_ur
i": "https://oauth2.googleapis.com/revoke", "id_token": {"iss": "https://accounts.goo
gle.com", "azp": "32555940559.apps.googleusercontent.com", "aud": "32555940559.apps.g
oogleusercontent.com", "sub": "113969137687744701622", "email": "dannydevarapalli6@gm
ail.com", "email_verified": true, "at_hash": "jrgryi6CI3OzNtf3crfE6Q", "iat": 1589473
713, "exp": 1589477313}, "id_token_jwt": "eyJhbGciOiJSUzI1NiIsImtpZCI6ImMxNzcxODE0YmE
2YTcwNjkzZmI5NDEyZGEzYzZlOTBjMmJmNWI5MjciLCJ0eXAiOiJKV1QifQ.eyJpc3MiOiJodHRwczovL2FjY
291bnRzLmdvb2dsZS5jb20iLCJhenAiOiIzMjU1NTk0MDU1OS5hcHBzLmdvb2dsZXVzZXJjb250ZW50LmNvbS
IsImF1ZCI6IjMyNTU1OTQwNTU5LmFwcHMuZ29vZ2xldXNlcmNvbnRlbnQuY29tIiwic3ViIjoiMTEzOTY5MTM
3Njg3NzQ0NzAxNjIyIiwiZW1haWwiOiJkYW5ueWRldmFyYXBhbGxpNkBnbWFpbC5jb20iLCJlbWFpbF92ZXJp
ZmllZCI6dHJ1ZSwiYXRfaGFzaCI6ImpyZ3J5aTZDSTNPek50ZjNjcmZFNlEiLCJpYXQiOjE1ODk0NzM3MTMsI
mV4cCI6MTU4OTQ3NzMxM30.UCYsQbAIYmDfjTH6xaiDB4KSq5fB-ug-DPtyFVMx9ilDSBUlvQ8pxzLHCXMli9
slniRSYrgNrAewp7qHc7skJTpJh34APsVsSB9WyEFnog5G0FAzX0pZ_clObYYl8BMGvXk6weXkF6GHmwSpmFL
ctVs7QEbY0j3ehKyWeDLn4bqyoZcEM_OYSX6ZZj6s6XvJZ-hV84G23gqs0ndzJHBsD5dkXFrGcmFsCNMCfwRq
tKoUoc3H25ceCdSJPlCQLA_Wu72ZddTWqFvgnLJDhnhodfhEZUIYyzAKPZnXdgPTrT9pPkMg8scI0bVQShUOJ
6SOyWLHsR257jhPhS5MapUNBw", "token_response": {"access_token": "ya29.a0AfH6SMAfrsGMg6
lBqN6LEZW484rDvGid96tpvYEFLT6pAWJpXNc6WyWDLaW5hKAjju46KDLPUkTD9PMn81Zm9wQN5rTrq3AGQEH
Z2GLSwu0gIqs3k3L6oBHUn2ttGiJFEXYS1FBrAEypIjq-nOb5axCpobIf05ooSfRUiT5NHIhX", "expires_
in": 3599, "scope": "https://www.googleapis.com/auth/appengine.admin https://www.goog
leapis.com/auth/userinfo.email https://www.googleapis.com/auth/accounts.reauth openid
https://www.googleapis.com/auth/cloud-platform https://www.googleapis.com/auth/comput
e https://www.googleapis.com/auth/drive", "token_type": "Bearer", "id_token": "eyJhbG
ciOiJSUzI1NiIsImtpZCI6ImMxNzcxODE0YmE2YTcwNjkzZmI5NDEyZGEzYzZlOTBjMmJmNWI5MjciLCJ0eXA
iOiJKV1QifQ.eyJpc3MiOiJodHRwczovL2FjY291bnRzLmdvb2dsZS5jb20iLCJhenAiOiIzMjU1NTk0MDU1O
S5hcHBzLmdvb2dsZXVzZXJjb250ZW50LmNvbSIsImF1ZCI6IjMyNTU1OTQwNTU5LmFwcHMuZ29vZ2xldXNlcm
NvbnRlbnQuY29tIiwic3ViIjoiMTEzOTY5MTM3Njg3NzQ0NzAxNjIyIiwiZW1haWwiOiJkYW5ueWRldmFyYXB
hbGxpNkBnbWFpbC5jb20iLCJlbWFpbF92ZXJpZmllZCI6dHJ1ZSwiYXRfaGFzaCI6ImpyZ3J5aTZDSTNPek50
ZjNjcmZFNlEiLCJpYXQiOjE1ODk0NzM3MTMsImV4cCI6MTU4OTQ3NzMxM30.UCYsQbAIYmDfjTH6xaiDB4KSq
5fB-ug-DPtyFVMx9ilDSBUlvQ8pxzLHCXMli9slniRSYrgNrAewp7qHc7skJTpJh34APsVsSB9WyEFnog5G0F
AzX0pZ_clObYYl8BMGvXk6weXkF6GHmwSpmFLctVs7QEbY0j3ehKyWeDLn4bqyoZcEM_OYSX6ZZj6s6XvJZ-h
V84G23gqs0ndzJHBsD5dkXFrGcmFsCNMCfwRqtKoUoc3H25ceCdSJPlCQLA_Wu72ZddTWqFvgnLJDhnhodfhE
ZUIYyzAKPZnXdgPTrT9pPkMg8scI0bVQShUOJ6SOyWLHsR257jhPhS5MapUNBw"}, "scopes": [], "toke
n_info_uri": null, "invalid": false, "_class": "GoogleCredentials", "_module": "oauth
2client.client"}'
mycreds_file = 'mycreds.json'

with open(mycreds_file, 'w') as f:
  f.write(mycreds_file_contents)

def authenticate_pydrive():
  gauth = GoogleAuth()

  # https://stackoverflow.com/a/24542604/5096199
  # Try to load saved client credentials
```

```
  gauth.LoadCredentialsFile(mycreds_file)
  if gauth.credentials is None:
    # Authenticate if they're not there
    google.colab.auth.authenticate_user()
    gauth.credentials = GoogleCredentials.get_application_default()
  elif gauth.access_token_expired:
    # Refresh them if expired
    gauth.Refresh()
  else:
    # Initialize the saved creds
    gauth.Authorize()
  # Save the current credentials to a file
  gauth.SaveCredentialsFile(mycreds_file)

  drive = GoogleDrive(gauth)
  return drive


def backup_pydrive():
  drive = authenticate_pydrive()
  paths = list(glob.iglob(os.path.join(dir_to_backup, '**'), recursive=True))
  print(paths)

  # Delete existing files
  files = drive.ListFile({'q': "'%s' in parents" % folder_id}).GetList()
  for file in files:
    if file['title'] in paths:
      file.Delete()

  for path in paths:
    if os.path.isdir(path) or os.stat(path).st_size == 0:
      continue
    file = drive.CreateFile({'title': path, 'parents':
           [{"kind": "drive#fileLink", "id": folder_id}]})
    file.SetContentFile(path)
    file.Upload()
    print('Backed up %s' % path)

def restore_pydrive():
  drive = authenticate_pydrive()
  files = drive.ListFile({'q': "'%s' in parents" % folder_id}).GetList()
  for file in files:
    os.makedirs(os.path.dirname(file['title']), exist_ok=True)
    file.GetContentFile(file['title'])
    print('Restored %s' % file['title'])

authenticate_pydrive()
!cat {mycreds_file}
```

```
{"access_token": "ya29.a0AfH6SMAfrsGMg6lBqN6LEZW484rDvGid96tpvYEFLT6pAWJpX
Nc6WyWDLaW5hKAjju46KDLPUkTD9PMn81Zm9wQN5rTrq3AGQEHZ2GLSwu0gIqs3k3L6oBHUn2t
tGiJFEXYS1FBrAEypIjq-nOb5axCpobIf05ooSfRUiT5NHIhX", "client_id": "32555940
559.apps.googleusercontent.com", "client_secret": "ZmssLNjJy2998hD4CTg2ejr
2", "refresh_token": "1//05nkqfGkJ9maPCgYIARAAGAUSNwF-L9IrGmqCGJIBvoEXeyvU
aLOBD9teJk_IQ2HFUPk6fhGTqCDhDW4WKzwi1erXV3c_EsJmMok", "token_expiry": nul
l, "token_uri": "https://oauth2.googleapis.com/token", "user_agent": "Pyth
on client library", "revoke_uri": "https://oauth2.googleapis.com/revoke",
"id_token": null, "id_token_jwt": null, "token_response": null, "scopes":
[], "token_info_uri": null, "invalid": false, "_class": "GoogleCredential
s", "_module": "oauth2client.client"}
```

In [ ]:

```
from keras import applications
from keras.layers import Conv2D, MaxPooling2D, Flatten, Dense, Dropout

vgg_model = applications.VGG16(weights='imagenet',
                               include_top=False,
                               input_shape=(224, 224, 3))


layer_dict = dict([(layer.name, layer) for layer in vgg_model.layers])

x = layer_dict['block2_pool'].output

x = Conv2D(filters=64, kernel_size=(3, 3), activation='relu')(x)
x = MaxPooling2D(pool_size=(2, 2))(x)
x = Flatten()(x)
x = Dense(256, activation='relu')(x)
x = Dropout(0.5)(x)
x = Dense(3, activation='softmax')(x)

from keras.models import Model
custom_model = Model(input=vgg_model.input, output=x)

# for layer in custom_model.layers[:7]:
#     layer.trainable = False
from keras.optimizers import Adam
adam = Adam(learning_rate = 0.0001)

custom_model.compile(loss='sparse_categorical_crossentropy',
                     optimizer=adam,
                     metrics=['accuracy'])

custom_model.summary()
```

```
Model: "model_3"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_3 (InputLayer)         (None, 224, 224, 3)       0
_________________________________________________________________
block1_conv1 (Conv2D)        (None, 224, 224, 64)      1792
_________________________________________________________________
block1_conv2 (Conv2D)        (None, 224, 224, 64)      36928
_________________________________________________________________
block1_pool (MaxPooling2D)   (None, 112, 112, 64)      0
_________________________________________________________________
block2_conv1 (Conv2D)        (None, 112, 112, 128)     73856
_________________________________________________________________
block2_conv2 (Conv2D)        (None, 112, 112, 128)     147584
_________________________________________________________________
block2_pool (MaxPooling2D)   (None, 56, 56, 128)       0
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 54, 54, 64)        73792
_________________________________________________________________
max_pooling2d_3 (MaxPooling2 (None, 27, 27, 64)        0
_________________________________________________________________
flatten_3 (Flatten)          (None, 46656)             0
_________________________________________________________________
dense_5 (Dense)              (None, 256)               11944192
_________________________________________________________________
dropout_3 (Dropout)          (None, 256)               0
_________________________________________________________________
dense_6 (Dense)              (None, 3)                 771
=================================================================
Total params: 12,278,915
Trainable params: 12,278,915
Non-trainable params: 0
_________________________________________________________________
```

```
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:21: UserWarni
ng: Update your `Model` call to the Keras 2 API: `Model(inputs=Tensor("i
n..., outputs=Tensor("de...)`
```

In [ ]:

```
import pickle
X = pickle.load(open("/content/drive/My Drive/Skin Cancer DATA/equal/X_equal.pickle",
"rb"))
y = pickle.load(open("/content/drive/My Drive/Skin Cancer DATA/equal/y_equal.pickle",
"rb"))
```

In [ ]:

```
history_vgg = custom_model.fit(X, y, batch_size = 32, epochs = 100, validation_split=
0.2,verbose = 1)



from keras.models import model_from_json
from keras.models import load_model
model_json = custom_model.to_json()
with open("/content/drive/My Drive/Skin Cancer DATA/modelvgg_img.json", "w") as json_fi
le :
        json_file.write(model_json)

custom_model.save_weights("/content/drive/My Drive/Skin Cancer DATA/modelvgg_img.h5")
print("Saved model to disk")

custom_model.save('/content/drive/My Drive/Skin Cancer DATA/CNNvgg_img.model')




import matplotlib.pyplot as plt


print(history_vgg.history.keys())
plt.figure(1)
plt.plot(history_vgg.history['accuracy'])
plt.plot(history_vgg.history['val_accuracy'])
plt.title('VGG16 with ImageNet accuracy')
plt.ylabel('accuracy')
plt.xlabel('epoch')
plt.legend(['Train', 'Validation'], loc='upper left')
plt.show()



print(history_vgg.history.keys())
plt.figure(2)
plt.plot(history_vgg.history['loss'])
plt.plot(history_vgg.history['val_loss'])
plt.title('VGG16 with ImageNet loss')
plt.ylabel('loss')
plt.xlabel('epoch')
plt.legend(['Train', 'Validation'], loc='upper left')
plt.show()
```

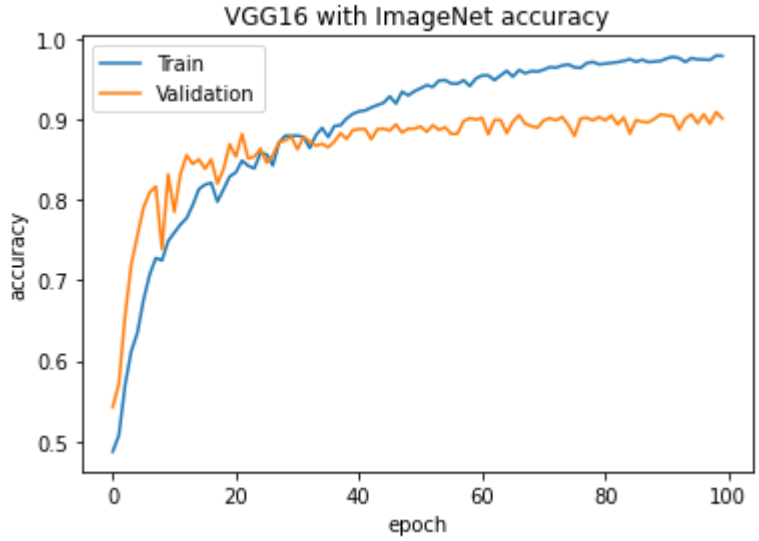
```
Train on 8516 samples, validate on 2129 samples
Epoch 1/100
8516/8516 [==============================] - 37s 4ms/step - loss: 3.8398 -
accuracy: 0.4872 - val_loss: 1.0611 - val_accuracy: 0.5425
Epoch 2/100
8516/8516 [==============================] - 37s 4ms/step - loss: 1.0505 -
accuracy: 0.5074 - val_loss: 0.8764 - val_accuracy: 0.5716
Epoch 3/100
8516/8516 [==============================] - 37s 4ms/step - loss: 0.8839 -
accuracy: 0.5695 - val_loss: 0.7894 - val_accuracy: 0.6552
Epoch 4/100
8516/8516 [==============================] - 37s 4ms/step - loss: 0.8269 -
accuracy: 0.6117 - val_loss: 0.7177 - val_accuracy: 0.7191
Epoch 5/100
8516/8516 [==============================] - 37s 4ms/step - loss: 0.8813 -
accuracy: 0.6346 - val_loss: 0.6677 - val_accuracy: 0.7553
Epoch 6/100
8516/8516 [==============================] - 37s 4ms/step - loss: 0.7237 -
accuracy: 0.6754 - val_loss: 0.6256 - val_accuracy: 0.7900
Epoch 7/100
8516/8516 [==============================] - 37s 4ms/step - loss: 0.6604 -
accuracy: 0.7070 - val_loss: 0.6022 - val_accuracy: 0.8093
Epoch 8/100
8516/8516 [==============================] - 37s 4ms/step - loss: 0.6292 -
accuracy: 0.7276 - val_loss: 0.5302 - val_accuracy: 0.8163
Epoch 9/100
8516/8516 [==============================] - 37s 4ms/step - loss: 0.6057 -
accuracy: 0.7246 - val_loss: 0.6644 - val_accuracy: 0.7388
Epoch 10/100
8516/8516 [==============================] - 37s 4ms/step - loss: 0.5645 -
accuracy: 0.7486 - val_loss: 0.5414 - val_accuracy: 0.8314
Epoch 11/100
8516/8516 [==============================] - 37s 4ms/step - loss: 0.5598 -
accuracy: 0.7592 - val_loss: 0.6223 - val_accuracy: 0.7849
Epoch 12/100
8516/8516 [==============================] - 37s 4ms/step - loss: 0.5347 -
accuracy: 0.7693 - val_loss: 0.5186 - val_accuracy: 0.8318
Epoch 13/100
8516/8516 [==============================] - 37s 4ms/step - loss: 0.5207 -
accuracy: 0.7770 - val_loss: 0.5029 - val_accuracy: 0.8553
Epoch 14/100
8516/8516 [==============================] - 37s 4ms/step - loss: 0.4916 -
accuracy: 0.7934 - val_loss: 0.4929 - val_accuracy: 0.8445
Epoch 15/100
8516/8516 [==============================] - 37s 4ms/step - loss: 0.4321 -
accuracy: 0.8133 - val_loss: 0.4895 - val_accuracy: 0.8502
Epoch 16/100
8516/8516 [==============================] - 37s 4ms/step - loss: 0.4351 -
accuracy: 0.8188 - val_loss: 0.5385 - val_accuracy: 0.8384
Epoch 17/100
8516/8516 [==============================] - 37s 4ms/step - loss: 0.4190 -
accuracy: 0.8209 - val_loss: 0.4788 - val_accuracy: 0.8502
Epoch 18/100
8516/8516 [==============================] - 37s 4ms/step - loss: 0.4883 -
accuracy: 0.7977 - val_loss: 0.5131 - val_accuracy: 0.8201
Epoch 19/100
8516/8516 [==============================] - 37s 4ms/step - loss: 0.4607 -
accuracy: 0.8129 - val_loss: 0.5540 - val_accuracy: 0.8380
Epoch 20/100
8516/8516 [==============================] - 37s 4ms/step - loss: 0.4042 -
accuracy: 0.8289 - val_loss: 0.4403 - val_accuracy: 0.8690
```

```
Epoch 21/100
8516/8516 [==============================] - 37s 4ms/step - loss: 0.4225 -
accuracy: 0.8340 - val_loss: 0.5498 - val_accuracy: 0.8535
Epoch 22/100
8516/8516 [==============================] - 37s 4ms/step - loss: 0.3442 -
accuracy: 0.8485 - val_loss: 0.4154 - val_accuracy: 0.8816
Epoch 23/100
8516/8516 [==============================] - 37s 4ms/step - loss: 0.3729 -
accuracy: 0.8422 - val_loss: 0.4603 - val_accuracy: 0.8511
Epoch 24/100
8516/8516 [==============================] - 37s 4ms/step - loss: 0.3776 -
accuracy: 0.8389 - val_loss: 0.5281 - val_accuracy: 0.8535
Epoch 25/100
8516/8516 [==============================] - 37s 4ms/step - loss: 0.3390 -
accuracy: 0.8581 - val_loss: 0.4488 - val_accuracy: 0.8633
Epoch 26/100
8516/8516 [==============================] - 37s 4ms/step - loss: 0.3472 -
accuracy: 0.8564 - val_loss: 0.4553 - val_accuracy: 0.8459
Epoch 27/100
8516/8516 [==============================] - 37s 4ms/step - loss: 0.3881 -
accuracy: 0.8426 - val_loss: 0.5034 - val_accuracy: 0.8553
Epoch 28/100
8516/8516 [==============================] - 37s 4ms/step - loss: 0.3060 -
accuracy: 0.8705 - val_loss: 0.4838 - val_accuracy: 0.8713
Epoch 29/100
8516/8516 [==============================] - 37s 4ms/step - loss: 0.2823 -
accuracy: 0.8795 - val_loss: 0.4975 - val_accuracy: 0.8736
Epoch 30/100
8516/8516 [==============================] - 37s 4ms/step - loss: 0.2883 -
accuracy: 0.8794 - val_loss: 0.4659 - val_accuracy: 0.8783
Epoch 31/100
8516/8516 [==============================] - 37s 4ms/step - loss: 0.2869 -
accuracy: 0.8795 - val_loss: 0.5246 - val_accuracy: 0.8628
Epoch 32/100
8516/8516 [==============================] - 37s 4ms/step - loss: 0.2889 -
accuracy: 0.8782 - val_loss: 0.5521 - val_accuracy: 0.8779
Epoch 33/100
8516/8516 [==============================] - 37s 4ms/step - loss: 0.3354 -
accuracy: 0.8643 - val_loss: 0.4495 - val_accuracy: 0.8727
Epoch 34/100
8516/8516 [==============================] - 37s 4ms/step - loss: 0.3134 -
accuracy: 0.8801 - val_loss: 0.4597 - val_accuracy: 0.8671
Epoch 35/100
8516/8516 [==============================] - 37s 4ms/step - loss: 0.2752 -
accuracy: 0.8893 - val_loss: 0.4953 - val_accuracy: 0.8694
Epoch 36/100
8516/8516 [==============================] - 37s 4ms/step - loss: 0.3599 -
accuracy: 0.8779 - val_loss: 0.5549 - val_accuracy: 0.8652
Epoch 37/100
8516/8516 [==============================] - 37s 4ms/step - loss: 0.2976 -
accuracy: 0.8915 - val_loss: 0.4424 - val_accuracy: 0.8722
Epoch 38/100
8516/8516 [==============================] - 37s 4ms/step - loss: 0.2770 -
accuracy: 0.8924 - val_loss: 0.5015 - val_accuracy: 0.8826
Epoch 39/100
8516/8516 [==============================] - 37s 4ms/step - loss: 0.2511 -
accuracy: 0.9010 - val_loss: 0.4892 - val_accuracy: 0.8755
Epoch 40/100
8516/8516 [==============================] - 37s 4ms/step - loss: 0.2365 -
accuracy: 0.9066 - val_loss: 0.6941 - val_accuracy: 0.8863
Epoch 41/100
```

```
8516/8516 [==============================] - 37s 4ms/step - loss: 0.2218 -
accuracy: 0.9097 - val_loss: 0.5586 - val_accuracy: 0.8877
Epoch 42/100
8516/8516 [==============================] - 37s 4ms/step - loss: 0.2095 -
accuracy: 0.9109 - val_loss: 0.5013 - val_accuracy: 0.8877
Epoch 43/100
8516/8516 [==============================] - 37s 4ms/step - loss: 0.2099 -
accuracy: 0.9145 - val_loss: 0.6295 - val_accuracy: 0.8751
Epoch 44/100
8516/8516 [==============================] - 37s 4ms/step - loss: 0.2019 -
accuracy: 0.9176 - val_loss: 0.6273 - val_accuracy: 0.8877
Epoch 45/100
8516/8516 [==============================] - 37s 4ms/step - loss: 0.1978 -
accuracy: 0.9204 - val_loss: 0.6410 - val_accuracy: 0.8882
Epoch 46/100
8516/8516 [==============================] - 37s 4ms/step - loss: 0.1798 -
accuracy: 0.9284 - val_loss: 0.5585 - val_accuracy: 0.8859
Epoch 47/100
8516/8516 [==============================] - 37s 4ms/step - loss: 0.2078 -
accuracy: 0.9194 - val_loss: 0.4713 - val_accuracy: 0.8934
Epoch 48/100
8516/8516 [==============================] - 37s 4ms/step - loss: 0.1556 -
accuracy: 0.9342 - val_loss: 0.5823 - val_accuracy: 0.8835
Epoch 49/100
8516/8516 [==============================] - 37s 4ms/step - loss: 0.1758 -
accuracy: 0.9295 - val_loss: 0.4860 - val_accuracy: 0.8882
Epoch 50/100
8516/8516 [==============================] - 37s 4ms/step - loss: 0.1684 -
accuracy: 0.9347 - val_loss: 0.5733 - val_accuracy: 0.8882
Epoch 51/100
8516/8516 [==============================] - 37s 4ms/step - loss: 0.1577 -
accuracy: 0.9384 - val_loss: 0.4859 - val_accuracy: 0.8910
Epoch 52/100
8516/8516 [==============================] - 37s 4ms/step - loss: 0.1350 -
accuracy: 0.9422 - val_loss: 0.4560 - val_accuracy: 0.8845
Epoch 53/100
8516/8516 [==============================] - 37s 4ms/step - loss: 0.1431 -
accuracy: 0.9401 - val_loss: 0.4698 - val_accuracy: 0.8924
Epoch 54/100
8516/8516 [==============================] - 37s 4ms/step - loss: 0.1363 -
accuracy: 0.9480 - val_loss: 0.5775 - val_accuracy: 0.8868
Epoch 55/100
8516/8516 [==============================] - 37s 4ms/step - loss: 0.1249 -
accuracy: 0.9484 - val_loss: 0.6678 - val_accuracy: 0.8901
Epoch 56/100
8516/8516 [==============================] - 37s 4ms/step - loss: 0.1331 -
accuracy: 0.9443 - val_loss: 0.7072 - val_accuracy: 0.8816
Epoch 57/100
8516/8516 [==============================] - 37s 4ms/step - loss: 0.1454 -
accuracy: 0.9441 - val_loss: 0.7836 - val_accuracy: 0.8821
Epoch 58/100
8516/8516 [==============================] - 37s 4ms/step - loss: 0.1213 -
accuracy: 0.9483 - val_loss: 0.6456 - val_accuracy: 0.8976
Epoch 59/100
8516/8516 [==============================] - 37s 4ms/step - loss: 0.1391 -
accuracy: 0.9412 - val_loss: 0.5578 - val_accuracy: 0.9014
Epoch 60/100
8516/8516 [==============================] - 37s 4ms/step - loss: 0.1260 -
accuracy: 0.9507 - val_loss: 0.4681 - val_accuracy: 0.8995
Epoch 61/100
8516/8516 [==============================] - 37s 4ms/step - loss: 0.1071 -
```

```
accuracy: 0.9543 - val_loss: 0.5903 - val_accuracy: 0.9018
Epoch 62/100
8516/8516 [==============================] - 37s 4ms/step - loss: 0.1109 -
accuracy: 0.9544 - val_loss: 0.7271 - val_accuracy: 0.8812
Epoch 63/100
8516/8516 [==============================] - 37s 4ms/step - loss: 0.1159 -
accuracy: 0.9484 - val_loss: 0.5420 - val_accuracy: 0.8990
Epoch 64/100
8516/8516 [==============================] - 37s 4ms/step - loss: 0.1189 -
accuracy: 0.9544 - val_loss: 0.5858 - val_accuracy: 0.8990
Epoch 65/100
8516/8516 [==============================] - 37s 4ms/step - loss: 0.1035 -
accuracy: 0.9600 - val_loss: 0.7683 - val_accuracy: 0.8826
Epoch 66/100
8516/8516 [==============================] - 37s 4ms/step - loss: 0.1212 -
accuracy: 0.9528 - val_loss: 0.5785 - val_accuracy: 0.8971
Epoch 67/100
8516/8516 [==============================] - 37s 4ms/step - loss: 0.0975 -
accuracy: 0.9611 - val_loss: 0.6071 - val_accuracy: 0.9051
Epoch 68/100
8516/8516 [==============================] - 37s 4ms/step - loss: 0.1128 -
accuracy: 0.9570 - val_loss: 0.5675 - val_accuracy: 0.8943
Epoch 69/100
8516/8516 [==============================] - 37s 4ms/step - loss: 0.1024 -
accuracy: 0.9597 - val_loss: 0.6415 - val_accuracy: 0.8915
Epoch 70/100
8516/8516 [==============================] - 37s 4ms/step - loss: 0.0948 -
accuracy: 0.9591 - val_loss: 0.6801 - val_accuracy: 0.8896
Epoch 71/100
8516/8516 [==============================] - 37s 4ms/step - loss: 0.1027 -
accuracy: 0.9615 - val_loss: 0.5828 - val_accuracy: 0.8985
Epoch 72/100
8516/8516 [==============================] - 37s 4ms/step - loss: 0.0822 -
accuracy: 0.9647 - val_loss: 0.6956 - val_accuracy: 0.9014
Epoch 73/100
8516/8516 [==============================] - 37s 4ms/step - loss: 0.0941 -
accuracy: 0.9641 - val_loss: 0.6289 - val_accuracy: 0.8985
Epoch 74/100
8516/8516 [==============================] - 37s 4ms/step - loss: 0.0801 -
accuracy: 0.9667 - val_loss: 0.5413 - val_accuracy: 0.9028
Epoch 75/100
8516/8516 [==============================] - 37s 4ms/step - loss: 0.0918 -
accuracy: 0.9681 - val_loss: 0.8941 - val_accuracy: 0.8929
Epoch 76/100
8516/8516 [==============================] - 37s 4ms/step - loss: 0.1098 -
accuracy: 0.9644 - val_loss: 0.7770 - val_accuracy: 0.8793
Epoch 77/100
8516/8516 [==============================] - 37s 4ms/step - loss: 0.1215 -
accuracy: 0.9637 - val_loss: 0.6464 - val_accuracy: 0.9009
Epoch 78/100
8516/8516 [==============================] - 37s 4ms/step - loss: 0.0716 -
accuracy: 0.9697 - val_loss: 0.5458 - val_accuracy: 0.9018
Epoch 79/100
8516/8516 [==============================] - 37s 4ms/step - loss: 0.0784 -
accuracy: 0.9711 - val_loss: 0.6515 - val_accuracy: 0.8985
Epoch 80/100
8516/8516 [==============================] - 37s 4ms/step - loss: 0.0791 -
accuracy: 0.9681 - val_loss: 0.6336 - val_accuracy: 0.9028
Epoch 81/100
8516/8516 [==============================] - 37s 4ms/step - loss: 0.0720 -
accuracy: 0.9690 - val_loss: 0.6039 - val_accuracy: 0.8985
```

```
Epoch 82/100
8516/8516 [==============================] - 37s 4ms/step - loss: 0.0755 -
accuracy: 0.9699 - val_loss: 0.6074 - val_accuracy: 0.9047
Epoch 83/100
8516/8516 [==============================] - 37s 4ms/step - loss: 0.0666 -
accuracy: 0.9709 - val_loss: 0.9204 - val_accuracy: 0.8934
Epoch 84/100
8516/8516 [==============================] - 37s 4ms/step - loss: 0.0679 -
accuracy: 0.9724 - val_loss: 0.8836 - val_accuracy: 0.9023
Epoch 85/100
8516/8516 [==============================] - 37s 4ms/step - loss: 0.0700 -
accuracy: 0.9745 - val_loss: 0.6668 - val_accuracy: 0.8821
Epoch 86/100
8516/8516 [==============================] - 37s 4ms/step - loss: 0.0722 -
accuracy: 0.9712 - val_loss: 0.7537 - val_accuracy: 0.8990
Epoch 87/100
8516/8516 [==============================] - 37s 4ms/step - loss: 0.0631 -
accuracy: 0.9739 - val_loss: 0.7058 - val_accuracy: 0.8967
Epoch 88/100
8516/8516 [==============================] - 37s 4ms/step - loss: 0.0816 -
accuracy: 0.9709 - val_loss: 0.6889 - val_accuracy: 0.8962
Epoch 89/100
8516/8516 [==============================] - 37s 4ms/step - loss: 0.0738 -
accuracy: 0.9716 - val_loss: 0.5718 - val_accuracy: 0.9004
Epoch 90/100
8516/8516 [==============================] - 37s 4ms/step - loss: 0.0681 -
accuracy: 0.9724 - val_loss: 0.7415 - val_accuracy: 0.9061
Epoch 91/100
8516/8516 [==============================] - 37s 4ms/step - loss: 0.0582 -
accuracy: 0.9756 - val_loss: 0.7053 - val_accuracy: 0.9047
Epoch 92/100
8516/8516 [==============================] - 37s 4ms/step - loss: 0.0592 -
accuracy: 0.9775 - val_loss: 0.7664 - val_accuracy: 0.9037
Epoch 93/100
8516/8516 [==============================] - 37s 4ms/step - loss: 0.0664 -
accuracy: 0.9759 - val_loss: 0.7811 - val_accuracy: 0.8873
Epoch 94/100
8516/8516 [==============================] - 37s 4ms/step - loss: 0.0734 -
accuracy: 0.9710 - val_loss: 0.7494 - val_accuracy: 0.9009
Epoch 95/100
8516/8516 [==============================] - 37s 4ms/step - loss: 0.0549 -
accuracy: 0.9758 - val_loss: 0.6171 - val_accuracy: 0.9061
Epoch 96/100
8516/8516 [==============================] - 37s 4ms/step - loss: 0.0693 -
accuracy: 0.9743 - val_loss: 0.5671 - val_accuracy: 0.8948
Epoch 97/100
8516/8516 [==============================] - 37s 4ms/step - loss: 0.0649 -
accuracy: 0.9742 - val_loss: 0.7489 - val_accuracy: 0.9061
Epoch 98/100
8516/8516 [==============================] - 37s 4ms/step - loss: 0.0612 -
accuracy: 0.9735 - val_loss: 0.6005 - val_accuracy: 0.8943
Epoch 99/100
8516/8516 [==============================] - 37s 4ms/step - loss: 0.0515 -
accuracy: 0.9790 - val_loss: 0.9055 - val_accuracy: 0.9089
Epoch 100/100
8516/8516 [==============================] - 37s 4ms/step - loss: 0.0533 -
accuracy: 0.9785 - val_loss: 0.6426 - val_accuracy: 0.9009
Saved model to disk
dict_keys(['val_loss', 'val_accuracy', 'loss', 'accuracy'])
```




```
dict_keys(['val_loss', 'val_accuracy', 'loss', 'accuracy'])
```