Class Challenge: Image Classification of COVID-19 X-rays

Task 2: AlexNet [Total points: 30]

Setup

- This assignment involves the following packages: 'matplotlib', 'numpy', and 'sklearn'.
- If you are using conda, use the following commands to install the above packages:

```
conda install matplotlib
conda install numpy
conda install -c anaconda scikit-learn
```

- If you are using pip, use use the following commands to install the above packages:

```
pip install matplotlib
pip install numpy
pip install sklearn
```

Data

Please download the data using the following link: COVID-19 (https://drive.google.com/file/d/1Y88tgqpQ1Pjko_7rntcPowOJs_QNOrJ-/view).

- After downloading 'Covid_Data_GradientCrescent.zip', unzip the file and you should see the following data structure:

|--all
|--------train
|--------test
|--two
|--------train
|--------test

- Put the 'all' folder, the 'two' folder and this python notebook in the **same directory** so that the following code can correctly locate the data.

[20 points] Multi-class Classification

In [1]:

```
import os

import tensorflow as tf
import numpy as np
import matplotlib.pyplot as plt
from tensorflow.keras.preprocessing.image import ImageDataGenerator

# os.environ['OMP_NUM_THREADS'] = '1'
# os.environ["CUDA_DEVICE_ORDER"]="PCI_BUS_ID"   # see issue #152
# os.environ["CUDA_VISIBLE_DEVICES"]="0"
tf.__version__
```

Out[1]:

```
'2.6.0'
```

In [2]:

```
print("Num GPUs Available: ", len(tf.config.list_physical_devices('GPU')))
tf.config.list_physical_devices('GPU')

from tensorflow.python.client import device_lib
print(device_lib.list_local_devices())
```

```
Num GPUs Available:  1
[name: "/device:CPU:0"
device_type: "CPU"
memory_limit: 268435456
locality {
}
incarnation: 12726373218004576618
, name: "/device:GPU:0"
device_type: "GPU"
memory_limit: 6252920832
locality {
  bus_id: 1
  links {
  }
}
incarnation: 8655891243073644095
physical_device_desc: "device: 0, name: NVIDIA GeForce RTX 2080 SUPE
R, pci bus id: 0000:01:00.0, compute capability: 7.5"
]
```

Load Image Data

In [3]:

```
DATA_LIST = os.listdir('all/train')
DATASET_PATH  = 'all/train'
TEST_DIR =  'all/test'
IMAGE_SIZE    = (224, 224)
NUM_CLASSES   = len(DATA_LIST)
BATCH_SIZE    = 10  # try reducing batch size or freeze more layers if your GPU runs out of memory
NUM_EPOCHS    = 100
LEARNING_RATE = 0.0001 # start off with high rate first 0.001 and experiment with reducing it gradually
```

Generate Training and Validation Batches

In [4]:

```
train_datagen = ImageDataGenerator(rescale=1./255,rotation_range=50,featurewise_
center = True,
                                   featurewise_std_normalization = True,width_sh
ift_range=0.2,
                                   height_shift_range=0.2,shear_range=0.25,zoom_
range=0.1,
                                   zca_whitening = True,channel_shift_range = 20
,
                                   horizontal_flip = True,vertical_flip = True,
                                   validation_split = 0.2,fill_mode='constant')


train_batches = train_datagen.flow_from_directory(DATASET_PATH,target_size=IMAGE
_SIZE,
                                                  shuffle=True,batch_size=BATCH_
SIZE,
                                                  subset = "training",seed=42,
                                                  class_mode="categorical")

valid_batches = train_datagen.flow_from_directory(DATASET_PATH,target_size=IMAGE
_SIZE,
                                                  shuffle=True,batch_size=BATCH_
SIZE,
                                                  subset = "validation",
                                                  seed=42,class_mode="categorica
l")
```

```
Found 216 images belonging to 4 classes.
Found 54 images belonging to 4 classes.
```

```
C:\Users\Li\anaconda3\lib\site-packages\keras_preprocessing\image\im
age_data_generator.py:342: UserWarning: This ImageDataGenerator spec
ifies `zca_whitening` which overrides setting of`featurewise_std_nor
malization`.
  warnings.warn('This ImageDataGenerator specifies '
```

[10 points] Build Model

Hint: Starting from a pre-trained model typically helps performance on a new task, e.g. starting with weights obtained by training on ImageNet.

In [6]:

```
from tensorflow import keras
model = keras.models.Sequential([
    keras.layers.Conv2D(filters=96, kernel_size=(11,11), strides=(4,4), activati
on='relu', input_shape=(224,224,3)),
    keras.layers.BatchNormalization(),
    keras.layers.MaxPool2D(pool_size=(3,3), strides=(2,2)),
    keras.layers.Conv2D(filters=256, kernel_size=(5,5), strides=(1,1), activatio
n='relu', padding="same"),
    keras.layers.BatchNormalization(),
    keras.layers.MaxPool2D(pool_size=(3,3), strides=(2,2)),
    keras.layers.Conv2D(filters=384, kernel_size=(3,3), strides=(1,1), activatio
n='relu', padding="same"),
    keras.layers.BatchNormalization(),
    keras.layers.Conv2D(filters=384, kernel_size=(3,3), strides=(1,1), activatio
n='relu', padding="same"),
    keras.layers.BatchNormalization(),
    keras.layers.Conv2D(filters=256, kernel_size=(3,3), strides=(1,1), activatio
n='relu', padding="same"),
    keras.layers.BatchNormalization(),
    keras.layers.MaxPool2D(pool_size=(3,3), strides=(2,2)),
    keras.layers.Flatten(),
    keras.layers.Dense(4096, activation='relu'),
    keras.layers.Dropout(0.5),
    keras.layers.Dense(4096, activation='relu'),
    keras.layers.Dropout(0.5),
    keras.layers.Dense(4, activation='softmax')
])

model.compile(loss='categorical_crossentropy', optimizer=tf.keras.optimizers.Ada
m(lr=LEARNING_RATE),metrics=['acc'])
```

```
C:\Users\Li\anaconda3\lib\site-packages\keras\optimizer_v2\optimizer
_v2.py:356: UserWarning: The `lr` argument is deprecated, use `learn
ing_rate` instead.
  "The `lr` argument is deprecated, use `learning_rate` instead.")
```

In [7]:

```
print(model.summary())
```

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #   
=================================================================
conv2d (Conv2D)              (None, 54, 54, 96)        34944     
_________________________________________________________________
batch_normalization (BatchNo (None, 54, 54, 96)        384       
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 26, 26, 96)        0         
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 26, 26, 256)       614656    
_________________________________________________________________
batch_normalization_1 (Batch (None, 26, 26, 256)       1024      
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 12, 12, 256)       0         
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 12, 12, 384)       885120    
_________________________________________________________________
batch_normalization_2 (Batch (None, 12, 12, 384)       1536      
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 12, 12, 384)       1327488   
_________________________________________________________________
batch_normalization_3 (Batch (None, 12, 12, 384)       1536      
_________________________________________________________________
conv2d_4 (Conv2D)            (None, 12, 12, 256)       884992    
_________________________________________________________________
batch_normalization_4 (Batch (None, 12, 12, 256)       1024      
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 5, 5, 256)         0         
_________________________________________________________________
flatten (Flatten)            (None, 6400)              0         
_________________________________________________________________
dense (Dense)                (None, 4096)              26218496  
_________________________________________________________________
dropout (Dropout)            (None, 4096)              0         
_________________________________________________________________
dense_1 (Dense)              (None, 4096)              16781312  
_________________________________________________________________
dropout_1 (Dropout)          (None, 4096)              0         
_________________________________________________________________
dense_2 (Dense)              (None, 4)                 16388     
=================================================================
Total params: 46,768,900
Trainable params: 46,766,148
Non-trainable params: 2,752
_________________________________________________________________
None
```

[5 points] Train Model

In [9]:

```
#FIT MODEL
print(len(train_batches))
print(len(valid_batches))

STEP_SIZE_TRAIN=train_batches.n//train_batches.batch_size
STEP_SIZE_VALID=valid_batches.n//valid_batches.batch_size

history=model.fit_generator(train_batches, steps_per_epoch =STEP_SIZE_TRAIN, val
idation_data = valid_batches, validation_steps = STEP_SIZE_VALID, epochs= NUM_EP
OCHS)
```

```
C:\Users\Li\anaconda3\lib\site-packages\keras\engine\training.py:197
2: UserWarning: `Model.fit_generator` is deprecated and will be remo
ved in a future version. Please use `Model.fit`, which supports gene
rators.
  warnings.warn('`Model.fit_generator` is deprecated and '
```

```
22
6
```

```
C:\Users\Li\anaconda3\lib\site-packages\keras_preprocessing\image\im
age_data_generator.py:720: UserWarning: This ImageDataGenerator spec
ifies `featurewise_center`, but it hasn't been fit on any training d
ata. Fit it first by calling `.fit(numpy_data)`.
  warnings.warn('This ImageDataGenerator specifies '
C:\Users\Li\anaconda3\lib\site-packages\keras_preprocessing\image\im
age_data_generator.py:739: UserWarning: This ImageDataGenerator spec
ifies `zca_whitening`, but it hasn't been fit on any training data.
Fit it first by calling `.fit(numpy_data)`.
  warnings.warn('This ImageDataGenerator specifies '
```

```
Epoch 1/100
21/21 [==============================] - 8s 255ms/step - loss: 4.82
15 - acc: 0.3204 - val_loss: 1.4077 - val_acc: 0.2400
Epoch 2/100
21/21 [==============================] - 5s 239ms/step - loss: 4.12
97 - acc: 0.3398 - val_loss: 1.5585 - val_acc: 0.2800
Epoch 3/100
21/21 [==============================] - 5s 238ms/step - loss: 2.73
25 - acc: 0.4369 - val_loss: 1.7217 - val_acc: 0.2600
Epoch 4/100
21/21 [==============================] - 5s 235ms/step - loss: 2.74
55 - acc: 0.4369 - val_loss: 1.9837 - val_acc: 0.2000
Epoch 5/100
21/21 [==============================] - 5s 234ms/step - loss: 2.50
71 - acc: 0.4563 - val_loss: 3.0141 - val_acc: 0.2400
Epoch 6/100
21/21 [==============================] - 5s 237ms/step - loss: 2.52
31 - acc: 0.4369 - val_loss: 4.0515 - val_acc: 0.2000
Epoch 7/100
21/21 [==============================] - 5s 237ms/step - loss: 2.58
52 - acc: 0.4660 - val_loss: 4.0158 - val_acc: 0.2200
```

```
Epoch 8/100
21/21 [==============================] - 5s 240ms/step - loss: 2.59
48 - acc: 0.5340 - val_loss: 4.0584 - val_acc: 0.2600
Epoch 9/100
21/21 [==============================] - 5s 237ms/step - loss: 2.51
59 - acc: 0.5243 - val_loss: 3.2596 - val_acc: 0.2400
Epoch 10/100
21/21 [==============================] - 5s 233ms/step - loss: 2.40
16 - acc: 0.5243 - val_loss: 3.0076 - val_acc: 0.3400
Epoch 11/100
21/21 [==============================] - 5s 238ms/step - loss: 2.45
81 - acc: 0.4854 - val_loss: 1.6260 - val_acc: 0.3000
Epoch 12/100
21/21 [==============================] - 5s 231ms/step - loss: 2.18
67 - acc: 0.5000 - val_loss: 5.4962 - val_acc: 0.2800
Epoch 13/100
21/21 [==============================] - 5s 237ms/step - loss: 1.75
52 - acc: 0.5534 - val_loss: 4.2334 - val_acc: 0.3400
Epoch 14/100
21/21 [==============================] - 5s 242ms/step - loss: 1.61
11 - acc: 0.6117 - val_loss: 3.8106 - val_acc: 0.3200
Epoch 15/100
21/21 [==============================] - 5s 235ms/step - loss: 1.92
08 - acc: 0.5680 - val_loss: 3.2543 - val_acc: 0.3800
Epoch 16/100
21/21 [==============================] - 5s 230ms/step - loss: 1.49
07 - acc: 0.5340 - val_loss: 5.6091 - val_acc: 0.2200
Epoch 17/100
21/21 [==============================] - 5s 238ms/step - loss: 1.71
11 - acc: 0.5194 - val_loss: 3.6573 - val_acc: 0.3400
Epoch 18/100
21/21 [==============================] - 5s 238ms/step - loss: 1.61
99 - acc: 0.5485 - val_loss: 2.0417 - val_acc: 0.4600
Epoch 19/100
21/21 [==============================] - 5s 232ms/step - loss: 1.73
17 - acc: 0.4709 - val_loss: 7.6606 - val_acc: 0.2600
Epoch 20/100
21/21 [==============================] - 5s 236ms/step - loss: 1.50
74 - acc: 0.5922 - val_loss: 5.5769 - val_acc: 0.3000
Epoch 21/100
21/21 [==============================] - 5s 241ms/step - loss: 1.61
57 - acc: 0.5534 - val_loss: 4.2102 - val_acc: 0.2000
Epoch 22/100
21/21 [==============================] - 5s 238ms/step - loss: 1.48
43 - acc: 0.5583 - val_loss: 5.7110 - val_acc: 0.2200
Epoch 23/100
21/21 [==============================] - 5s 235ms/step - loss: 1.53
78 - acc: 0.5534 - val_loss: 4.3639 - val_acc: 0.4400
Epoch 24/100
21/21 [==============================] - 5s 231ms/step - loss: 1.47
35 - acc: 0.5728 - val_loss: 2.2649 - val_acc: 0.4000
Epoch 25/100
21/21 [==============================] - 5s 237ms/step - loss: 1.42
```

```
72 - acc: 0.5777 - val_loss: 1.4866 - val_acc: 0.3600
Epoch 26/100
21/21 [==============================] - 5s 236ms/step - loss: 1.32
18 - acc: 0.5631 - val_loss: 1.5364 - val_acc: 0.4600
Epoch 27/100
21/21 [==============================] - 5s 235ms/step - loss: 1.27
39 - acc: 0.5728 - val_loss: 3.7323 - val_acc: 0.3200
Epoch 28/100
21/21 [==============================] - 5s 236ms/step - loss: 1.21
45 - acc: 0.6117 - val_loss: 2.6018 - val_acc: 0.2600
Epoch 29/100
21/21 [==============================] - 5s 238ms/step - loss: 1.17
25 - acc: 0.6456 - val_loss: 2.9251 - val_acc: 0.4800
Epoch 30/100
21/21 [==============================] - 5s 226ms/step - loss: 1.29
45 - acc: 0.5291 - val_loss: 1.1911 - val_acc: 0.6000
Epoch 31/100
21/21 [==============================] - 5s 233ms/step - loss: 1.32
50 - acc: 0.5340 - val_loss: 2.3267 - val_acc: 0.3600
Epoch 32/100
21/21 [==============================] - 5s 237ms/step - loss: 1.02
52 - acc: 0.5825 - val_loss: 1.6893 - val_acc: 0.4800
Epoch 33/100
21/21 [==============================] - 5s 238ms/step - loss: 1.08
42 - acc: 0.5922 - val_loss: 3.1281 - val_acc: 0.3800
Epoch 34/100
21/21 [==============================] - 5s 238ms/step - loss: 1.12
28 - acc: 0.6019 - val_loss: 1.5977 - val_acc: 0.4600
Epoch 35/100
21/21 [==============================] - 5s 241ms/step - loss: 1.03
67 - acc: 0.6117 - val_loss: 2.8055 - val_acc: 0.3600
Epoch 36/100
21/21 [==============================] - 5s 233ms/step - loss: 1.08
97 - acc: 0.5680 - val_loss: 2.6730 - val_acc: 0.3000
Epoch 37/100
21/21 [==============================] - 5s 239ms/step - loss: 0.86
13 - acc: 0.6359 - val_loss: 4.1116 - val_acc: 0.2000
Epoch 38/100
21/21 [==============================] - 5s 236ms/step - loss: 1.06
99 - acc: 0.6117 - val_loss: 2.4060 - val_acc: 0.4200
Epoch 39/100
21/21 [==============================] - 5s 235ms/step - loss: 0.79
67 - acc: 0.6942 - val_loss: 2.4584 - val_acc: 0.3600
Epoch 40/100
21/21 [==============================] - 5s 235ms/step - loss: 0.96
40 - acc: 0.6165 - val_loss: 1.2635 - val_acc: 0.5800
Epoch 41/100
21/21 [==============================] - 5s 236ms/step - loss: 1.08
71 - acc: 0.5971 - val_loss: 2.2130 - val_acc: 0.4400
Epoch 42/100
21/21 [==============================] - 5s 236ms/step - loss: 0.98
06 - acc: 0.6359 - val_loss: 1.7818 - val_acc: 0.4000
Epoch 43/100
```

```
21/21 [==============================] - 5s 236ms/step - loss: 1.04
76 - acc: 0.5905 - val_loss: 2.0843 - val_acc: 0.3000
Epoch 44/100
21/21 [==============================] - 5s 236ms/step - loss: 0.87
78 - acc: 0.6068 - val_loss: 1.9924 - val_acc: 0.3800
Epoch 45/100
21/21 [==============================] - 5s 236ms/step - loss: 0.76
34 - acc: 0.6748 - val_loss: 3.5316 - val_acc: 0.3000
Epoch 46/100
21/21 [==============================] - 5s 231ms/step - loss: 0.96
05 - acc: 0.6165 - val_loss: 1.8319 - val_acc: 0.4200
Epoch 47/100
21/21 [==============================] - 5s 238ms/step - loss: 0.93
66 - acc: 0.6214 - val_loss: 0.9257 - val_acc: 0.6000
Epoch 48/100
21/21 [==============================] - 5s 233ms/step - loss: 0.94
89 - acc: 0.6214 - val_loss: 1.1202 - val_acc: 0.5400
Epoch 49/100
21/21 [==============================] - 5s 234ms/step - loss: 0.84
07 - acc: 0.6650 - val_loss: 4.4054 - val_acc: 0.2800
Epoch 50/100
21/21 [==============================] - 5s 230ms/step - loss: 0.89
85 - acc: 0.6553 - val_loss: 1.4941 - val_acc: 0.4200
Epoch 51/100
21/21 [==============================] - 5s 237ms/step - loss: 0.90
61 - acc: 0.6359 - val_loss: 1.2352 - val_acc: 0.4200
Epoch 52/100
21/21 [==============================] - 5s 234ms/step - loss: 0.75
34 - acc: 0.6699 - val_loss: 1.0490 - val_acc: 0.5200
Epoch 53/100
21/21 [==============================] - 5s 236ms/step - loss: 0.99
39 - acc: 0.5922 - val_loss: 0.8031 - val_acc: 0.6400
Epoch 54/100
21/21 [==============================] - 5s 235ms/step - loss: 0.96
45 - acc: 0.6117 - val_loss: 1.1666 - val_acc: 0.4800
Epoch 55/100
21/21 [==============================] - 5s 230ms/step - loss: 0.98
08 - acc: 0.6262 - val_loss: 1.5473 - val_acc: 0.2400
Epoch 56/100
21/21 [==============================] - 5s 232ms/step - loss: 0.95
42 - acc: 0.6262 - val_loss: 1.8876 - val_acc: 0.4200
Epoch 57/100
21/21 [==============================] - 5s 234ms/step - loss: 1.01
49 - acc: 0.6165 - val_loss: 2.2951 - val_acc: 0.4000
Epoch 58/100
21/21 [==============================] - 5s 240ms/step - loss: 0.79
63 - acc: 0.6699 - val_loss: 1.4127 - val_acc: 0.4600
Epoch 59/100
21/21 [==============================] - 5s 232ms/step - loss: 0.88
24 - acc: 0.5874 - val_loss: 4.3082 - val_acc: 0.2200
Epoch 60/100
21/21 [==============================] - 5s 234ms/step - loss: 0.77
91 - acc: 0.6748 - val_loss: 3.5294 - val_acc: 0.2400
```

```
Epoch 61/100
21/21 [==============================] - 5s 233ms/step - loss: 0.80
23 - acc: 0.6553 - val_loss: 1.6017 - val_acc: 0.3800
Epoch 62/100
21/21 [==============================] - 5s 237ms/step - loss: 0.82
77 - acc: 0.6699 - val_loss: 1.3177 - val_acc: 0.5200
Epoch 63/100
21/21 [==============================] - 5s 226ms/step - loss: 0.85
62 - acc: 0.6602 - val_loss: 2.5455 - val_acc: 0.3200
Epoch 64/100
21/21 [==============================] - 5s 232ms/step - loss: 0.77
71 - acc: 0.6602 - val_loss: 1.3651 - val_acc: 0.5000
Epoch 65/100
21/21 [==============================] - 5s 238ms/step - loss: 0.93
41 - acc: 0.6699 - val_loss: 1.4941 - val_acc: 0.4400
Epoch 66/100
21/21 [==============================] - 5s 238ms/step - loss: 0.87
99 - acc: 0.6456 - val_loss: 5.9702 - val_acc: 0.2800
Epoch 67/100
21/21 [==============================] - 5s 238ms/step - loss: 0.82
11 - acc: 0.6456 - val_loss: 1.4320 - val_acc: 0.3800
Epoch 68/100
21/21 [==============================] - 5s 230ms/step - loss: 0.73
80 - acc: 0.6893 - val_loss: 1.3813 - val_acc: 0.4400
Epoch 69/100
21/21 [==============================] - 5s 231ms/step - loss: 0.76
03 - acc: 0.6650 - val_loss: 1.6980 - val_acc: 0.3400
Epoch 70/100
21/21 [==============================] - 5s 231ms/step - loss: 0.71
71 - acc: 0.6796 - val_loss: 4.2800 - val_acc: 0.4000
Epoch 71/100
21/21 [==============================] - 5s 231ms/step - loss: 0.80
87 - acc: 0.6408 - val_loss: 7.1466 - val_acc: 0.4800
Epoch 72/100
21/21 [==============================] - 5s 239ms/step - loss: 0.74
75 - acc: 0.7184 - val_loss: 6.5452 - val_acc: 0.2800
Epoch 73/100
21/21 [==============================] - 5s 239ms/step - loss: 0.79
55 - acc: 0.6699 - val_loss: 1.4028 - val_acc: 0.5800
Epoch 74/100
21/21 [==============================] - 5s 236ms/step - loss: 0.67
79 - acc: 0.6893 - val_loss: 1.2342 - val_acc: 0.5000
Epoch 75/100
21/21 [==============================] - 5s 238ms/step - loss: 0.88
65 - acc: 0.6262 - val_loss: 1.8677 - val_acc: 0.4400
Epoch 76/100
21/21 [==============================] - 5s 238ms/step - loss: 0.70
29 - acc: 0.6942 - val_loss: 1.2583 - val_acc: 0.5600
Epoch 77/100
21/21 [==============================] - 5s 238ms/step - loss: 0.74
00 - acc: 0.6942 - val_loss: 2.7565 - val_acc: 0.2600
Epoch 78/100
21/21 [==============================] - 5s 241ms/step - loss: 0.86
```

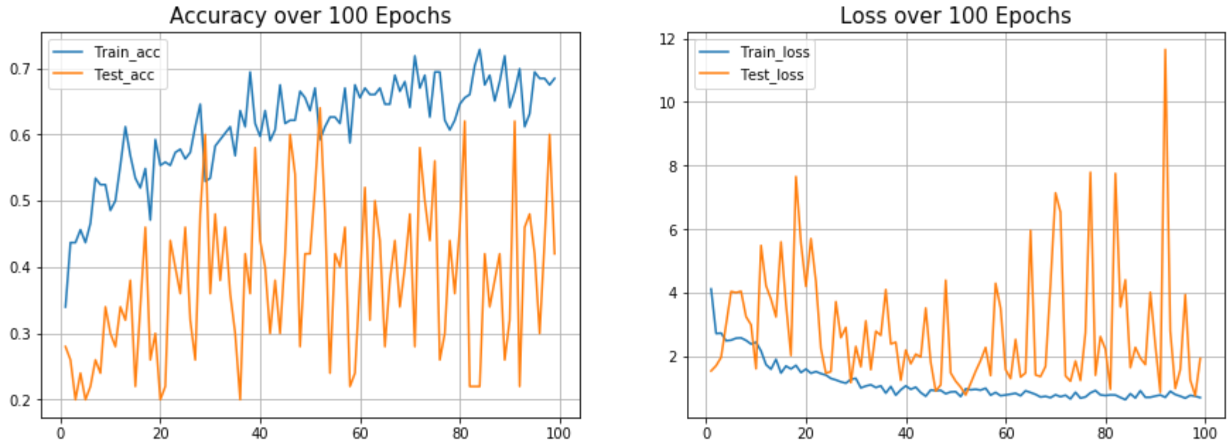
```
94 - acc: 0.6214 - val_loss: 7.7952 - val_acc: 0.3000
Epoch 79/100
21/21 [==============================] - 5s 235ms/step - loss: 0.94
16 - acc: 0.6068 - val_loss: 1.4215 - val_acc: 0.4400
Epoch 80/100
21/21 [==============================] - 5s 239ms/step - loss: 0.80
51 - acc: 0.6214 - val_loss: 2.6372 - val_acc: 0.3600
Epoch 81/100
21/21 [==============================] - 5s 236ms/step - loss: 0.78
54 - acc: 0.6456 - val_loss: 2.2402 - val_acc: 0.4600
Epoch 82/100
21/21 [==============================] - 5s 234ms/step - loss: 0.80
42 - acc: 0.6553 - val_loss: 0.9815 - val_acc: 0.6200
Epoch 83/100
21/21 [==============================] - 5s 240ms/step - loss: 0.80
04 - acc: 0.6602 - val_loss: 7.7574 - val_acc: 0.2200
Epoch 84/100
21/21 [==============================] - 5s 229ms/step - loss: 0.72
40 - acc: 0.7039 - val_loss: 3.5685 - val_acc: 0.2200
Epoch 85/100
21/21 [==============================] - 5s 237ms/step - loss: 0.65
45 - acc: 0.7282 - val_loss: 4.4165 - val_acc: 0.2200
Epoch 86/100
21/21 [==============================] - 5s 239ms/step - loss: 0.84
18 - acc: 0.6748 - val_loss: 1.6613 - val_acc: 0.4200
Epoch 87/100
21/21 [==============================] - 5s 237ms/step - loss: 0.70
77 - acc: 0.6893 - val_loss: 2.2972 - val_acc: 0.3400
Epoch 88/100
21/21 [==============================] - 5s 241ms/step - loss: 0.93
60 - acc: 0.6505 - val_loss: 1.9593 - val_acc: 0.3800
Epoch 89/100
21/21 [==============================] - 5s 235ms/step - loss: 0.72
83 - acc: 0.6796 - val_loss: 1.7564 - val_acc: 0.4200
Epoch 90/100
21/21 [==============================] - 5s 233ms/step - loss: 0.73
04 - acc: 0.7184 - val_loss: 4.0258 - val_acc: 0.2600
Epoch 91/100
21/21 [==============================] - 5s 233ms/step - loss: 0.76
88 - acc: 0.6408 - val_loss: 2.3688 - val_acc: 0.3200
Epoch 92/100
21/21 [==============================] - 5s 238ms/step - loss: 0.80
23 - acc: 0.6650 - val_loss: 0.8875 - val_acc: 0.6200
Epoch 93/100
21/21 [==============================] - 5s 241ms/step - loss: 0.72
71 - acc: 0.6990 - val_loss: 11.6511 - val_acc: 0.2200
Epoch 94/100
21/21 [==============================] - 5s 236ms/step - loss: 0.91
90 - acc: 0.6117 - val_loss: 2.8103 - val_acc: 0.4600
Epoch 95/100
21/21 [==============================] - 5s 241ms/step - loss: 0.81
67 - acc: 0.6311 - val_loss: 1.0039 - val_acc: 0.4800
Epoch 96/100
```

```
21/21 [==============================] - 5s 236ms/step - loss: 0.76
45 - acc: 0.6942 - val_loss: 1.5999 - val_acc: 0.4200
Epoch 97/100
21/21 [==============================] - 5s 236ms/step - loss: 0.70
12 - acc: 0.6845 - val_loss: 3.9530 - val_acc: 0.3000
Epoch 98/100
21/21 [==============================] - 5s 237ms/step - loss: 0.79
06 - acc: 0.6845 - val_loss: 1.2719 - val_acc: 0.4400
Epoch 99/100
21/21 [==============================] - 5s 239ms/step - loss: 0.76
34 - acc: 0.6748 - val_loss: 0.7836 - val_acc: 0.6000
Epoch 100/100
21/21 [==============================] - 5s 242ms/step - loss: 0.71
93 - acc: 0.6845 - val_loss: 1.9475 - val_acc: 0.4200
```

[5 points] Plot Accuracy and Loss During Training

In [11]:

```python
import matplotlib.pyplot as plt

def plot_acc_loss(history, epochs):
    acc = history.history['acc']
    val_acc = history.history['val_acc']
    loss = history.history['loss']
    val_loss = history.history['val_loss']

    plt.figure(figsize=(15, 5))

    plt.subplot(121)
    plt.plot(range(1,epochs), acc[1:], label='Train_acc')
    plt.plot(range(1,epochs), val_acc[1:], label='Test_acc')
    plt.title('Accuracy over ' + str(epochs) + ' Epochs', size=15)
    plt.grid(True)
    plt.legend()

    plt.subplot(122)
    plt.plot(range(1,epochs), loss[1:], label='Train_loss')
    plt.plot(range(1,epochs), val_loss[1:], label='Test_loss')
    plt.title('Loss over ' + str(epochs) + ' Epochs', size=15)
    plt.grid(True)
    plt.legend()

    plt.show()


plot_acc_loss(history, 100)
```



Accuracy over 100 Epochs

Loss over 100 Epochs

Testing Model

In [12]:

```
test_datagen = ImageDataGenerator(rescale=1. / 255)

eval_generator = test_datagen.flow_from_directory(TEST_DIR,target_size=IMAGE_SIZ
E,
                                                 batch_size=1,shuffle=True,seed
=42,class_mode="categorical")
eval_generator.reset()
print(len(eval_generator))
x = model.evaluate_generator(eval_generator,steps = np.ceil(len(eval_generator))
,
                           use_multiprocessing = False,verbose = 1,workers=1)
print('Test loss:' , x[0])
print('Test accuracy:',x[1])
```

```
Found 36 images belonging to 4 classes.
36
 1/36 [..............................] - ETA: 5s - loss: 4.9422 - ac
c: 0.0000e+00

C:\Users\Li\anaconda3\lib\site-packages\keras\engine\training.py:200
6: UserWarning: `Model.evaluate_generator` is deprecated and will be
removed in a future version. Please use `Model.evaluate`, which supp
orts generators.
  warnings.warn('`Model.evaluate_generator` is deprecated and '

36/36 [==============================] - 1s 16ms/step - loss: 2.4918
- acc: 0.3611
Test loss: 2.491814613342285
Test accuracy: 0.3611111044883728
```

[10 points] TSNE Plot

t-Distributed Stochastic Neighbor Embedding (t-SNE) is a widely used technique for dimensionality reduction that is particularly well suited for the visualization of high-dimensional datasets. After training is complete, extract features from a specific deep layer of your choice, use t-SNE to reduce the dimensionality of your extracted features to 2 dimensions and plot the resulting 2D features.

In [16]:

```
from sklearn.manifold import TSNE

intermediate_layer_model = tf.keras.models.Model(inputs=model.input,
                                        outputs=model.get_layer('dense').output)
tsne_eval_generator = test_datagen.flow_from_directory(DATASET_PATH,target_size=
IMAGE_SIZE,
                                                  batch_size=1,shuffle=False,see
d=42,class_mode="categorical")
print(tsne_eval_generator.class_indices)

outputs = intermediate_layer_model.predict_generator(tsne_eval_generator,270,ver
bose=1)
features = TSNE(n_components=2).fit_transform(outputs)
label = tsne_eval_generator.classes


covid_x, covid_y, normal_x, normal_y, pneumonia_bac_x, pneumonia_bac_y, pneumoni
a_vir_x, pneumonia_vir_y = [],[],[],[],[],[],[],[]

plt.figure()
for index in range(len(features)):
    if label[index] == 0:
        covid_x.append(features[index, 0])
        covid_y.append(features[index, 1])
    elif label[index] == 1:
        normal_x.append(features[index, 0])
        normal_y.append(features[index, 1])
    elif label[index] == 2:
        pneumonia_bac_x.append(features[index, 0])
        pneumonia_bac_y.append(features[index, 1])
    else:
        pneumonia_vir_x.append(features[index, 0])
        pneumonia_vir_y.append(features[index, 1])

plt.plot(covid_x, covid_y,'bo', ms=4, label="COVID-19")
plt.plot(normal_x, normal_y,'yo', ms=4, label="Normal")
plt.plot(pneumonia_bac_x, pneumonia_bac_y, 'go', ms=4, label="Pneumonia_bac")
plt.plot(pneumonia_vir_x, pneumonia_vir_y, 'ro', ms=4, label="Pneumonia_vir")

plt.legend(loc='lower right')
```

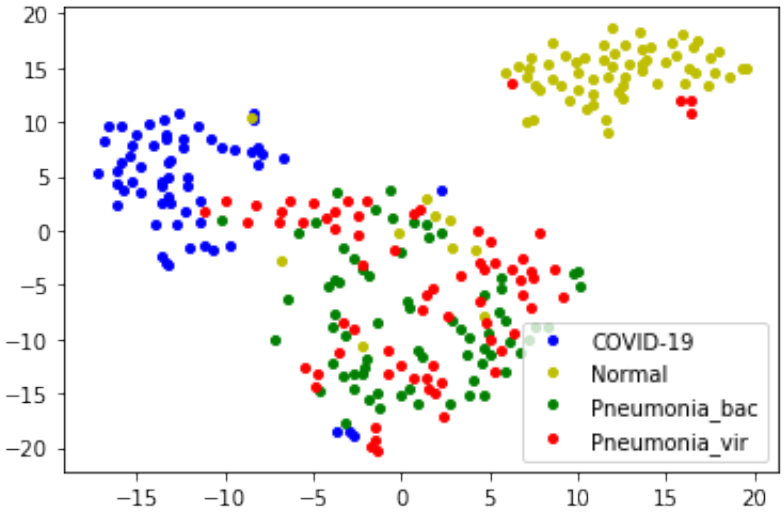
```
Found 270 images belonging to 4 classes.
{'covid': 0, 'normal': 1, 'pneumonia_bac': 2, 'pneumonia_vir': 3}
  4/270 [..............................] - ETA: 4s
```

```
C:\Users\Li\anaconda3\lib\site-packages\keras\engine\training.py:203
5: UserWarning: `Model.predict_generator` is deprecated and will be
removed in a future version. Please use `Model.predict`, which suppo
rts generators.
  warnings.warn('`Model.predict_generator` is deprecated and '
```

```
270/270 [==============================] - 3s 12ms/step
```

Out[16]:

```
<matplotlib.legend.Legend at 0x267585dbbc8>
```


● COVID-19
● Normal
● Pneumonia_bac
● Pneumonia_vir