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

Task 2 [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_VISIBLE_DEVICES'] = '-1'
tf.__version__
```

Out[1]:
```
'2.3.1'
```

Load Image Data

In [2]:
```
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
NUM_EPOCHS    = 100
LEARNING_RATE = 0.0001 # start off with high rate first 0.001 and expe
```

Generate Training and Validation Batches

In [4]:
```
train_datagen = ImageDataGenerator(rescale=1./255,rotation_range=50,fe
                                   featurewise_std_normalization = Tru
                                   height_shift_range=0.2,shear_range=
                                   zca_whitening = True,channel_shift_
                                   horizontal_flip = True,vertical_fli
                                   validation_split = 0.2,fill_mode='c


train_batches = train_datagen.flow_from_directory(DATASET_PATH,target_
                                                  shuffle=True,batch_s
                                                  subset = "training",
                                                  class_mode="categori

valid_batches = train_datagen.flow_from_directory(DATASET_PATH,target_
                                                  shuffle=True,batch_s
                                                  subset = "validation
                                                  seed=42,class_mode="
```

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

[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 [26]:
```
VGG16 = tf.keras.applications.VGG16(weights="imagenet", include_top=Fa
VGG16.trainable = False

# model 1:
model = tf.keras.Sequential([
    VGG16,
    tf.keras.layers.Flatten(),
    tf.keras.layers.Dense(256, activation='relu', name='dense_feature'
    #tf.keras.layers.Dropout(0.2),
    tf.keras.layers.Dense(4, activation='softmax'),
])
model.summary()

print("\n --------- \n")

Inception = tf.keras.applications.InceptionV3(weights="imagenet", incl
Inception.trainable = False

# model 2:
model2 = tf.keras.Sequential([
    Inception,
    tf.keras.layers.Flatten(),
```

```
    tf.keras.layers.Dense(256, activation='relu', name='dense_feature'
    tf.keras.layers.Dense(4, activation='softmax')
])
model2.summary()
```

```
Model: "sequential_8"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
vgg16 (Functional)           (None, 7, 7, 512)         14714688
_________________________________________________________________
flatten_8 (Flatten)          (None, 25088)             0
_________________________________________________________________
dense_feature (Dense)        (None, 256)               6422784
_________________________________________________________________
dense_8 (Dense)              (None, 4)                 1028
=================================================================
Total params: 21,138,500
Trainable params: 6,423,812
Non-trainable params: 14,714,688
_________________________________________________________________

 _________

Model: "sequential_9"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
inception_v3 (Functional)    (None, 5, 5, 2048)        21802784
_________________________________________________________________
flatten_9 (Flatten)          (None, 51200)             0
_________________________________________________________________
dense_feature (Dense)        (None, 256)               13107456
_________________________________________________________________
dense_9 (Dense)              (None, 4)                 1028
=================================================================
Total params: 34,911,268
Trainable params: 13,108,484
Non-trainable params: 21,802,784
_________________________________________________________________
```

[5 points] Train Model

In [27]:
```
# FIT MODEL 1
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
```

```
opt = tf.keras.optimizers.Adam(learning_rate=LEARNING_RATE)
model.compile(optimizer=opt,
              loss='categorical_crossentropy',
              metrics=['accuracy'])

#need to specify step size
history = model.fit(train_batches, batch_size=BATCH_SIZE, steps_per_ep
                    epochs=NUM_EPOCHS, validation_data=valid_batches,
```

```
22
6
Epoch 1/100
21/21 [==============================] - 49s 2s/step - loss: 1.4548 -
accuracy: 0.3932 - val_loss: 1.3317 - val_accuracy: 0.3800
Epoch 2/100
21/21 [==============================] - 55s 3s/step - loss: 1.2023 -
accuracy: 0.4320 - val_loss: 1.2437 - val_accuracy: 0.4000
Epoch 3/100
21/21 [==============================] - 77s 4s/step - loss: 1.0604 -
accuracy: 0.4951 - val_loss: 1.0137 - val_accuracy: 0.4800
Epoch 4/100
21/21 [==============================] - 53s 3s/step - loss: 0.9791 -
accuracy: 0.5874 - val_loss: 1.0192 - val_accuracy: 0.5000
Epoch 5/100
21/21 [==============================] - 76s 4s/step - loss: 0.9570 -
accuracy: 0.6311 - val_loss: 1.0725 - val_accuracy: 0.5200
Epoch 6/100
21/21 [==============================] - 61s 3s/step - loss: 0.9692 -
accuracy: 0.5825 - val_loss: 1.0386 - val_accuracy: 0.5400
Epoch 7/100
21/21 [==============================] - 76s 4s/step - loss: 0.8479 -
accuracy: 0.6262 - val_loss: 0.7860 - val_accuracy: 0.6400
Epoch 8/100
21/21 [==============================] - 66s 3s/step - loss: 0.9470 -
accuracy: 0.5922 - val_loss: 0.8060 - val_accuracy: 0.6600
Epoch 9/100
21/21 [==============================] - 67s 3s/step - loss: 0.8336 -
accuracy: 0.6505 - val_loss: 0.9031 - val_accuracy: 0.5400
Epoch 10/100
21/21 [==============================] - 57s 3s/step - loss: 0.8465 -
accuracy: 0.6262 - val_loss: 0.8460 - val_accuracy: 0.6400
Epoch 11/100
21/21 [==============================] - 53s 3s/step - loss: 0.7528 -
accuracy: 0.6845 - val_loss: 0.9090 - val_accuracy: 0.5400
Epoch 12/100
21/21 [==============================] - 48s 2s/step - loss: 0.8079 -
accuracy: 0.6699 - val_loss: 0.9608 - val_accuracy: 0.6400
Epoch 13/100
21/21 [==============================] - 64s 3s/step - loss: 0.7343 -
```

```
accuracy: 0.6796 - val_loss: 0.9493 - val_accuracy: 0.6400
Epoch 14/100
21/21 [==============================] - 50s 2s/step - loss: 0.8070 -
accuracy: 0.6796 - val_loss: 1.0286 - val_accuracy: 0.5000
Epoch 15/100
21/21 [==============================] - 58s 3s/step - loss: 0.7071 -
accuracy: 0.7282 - val_loss: 0.7251 - val_accuracy: 0.6400
Epoch 16/100
21/21 [==============================] - 52s 2s/step - loss: 0.8350 -
accuracy: 0.6602 - val_loss: 0.9295 - val_accuracy: 0.5800
Epoch 17/100
21/21 [==============================] - 56s 3s/step - loss: 0.7188 -
accuracy: 0.7282 - val_loss: 0.7466 - val_accuracy: 0.6600
Epoch 18/100
21/21 [==============================] - 57s 3s/step - loss: 0.6820 -
accuracy: 0.7238 - val_loss: 0.8532 - val_accuracy: 0.5800
Epoch 19/100
21/21 [==============================] - 61s 3s/step - loss: 0.6824 -
accuracy: 0.6845 - val_loss: 0.7934 - val_accuracy: 0.5800
Epoch 20/100
21/21 [==============================] - 55s 3s/step - loss: 0.6921 -
accuracy: 0.7379 - val_loss: 0.8642 - val_accuracy: 0.6400
Epoch 21/100
21/21 [==============================] - 63s 3s/step - loss: 0.6920 -
accuracy: 0.7379 - val_loss: 0.7907 - val_accuracy: 0.6400
Epoch 22/100
21/21 [==============================] - 56s 3s/step - loss: 0.6931 -
accuracy: 0.7087 - val_loss: 0.8698 - val_accuracy: 0.6000
Epoch 23/100
21/21 [==============================] - 60s 3s/step - loss: 0.6799 -
accuracy: 0.7476 - val_loss: 0.7940 - val_accuracy: 0.6400
Epoch 24/100
21/21 [==============================] - 52s 2s/step - loss: 0.6361 -
accuracy: 0.7282 - val_loss: 0.7907 - val_accuracy: 0.6400
Epoch 25/100
21/21 [==============================] - 55s 3s/step - loss: 0.6711 -
accuracy: 0.7136 - val_loss: 0.9118 - val_accuracy: 0.6400
Epoch 26/100
21/21 [==============================] - 53s 3s/step - loss: 0.7884 -
accuracy: 0.6408 - val_loss: 0.7857 - val_accuracy: 0.7400
Epoch 27/100
21/21 [==============================] - 53s 3s/step - loss: 0.6712 -
accuracy: 0.7039 - val_loss: 0.7449 - val_accuracy: 0.6400
Epoch 28/100
21/21 [==============================] - 52s 2s/step - loss: 0.6988 -
accuracy: 0.6990 - val_loss: 0.7444 - val_accuracy: 0.6800
Epoch 29/100
21/21 [==============================] - 57s 3s/step - loss: 0.6423 -
accuracy: 0.7330 - val_loss: 0.7673 - val_accuracy: 0.7000
Epoch 30/100
```

```
21/21 [==============================] - 50s 2s/step - loss: 0.6124 -
accuracy: 0.7621 - val_loss: 0.7771 - val_accuracy: 0.6000
Epoch 31/100
21/21 [==============================] - 54s 3s/step - loss: 0.7234 -
accuracy: 0.7039 - val_loss: 0.8142 - val_accuracy: 0.6400
Epoch 32/100
21/21 [==============================] - 59s 3s/step - loss: 0.7140 -
accuracy: 0.6990 - val_loss: 0.6499 - val_accuracy: 0.6600
Epoch 33/100
21/21 [==============================] - 73s 3s/step - loss: 0.6272 -
accuracy: 0.7524 - val_loss: 0.6482 - val_accuracy: 0.7200
Epoch 34/100
21/21 [==============================] - 53s 3s/step - loss: 0.6400 -
accuracy: 0.7233 - val_loss: 0.7456 - val_accuracy: 0.6800
Epoch 35/100
21/21 [==============================] - 52s 2s/step - loss: 0.6080 -
accuracy: 0.8010 - val_loss: 0.7076 - val_accuracy: 0.6800
Epoch 36/100
21/21 [==============================] - 78s 4s/step - loss: 0.6637 -
accuracy: 0.7087 - val_loss: 0.8434 - val_accuracy: 0.6600
Epoch 37/100
21/21 [==============================] - 49s 2s/step - loss: 0.6286 -
accuracy: 0.7621 - val_loss: 0.6974 - val_accuracy: 0.6600
Epoch 38/100
21/21 [==============================] - 54s 3s/step - loss: 0.6601 -
accuracy: 0.7233 - val_loss: 0.8850 - val_accuracy: 0.6200
Epoch 39/100
21/21 [==============================] - 60s 3s/step - loss: 0.6064 -
accuracy: 0.7670 - val_loss: 0.8083 - val_accuracy: 0.6600
Epoch 40/100
21/21 [==============================] - 51s 2s/step - loss: 0.6630 -
accuracy: 0.7184 - val_loss: 0.9370 - val_accuracy: 0.6000
Epoch 41/100
21/21 [==============================] - 53s 3s/step - loss: 0.6871 -
accuracy: 0.7238 - val_loss: 0.7609 - val_accuracy: 0.6600
Epoch 42/100
21/21 [==============================] - 59s 3s/step - loss: 0.5353 -
accuracy: 0.7670 - val_loss: 0.7157 - val_accuracy: 0.6800
Epoch 43/100
21/21 [==============================] - 48s 2s/step - loss: 0.5703 -
accuracy: 0.7767 - val_loss: 0.6260 - val_accuracy: 0.6800
Epoch 44/100
21/21 [==============================] - 46s 2s/step - loss: 0.5828 -
accuracy: 0.7379 - val_loss: 0.6172 - val_accuracy: 0.7800
Epoch 45/100
21/21 [==============================] - 47s 2s/step - loss: 0.6353 -
accuracy: 0.7427 - val_loss: 0.6846 - val_accuracy: 0.7200
Epoch 46/100
21/21 [==============================] - 48s 2s/step - loss: 0.6901 -
accuracy: 0.7087 - val_loss: 0.6730 - val_accuracy: 0.7200
```

```
Epoch 47/100
21/21 [==============================] - 52s 2s/step - loss: 0.6029 -
accuracy: 0.7670 - val_loss: 0.5067 - val_accuracy: 0.8200
Epoch 48/100
21/21 [==============================] - 55s 3s/step - loss: 0.6061 -
accuracy: 0.7282 - val_loss: 0.7658 - val_accuracy: 0.6000
Epoch 49/100
21/21 [==============================] - 48s 2s/step - loss: 0.5393 -
accuracy: 0.8010 - val_loss: 0.7522 - val_accuracy: 0.6600
Epoch 50/100
21/21 [==============================] - 47s 2s/step - loss: 0.5671 -
accuracy: 0.7767 - val_loss: 0.6884 - val_accuracy: 0.7200
Epoch 51/100
21/21 [==============================] - 54s 3s/step - loss: 0.5536 -
accuracy: 0.7913 - val_loss: 0.6973 - val_accuracy: 0.6800
Epoch 52/100
21/21 [==============================] - 54s 3s/step - loss: 0.5581 -
accuracy: 0.8058 - val_loss: 0.6715 - val_accuracy: 0.6800
Epoch 53/100
21/21 [==============================] - 50s 2s/step - loss: 0.6045 -
accuracy: 0.7087 - val_loss: 0.6566 - val_accuracy: 0.7200
Epoch 54/100
21/21 [==============================] - 54s 3s/step - loss: 0.5256 -
accuracy: 0.8010 - val_loss: 0.5925 - val_accuracy: 0.7600
Epoch 55/100
21/21 [==============================] - 50s 2s/step - loss: 0.5527 -
accuracy: 0.7667 - val_loss: 0.7914 - val_accuracy: 0.7000
Epoch 56/100
21/21 [==============================] - 48s 2s/step - loss: 0.6414 -
accuracy: 0.7282 - val_loss: 0.6798 - val_accuracy: 0.6400
Epoch 57/100
21/21 [==============================] - 48s 2s/step - loss: 0.5846 -
accuracy: 0.7379 - val_loss: 0.7369 - val_accuracy: 0.6000
Epoch 58/100
21/21 [==============================] - 47s 2s/step - loss: 0.6305 -
accuracy: 0.7233 - val_loss: 0.7294 - val_accuracy: 0.6000
Epoch 59/100
21/21 [==============================] - 47s 2s/step - loss: 0.5562 -
accuracy: 0.7670 - val_loss: 0.7629 - val_accuracy: 0.6800
Epoch 60/100
21/21 [==============================] - 49s 2s/step - loss: 0.5508 -
accuracy: 0.7913 - val_loss: 0.7212 - val_accuracy: 0.6000
Epoch 61/100
21/21 [==============================] - 51s 2s/step - loss: 0.5945 -
accuracy: 0.7330 - val_loss: 0.6630 - val_accuracy: 0.7200
Epoch 62/100
21/21 [==============================] - 50s 2s/step - loss: 0.5712 -
accuracy: 0.7233 - val_loss: 0.8970 - val_accuracy: 0.6400
Epoch 63/100
```

```
21/21 [==============================] - 46s 2s/step - loss: 0.5357 -
accuracy: 0.7573 - val_loss: 0.6657 - val_accuracy: 0.7200
Epoch 64/100
21/21 [==============================] - 51s 2s/step - loss: 0.5916 -
accuracy: 0.7427 - val_loss: 0.6654 - val_accuracy: 0.6400
Epoch 65/100
21/21 [==============================] - 48s 2s/step - loss: 0.5359 -
accuracy: 0.7621 - val_loss: 0.6856 - val_accuracy: 0.6800
Epoch 66/100
21/21 [==============================] - 48s 2s/step - loss: 0.5299 -
accuracy: 0.8010 - val_loss: 0.5930 - val_accuracy: 0.7200
Epoch 67/100
21/21 [==============================] - 54s 3s/step - loss: 0.5897 -
accuracy: 0.7670 - val_loss: 0.7419 - val_accuracy: 0.6600
Epoch 68/100
21/21 [==============================] - 47s 2s/step - loss: 0.5257 -
accuracy: 0.7767 - val_loss: 0.7219 - val_accuracy: 0.6000
Epoch 69/100
21/21 [==============================] - 46s 2s/step - loss: 0.5123 -
accuracy: 0.7864 - val_loss: 0.6625 - val_accuracy: 0.7400
Epoch 70/100
21/21 [==============================] - 48s 2s/step - loss: 0.5404 -
accuracy: 0.7427 - val_loss: 0.6980 - val_accuracy: 0.7400
Epoch 71/100
21/21 [==============================] - 56s 3s/step - loss: 0.5593 -
accuracy: 0.7816 - val_loss: 0.6116 - val_accuracy: 0.7600
Epoch 72/100
21/21 [==============================] - 50s 2s/step - loss: 0.4897 -
accuracy: 0.8204 - val_loss: 0.5864 - val_accuracy: 0.7400
Epoch 73/100
21/21 [==============================] - 50s 2s/step - loss: 0.4585 -
accuracy: 0.7864 - val_loss: 0.7322 - val_accuracy: 0.6400
Epoch 74/100
21/21 [==============================] - 51s 2s/step - loss: 0.4423 -
accuracy: 0.8350 - val_loss: 0.6169 - val_accuracy: 0.7800
Epoch 75/100
21/21 [==============================] - 56s 3s/step - loss: 0.5198 -
accuracy: 0.7864 - val_loss: 0.6075 - val_accuracy: 0.7600
Epoch 76/100
21/21 [==============================] - 47s 2s/step - loss: 0.5698 -
accuracy: 0.7864 - val_loss: 0.6192 - val_accuracy: 0.7400
Epoch 77/100
21/21 [==============================] - 49s 2s/step - loss: 0.5273 -
accuracy: 0.7718 - val_loss: 0.8016 - val_accuracy: 0.6600
Epoch 78/100
21/21 [==============================] - 48s 2s/step - loss: 0.6575 -
accuracy: 0.7379 - val_loss: 0.8568 - val_accuracy: 0.6200
Epoch 79/100
21/21 [==============================] - 53s 3s/step - loss: 0.5183 -
accuracy: 0.7864 - val_loss: 0.7508 - val_accuracy: 0.6600
```

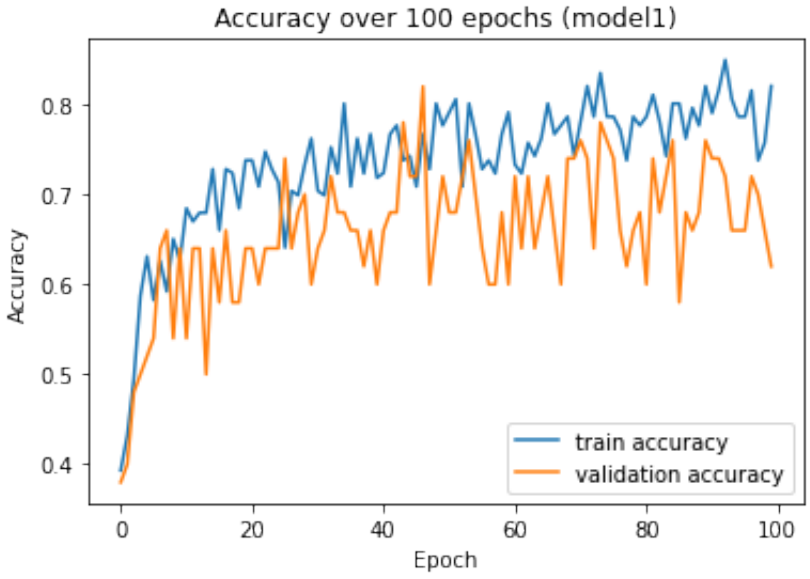
```
Epoch 80/100
21/21 [==============================] - 54s 3s/step - loss: 0.5327 -
accuracy: 0.7767 - val_loss: 0.5801 - val_accuracy: 0.6800
Epoch 81/100
21/21 [==============================] - 57s 3s/step - loss: 0.4947 -
accuracy: 0.7864 - val_loss: 0.7617 - val_accuracy: 0.6000
Epoch 82/100
21/21 [==============================] - 56s 3s/step - loss: 0.4637 -
accuracy: 0.8107 - val_loss: 0.6378 - val_accuracy: 0.7400
Epoch 83/100
21/21 [==============================] - 50s 2s/step - loss: 0.5072 -
accuracy: 0.7816 - val_loss: 0.7649 - val_accuracy: 0.6800
Epoch 84/100
21/21 [==============================] - 49s 2s/step - loss: 0.5157 -
accuracy: 0.7427 - val_loss: 0.5973 - val_accuracy: 0.7200
Epoch 85/100
21/21 [==============================] - 51s 2s/step - loss: 0.4970 -
accuracy: 0.8010 - val_loss: 0.6219 - val_accuracy: 0.7600
Epoch 86/100
21/21 [==============================] - 52s 2s/step - loss: 0.4694 -
accuracy: 0.8010 - val_loss: 0.8287 - val_accuracy: 0.5800
Epoch 87/100
21/21 [==============================] - 55s 3s/step - loss: 0.4958 -
accuracy: 0.7621 - val_loss: 0.6893 - val_accuracy: 0.6800
Epoch 88/100
21/21 [==============================] - 65s 3s/step - loss: 0.5504 -
accuracy: 0.7961 - val_loss: 0.6911 - val_accuracy: 0.6600
Epoch 89/100
21/21 [==============================] - 51s 2s/step - loss: 0.5491 -
accuracy: 0.7767 - val_loss: 0.5618 - val_accuracy: 0.6800
Epoch 90/100
21/21 [==============================] - 50s 2s/step - loss: 0.4690 -
accuracy: 0.8204 - val_loss: 0.6732 - val_accuracy: 0.7600
Epoch 91/100
21/21 [==============================] - 52s 2s/step - loss: 0.4886 -
accuracy: 0.7905 - val_loss: 0.6011 - val_accuracy: 0.7400
Epoch 92/100
21/21 [==============================] - 49s 2s/step - loss: 0.4794 -
accuracy: 0.8155 - val_loss: 0.5544 - val_accuracy: 0.7400
Epoch 93/100
21/21 [==============================] - 51s 2s/step - loss: 0.4062 -
accuracy: 0.8495 - val_loss: 0.5977 - val_accuracy: 0.7200
Epoch 94/100
21/21 [==============================] - 51s 2s/step - loss: 0.4816 -
accuracy: 0.8058 - val_loss: 0.8238 - val_accuracy: 0.6600
Epoch 95/100
21/21 [==============================] - 51s 2s/step - loss: 0.5066 -
accuracy: 0.7864 - val_loss: 0.8471 - val_accuracy: 0.6600
Epoch 96/100
21/21 [==============================] - 50s 2s/step - loss: 0.4684 -
```

```
accuracy: 0.7864 - val_loss: 0.6648 - val_accuracy: 0.6600
Epoch 97/100
21/21 [==============================] - 49s 2s/step - loss: 0.4563 -
accuracy: 0.8155 - val_loss: 0.5845 - val_accuracy: 0.7200
Epoch 98/100
21/21 [==============================] - 47s 2s/step - loss: 0.5413 -
accuracy: 0.7379 - val_loss: 0.6071 - val_accuracy: 0.7000
Epoch 99/100
21/21 [==============================] - 58s 3s/step - loss: 0.5566 -
accuracy: 0.7573 - val_loss: 0.8869 - val_accuracy: 0.6600
Epoch 100/100
21/21 [==============================] - 47s 2s/step - loss: 0.4420 -
accuracy: 0.8204 - val_loss: 0.7724 - val_accuracy: 0.6200
```

[5 points] Plot Accuracy and Loss During Training

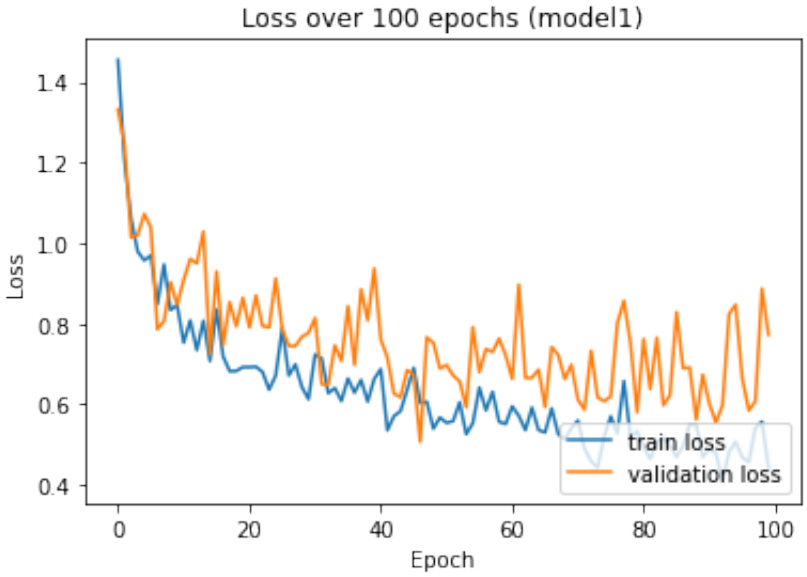
In [28]:
```
import matplotlib.pyplot as plt

plt.plot(history.history['accuracy'], label='train accuracy')
plt.plot(history.history['val_accuracy'], label = 'validation accuracy
plt.xlabel('Epoch')
plt.ylabel('Accuracy')
plt.legend(loc='lower right')
plt.title('Accuracy over ' + str(NUM_EPOCHS) + ' epochs (model1)')
#print(str(history.history['loss']))
```

Out[28]: Text(0.5, 1.0, 'Accuracy over 100 epochs (model1)')



In [29]:
```
plt.plot(history.history['loss'], label='train loss')
plt.plot(history.history['val_loss'], label = 'validation loss')
plt.xlabel('Epoch')
plt.ylabel('Loss')
plt.legend(loc='lower right')
plt.title('Loss over ' + str(NUM_EPOCHS) + ' epochs (model1)')
```

Out[29]: Text(0.5, 1.0, 'Loss over 100 epochs (model1)')



In [17]:
```
# FIT MODEL 2

opt = tf.keras.optimizers.Adam(learning_rate=LEARNING_RATE)
model2.compile(optimizer=opt,
               loss='categorical_crossentropy',
               metrics=['accuracy'])

#need to specify step size
history2 = model2.fit(train_batches, batch_size=BATCH_SIZE, steps_per_
                      epochs=NUM_EPOCHS, validation_data=valid_batches,
```

```
Epoch 1/100
21/21 [==============================] - 18s 846ms/step - loss: 0.893
0 - accuracy: 0.6505 - val_loss: 0.9066 - val_accuracy: 0.6200
Epoch 2/100
21/21 [==============================] - 14s 665ms/step - loss: 0.835
0 - accuracy: 0.6845 - val_loss: 0.8473 - val_accuracy: 0.5600
Epoch 3/100
21/21 [==============================] - 14s 656ms/step - loss: 0.717
9 - accuracy: 0.7184 - val_loss: 0.6429 - val_accuracy: 0.7600
Epoch 4/100
21/21 [==============================] - 18s 875ms/step - loss: 0.628
1 - accuracy: 0.7379 - val_loss: 0.8378 - val_accuracy: 0.5600
```

```
Epoch 5/100
21/21 [==============================] - 16s 750ms/step - loss: 0.649
7 - accuracy: 0.7233 - val_loss: 0.7033 - val_accuracy: 0.7200
Epoch 6/100
21/21 [==============================] - 16s 759ms/step - loss: 0.629
4 - accuracy: 0.7573 - val_loss: 0.9428 - val_accuracy: 0.7000
Epoch 7/100
21/21 [==============================] - 14s 657ms/step - loss: 0.596
6 - accuracy: 0.7427 - val_loss: 0.9722 - val_accuracy: 0.5600
Epoch 8/100
21/21 [==============================] - 15s 723ms/step - loss: 0.620
9 - accuracy: 0.7330 - val_loss: 0.6998 - val_accuracy: 0.6800
Epoch 9/100
21/21 [==============================] - 16s 739ms/step - loss: 0.567
8 - accuracy: 0.7524 - val_loss: 0.7241 - val_accuracy: 0.7000
Epoch 10/100
21/21 [==============================] - 17s 824ms/step - loss: 0.532
0 - accuracy: 0.7767 - val_loss: 0.9064 - val_accuracy: 0.6200
Epoch 11/100
21/21 [==============================] - 14s 688ms/step - loss: 0.649
9 - accuracy: 0.6942 - val_loss: 1.0875 - val_accuracy: 0.6200
Epoch 12/100
21/21 [==============================] - 14s 671ms/step - loss: 0.684
9 - accuracy: 0.7087 - val_loss: 0.6445 - val_accuracy: 0.7400
Epoch 13/100
21/21 [==============================] - 14s 689ms/step - loss: 0.505
1 - accuracy: 0.7573 - val_loss: 1.2618 - val_accuracy: 0.5200
Epoch 14/100
21/21 [==============================] - 17s 812ms/step - loss: 0.542
5 - accuracy: 0.7816 - val_loss: 0.7977 - val_accuracy: 0.6400
Epoch 15/100
21/21 [==============================] - 15s 720ms/step - loss: 0.488
7 - accuracy: 0.7864 - val_loss: 0.7204 - val_accuracy: 0.7000
Epoch 16/100
21/21 [==============================] - 14s 666ms/step - loss: 0.586
8 - accuracy: 0.7379 - val_loss: 0.8388 - val_accuracy: 0.5400
Epoch 17/100
21/21 [==============================] - 13s 600ms/step - loss: 0.653
6 - accuracy: 0.7330 - val_loss: 0.8208 - val_accuracy: 0.6000
Epoch 18/100
21/21 [==============================] - 13s 635ms/step - loss: 0.574
7 - accuracy: 0.7621 - val_loss: 0.7327 - val_accuracy: 0.7200
Epoch 19/100
21/21 [==============================] - 14s 672ms/step - loss: 0.521
9 - accuracy: 0.7573 - val_loss: 0.6458 - val_accuracy: 0.6600
Epoch 20/100
21/21 [==============================] - 14s 662ms/step - loss: 0.528
9 - accuracy: 0.7961 - val_loss: 0.7567 - val_accuracy: 0.6000
Epoch 21/100
21/21 [==============================] - 13s 606ms/step - loss: 0.526
```

```
9 - accuracy: 0.7714 - val_loss: 0.6875 - val_accuracy: 0.6600
Epoch 22/100
21/21 [==============================] - 13s 625ms/step - loss: 0.595
3 - accuracy: 0.7476 - val_loss: 0.5381 - val_accuracy: 0.7400
Epoch 23/100
21/21 [==============================] - 12s 592ms/step - loss: 0.565
3 - accuracy: 0.7282 - val_loss: 0.9904 - val_accuracy: 0.5800
Epoch 24/100
21/21 [==============================] - 13s 619ms/step - loss: 0.555
7 - accuracy: 0.7718 - val_loss: 0.8862 - val_accuracy: 0.6000
Epoch 25/100
21/21 [==============================] - 15s 725ms/step - loss: 0.461
1 - accuracy: 0.7427 - val_loss: 0.7874 - val_accuracy: 0.5800
Epoch 26/100
21/21 [==============================] - 18s 834ms/step - loss: 0.455
7 - accuracy: 0.7961 - val_loss: 0.8788 - val_accuracy: 0.6000
Epoch 27/100
21/21 [==============================] - 18s 873ms/step - loss: 0.548
2 - accuracy: 0.7718 - val_loss: 0.9758 - val_accuracy: 0.7200
Epoch 28/100
21/21 [==============================] - 13s 614ms/step - loss: 0.563
3 - accuracy: 0.7524 - val_loss: 0.6985 - val_accuracy: 0.7200
Epoch 29/100
21/21 [==============================] - 13s 618ms/step - loss: 0.513
0 - accuracy: 0.7573 - val_loss: 0.7428 - val_accuracy: 0.6600
Epoch 30/100
21/21 [==============================] - 15s 729ms/step - loss: 0.534
3 - accuracy: 0.7913 - val_loss: 0.6440 - val_accuracy: 0.7000
Epoch 31/100
21/21 [==============================] - 15s 732ms/step - loss: 0.522
6 - accuracy: 0.7816 - val_loss: 0.7522 - val_accuracy: 0.6200
Epoch 32/100
21/21 [==============================] - 13s 628ms/step - loss: 0.520
0 - accuracy: 0.7621 - val_loss: 0.6871 - val_accuracy: 0.6400
Epoch 33/100
21/21 [==============================] - 13s 623ms/step - loss: 0.390
5 - accuracy: 0.8398 - val_loss: 0.8762 - val_accuracy: 0.6200
Epoch 34/100
21/21 [==============================] - 14s 660ms/step - loss: 0.445
7 - accuracy: 0.7913 - val_loss: 1.0864 - val_accuracy: 0.6200
Epoch 35/100
21/21 [==============================] - 13s 623ms/step - loss: 0.457
5 - accuracy: 0.7961 - val_loss: 0.6050 - val_accuracy: 0.7200
Epoch 36/100
21/21 [==============================] - 15s 728ms/step - loss: 0.434
7 - accuracy: 0.7816 - val_loss: 0.6013 - val_accuracy: 0.7400
Epoch 37/100
21/21 [==============================] - 15s 732ms/step - loss: 0.448
6 - accuracy: 0.8204 - val_loss: 0.7341 - val_accuracy: 0.6800
Epoch 38/100
```

```
21/21 [==============================] - 16s 757ms/step - loss: 0.491
4 - accuracy: 0.8010 - val_loss: 0.5023 - val_accuracy: 0.7200
Epoch 39/100
21/21 [==============================] - 17s 829ms/step - loss: 0.448
1 - accuracy: 0.7864 - val_loss: 0.5813 - val_accuracy: 0.6600
Epoch 40/100
21/21 [==============================] - 15s 715ms/step - loss: 0.525
7 - accuracy: 0.7816 - val_loss: 0.8509 - val_accuracy: 0.6200
Epoch 41/100
21/21 [==============================] - 13s 621ms/step - loss: 0.405
5 - accuracy: 0.7961 - val_loss: 0.7969 - val_accuracy: 0.6600
Epoch 42/100
21/21 [==============================] - 14s 651ms/step - loss: 0.492
2 - accuracy: 0.8010 - val_loss: 0.7586 - val_accuracy: 0.7400
Epoch 43/100
21/21 [==============================] - 21s 994ms/step - loss: 0.481
6 - accuracy: 0.8301 - val_loss: 0.6643 - val_accuracy: 0.6400
Epoch 44/100
21/21 [==============================] - 25s 1s/step - loss: 0.3914 -
accuracy: 0.8010 - val_loss: 0.5099 - val_accuracy: 0.7000
Epoch 45/100
21/21 [==============================] - 18s 838ms/step - loss: 0.404
1 - accuracy: 0.8592 - val_loss: 0.7681 - val_accuracy: 0.7400
Epoch 46/100
21/21 [==============================] - 15s 726ms/step - loss: 0.366
3 - accuracy: 0.8447 - val_loss: 0.7319 - val_accuracy: 0.6600
Epoch 47/100
21/21 [==============================] - 15s 700ms/step - loss: 0.456
8 - accuracy: 0.8000 - val_loss: 0.7366 - val_accuracy: 0.6800
Epoch 48/100
21/21 [==============================] - 16s 780ms/step - loss: 0.422
1 - accuracy: 0.8058 - val_loss: 0.9282 - val_accuracy: 0.6800
Epoch 49/100
21/21 [==============================] - 23s 1s/step - loss: 0.4431 -
accuracy: 0.8447 - val_loss: 0.6251 - val_accuracy: 0.7200
Epoch 50/100
21/21 [==============================] - 22s 1s/step - loss: 0.4204 -
accuracy: 0.8495 - val_loss: 0.5092 - val_accuracy: 0.7000
Epoch 51/100
21/21 [==============================] - 14s 667ms/step - loss: 0.385
3 - accuracy: 0.8333 - val_loss: 0.8369 - val_accuracy: 0.7200
Epoch 52/100
21/21 [==============================] - 15s 719ms/step - loss: 0.458
7 - accuracy: 0.7913 - val_loss: 0.7711 - val_accuracy: 0.7000
Epoch 53/100
21/21 [==============================] - 14s 688ms/step - loss: 0.463
4 - accuracy: 0.7767 - val_loss: 0.9715 - val_accuracy: 0.6600
Epoch 54/100
21/21 [==============================] - 15s 715ms/step - loss: 0.354
8 - accuracy: 0.8689 - val_loss: 0.7771 - val_accuracy: 0.6600
```

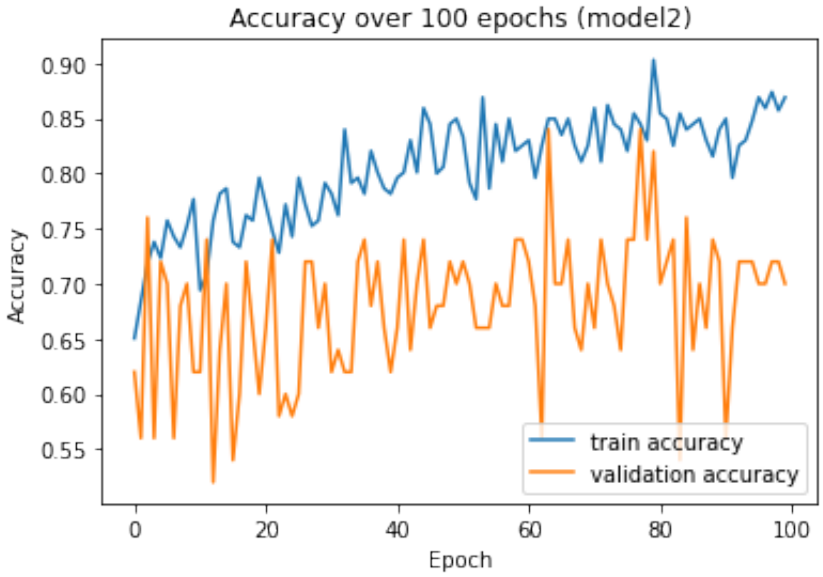
```
Epoch 55/100
21/21 [==============================] - 15s 729ms/step - loss: 0.443
1 - accuracy: 0.7864 - val_loss: 0.6714 - val_accuracy: 0.6600
Epoch 56/100
21/21 [==============================] - 15s 723ms/step - loss: 0.386
8 - accuracy: 0.8447 - val_loss: 0.5384 - val_accuracy: 0.7000
Epoch 57/100
21/21 [==============================] - 15s 733ms/step - loss: 0.411
9 - accuracy: 0.8107 - val_loss: 0.7462 - val_accuracy: 0.6800
Epoch 58/100
21/21 [==============================] - 15s 724ms/step - loss: 0.397
7 - accuracy: 0.8495 - val_loss: 0.7064 - val_accuracy: 0.6800
Epoch 59/100
21/21 [==============================] - 15s 730ms/step - loss: 0.470
9 - accuracy: 0.8204 - val_loss: 0.6579 - val_accuracy: 0.7400
Epoch 60/100
21/21 [==============================] - 15s 704ms/step - loss: 0.444
7 - accuracy: 0.8252 - val_loss: 0.5647 - val_accuracy: 0.7400
Epoch 61/100
21/21 [==============================] - 14s 679ms/step - loss: 0.450
4 - accuracy: 0.8301 - val_loss: 0.6716 - val_accuracy: 0.7200
Epoch 62/100
21/21 [==============================] - 15s 712ms/step - loss: 0.494
0 - accuracy: 0.7961 - val_loss: 0.6642 - val_accuracy: 0.6800
Epoch 63/100
21/21 [==============================] - 13s 622ms/step - loss: 0.383
3 - accuracy: 0.8252 - val_loss: 1.2110 - val_accuracy: 0.5600
Epoch 64/100
21/21 [==============================] - 16s 746ms/step - loss: 0.352
3 - accuracy: 0.8495 - val_loss: 0.6830 - val_accuracy: 0.8400
Epoch 65/100
21/21 [==============================] - 14s 646ms/step - loss: 0.397
0 - accuracy: 0.8495 - val_loss: 0.5177 - val_accuracy: 0.7000
Epoch 66/100
21/21 [==============================] - 13s 606ms/step - loss: 0.388
0 - accuracy: 0.8350 - val_loss: 0.8651 - val_accuracy: 0.7000
Epoch 67/100
21/21 [==============================] - 15s 722ms/step - loss: 0.380
5 - accuracy: 0.8495 - val_loss: 0.5113 - val_accuracy: 0.7400
Epoch 68/100
21/21 [==============================] - 14s 655ms/step - loss: 0.431
8 - accuracy: 0.8252 - val_loss: 0.7158 - val_accuracy: 0.6600
Epoch 69/100
21/21 [==============================] - 13s 637ms/step - loss: 0.477
5 - accuracy: 0.8107 - val_loss: 0.8772 - val_accuracy: 0.6400
Epoch 70/100
21/21 [==============================] - 14s 653ms/step - loss: 0.378
2 - accuracy: 0.8252 - val_loss: 0.6905 - val_accuracy: 0.7000
Epoch 71/100
```

```
21/21 [==============================] - 16s 776ms/step - loss: 0.355
4 - accuracy: 0.8592 - val_loss: 1.0911 - val_accuracy: 0.6600
Epoch 72/100
21/21 [==============================] - 15s 735ms/step - loss: 0.408
8 - accuracy: 0.8107 - val_loss: 0.5222 - val_accuracy: 0.7400
Epoch 73/100
21/21 [==============================] - 15s 712ms/step - loss: 0.305
7 - accuracy: 0.8619 - val_loss: 0.8914 - val_accuracy: 0.7000
Epoch 74/100
21/21 [==============================] - 17s 791ms/step - loss: 0.382
2 - accuracy: 0.8447 - val_loss: 0.6356 - val_accuracy: 0.6800
Epoch 75/100
21/21 [==============================] - 13s 631ms/step - loss: 0.342
2 - accuracy: 0.8398 - val_loss: 0.8168 - val_accuracy: 0.6400
Epoch 76/100
21/21 [==============================] - 16s 747ms/step - loss: 0.434
4 - accuracy: 0.8204 - val_loss: 0.6285 - val_accuracy: 0.7400
Epoch 77/100
21/21 [==============================] - 15s 717ms/step - loss: 0.368
7 - accuracy: 0.8544 - val_loss: 0.9801 - val_accuracy: 0.7400
Epoch 78/100
21/21 [==============================] - 17s 801ms/step - loss: 0.372
9 - accuracy: 0.8447 - val_loss: 0.4505 - val_accuracy: 0.8400
Epoch 79/100
21/21 [==============================] - 15s 720ms/step - loss: 0.384
0 - accuracy: 0.8301 - val_loss: 0.5932 - val_accuracy: 0.7400
Epoch 80/100
21/21 [==============================] - 13s 613ms/step - loss: 0.316
3 - accuracy: 0.9029 - val_loss: 0.5324 - val_accuracy: 0.8200
Epoch 81/100
21/21 [==============================] - 15s 691ms/step - loss: 0.388
9 - accuracy: 0.8544 - val_loss: 0.5603 - val_accuracy: 0.7000
Epoch 82/100
21/21 [==============================] - 15s 737ms/step - loss: 0.371
9 - accuracy: 0.8495 - val_loss: 0.6993 - val_accuracy: 0.7200
Epoch 83/100
21/21 [==============================] - 15s 707ms/step - loss: 0.374
0 - accuracy: 0.8252 - val_loss: 0.8669 - val_accuracy: 0.7400
Epoch 84/100
21/21 [==============================] - 14s 649ms/step - loss: 0.327
3 - accuracy: 0.8544 - val_loss: 1.1346 - val_accuracy: 0.5400
Epoch 85/100
21/21 [==============================] - 13s 619ms/step - loss: 0.375
7 - accuracy: 0.8398 - val_loss: 0.6727 - val_accuracy: 0.7600
Epoch 86/100
21/21 [==============================] - 17s 805ms/step - loss: 0.309
0 - accuracy: 0.8447 - val_loss: 0.7651 - val_accuracy: 0.6400
Epoch 87/100
21/21 [==============================] - 15s 727ms/step - loss: 0.377
3 - accuracy: 0.8495 - val_loss: 0.9104 - val_accuracy: 0.7000
```

```
Epoch 88/100
21/21 [==============================] - 16s 774ms/step - loss: 0.401
4 - accuracy: 0.8301 - val_loss: 0.5944 - val_accuracy: 0.6600
Epoch 89/100
21/21 [==============================] - 16s 748ms/step - loss: 0.332
2 - accuracy: 0.8155 - val_loss: 0.4445 - val_accuracy: 0.7400
Epoch 90/100
21/21 [==============================] - 15s 716ms/step - loss: 0.356
8 - accuracy: 0.8398 - val_loss: 0.8321 - val_accuracy: 0.7200
Epoch 91/100
21/21 [==============================] - 15s 723ms/step - loss: 0.323
0 - accuracy: 0.8495 - val_loss: 0.9624 - val_accuracy: 0.5600
Epoch 92/100
21/21 [==============================] - 15s 694ms/step - loss: 0.379
8 - accuracy: 0.7961 - val_loss: 0.8240 - val_accuracy: 0.6600
Epoch 93/100
21/21 [==============================] - 15s 725ms/step - loss: 0.377
9 - accuracy: 0.8252 - val_loss: 0.7359 - val_accuracy: 0.7200
Epoch 94/100
21/21 [==============================] - 13s 598ms/step - loss: 0.350
8 - accuracy: 0.8301 - val_loss: 0.7483 - val_accuracy: 0.7200
Epoch 95/100
21/21 [==============================] - 17s 805ms/step - loss: 0.320
2 - accuracy: 0.8476 - val_loss: 0.6016 - val_accuracy: 0.7200
Epoch 96/100
21/21 [==============================] - 16s 745ms/step - loss: 0.345
6 - accuracy: 0.8689 - val_loss: 0.6534 - val_accuracy: 0.7000
Epoch 97/100
21/21 [==============================] - 13s 616ms/step - loss: 0.365
5 - accuracy: 0.8592 - val_loss: 0.7623 - val_accuracy: 0.7000
Epoch 98/100
21/21 [==============================] - 15s 703ms/step - loss: 0.337
8 - accuracy: 0.8738 - val_loss: 0.7639 - val_accuracy: 0.7200
Epoch 99/100
21/21 [==============================] - 16s 751ms/step - loss: 0.316
8 - accuracy: 0.8571 - val_loss: 0.8615 - val_accuracy: 0.7200
Epoch 100/100
21/21 [==============================] - 13s 604ms/step - loss: 0.326
4 - accuracy: 0.8689 - val_loss: 0.8074 - val_accuracy: 0.7000
```

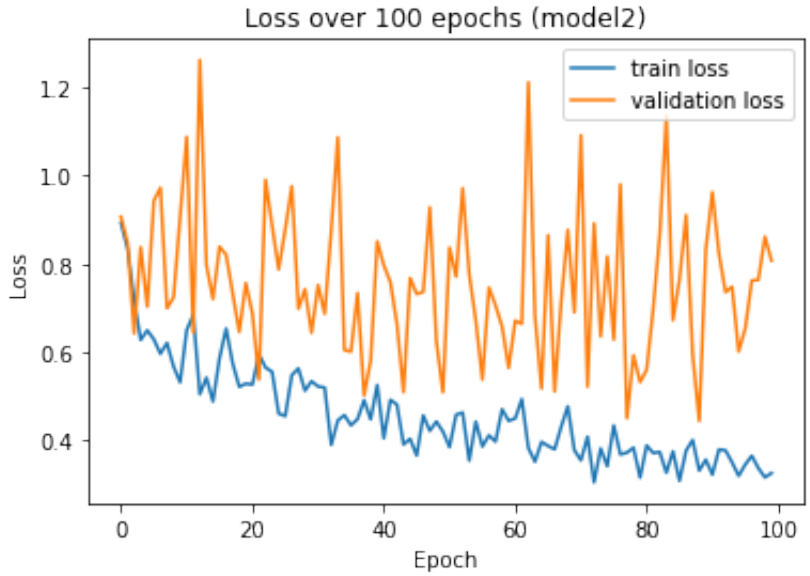
In [18]:
```
plt.plot(history2.history['accuracy'], label='train accuracy')
plt.plot(history2.history['val_accuracy'], label = 'validation accurac
plt.xlabel('Epoch')
plt.ylabel('Accuracy')
plt.legend(loc='lower right')
plt.title('Accuracy over ' + str(NUM_EPOCHS) + ' epochs (model2)')
```

Out[18]:
```
Text(0.5, 1.0, 'Accuracy over 100 epochs (model2)')
```



In [22]:
```
plt.plot(history2.history['loss'], label='train loss')
plt.plot(history2.history['val_loss'], label = 'validation loss')
plt.xlabel('Epoch')
plt.ylabel('Loss')
plt.legend(loc='upper right')
plt.title('Loss over ' + str(NUM_EPOCHS) + ' epochs (model2)')
```

Out[22]:
```
Text(0.5, 1.0, 'Loss over 100 epochs (model2)')
```



Testing Model

In [30]:
```
test_datagen = ImageDataGenerator(rescale=1. / 255)

eval_generator = test_datagen.flow_from_directory(TEST_DIR,target_size
                                                  batch_size=1,shuffle
eval_generator.reset()
print(len(eval_generator))
x = model.evaluate_generator(eval_generator,steps = np.ceil(len(eval_g
                             use_multiprocessing = False,verbose = 1,wor
print('Test loss:' , x[0])
print('Test accuracy:',x[1])
```

```
Found 36 images belonging to 4 classes.
36
36/36 [==============================] - 7s 183ms/step - loss: 0.7414
- accuracy: 0.7222
Test loss: 0.7414432764053345
Test accuracy: 0.7222222089767456
```

[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 [50]:
```
from sklearn.manifold import TSNE
from tensorflow.keras import models

intermediate_layer_model = models.Model(inputs=model.input,
                                        outputs=model.get_layer('dense

tsne_eval_generator = test_datagen.flow_from_directory(DATASET_PATH,ta
                                                       batch_size=1,shuffle

labels = tsne_eval_generator.labels
colors = []
for i in range(len(labels)):
    if labels[i] == 0:
        colors += 'b'
    elif labels[i] == 1:
        colors += 'g'
    elif labels[i] == 2:
        colors += 'r'
    else:
        colors += 'y'

layer = intermediate_layer_model.predict(tsne_eval_generator)
tsne = TSNE(n_components=2)
intermediate_tsne = tsne.fit_transform(layer)

print("\nLEGEND:")
print("blue = covid")
print("green = normal")
print("red = pneumonia_bacterial")
print("yellow = pneumonia_viral")

plt.figure(figsize=(8, 8))
plt.scatter(x = intermediate_tsne[:,0], y=intermediate_tsne[:,1], c=co
plt.show()
```

```
Found 270 images belonging to 4 classes.

LEGEND:
blue = covid
```

```
green = normal
red = pneumonia_bacterial
yellow = pneumonia_viral
```