In [31]:

```
#preprocessing images
import tensorflow as tf
from tensorflow.keras import layers, models,preprocessing,regularizers,callbacks
from matplotlib import pyplot as plt

import tensorflow as tf
from tensorflow.keras import Sequential
from tensorflow.keras.layers import Conv2D,MaxPool2D,Dropout,Flatten,Dense,BatchNormalizati
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.callbacks import EarlyStopping

#datagen = preprocessing.image.ImageDataGenerator()
datagen = ImageDataGenerator(validation_split=0.2)# normalisation
train_generator = datagen.flow_from_directory(
        'D:/mit wpu/DL Project/cnn/train/',
        batch_size=10,
        target_size=(512,512),
        class_mode='binary')




'''val_generator = datagen.flow_from_directory(
        'D:/mit wpu/DL Project/cnn/test/',
        target_size=(512,512),
        class_mode='binary')

'''


val_generator = datagen.flow_from_directory(
        'D:/mit wpu/DL Project/cnn/train/',
        batch_size=10,
        target_size=(512,512),
        class_mode='binary',subset='validation')
```

```
Found 4077 images belonging to 2 classes.
Found 814 images belonging to 2 classes.
```

In [33]:

```
model = models.Sequential()
model.add(layers.Conv2D(32, (3,3), activation='relu', input_shape=(512, 512, 3)))
model.add(layers.MaxPooling2D((2, 2)))
model.add(layers.Conv2D(64, (3, 3), activation='relu',kernel_regularizer=regularizers.l2(0.
model.add(layers.MaxPooling2D((2,2)))
model.add(Flatten())
model.add(layers.Dense(256, activation='relu',kernel_regularizer=regularizers.l2(0.01)))
model.add(layers.Dropout(0.2))
model.add(layers.Dense(128, activation='relu',kernel_regularizer=regularizers.l2(0.001)))
model.add(layers.Dropout(0.2))
model.add(layers.Dense(75, activation='relu',kernel_regularizer=regularizers.l2(0.001)))
model.add(layers.Dropout(0.2))
model.add(layers.Dense(35, activation='relu',kernel_regularizer=regularizers.l2(0.001)))
model.add(layers.Dropout(0.5))
model.add(layers.Dense(10, activation='relu',kernel_regularizer=regularizers.l2(0.001)))
model.add(layers.Dense(1, activation='sigmoid'))
model.compile(optimizer=tf.keras.optimizers.SGD(),
              loss=tf.keras.losses.BinaryCrossentropy(),
              metrics=['accuracy',
                       tf.keras.metrics.TrueNegatives(),
                       tf.keras.metrics.TruePositives(),
                       tf.keras.metrics.FalseNegatives(),
                       tf.keras.metrics.FalsePositives()])
```

In [34]:

```
model.summary()
```

```
Model: "sequential_4"
_________________________________________________________________
Layer (type)                 Output Shape              Param #   
=================================================================
conv2d_11 (Conv2D)           (None, 510, 510, 32)      896       
_________________________________________________________________
max_pooling2d_11 (MaxPooling (None, 255, 255, 32)      0         
_________________________________________________________________
conv2d_12 (Conv2D)           (None, 253, 253, 64)      18496     
_________________________________________________________________
max_pooling2d_12 (MaxPooling (None, 126, 126, 64)      0         
_________________________________________________________________
flatten_4 (Flatten)          (None, 1016064)           0         
_________________________________________________________________
dense_9 (Dense)              (None, 256)               260112640 
_________________________________________________________________
dropout_14 (Dropout)         (None, 256)               0         
_________________________________________________________________
dense_10 (Dense)             (None, 128)               32896     
_________________________________________________________________
dropout_15 (Dropout)         (None, 128)               0         
_________________________________________________________________
dense_11 (Dense)             (None, 75)                9675      
_________________________________________________________________
dropout_16 (Dropout)         (None, 75)                0         
_________________________________________________________________
dense_12 (Dense)             (None, 35)                2660      
_________________________________________________________________
dropout_17 (Dropout)         (None, 35)                0         
_________________________________________________________________
dense_13 (Dense)             (None, 10)                360       
_________________________________________________________________
dense_14 (Dense)             (None, 1)                 11        
=================================================================
Total params: 260,177,634
Trainable params: 260,177,634
Non-trainable params: 0
_________________________________________________________________
```

In [35]:

```
model.compile(optimizer='adam',loss='binary_crossentropy',metrics=['accuracy',
                       tf.keras.metrics.TrueNegatives(),
                       tf.keras.metrics.TruePositives(),
                       tf.keras.metrics.FalseNegatives(),
                       tf.keras.metrics.FalsePositives()])

early_stop = EarlyStopping(monitor='val_loss',patience=2)
```

In [36]:

```
'''history = model.fit_generator(generator = train_generator,
                             steps_per_epoch = len(train_generator),
                              epochs =20,
                              validation_data = val_generator,
                              validation_steps=len(val_generator),
                             callbacks=[early_stop])

'''
```

Out[36]:

```
'history = model.fit_generator(generator = train_generator,\n
steps_per_epoch = len(train_generator),\n                              epoch
s =20,\n                               validation_data = val_generator,\n
validation_steps=len(val_generator),\n                              callbacks
=[early_stop])\n                              \n'
```

In [37]:

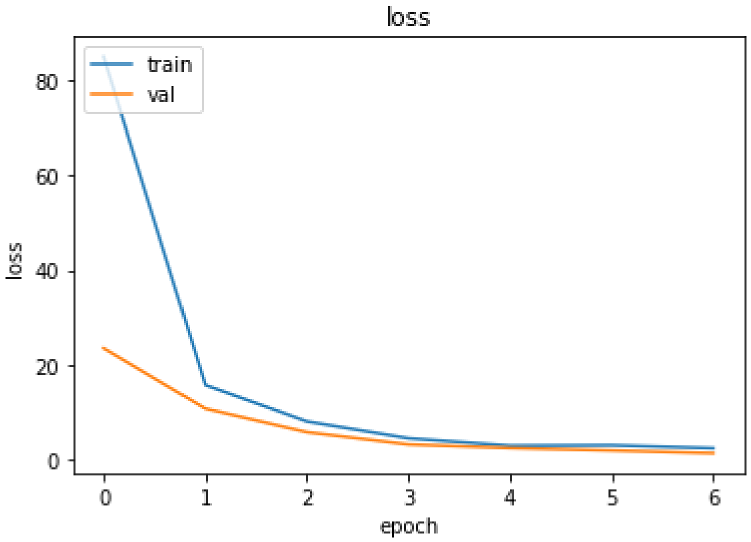
```
# fititing
history=model.fit_generator(train_generator,epochs=7,validation_data=val_generator,shuffle=
          callbacks=[callbacks.EarlyStopping(monitor='val_acc', patience=5,restore_best_we
# using early stopping method to train the model to achieve maximum accuracy
```

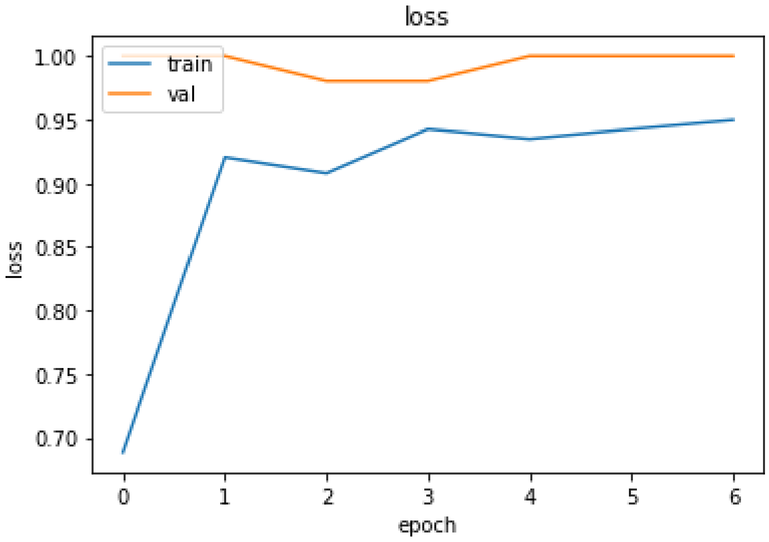
```
Epoch 1/7
408/408 [==============================] - ETA: 0s - loss: 84.9565 - accurac
y: 0.6885 - true_negatives_3: 1572.0000 - true_positives_3: 1235.0000 - fals
e_negatives_3: 829.0000 - false_positives_3: 441.0000WARNING:tensorflow:Earl
y stopping conditioned on metric `val_acc` which is not available. Available
metrics are: loss,accuracy,true_negatives_3,true_positives_3,false_negatives
_3,false_positives_3,val_loss,val_accuracy,val_true_negatives_3,val_true_pos
itives_3,val_false_negatives_3,val_false_positives_3
408/408 [==============================] - 2467s 6s/step - loss: 84.9565 - a
ccuracy: 0.6885 - true_negatives_3: 1572.0000 - true_positives_3: 1235.0000
- false_negatives_3: 829.0000 - false_positives_3: 441.0000 - val_loss: 23.4
768 - val_accuracy: 1.0000 - val_true_negatives_3: 402.0000 - val_true_posit
ives_3: 412.0000 - val_false_negatives_3: 0.0000e+00 - val_false_positives_
3: 0.0000e+00
Epoch 2/7
408/408 [==============================] - ETA: 0s - loss: 15.6727 - accurac
y: 0.9205 - true_negatives_3: 1761.0000 - true_positives_3: 1992.0000 - fals
e_negatives_3: 72.0000 - false_positives_3: 252.0000   ETA: 23:43 - loss: 1
8.WARNING:tensorflow:Early stopping conditioned on metric `val_acc` which is
not available. Available metrics are: loss,accuracy,true_negatives_3,true_po
sitives_3,false_negatives_3,false_positives_3,val_loss,val_accuracy,val_true
_negatives_3,val_true_positives_3,val_false_negatives_3,val_false_positives_
3
408/408 [==============================] - 3018s 7s/step - loss: 15.6727 - a
ccuracy: 0.9205 - true_negatives_3: 1761.0000 - true_positives_3: 1992.0000
- false_negatives_3: 72.0000 - false_positives_3: 252.0000 - val_loss: 10.70
62 - val_accuracy: 1.0000 - val_true_negatives_3: 402.0000 - val_true_positi
ves_3: 412.0000 - val_false_negatives_3: 0.0000e+00 - val_false_positives_3:
0.0000e+00
Epoch 3/7
408/408 [==============================] - ETA: 0s - loss: 8.0260 - accurac
y: 0.9080 - true_negatives_3: 1766.0000 - true_positives_3: 1936.0000 - fals
e_negatives_3: 128.0000 - false_positives_3: 247.0000   ETA: 3:04 - loss: 8.
1779 - accuracy: 0.9052 - true_negatives_3: 1627.0000 - true_positives_3: 17
92.0000 - falseWARNING:tensorflow:Early stopping conditioned on metric `val_
acc` which is not available. Available metrics are: loss,accuracy,true_negat
ives_3,true_positives_3,false_negatives_3,false_positives_3,val_loss,val_acc
uracy,val_true_negatives_3,val_true_positives_3,val_false_negatives_3,val_fa
lse_positives_3
408/408 [==============================] - 2605s 6s/step - loss: 8.0260 - ac
curacy: 0.9080 - true_negatives_3: 1766.0000 - true_positives_3: 1936.0000 -
false_negatives_3: 128.0000 - false_positives_3: 247.0000 - val_loss: 5.7683
- val_accuracy: 0.9803 - val_true_negatives_3: 402.0000 - val_true_positives
_3: 396.0000 - val_false_negatives_3: 16.0000 - val_false_positives_3: 0.000
0e+00
Epoch 4/7
408/408 [==============================] - ETA: 0s - loss: 4.4792 - accurac
y: 0.9426 - true_negatives_3: 1846.0000 - true_positives_3: 1997.0000 - fals
e_negatives_3: 67.0000 - false_positives_3: 167.0000  - ETA: 2:51 - loss: 4.
5643 - accuracy: 0.9405 - true_negatives_3: 1717.0000 - true_positives_3: 18
54.0000 - false_negatives_3: 6WARNING:tensorflow:Early stopping conditioned
on metric `val_acc` which is not available. Available metrics are: loss,accu
racy,true_negatives_3,true_positives_3,false_negatives_3,false_positives_3,v
```

```
al_loss,val_accuracy,val_true_negatives_3,val_true_positives_3,val_false_neg
atives_3,val_false_positives_3
408/408 [==============================] - 2599s 6s/step - loss: 4.4792 - ac
curacy: 0.9426 - true_negatives_3: 1846.0000 - true_positives_3: 1997.0000 -
false_negatives_3: 67.0000 - false_positives_3: 167.0000 - val_loss: 3.1747
- val_accuracy: 0.9803 - val_true_negatives_3: 402.0000 - val_true_positives
_3: 396.0000 - val_false_negatives_3: 16.0000 - val_false_positives_3: 0.000
0e+00
Epoch 5/7
408/408 [==============================] - ETA: 0s - loss: 2.9215 - accurac
y: 0.9348 - true_negatives_3: 1842.0000 - true_positives_3: 1969.0000 - fals
e_negatives_3: 95.0000 - false_positives_3: 171.0000WARNING:tensorflow:Early
stopping conditioned on metric `val_acc` which is not available. Available m
etrics are: loss,accuracy,true_negatives_3,true_positives_3,false_negatives_
3,false_positives_3,val_loss,val_accuracy,val_true_negatives_3,val_true_posi
tives_3,val_false_negatives_3,val_false_positives_3
408/408 [==============================] - 2664s 7s/step - loss: 2.9215 - ac
curacy: 0.9348 - true_negatives_3: 1842.0000 - true_positives_3: 1969.0000 -
false_negatives_3: 95.0000 - false_positives_3: 171.0000 - val_loss: 2.4192
- val_accuracy: 1.0000 - val_true_negatives_3: 402.0000 - val_true_positives
_3: 412.0000 - val_false_negatives_3: 0.0000e+00 - val_false_positives_3: 0.
0000e+00
Epoch 6/7
408/408 [==============================] - ETA: 0s - loss: 3.0047 - accurac
y: 0.9426 - true_negatives_3: 1860.0000 - true_positives_3: 1983.0000 - fals
e_negatives_3: 81.0000 - false_positives_3: 153.0000WARNING:tensorflow:Early
stopping conditioned on metric `val_acc` which is not available. Available m
etrics are: loss,accuracy,true_negatives_3,true_positives_3,false_negatives_
3,false_positives_3,val_loss,val_accuracy,val_true_negatives_3,val_true_posi
tives_3,val_false_negatives_3,val_false_positives_3
408/408 [==============================] - 2641s 6s/step - loss: 3.0047 - ac
curacy: 0.9426 - true_negatives_3: 1860.0000 - true_positives_3: 1983.0000 -
false_negatives_3: 81.0000 - false_positives_3: 153.0000 - val_loss: 1.9025
- val_accuracy: 1.0000 - val_true_negatives_3: 402.0000 - val_true_positives
_3: 412.0000 - val_false_negatives_3: 0.0000e+00 - val_false_positives_3: 0.
0000e+00
Epoch 7/7
408/408 [==============================] - ETA: 0s - loss: 2.3908 - accurac
y: 0.9497 - true_negatives_3: 1867.0000 - true_positives_3: 2005.0000 - fals
e_negatives_3: 59.0000 - false_positives_3: 146.0000    ETA: 11:24 - loss:
2.5516 - accuracy: 0.9475 - true_negatives_3: 1383.0000 - true - ETA: 3:55 -
loss: 2.4715 - accuracy: 0.9485 - true_negatives_3: 1698.0000 - true_positiv
es_3: 1837.0000 - false_negativeWARNING:tensorflow:Early stopping conditione
d on metric `val_acc` which is not available. Available metrics are: loss,ac
curacy,true_negatives_3,true_positives_3,false_negatives_3,false_positives_
3,val_loss,val_accuracy,val_true_negatives_3,val_true_positives_3,val_false_
negatives_3,val_false_positives_3
408/408 [==============================] - 2851s 7s/step - loss: 2.3908 - ac
curacy: 0.9497 - true_negatives_3: 1867.0000 - true_positives_3: 2005.0000 -
false_negatives_3: 59.0000 - false_positives_3: 146.0000 - val_loss: 1.3371
- val_accuracy: 1.0000 - val_true_negatives_3: 402.0000 - val_true_positives
_3: 412.0000 - val_false_negatives_3: 0.0000e+00 - val_false_positives_3: 0.
0000e+00
```

In [38]:

```
plt.plot(history.history['loss'])
plt.plot(history.history['val_loss'])
plt.title('loss')
plt.ylabel('loss')
plt.xlabel('epoch')
plt.legend(['train', 'val'], loc='upper left')
plt.show()
plt.plot(history.history['accuracy'])
plt.plot(history.history['val_accuracy'])
plt.title('loss')
plt.ylabel('loss')
plt.xlabel('epoch')
plt.legend(['train', 'val'], loc='upper left')
plt.show()
```





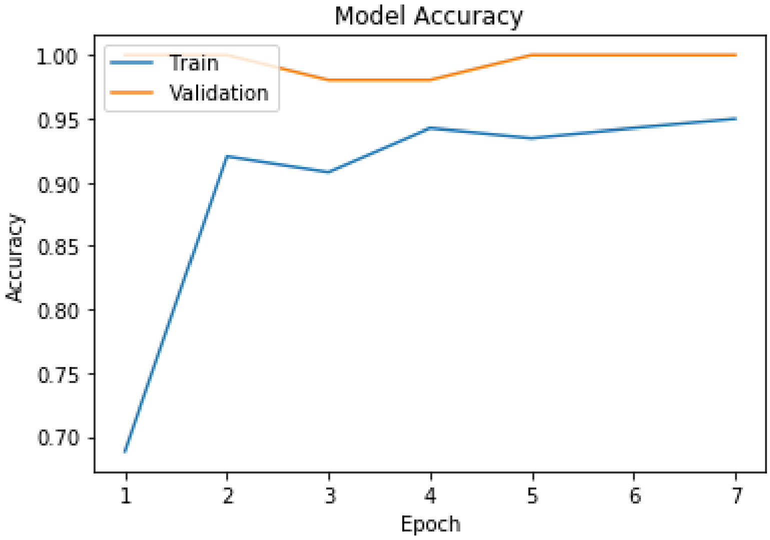
In [39]:

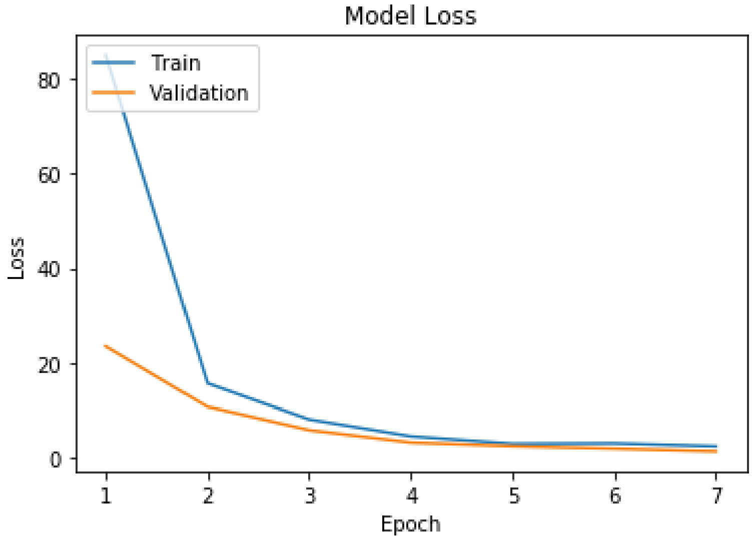
```
def plotLearningCurve(history,epochs):
  epochRange = range(1,epochs+1)
  plt.plot(epochRange,history.history['accuracy'])
  plt.plot(epochRange,history.history['val_accuracy'])
  plt.title('Model Accuracy')
  plt.xlabel('Epoch')
  plt.ylabel('Accuracy')
  plt.legend(['Train','Validation'],loc='upper left')
  plt.show()

  plt.plot(epochRange,history.history['loss'])
  plt.plot(epochRange,history.history['val_loss'])
  plt.title('Model Loss')
  plt.xlabel('Epoch')
  plt.ylabel('Loss')
  plt.legend(['Train','Validation'],loc='upper left')
  plt.show()
```

In [40]:

```
plotLearningCurve(history,7)
```





In [41]:

```
test = datagen.flow_from_directory(
        'A:/project/',
        color_mode='rgb',
        target_size=(512,512),
        class_mode='binary')



#testing
loss,accuracy,tn,tp,fn,fp=model.evaluate_generator(test)
print(accuracy,tn,tp,fn,fp)
```

```
Found 1000 images belonging to 2 classes.
WARNING:tensorflow:From <ipython-input-41-3f0d89588caa>:10: Model.evaluate_g
enerator (from tensorflow.python.keras.engine.training) is deprecated and wi
ll be removed in a future version.
Instructions for updating:
Please use Model.evaluate, which supports generators.
0.9570000171661377 468.0 489.0 11.0 32.0
```

In [ ]:

```

```