

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

```
Mounted at /content/drive
```

```
!pip install keras==2.10.0
!pip install tensorflow==2.10.0
!pip install h5py==3.7.0
```

```
Requirement already satisfied: protobuf<3.20,>=3.9.2 in /usr/local/lib/python3.8/dist-packages (from tensorflow==2.10.0) (3.19.5)
Requirement already satisfied: numpy>=1.20 in /usr/local/lib/python3.8/dist-packages (from tensorflow==2.10.0) (1.21.0)
Requirement already satisfied: tensorflow-io-gcs-filesystem>=0.23.1 in /usr/local/lib/python3.8/dist-packages (from tensorflow==2.10.0) (0.23.1)
Requirement already satisfied: grpcio<2.0,>=1.24.3 in /usr/local/lib/python3.8/dist-packages (from tensorflow==2.10.0) (1.24.3)
Requirement already satisfied: typing-extensions>=3.6.6 in /usr/local/lib/python3.8/dist-packages (from tensorflow==2.10.0) (3.7.4)
Requirement already satisfied: termcolor>=1.1.0 in /usr/local/lib/python3.8/dist-packages (from tensorflow==2.10.0) (1.1.0)
Requirement already satisfied: wheel<1.0,>=0.23.0 in /usr/local/lib/python3.8/dist-packages (from tensorflow==2.10.0) (0.23.0)
Requirement already satisfied: requests<3,>=2.21.0 in /usr/local/lib/python3.8/dist-packages (from tensorflow==2.10.0) (2.21.0)
Requirement already satisfied: tensorboard-data-server<0.7.0,>=0.6.0 in /usr/local/lib/python3.8/dist-packages (from tensorflow==2.10.0) (0.6.0)
Requirement already satisfied: google-auth<3,>=1.6.3 in /usr/local/lib/python3.8/dist-packages (from tensorflow==2.10.0) (1.6.3)
Requirement already satisfied: google-auth-oauthlib<0.5,>=0.4.1 in /usr/local/lib/python3.8/dist-packages (from tensorflow==2.10.0) (0.4.1)
Requirement already satisfied: werkzeug>=1.0.1 in /usr/local/lib/python3.8/dist-packages (from tensorflow==2.10.0) (1.0.1)
Requirement already satisfied: markdown>=2.6.8 in /usr/local/lib/python3.8/dist-packages (from tensorflow==2.10.0) (2.6.8)
Requirement already satisfied: tensorboard-plugin-wit>=1.6.0 in /usr/local/lib/python3.8/dist-packages (from tensorflow==2.10.0) (1.6.0)
Requirement already satisfied: cachetools<6.0,>=2.0.0 in /usr/local/lib/python3.8/dist-packages (from tensorflow==2.10.0) (4.1.0)
Requirement already satisfied: rsa<5,>=3.1.4 in /usr/local/lib/python3.8/dist-packages (from tensorflow==2.10.0) (4.7.1)
Requirement already satisfied: pyasn1-modules>=0.2.1 in /usr/local/lib/python3.8/dist-packages (from tensorflow==2.10.0) (0.2.1)
Requirement already satisfied: requests-oauthlib>=0.7.0 in /usr/local/lib/python3.8/dist-packages (from tensorflow==2.10.0) (0.7.0)
Requirement already satisfied: importlib-metadata>=4.4 in /usr/local/lib/python3.8/dist-packages (from tensorflow==2.10.0) (4.4.0)
Requirement already satisfied: zipp>=0.5 in /usr/local/lib/python3.8/dist-packages (from tensorflow==2.10.0) (0.5.2)
Requirement already satisfied: pyasn1<0.5.0,>=0.4.6 in /usr/local/lib/python3.8/dist-packages (from tensorflow==2.10.0) (0.4.8)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.8/dist-packages (from tensorflow==2.10.0) (2019.9.11)
Requirement already satisfied: urllib3!=1.25.0,!1.25.1,<1.26,>=1.21.1 in /usr/local/lib/python3.8/dist-packages (from tensorflow==2.10.0) (1.25.1)
Requirement already satisfied: chardet<4,>=3.0.2 in /usr/local/lib/python3.8/dist-packages (from tensorflow==2.10.0) (3.0.2)
Requirement already satisfied: idna<3,>=2.5 in /usr/local/lib/python3.8/dist-packages (from tensorflow==2.10.0) (2.5)
Requirement already satisfied: oauthlib>=3.0.0 in /usr/local/lib/python3.8/dist-packages (from tensorflow==2.10.0) (3.0.0)
Requirement already satisfied: pyparsing!=3.0.5,>=2.0.2 in /usr/local/lib/python3.8/dist-packages (from tensorflow==2.10.0) (2.0.2)
Installing collected packages: tensorflow-estimator, tensorboard, flatbuffers, tensorflow
Attempting uninstall: tensorflow-estimator
Found existing installation: tensorflow-estimator 2.9.0
Uninstalling tensorflow-estimator-2.9.0:
Successfully uninstalled tensorflow-estimator-2.9.0
Attempting uninstall: tensorboard
Found existing installation: tensorboard 2.9.1
Uninstalling tensorboard-2.9.1:
Successfully uninstalled tensorboard-2.9.1
Attempting uninstall: flatbuffers
Found existing installation: flatbuffers 1.12
Uninstalling flatbuffers-1.12:
Successfully uninstalled flatbuffers-1.12
Attempting uninstall: tensorflow
Found existing installation: tensorflow 2.9.2
Uninstalling tensorflow-2.9.2:
Successfully uninstalled tensorflow-2.9.2
Successfully installed flatbuffers-22.12.6 tensorboard-2.10.1 tensorflow-2.10.0 tensorflow-estimator-2.10.0
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Collecting h5py==3.7.0
  Downloading h5py-3.7.0-cp37-cp38-manylinux_2_12_x86_64.manylinux2010_x86_64.whl (4.5 MB)
    |██████████| 4.5 MB 29.9 MB/s
Requirement already satisfied: numpy>=1.14.5 in /usr/local/lib/python3.8/dist-packages (from h5py==3.7.0) (1.21.6)
Installing collected packages: h5py
  Attempting uninstall: h5py
    Found existing installation: h5py 3.1.0
    Uninstalling h5py-3.1.0:
      Successfully uninstalled h5py-3.1.0
  Successfully installed h5py-3.7.0
```

```
#!unzip /content/drive/MyDrive/pcos_classification/Data.zip -d /content/drive/MyDrive/pcos_classification/Data
```

```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import os
import cv2
from sklearn.model_selection import train_test_split
from tqdm import tqdm
from PIL import Image
import io
import tensorflow as tf
from tensorflow import keras
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from sklearn.utils import shuffle
```

```

from sklearn.model_selection import train_test_split
from tensorflow.keras.applications import EfficientNetB2
from keras.layers import GlobalAveragePooling2D, Dropout, Dense
from keras.models import Model
from tensorflow.keras import layers
from keras.layers import Conv2D, Input, ZeroPadding2D, BatchNormalization, Activation, MaxPooling2D, Flatten, Dense
from tensorflow.keras.callbacks import EarlyStopping, ReduceLROnPlateau, TensorBoard, ModelCheckpoint
from keras.layers import Dense, Dropout, Flatten, Conv2D, MaxPool2D, BatchNormalization, AveragePooling2D, GlobalAveragePoolin
from sklearn.metrics import classification_report, confusion_matrix
from IPython.display import display, clear_output
import ipywidgets as widgets
from tensorflow.keras.utils import plot_model
from sklearn.metrics import confusion_matrix, classification_report
import itertools
from tensorflow.keras import layers, models, optimizers
from keras.utils.np_utils import to_categorical
from keras.models import Sequential
from keras.applications.vgg19 import VGG19
from keras.layers import Dense, Dropout, Flatten, Conv2D, MaxPool2D
from tensorflow.keras.optimizers import RMSprop, Adam, SGD
from keras.preprocessing.image import ImageDataGenerator
from keras.callbacks import ReduceLROnPlateau
from keras.callbacks import EarlyStopping

```

### Loading Data

```

labels = ['infected', 'notinfected']
X_train = []
Y_train = []
X_test = []
Y_test = []
image_size=150 #sizeof image

for label in labels:
    trainPath = os.path.join('/content/drive/MyDrive/pcos_classification/Data/train',label)
    for file in tqdm(os.listdir(trainPath)):
        image = cv2.imread(os.path.join(trainPath, file))
        image = cv2.resize(image, (image_size, image_size))
        X_train.append(image)
        Y_train.append(label)

    testPath = os.path.join('/content/drive/MyDrive/pcos_classification/Data/test',label)
    for file in tqdm(os.listdir(testPath)):
        image = cv2.imread(os.path.join(testPath, file))
        image = cv2.resize(image, (image_size, image_size))
        X_test.append(image)
        Y_test.append(label)

X_train = np.array(X_train)
X_test = np.array(X_test)

100%|██████████| 781/781 [00:16<00:00, 46.81it/s]
100%|██████████| 98/98 [00:56<00:00, 1.74it/s]
100%|██████████| 1141/1141 [00:26<00:00, 43.03it/s]
100%|██████████| 98/98 [00:57<00:00, 1.69it/s]

```

```

print(X_train.shape)
print(X_test.shape)

(1922, 150, 150, 3)
(196, 150, 150, 3)

```

```

X_train, Y_train = shuffle(X_train, Y_train, random_state=10)
X_train.shape

(1922, 150, 150, 3)

```

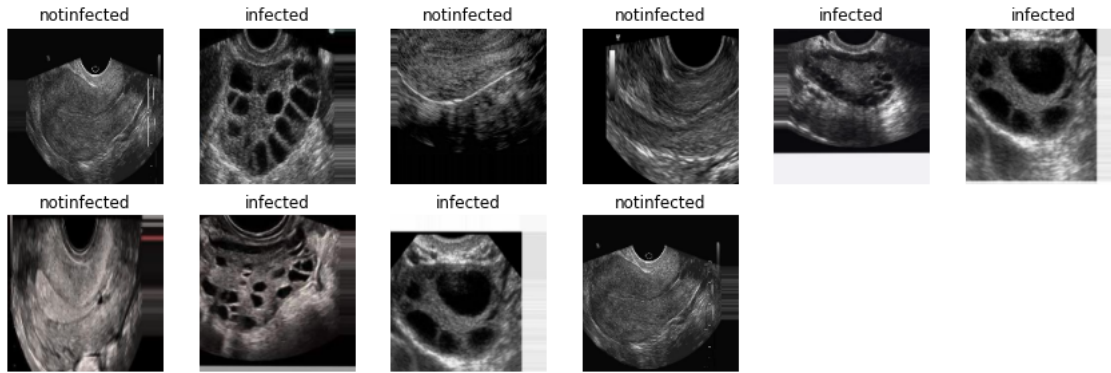
### Data Visualization

```

plt.figure(figsize=(15, 15))
class_names = Y_train
for i in range(0,10):
    for i in range(10):
        ax = plt.subplot(6, 6, i + 1)
        plt.imshow(X_train[i])
        plt.title(Y_train[i])
        plt.axis("off")

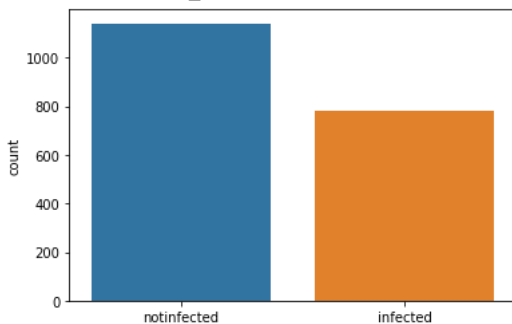
```

```
<ipython-input-10-7a626869507d>:5: MatplotlibDeprecationWarning: Adding an axes using the same arguments as a previous a
ax = plt.subplot(6, 6, i + 1)
```



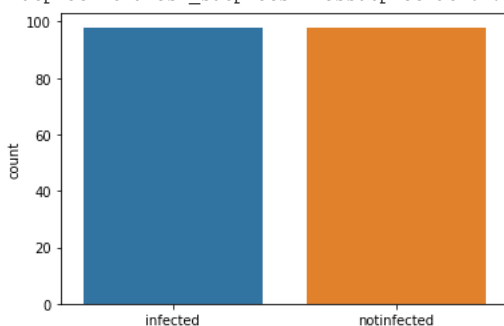
```
sns.countplot(Y_train)
```

```
/usr/local/lib/python3.8/dist-packages/seaborn/_decorators.py:36: FutureWarning: Pass the following variable as a keyword
warnings.warn(
<matplotlib.axes._subplots.AxesSubplot at 0x7f5ea6820370>
```



```
sns.countplot(Y_test)
```

```
/usr/local/lib/python3.8/dist-packages/seaborn/_decorators.py:36: FutureWarning: Pass the following variable as a keyword
warnings.warn(
<matplotlib.axes._subplots.AxesSubplot at 0x7f5ea4f74fa0>
```



```
#label encoding
```

```
y_train_ = []
```

```
for i in Y_train:
```

```
    y_train_.append(labels.index(i))
```

```
Y_train = y_train_
```

```
Y_train = tf.keras.utils.to_categorical(Y_train)
```

```
y_test_ = []
```

```
for i in Y_test:
```

```
    y_test_.append(labels.index(i))
```

```
Y_test = y_test_
```

```
Y_test = tf.keras.utils.to_categorical(Y_test)
```

```
x_train, x_val, y_train, y_val = train_test_split(X_train, Y_train, test_size=0.1, random_state=28) #splitting data
```

```
#Image augmentation
```

```
datagen = ImageDataGenerator(rescale = 1./255,
                             shear_range = 0.2,
                             zoom_range = 0.2,
                             horizontal_flip = True,
```

```

        vertical_flip=True,
        rotation_range=30,
        validation_split=0.3,
        fill_mode='nearest'
    )

batch_size = 32
nb_train_samples = x_train.shape[0]
nb_validation_samples = x_val.shape[0]
print(nb_train_samples)
print(nb_validation_samples)
training_generator = datagen.flow(x_train, y_train, batch_size=batch_size)
validation_generator = datagen.flow(x_val, y_val, batch_size=batch_size)

```

```

1729
193

```

## Resnet50

```

from tensorflow.keras.applications import resnet50
from tensorflow.keras.applications.imagenet_utils import preprocess_input

conv_model = resnet50.ResNet50(weights='imagenet', include_top=False, input_shape = (150,150,3))

for layer in conv_model.layers[:-3]:
    layer.trainable=False #The role of the embedding layer is to map a category into a dense space in a way that is useful for

resnet_model = models.Sequential()
resnet_model.add(layers.Conv2D(32,(3,3),activation = 'relu',name = 'Conv_',input_shape = (150,150,3)))
resnet_model.add(layers.Conv2D(32,(3,3),activation = 'relu',name = 'Conv_2',padding = 'same'))
resnet_model.add(layers.Conv2D(32,(3,3),activation = 'relu',name = 'Conv_3',padding = 'same'))
resnet_model.add(layers.BatchNormalization())
resnet_model.add(layers.MaxPooling2D((2,2),name = 'max_1'))
resnet_model.add(layers.Conv2D(64,(3,3),activation = 'relu',name = 'Conv_4',padding='same'))
resnet_model.add(layers.Conv2D(64,(3,3),activation = 'relu',name = 'Conv_5',padding='same'))
resnet_model.add(layers.BatchNormalization())
resnet_model.add(layers.MaxPooling2D((2,2),name = 'max_2'))

resnet_model.add(layers.Conv2D(128,(3,3),activation='relu'))
resnet_model.add(layers.BatchNormalization())
resnet_model.add(layers.MaxPooling2D((2,2)))
resnet_model.add(layers.Conv2D(128,(3,3),activation='relu'))
resnet_model.add(layers.BatchNormalization())
resnet_model.add(layers.Flatten())
resnet_model.add(layers.Dense(512,activation = 'relu',name = 'L1',))
resnet_model.add(layers.BatchNormalization())
resnet_model.add(layers.Dense(256,activation = 'relu',name = 'L2'))
resnet_model.add(layers.BatchNormalization())
resnet_model.add(layers.Dense(256,activation = 'relu',name = 'L3'))
resnet_model.add(layers.BatchNormalization())
resnet_model.add(layers.Dense(128,activation = 'relu' ,name ='L4'))
resnet_model.add(layers.BatchNormalization())
resnet_model.add(layers.Dense(2,activation = 'softmax',name = 'output'))

```

Downloading data from [https://storage.googleapis.com/tensorflow/keras-applications/resnet/resnet50\\_weights\\_tf\\_dim\\_ordering\\_94765736/94765736](https://storage.googleapis.com/tensorflow/keras-applications/resnet/resnet50_weights_tf_dim_ordering_94765736/94765736) [=====] - 4s 0us/step

```

resnet_model.compile(optimizer = optimizers.SGD(learning_rate=.0001) ,
                    loss='categorical_crossentropy',
                    metrics=['acc'])

```

```
resnet_model.summary()
```

Model: "sequential"

Layer (type)	Output Shape	Param #
=====		
Conv_ (Conv2D)	(None, 148, 148, 32)	896
Conv_2 (Conv2D)	(None, 148, 148, 32)	9248
Conv_3 (Conv2D)	(None, 148, 148, 32)	9248
batch_normalization (Batch Normalization)	(None, 148, 148, 32)	128
max_1 (MaxPooling2D)	(None, 74, 74, 32)	0
Conv_4 (Conv2D)	(None, 74, 74, 64)	18496

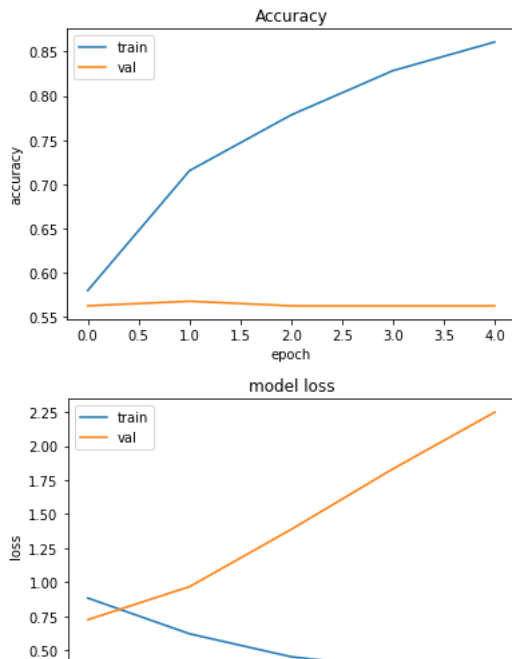
Conv_5 (Conv2D)	(None, 74, 74, 64)	36928
batch_normalization_1 (Batch Normalization)	(None, 74, 74, 64)	256
max_2 (MaxPooling2D)	(None, 37, 37, 64)	0
conv2d (Conv2D)	(None, 35, 35, 128)	73856
batch_normalization_2 (Batch Normalization)	(None, 35, 35, 128)	512
max_pooling2d (MaxPooling2D)	(None, 17, 17, 128)	0
conv2d_1 (Conv2D)	(None, 15, 15, 128)	147584
batch_normalization_3 (Batch Normalization)	(None, 15, 15, 128)	512
flatten (Flatten)	(None, 28800)	0
L1 (Dense)	(None, 512)	14746112
batch_normalization_4 (Batch Normalization)	(None, 512)	2048
L2 (Dense)	(None, 256)	131328
batch_normalization_5 (Batch Normalization)	(None, 256)	1024
L3 (Dense)	(None, 256)	65792
batch_normalization_6 (Batch Normalization)	(None, 256)	1024
L4 (Dense)	(None, 128)	32896
batch_normalization_7 (Batch Normalization)	(None, 128)	512

```
history1 = resnet_model.fit_generator(training_generator,
                                     steps_per_epoch = nb_train_samples // batch_size,
                                     epochs = 5,
                                     verbose = 1,
                                     validation_data = validation_generator,
                                     validation_steps = nb_validation_samples // batch_size)
```

```
<ipython-input-21-e45c2e522de0>:1: UserWarning: `Model.fit_generator` is deprecated and will be removed in a future version.
history1 = resnet_model.fit_generator(training_generator,
Epoch 1/5
54/54 [=====] - 22s 195ms/step - loss: 0.8827 - acc: 0.5798 - val_loss: 0.7245 - val_acc: 0.5625
Epoch 2/5
54/54 [=====] - 10s 181ms/step - loss: 0.6205 - acc: 0.7154 - val_loss: 0.9673 - val_acc: 0.5677
Epoch 3/5
54/54 [=====] - 10s 183ms/step - loss: 0.4516 - acc: 0.7784 - val_loss: 1.3881 - val_acc: 0.5625
Epoch 4/5
54/54 [=====] - 10s 183ms/step - loss: 0.3687 - acc: 0.8285 - val_loss: 1.8326 - val_acc: 0.5625
Epoch 5/5
54/54 [=====] - 10s 188ms/step - loss: 0.3060 - acc: 0.8609 - val_loss: 2.2487 - val_acc: 0.5625
```

```
#accuracy and loss plot
plt.plot(history1.history['acc'])
plt.plot(history1.history['val_acc'])
plt.title('Accuracy')
plt.ylabel('accuracy')
plt.xlabel('epoch')
plt.legend(['train', 'val'], loc='upper left')
plt.show()
```

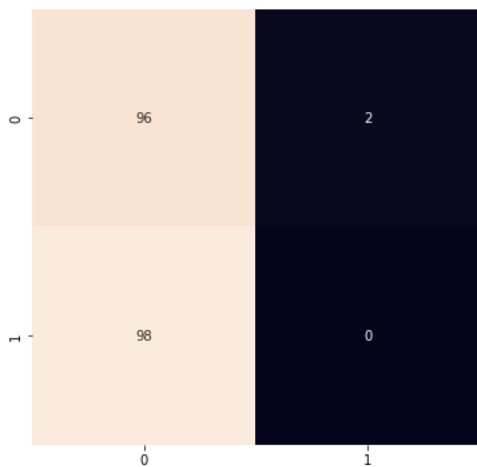
```
#loss plot
plt.plot(history1.history['loss'])
plt.plot(history1.history['val_loss'])
plt.title('model loss')
plt.ylabel('loss')
plt.xlabel('epoch')
plt.legend(['train', 'val'], loc='upper left')
plt.show()
```



```
pred = resnet_model.predict(X_test)
pred = np.argmax(pred,axis=1)
y_test_new = np.argmax(Y_test,axis=1)
```

```
cmat = confusion_matrix(y_test_new,pred)
plt.figure(figsize=(6,6))
sns.heatmap(cmat, annot = True, cbar = False);
```

7/7 [=====] - 1s 63ms/step



```
print(classification_report(y_test_new,pred))
```

	precision	recall	f1-score	support
0	0.49	0.98	0.66	98
1	0.00	0.00	0.00	98
accuracy			0.49	196
macro avg	0.25	0.49	0.33	196
weighted avg	0.25	0.49	0.33	196

```
from sklearn.metrics import precision_recall_fscore_support
res = []
for l in range(2):
    prec,recall,_,_ = precision_recall_fscore_support(y_test_new==l,
                                                    pred==l,
                                                    pos_label=True,average=None)
    res.append([l,recall[0],recall[1]])

pd.DataFrame(res,columns = ['class','sensitivity','specificity'])
```

```
class sensitivity specificity
```



## EfficientNet- B2

```
base_model = EfficientNetB2(weights='imagenet', include_top=False, input_shape=(image_size, image_size, 3))
```

Downloading data from [https://storage.googleapis.com/keras-applications/efficientnetb2\\_notop.h5](https://storage.googleapis.com/keras-applications/efficientnetb2_notop.h5)  
31790344/31790344 [=====] - 2s 0us/step

```
model2 = base_model.output
model2 = GlobalAveragePooling2D()(model2)
model2 = Dropout(0.4)(model2)
model2 = Dense(2, activation='softmax')(model2)
model2 = Model(inputs = base_model.input, outputs=model2)
```

```
model2.compile(loss='categorical_crossentropy', optimizer=Adam(0.1), metrics=['accuracy'])
```

```
model2.summary()
```

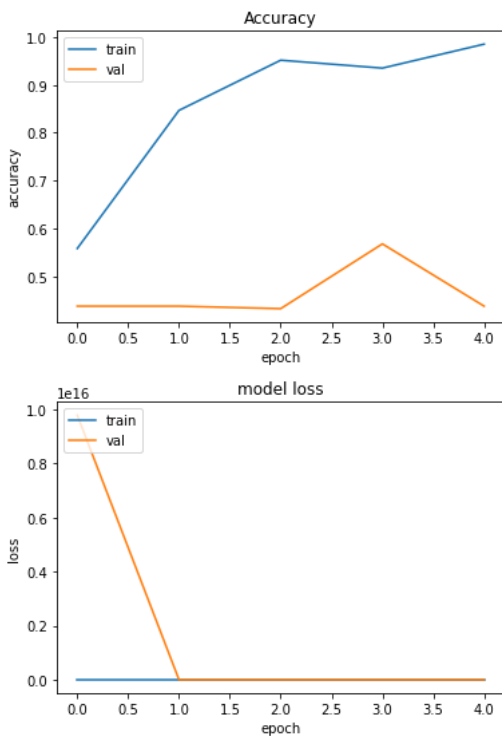
block2a_expand_conv (Conv2D)	(None, 75, 75, 96)	1536	['block1b_add[0][0]']
block2a_expand_bn (BatchNormalization)	(None, 75, 75, 96)	384	['block2a_expand_conv[0][0]']
block2a_expand_activation (Activation)	(None, 75, 75, 96)	0	['block2a_expand_bn[0][0]']
block2a_dwconv_pad (ZeroPadding2D)	(None, 77, 77, 96)	0	['block2a_expand_activation[0][0]']
block2a_dwconv (DepthwiseConv2D)	(None, 38, 38, 96)	864	['block2a_dwconv_pad[0][0]']
block2a_bn (BatchNormalization)	(None, 38, 38, 96)	384	['block2a_dwconv[0][0]']
block2a_activation (Activation)	(None, 38, 38, 96)	0	['block2a_bn[0][0]']
block2a_se_squeeze (GlobalAveragePooling2D)	(None, 96)	0	['block2a_activation[0][0]']
block2a_se_reshape (Reshape)	(None, 1, 1, 96)	0	['block2a_se_squeeze[0][0]']
block2a_se_reduce (Conv2D)	(None, 1, 1, 4)	388	['block2a_se_reshape[0][0]']
block2a_se_expand (Conv2D)	(None, 1, 1, 96)	480	['block2a_se_reduce[0][0]']
block2a_se_excite (Multiply)	(None, 38, 38, 96)	0	['block2a_activation[0][0]', 'block2a_se_expand[0][0]']
block2a_project_conv (Conv2D)	(None, 38, 38, 24)	2304	['block2a_se_excite[0][0]']
block2a_project_bn (BatchNormalization)	(None, 38, 38, 24)	96	['block2a_project_conv[0][0]']
block2b_expand_conv (Conv2D)	(None, 38, 38, 144)	3456	['block2a_project_bn[0][0]']
block2b_expand_bn (BatchNormalization)	(None, 38, 38, 144)	576	['block2b_expand_conv[0][0]']
block2b_expand_activation (Activation)	(None, 38, 38, 144)	0	['block2b_expand_bn[0][0]']
block2b_dwconv (DepthwiseConv2D)	(None, 38, 38, 144)	1296	['block2b_expand_activation[0][0]']
block2b_bn (BatchNormalization)	(None, 38, 38, 144)	576	['block2b_dwconv[0][0]']
block2b_activation (Activation)	(None, 38, 38, 144)	0	['block2b_bn[0][0]']
block2b_se_squeeze (GlobalAveragePooling2D)	(None, 144)	0	['block2b_activation[0][0]']
block2b_se_reshape (Reshape)	(None, 1, 1, 144)	0	['block2b_se_squeeze[0][0]']

```
history2 = model2.fit_generator(training_generator,
                                steps_per_epoch = nb_train_samples // batch_size,
                                epochs = 5,
                                verbose = 1,
                                validation_data = validation_generator,
                                validation_steps = nb_validation_samples // batch_size)
```

```
<ipython-input-30-ace8fa9073de>:1: UserWarning: `Model.fit_generator` is deprecated and will be removed in a future version
history2 = model2.fit_generator(training_generator,
Epoch 1/5
54/54 [=====] - 31s 292ms/step - loss: 6.2870 - accuracy: 0.5580 - val_loss: 9793284570349568.00
Epoch 2/5
54/54 [=====] - 13s 232ms/step - loss: 1.2227 - accuracy: 0.8468 - val_loss: 284496544.0000 - val_
Epoch 3/5
54/54 [=====] - 14s 262ms/step - loss: 0.4921 - accuracy: 0.9517 - val_loss: 2726200.2500 - val_
Epoch 4/5
54/54 [=====] - 13s 232ms/step - loss: 0.5393 - accuracy: 0.9352 - val_loss: 572822.8125 - val_
Epoch 5/5
54/54 [=====] - 13s 236ms/step - loss: 0.1215 - accuracy: 0.9853 - val_loss: 18233.6738 - val_ac
```

```
#accuracy and loss plot
plt.plot(history2.history['accuracy'])
plt.plot(history2.history['val_accuracy'])
plt.title('Accuracy')
plt.ylabel('accuracy')
plt.xlabel('epoch')
plt.legend(['train', 'val'], loc='upper left')
plt.show()
```

```
#loss plot
plt.plot(history2.history['loss'])
plt.plot(history2.history['val_loss'])
plt.title('model loss')
plt.ylabel('loss')
plt.xlabel('epoch')
plt.legend(['train', 'val'], loc='upper left')
plt.show()
```

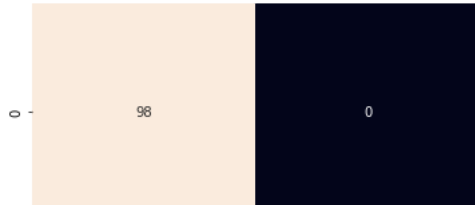


```
pred = model2.predict(X_test)
pred = np.argmax(pred,axis=1)
y_test_new = np.argmax(Y_test,axis=1)

cmat = confusion_matrix(y_test_new,pred)
plt.figure(figsize=(6,6))
sns.heatmap(cmat, annot = True, cbar = False);
```



7/7 [=====] - 3s 133ms/step



```
print(classification_report(y_test_new, pred))
```

	precision	recall	f1-score	support
0	0.50	1.00	0.67	98
1	0.00	0.00	0.00	98
accuracy			0.50	196
macro avg	0.25	0.50	0.33	196
weighted avg	0.25	0.50	0.33	196

```
/usr/local/lib/python3.8/dist-packages/sklearn/metrics/_classification.py:1318: UndefinedMetricWarning: Precision and F-score  
_warn_prf(average, modifier, msg_start, len(result))  
/usr/local/lib/python3.8/dist-packages/sklearn/metrics/_classification.py:1318: UndefinedMetricWarning: Precision and F-score  
_warn_prf(average, modifier, msg_start, len(result))  
/usr/local/lib/python3.8/dist-packages/sklearn/metrics/_classification.py:1318: UndefinedMetricWarning: Precision and F-score  
_warn_prf(average, modifier, msg_start, len(result))
```

```
from sklearn.metrics import precision_recall_fscore_support  
res = []  
for l in range(2):  
    prec, recall, _, _ = precision_recall_fscore_support(y_test_new==l,  
                                                         pred==l,  
                                                         pos_label=True, average=None)  
    res.append([l, recall[0], recall[1]])  
  
pd.DataFrame(res, columns = ['class', 'sensitivity', 'specificity'])
```

```
/usr/local/lib/python3.8/dist-packages/sklearn/metrics/_classification.py:1318: UndefinedMetricWarning: Precision and F-score  
_warn_prf(average, modifier, msg_start, len(result))  
/usr/local/lib/python3.8/dist-packages/sklearn/metrics/_classification.py:1318: UndefinedMetricWarning: Precision and F-score  
_warn_prf(average, modifier, msg_start, len(result))
```

	class	sensitivity	specificity
0	0	0.0	1.0
1	1	1.0	0.0

## CNN Model

```
model = Sequential()  
model.add(Conv2D(15, (5,5),padding='valid',activation='relu',input_shape=(150,150,3)))  
model.add(MaxPooling2D(pool_size=(5,5)))  
model.add(Conv2D(12, (4,4),padding='valid',activation='relu'))  
model.add(MaxPooling2D(pool_size=(4,4)))  
model.add(Dropout(0.65))  
model.add(Conv2D(8, (1,1),padding='valid',activation='relu'))  
model.add(MaxPooling2D(pool_size=(1,1)))  
model.add(Dropout(0.55))  
model.add(Flatten())  
model.add(Dense(2,activation='softmax'))
```

```
model.compile(optimizer='adam', loss="categorical_crossentropy", metrics=['accuracy'])
```

```
model.summary()
```

```
Model: "sequential_1"
```

Layer (type)	Output Shape	Param #
conv2d_2 (Conv2D)	(None, 146, 146, 15)	1140
max_pooling2d_1 (MaxPooling 2D)	(None, 29, 29, 15)	0
conv2d_3 (Conv2D)	(None, 26, 26, 12)	2892
max_pooling2d_2 (MaxPooling 2D)	(None, 6, 6, 12)	0

2D)

dropout_1 (Dropout)	(None, 6, 6, 12)	0
conv2d_4 (Conv2D)	(None, 6, 6, 8)	104
max_pooling2d_3 (MaxPooling 2D)	(None, 6, 6, 8)	0
dropout_2 (Dropout)	(None, 6, 6, 8)	0
flatten_1 (Flatten)	(None, 288)	0
dense_1 (Dense)	(None, 2)	578

```

=====
Total params: 4,714
Trainable params: 4,714
Non-trainable params: 0

```

```

history3 = model.fit_generator(training_generator,
                               steps_per_epoch = nb_train_samples // batch_size,
                               epochs = 5,
                               verbose = 1,
                               validation_data = validation_generator,
                               validation_steps = nb_validation_samples // batch_size)

```

Epoch 1/5

<ipython-input-45-36cb8bb97a7d>:1: UserWarning: `Model.fit\_generator` is deprecated and will be removed in a future version

```

history3 = model.fit_generator(training_generator,
54/54 [=====] - 9s 164ms/step - loss: 0.1509 - accuracy: 0.9481 - val_loss: 0.0540 - val_accuracy: 0.9540
Epoch 2/5
54/54 [=====] - 9s 163ms/step - loss: 0.1222 - accuracy: 0.9593 - val_loss: 0.0546 - val_accuracy: 0.9546
Epoch 3/5
54/54 [=====] - 9s 160ms/step - loss: 0.1219 - accuracy: 0.9646 - val_loss: 0.0311 - val_accuracy: 0.9646
Epoch 4/5
54/54 [=====] - 9s 163ms/step - loss: 0.1034 - accuracy: 0.9717 - val_loss: 0.0249 - val_accuracy: 0.9717
Epoch 5/5
54/54 [=====] - 10s 176ms/step - loss: 0.0783 - accuracy: 0.9770 - val_loss: 0.0223 - val_accuracy: 0.9770

```

```
model.save('/content/drive/MyDrive/pcos_classification/models/cnn.h5')
```

#accuracy and loss plot

```

plt.plot(history3.history['accuracy'])
plt.plot(history3.history['val_accuracy'])
plt.title('Accuracy')
plt.ylabel('accuracy')
plt.xlabel('epoch')
plt.legend(['train', 'val'], loc='upper left')
plt.show()

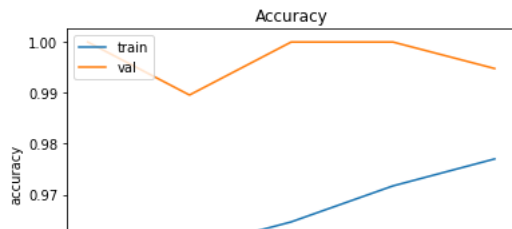
```

#loss plot

```

plt.plot(history3.history['loss'])
plt.plot(history3.history['val_loss'])
plt.title('model loss')
plt.ylabel('loss')
plt.xlabel('epoch')
plt.legend(['train', 'val'], loc='upper left')
plt.show()

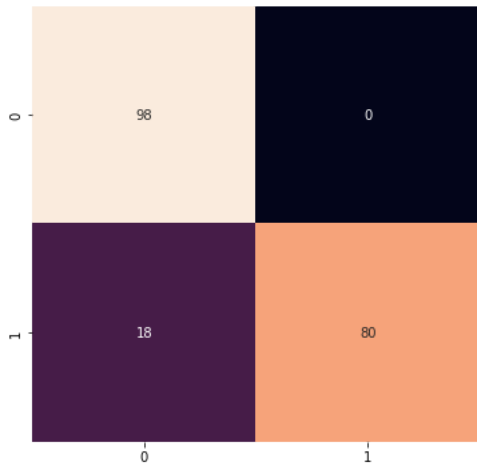
```



```
pred = model.predict(X_test)
pred = np.argmax(pred,axis=1)
y_test_new = np.argmax(Y_test,axis=1)

cmat = confusion_matrix(y_test_new,pred)
plt.figure(figsize=(6,6))
sns.heatmap(cmat, annot = True, cbar = False);
```

7/7 [=====] - 0s 5ms/step



```
print(classification_report(y_test_new,pred))
```

	precision	recall	f1-score	support
0	0.84	1.00	0.92	98
1	1.00	0.82	0.90	98
accuracy			0.91	196
macro avg	0.92	0.91	0.91	196
weighted avg	0.92	0.91	0.91	196

```
from sklearn.metrics import precision_recall_fscore_support
res = []
for l in range(2):
    prec,recall,_,_ = precision_recall_fscore_support(y_test_new==l,
                                                    pred==l,
                                                    pos_label=True,average=None)
    res.append([l,recall[0],recall[1]])

pd.DataFrame(res,columns = ['class','sensitivity','specificity'])
```

	class	sensitivity	specificity
0	0	0.816327	1.000000
1	1	1.000000	0.816327



✓ 0s completed at 6:25 PM

● ✕