# Library Import

```
from google.colab import drive
#drive.mount('/content/drive')
drive.mount("/content/drive", force_remount=True)
```

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

```
import glob, random
import keras, os, sys
import tensorflow as tf
import cv2
import matplotlib.pyplot as plt
import matplotlib.image as mpimg
import numpy as np
import pandas as pd
import imageio
import warnings
warnings.filterwarnings("ignore")
import matplotlib.pyplot as plt

from PIL import Image

from astropy.io import fits
from astropy.visualization import astropy_mpl_style
plt.style.use(astropy_mpl_style)
from astropy.utils.data import get_pkg_data_filename



from imblearn.over_sampling import SMOTE
from collections import Counter
from sklearn.decomposition import PCA
from sklearn.preprocessing import StandardScaler
from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import train_test_split
from sklearn.svm import SVC
from sklearn.neighbors import KNeighborsClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score


from tensorflow.keras import backend as K
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.models import Sequential, Model
from tensorflow.keras.layers import Conv1D, Conv2D, MaxPool2D, Dense, Flatten, Layer, Input, Dropout
from tensorflow.python.util import deprecation
deprecation._PRINT_DEPRECATION_WARNINGS = True
from tensorflow.keras.initializers import glorot_uniform
from tensorflow.keras import initializers, layers
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.applications.vgg16 import VGG16
from tensorflow.keras.applications.vgg19 import VGG19
from tensorflow.keras.applications.vgg19 import preprocess_input
from tensorflow.keras.applications.vgg16 import preprocess_input
```

```
exoTrain = glob.glob("/content/drive/MyDrive/exoplanet/Data/trainingData/ConfirmedExoplanets/*.png")
print ("Total of %d  images." % len(exoTrain))
print ('the filenames:\n')
print("\n".join(exoTrain[:5]))
```

```
Total of 15  images.
the filenames:

/content/drive/MyDrive/exoplanet/Data/trainingData/ConfirmedExoplanets/kplr010874614-2012179063303_llc1655377743.png
/content/drive/MyDrive/exoplanet/Data/trainingData/ConfirmedExoplanets/kplr010874614-2012277125453_llc1655377743.png
/content/drive/MyDrive/exoplanet/Data/trainingData/ConfirmedExoplanets/kplr010874614-2011271113734_llc1655377742.png
/content/drive/MyDrive/exoplanet/Data/trainingData/ConfirmedExoplanets/kplr010874614-2012088054726_llc1655377742.png
/content/drive/MyDrive/exoplanet/Data/trainingData/ConfirmedExoplanets/kplr010874614-2011177032512_llc1655377741.png
```

```
epTest = glob.glob("/content/drive/MyDrive/exoplanet/Data/testingData/ConfirmedExoplanets/*.png")
print ("Total of %d  images." % len(epTest))
```

```
print ('the filenames:\n')
print("\n".join(epTest[:5]))
```

```
Total of 4  images.
the filenames:

/content/drive/MyDrive/exoplanet/Data/testingData/ConfirmedExoplanets/kplr010874614-2013131215648_llc1655377744.png
/content/drive/MyDrive/exoplanet/Data/testingData/ConfirmedExoplanets/kplr010874614-2012179063303_llc1655377743.png
/content/drive/MyDrive/exoplanet/Data/testingData/ConfirmedExoplanets/kplr010874614-2012277125453_llc1655377743.png
/content/drive/MyDrive/exoplanet/Data/testingData/ConfirmedExoplanets/kplr010874614-2013098041711_llc1655377744.png
```

```
fpTrain = glob.glob("/content/drive/MyDrive/exoplanet/Data/trainingData/FalsePositiveExoplanets/*.png")
print ("Total of %d  images." % len(fpTrain))
print ('the filenames:\n')
print("\n".join(fpTrain[:5]))
```

```
Total of 14  images.
the filenames:

/content/drive/MyDrive/exoplanet/Data/trainingData/FalsePositiveExoplanets/kplr000892772-2012088054726_llc1655377853.png
/content/drive/MyDrive/exoplanet/Data/trainingData/FalsePositiveExoplanets/kplr000892772-2012179063303_llc1655377853.png
/content/drive/MyDrive/exoplanet/Data/trainingData/FalsePositiveExoplanets/kplr000892772-2012277125453_llc1655377855.png
/content/drive/MyDrive/exoplanet/Data/trainingData/FalsePositiveExoplanets/kplr000892772-2012004120508_llc1655377852.png
/content/drive/MyDrive/exoplanet/Data/trainingData/FalsePositiveExoplanets/kplr000892772-2011271113734_llc1655377852.png
```

```
fpTest = glob.glob("/content/drive/MyDrive/exoplanet/Data/testingData/FalsePositiveExoplanets/*.png")
print ("Total of %d  images." % len(fpTest))
print ('the filenames:\n')
print("\n".join(fpTest[:5]))
```

```
Total of 3  images.
the filenames:

/content/drive/MyDrive/exoplanet/Data/testingData/FalsePositiveExoplanets/kplr000892772-2013131215648_llc1655377855.png
/content/drive/MyDrive/exoplanet/Data/testingData/FalsePositiveExoplanets/kplr000892772-2013011073258_llc1655377854.png
/content/drive/MyDrive/exoplanet/Data/testingData/FalsePositiveExoplanets/kplr000892772-2013098041711_llc1655377855.png
```

```
kepler = pd.read_csv('/content/drive/MyDrive/exoplanet/keplerData.csv', skiprows=144)
kepler.head()
```

| | rowid | kepid | kepoi_name | kepler_name | koi_disposition | koi_vet_stat | koi_vet_dat |
|---|---|---|---|---|---|---|---|
| **0** | 1 | 11446443 | K00001.01 | Kepler-1 b | CONFIRMED | Done | 2018-08-1 |
| **1** | 2 | 10666592 | K00002.01 | Kepler-2 b | CONFIRMED | Done | 2018-08-1 |
| **2** | 3 | 10748390 | K00003.01 | Kepler-3 b | CONFIRMED | Done | 2018-08-1 |
| **3** | 4 | 3861595 | K00004.01 | Kepler-1658 b | CONFIRMED | Done | 2018-08-1 |
| **4** | 5 | 8554498 | K00005.01 | NaN | CANDIDATE | Done | 2018-08-1 |

5 rows × 141 columns

```
kepler = kepler.drop(['rowid'],1).reset_index(drop=True)
kepler.head()
```

| | kepid | kepoi_name | kepler_name | koi_disposition | koi_vet_stat | koi_vet_date | koi_ |
|---|---|---|---|---|---|---|---|
| **0** | 11446443 | K00001.01 | Kepler-1 b | CONFIRMED | Done | 2018-08-16 | |
| **1** | 10666592 | K00002.01 | Kepler-2 b | CONFIRMED | Done | 2018-08-16 | |
| **2** | 10748390 | K00003.01 | Kepler-3 b | CONFIRMED | Done | 2018-08-16 | |
| **3** | 3861595 | K00004.01 | Kepler-1658 b | CONFIRMED | Done | 2018-08-16 | |
| **4** | 8554498 | K00005.01 | NaN | CANDIDATE | Done | 2018-08-16 | |

5 rows × 140 columns

## Data Processing

```
TRAIN_DIR = '/content/drive/MyDrive/exoplanet/Data/trainingData/'
TEST_DIR = '/content/drive/MyDrive/exoplanet/Data/testingData/'
CATEGORIES = ['ConfirmedExoplanets','FalsePositiveExoplanets']
IMG_SIZE=100
scores= []
```

```
training_data=[]
def create_training_data():
    for category in CATEGORIES:
        path=os.path.join(TRAIN_DIR, category)
        class_num=CATEGORIES.index(category)
        for img in os.listdir(path):
            try:
                img_array=cv2.imread(os.path.join(path,img))
                new_array=cv2.resize(img_array,(IMG_SIZE,IMG_SIZE))
                training_data.append([new_array,class_num])
            except Exception as e:
                pass

create_training_data()

print(len(training_data))
```

```
29
```

```
lenofimage = len(training_data)
X_train=[]
y_train=[]

for categories, label in training_data:
    X_train.append(categories)
    y_train.append(label)
```

```
X_train= np.array(X_train).reshape(lenofimage,-1)
```

```
X_train = X_train/255.0
X_train.shape
```

```
(29, 30000)
```

```
y_train=np.array(y_train)
y_train.shape
```

```
(29,)
```

```
testing_data=[]
def create_testing_data():
    for category in CATEGORIES:
        path=os.path.join(TEST_DIR, category)
        class_num=CATEGORIES.index(category)
        for img in os.listdir(path):
            try:
                img_array=cv2.imread(os.path.join(path,img))
                new_array=cv2.resize(img_array,(IMG_SIZE,IMG_SIZE))
                testing_data.append([new_array,class_num])
            except Exception as e:
                pass

create_testing_data()

print(len(testing_data))
```

```
7
```

```
lenofimage = len(testing_data)
X_test=[]
y_test=[]
```

```
for categories, label in testing_data:
    X_test.append(categories)
    y_test.append(label)



X_test= np.array(X_test).reshape(lenofimage,-1)


X_test = X_test/255.0
print(X_test.shape)
```

```
(7, 30000)
```

```
y_test=np.array(y_test)
y_test.shape
```

```
(7,)
```

```
kepler.info()
```

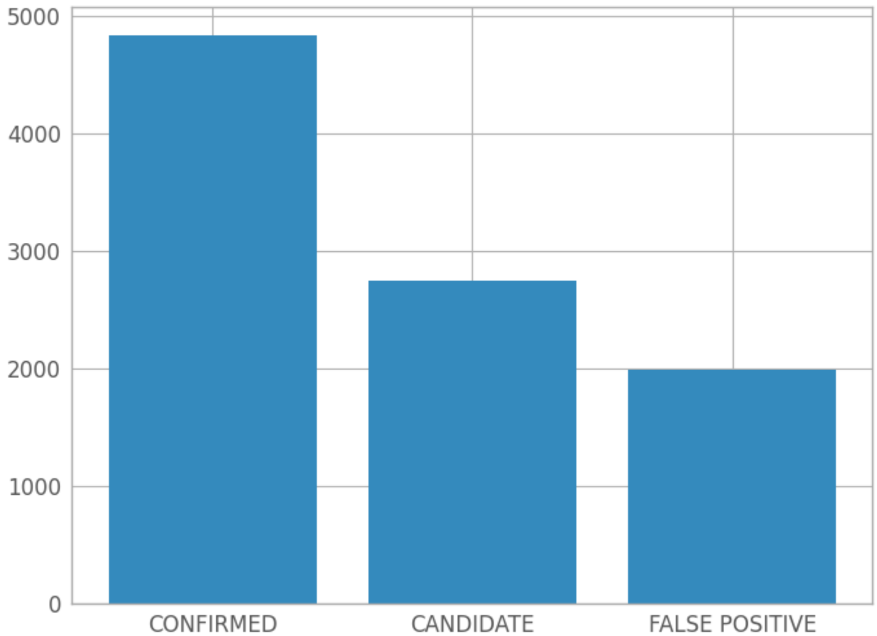
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9564 entries, 0 to 9563
Columns: 140 entries, kepid to koi_dikco_msky_err
dtypes: float64(124), int64(6), object(10)
memory usage: 10.2+ MB
```

```
pd.DataFrame(round((kepler.isnull().sum() * 100/ len(kepler)),2).sort_values(ascending=False)).head(30)
```

| | 0 |
|---|---|
| **koi_ldm_coeff1** | 100.0 |
| **koi_model_chisq** | 100.0 |

```
classes=kepler.koi_disposition.unique()
counts = kepler.koi_disposition.value_counts().to_list()
plt.bar(classes,counts)
print(kepler.koi_disposition.value_counts())
```

```
FALSE POSITIVE    4839
CONFIRMED         2741
CANDIDATE         1984
Name: koi_disposition, dtype: int64
```




```
# Dropping rows with more than 80% data missing
kepler = kepler.dropna(thresh=len(kepler) * .80, axis=1)
kepler.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9564 entries, 0 to 9563
Data columns (total 34 columns):
 #   Column             Non-Null Count  Dtype  
---  ------             --------------  -----  
 0   kepid              9564 non-null   int64  
 1   kepoi_name         9564 non-null   object 
 2   koi_disposition    9564 non-null   object 
 3   koi_vet_stat       9564 non-null   object 
 4   koi_vet_date       9564 non-null   object 
 5   koi_pdisposition   9564 non-null   object 
 6   koi_fpflag_nt      9564 non-null   int64  
 7   koi_fpflag_ss      9564 non-null   int64  
 8   koi_fpflag_co      9564 non-null   int64  
 9   koi_fpflag_ec      9564 non-null   int64  
 10  koi_disp_prov      9564 non-null   object 
 11  koi_period         9564 non-null   float64
 12  koi_period_err1    9103 non-null   float64
 13  koi_period_err2    9103 non-null   float64
 14  koi_time0bk        9564 non-null   float64
 15  koi_time0bk_err1   9110 non-null   float64
 16  koi_time0bk_err2   9110 non-null   float64
 17  koi_time0          9564 non-null   float64
 18  koi_time0_err1     9110 non-null   float64
 19  koi_time0_err2     9110 non-null   float64
 20  koi_parm_prov      9564 non-null   object 
 21  koi_count          9564 non-null   int64  
 22  koi_tce_delivname  9564 non-null   object 
 23  koi_datalink_dvr   8478 non-null   object 
 24  ra                 9564 non-null   float64
 25  dec                9564 non-null   float64
 26  koi_kepmag         9563 non-null   float64
 27  koi_gmag           9523 non-null   float64
 28  koi_rmag           9555 non-null   float64
```

```
 29  koi_imag           9410 non-null   float64
 30  koi_zmag           8951 non-null   float64
 31  koi_jmag           9539 non-null   float64
 32  koi_hmag           9539 non-null   float64
 33  koi_kmag           9539 non-null   float64
dtypes: float64(19), int64(6), object(9)
memory usage: 2.5+ MB
```

```
for column in kepler.columns:
    print(column)
    print(kepler[column].unique())
```

```
koi_fpflag_ec
[0 1]
koi_disp_prov
['q1_q17_dr25_sup_koi']
koi_period
[  2.47061338   2.20473542   4.88780308 ...   1.75647084 272.54288087
 229.957537  ]
koi_period_err1
[2.700e-08 4.300e-08 4.660e-07 ... 2.276e-02 3.684e-03 6.728e-03]
koi_period_err2
[-2.700e-08 -4.300e-08 -4.660e-07 ... -2.276e-02 -3.684e-03 -6.728e-03]
koi_time0bk
[122.763305  121.3585417 124.8130808 ... 132.02757   349.7527344
 326.0184   ]
koi_time0bk_err1
[8.70e-06 1.60e-05 7.51e-05 ... 6.48e-02 6.43e-02 6.23e-02]
koi_time0bk_err2
[-8.70e-06 -1.60e-05 -7.51e-05 ... -6.48e-02 -6.43e-02 -6.23e-02]
koi_time0
[2454955.763 2454954.359 2454957.813 ... 2454965.028 2455182.753
 2455159.018]
koi_time0_err1
[8.70e-06 1.60e-05 7.51e-05 ... 6.48e-02 6.43e-02 6.23e-02]
koi_time0_err2
[-8.70e-06 -1.60e-05 -7.51e-05 ... -6.48e-02 -6.43e-02 -6.23e-02]
koi_parm_prov
['q1_q17_dr25_sup_koi']
koi_count
[1 2 3 5 6 4 7]
koi_tce_delivname
['q1_q17_dr25_tce']
koi_datalink_dvr
['011/011446/011446443/dv/kplr011446443-20160209194854_dvr.pdf'
 '010/010666/010666592/dv/kplr010666592-20160209194854_dvr.pdf'
 '010/010748/010748390/dv/kplr010748390-20160209194854_dvr.pdf' ...
 '011/011923/011923074/dv/kplr011923074-20160209194854_dvr.pdf'
 '012/012117/012117215/dv/kplr012117215-20160209194854_dvr.pdf'
 '012/012168/012168280/dv/kplr012168280-20160209194854_dvr.pdf']
ra
[286.80847 292.24728 297.70935 ... 296.14072 294.92795 295.97794]
dec
[49.316399 47.969521 48.080853 ... 50.279949 50.662369 50.771481]
koi_kepmag
[11.338 10.463  9.174 ... 16.652 10.736 14.723]
koi_gmag
[11.736 10.935 10.665 ... 17.378 11.375 10.714]
koi_rmag
[11.275 10.49   9.479 ... 10.67  15.062 10.435]
koi_imag
[11.168    nan 11.294 ... 16.341 10.463 10.412]
koi_zmag
[11.126    nan 11.305 ... 16.27  10.312 10.409]
koi_jmag
[10.232  9.555  7.608 ... 15.234  9.286  9.618]
koi_hmag
[ 9.92   9.344  7.131 ... 13.62   8.834  9.448]
koi_kmag
[ 9.846  9.334  7.009 ... 12.599 14.744  8.772]
```

```
kepler.drop(['kepid','kepoi_name','koi_vet_stat','koi_vet_date','koi_disp_prov','koi_pdisposition','koi_datalink_dvr',
                      'koi_parm_prov','koi_tce_delivname'], inplace=True, axis=1)
```

```
kepler.dropna(inplace=True)
kepler.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 8515 entries, 0 to 9563
Data columns (total 25 columns):
 #   Column            Non-Null Count  Dtype
```

```
 ---  ------            --------------  -----
  0   koi_disposition   8515 non-null   object
  1   koi_fpflag_nt     8515 non-null   int64
  2   koi_fpflag_ss     8515 non-null   int64
  3   koi_fpflag_co     8515 non-null   int64
  4   koi_fpflag_ec     8515 non-null   int64
  5   koi_period        8515 non-null   float64
  6   koi_period_err1   8515 non-null   float64
  7   koi_period_err2   8515 non-null   float64
  8   koi_time0bk       8515 non-null   float64
  9   koi_time0bk_err1  8515 non-null   float64
  10  koi_time0bk_err2  8515 non-null   float64
  11  koi_time0         8515 non-null   float64
  12  koi_time0_err1    8515 non-null   float64
  13  koi_time0_err2    8515 non-null   float64
  14  koi_count         8515 non-null   int64
  15  ra                8515 non-null   float64
  16  dec               8515 non-null   float64
  17  koi_kepmag        8515 non-null   float64
  18  koi_gmag          8515 non-null   float64
  19  koi_rmag          8515 non-null   float64
  20  koi_imag          8515 non-null   float64
  21  koi_zmag          8515 non-null   float64
  22  koi_jmag          8515 non-null   float64
  23  koi_hmag          8515 non-null   float64
  24  koi_kmag          8515 non-null   float64
dtypes: float64(19), int64(5), object(1)
memory usage: 1.7+ MB
```

## Label encoding

```
le = LabelEncoder()
```

```
kepler['koi_disposition'] = le.fit_transform(kepler['koi_disposition'])
```

```
keplerX= kepler.drop(['koi_disposition'],axis=1)
keplerY = kepler['koi_disposition']
```

## Dimension Reduction

```
pca = PCA(n_components=0.95)
```

```
# fit and transform data
X_train = pca.fit_transform(X_train)
X_train.shape
```

```
(29, 26)
```

```
X_test = pca.transform(X_test)
X_test.shape
```

```
(7, 26)
```

pca = PCA(n_components=15) keplerX = pca.fit_transform(keplerX) keplerX.shape

```
sc = StandardScaler()
keplerX = sc.fit_transform(keplerX)
keplerX.shape
```

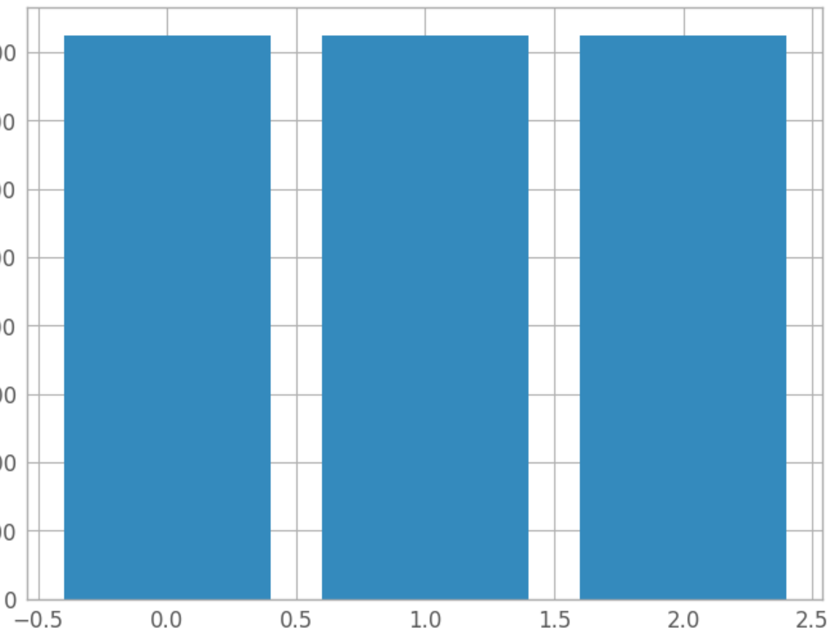
```
(8515, 24)
```

## Class Balancing

```
oversample = SMOTE()
keplerX, keplerY = oversample.fit_resample(keplerX, keplerY)
```

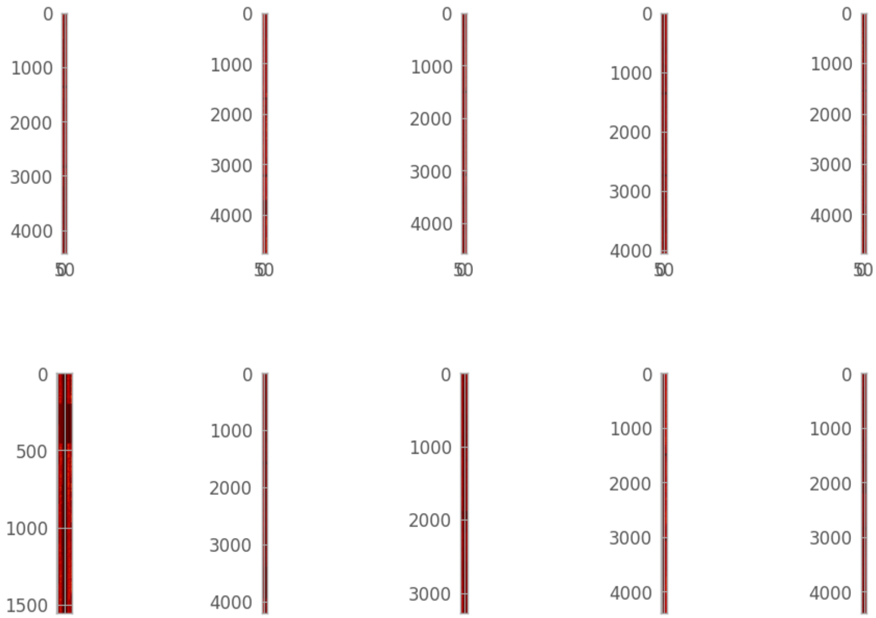
```
counter = Counter(keplerY)
for k,v in counter.items():
    per = v / len(keplerY) * 100
    print('Class=%d, n=%d (%.3f%%)' % (k, v, per))
plt.bar(counter.keys(), counter.values())
plt.show()
```

```
Class=1, n=4124 (33.333%)
Class=0, n=4124 (33.333%)
Class=2, n=4124 (33.333%)
```

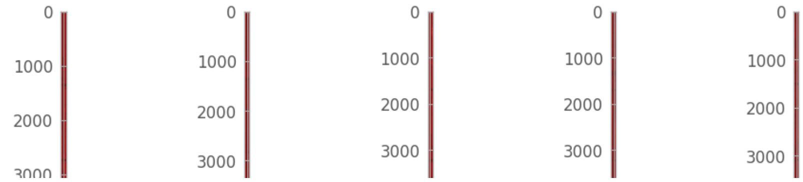



## Astropy

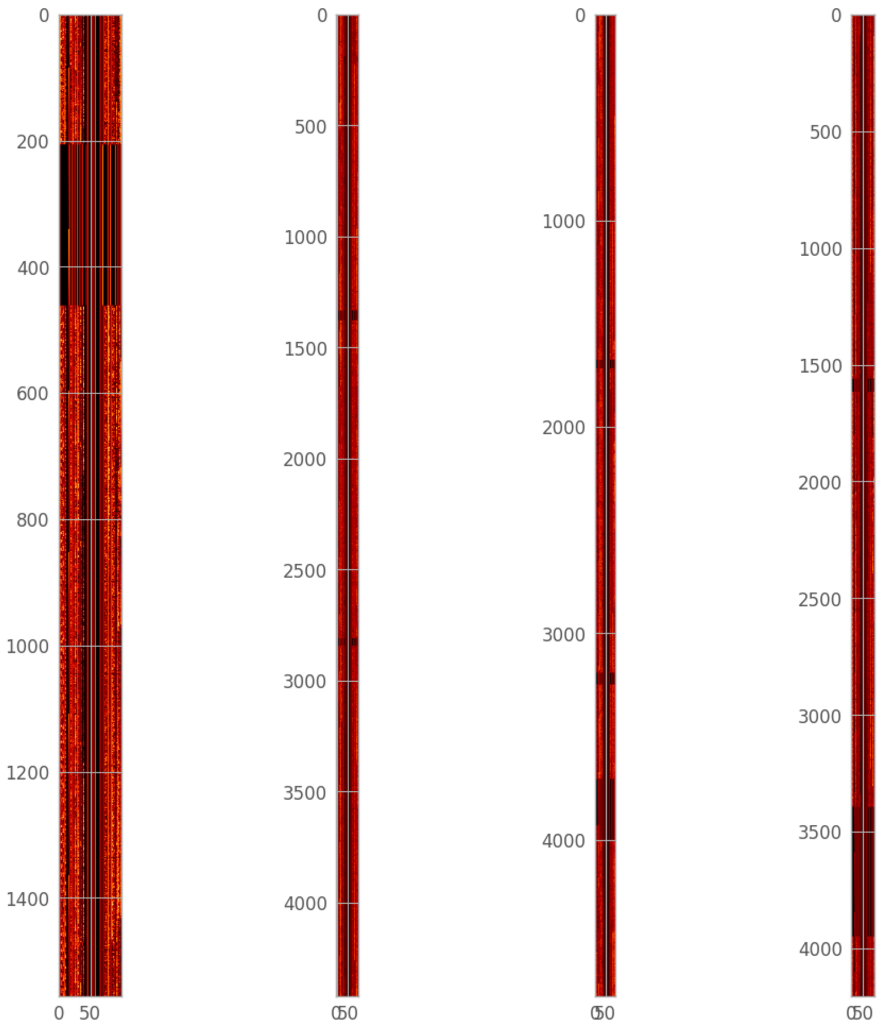
```
_, axs = plt.subplots(3, 5, figsize=(12, 12))
axs = axs.flatten()
for img, ax in zip(exoTrain, axs):
    image = Image.open(img)
    ax.imshow(image)
plt.show()
```



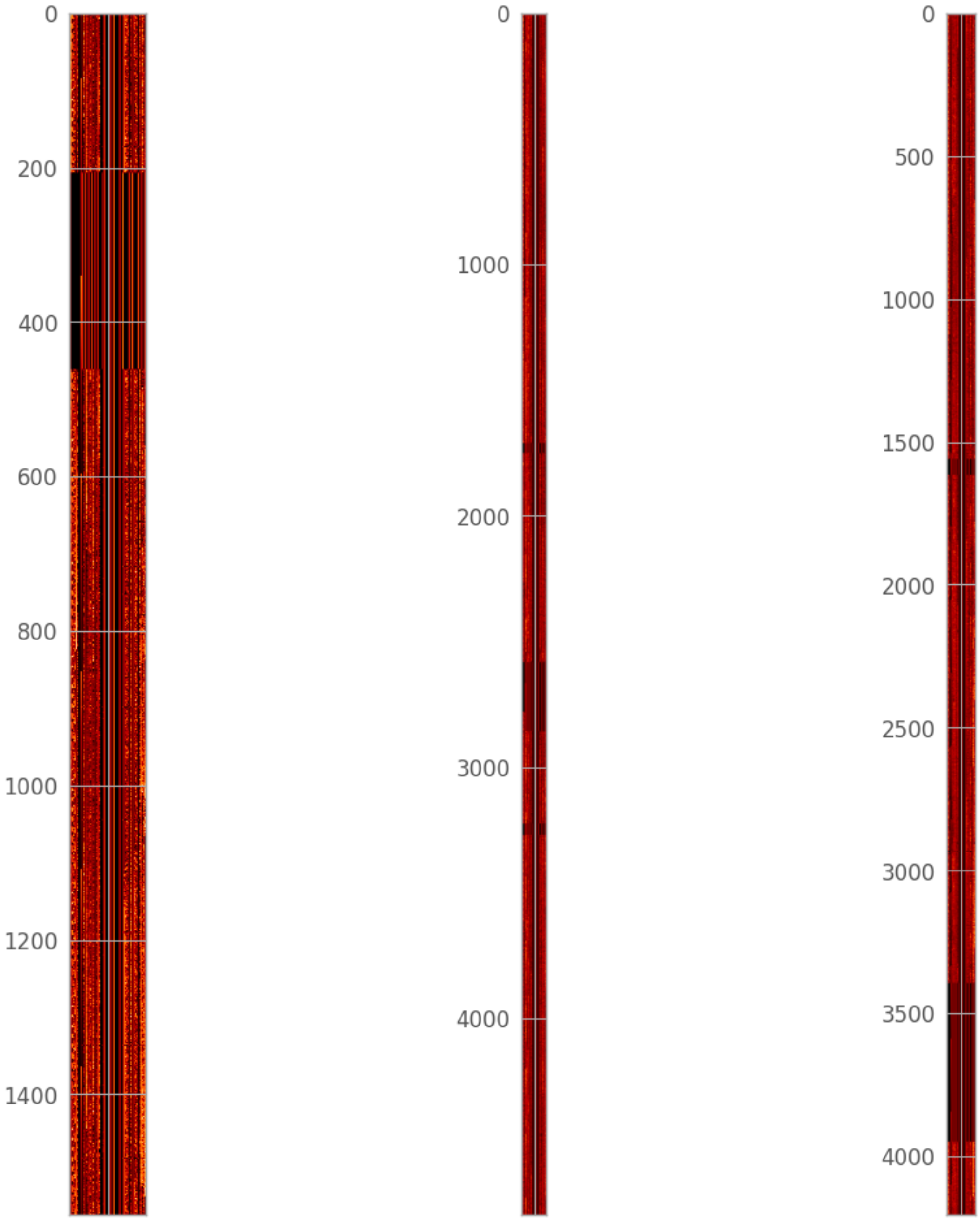
```
_, axs = plt.subplots(3, 5, figsize=(12, 12))
axs = axs.flatten()
for img, ax in zip(fpTrain, axs):
    image = Image.open(img)
    ax.imshow(image)
plt.show()
```




```
_, axs = plt.subplots(1, 4, figsize=(12, 12))
axs = axs.flatten()
for img, ax in zip(epTest, axs):
    image = Image.open(img)
    ax.imshow(image)
plt.show()
```




```
_, axs = plt.subplots(1, 3, figsize=(12, 12))
axs = axs.flatten()
for img, ax in zip(fpTest, axs):
    image = Image.open(img)
    ax.imshow(image)
plt.show()
```




```
confirmedExoFits = glob.glob("/content/drive/MyDrive/exoplanet/rawdata/confirmedExoplanets/010874614/*.fits")
print ("Total of %d  images." % len(confirmedExoFits))
print ('the filenames:\n')
print("\n".join(confirmedExoFits[:5]))
```

```
Total of 15  images.
the filenames:

/content/drive/MyDrive/exoplanet/rawdata/confirmedExoplanets/010874614/kplr010874614-2013131215648_llc.fits
/content/drive/MyDrive/exoplanet/rawdata/confirmedExoplanets/010874614/kplr010874614-2013098041711_llc.fits
/content/drive/MyDrive/exoplanet/rawdata/confirmedExoplanets/010874614/kplr010874614-2012277125453_llc.fits
/content/drive/MyDrive/exoplanet/rawdata/confirmedExoplanets/010874614/kplr010874614-2010078095331_llc.fits
/content/drive/MyDrive/exoplanet/rawdata/confirmedExoplanets/010874614/kplr010874614-2010174085026_llc.fits
```

```
fpExoFits = glob.glob("/content/drive/MyDrive/exoplanet/rawdata/falsePositiveExoplanets/000892772/*.fits")
print ("Total of %d  images." % len(fpExoFits))
print ('the filenames:\n')
print("\n".join(fpExoFits[:5]))
```

```
Total of 14  images.
the filenames:

/content/drive/MyDrive/exoplanet/rawdata/falsePositiveExoplanets/000892772/kplr000892772-2013131215648_llc.fits
/content/drive/MyDrive/exoplanet/rawdata/falsePositiveExoplanets/000892772/kplr000892772-2012277125453_llc.fits
/content/drive/MyDrive/exoplanet/rawdata/falsePositiveExoplanets/000892772/kplr000892772-2013011073258_llc.fits
/content/drive/MyDrive/exoplanet/rawdata/falsePositiveExoplanets/000892772/kplr000892772-2013098041711_llc.fits
/content/drive/MyDrive/exoplanet/rawdata/falsePositiveExoplanets/000892772/kplr000892772-2011271113734_llc.fits
```

```
for file in confirmedExoFits:
    hdul = fits.open(file)
    hdul.info()
```

```
  0  PRIMARY       1 PrimaryHDU      58   ()
  1  LIGHTCURVE    1 BinTableHDU    161   4397R x 20C   [D, E, J, E, E, E, E, E, E, J, D, E, D, E, D, E, D, E, E, E]
  2  APERTURE      1 ImageHDU        48   (8, 8)   int32
Filename: /content/drive/MyDrive/exoplanet/rawdata/confirmedExoplanets/010874614/kplr010874614-2010174085026_llc.fits
No.    Name      Ver    Type      Cards   Dimensions   Format
  0  PRIMARY       1 PrimaryHDU      58   ()
  1  LIGHTCURVE    1 BinTableHDU    161   4634R x 20C   [D, E, J, E, E, E, E, E, E, J, D, E, D, E, D, E, D, E, E, E]
  2  APERTURE      1 ImageHDU        48   (8, 8)   int32
Filename: /content/drive/MyDrive/exoplanet/rawdata/confirmedExoplanets/010874614/kplr010874614-2010265121752_llc.fits
No.    Name      Ver    Type      Cards   Dimensions   Format
  0  PRIMARY       1 PrimaryHDU      58   ()
  1  LIGHTCURVE    1 BinTableHDU    161   4398R x 20C   [D, E, J, E, E, E, E, E, E, J, D, E, D, E, D, E, D, E, E, E]
  2  APERTURE      1 ImageHDU        48   (9, 7)   int32
Filename: /content/drive/MyDrive/exoplanet/rawdata/confirmedExoplanets/010874614/kplr010874614-2012088054726_llc.fits
No.    Name      Ver    Type      Cards   Dimensions   Format
  0  PRIMARY       1 PrimaryHDU      58   ()
  1  LIGHTCURVE    1 BinTableHDU    161   4044R x 20C   [D, E, J, E, E, E, E, E, E, J, D, E, D, E, D, E, D, E, E, E]
  2  APERTURE      1 ImageHDU        48   (8, 8)   int32
Filename: /content/drive/MyDrive/exoplanet/rawdata/confirmedExoplanets/010874614/kplr010874614-2011271113734_llc.fits
No.    Name      Ver    Type      Cards   Dimensions   Format
  0  PRIMARY       1 PrimaryHDU      58   ()
  1  LIGHTCURVE    1 BinTableHDU    161   4573R x 20C   [D, E, J, E, E, E, E, E, E, J, D, E, D, E, D, E, D, E, E, E]
  2  APERTURE      1 ImageHDU        48   (9, 7)   int32
Filename: /content/drive/MyDrive/exoplanet/rawdata/confirmedExoplanets/010874614/kplr010874614-2012179063303_llc.fits
No.    Name      Ver    Type      Cards   Dimensions   Format
  0  PRIMARY       1 PrimaryHDU      58   ()
  1  LIGHTCURVE    1 BinTableHDU    161   4421R x 20C   [D, E, J, E, E, E, E, E, E, J, D, E, D, E, D, E, D, E, E, E]
  2  APERTURE      1 ImageHDU        48   (8, 8)   int32
Filename: /content/drive/MyDrive/exoplanet/rawdata/confirmedExoplanets/010874614/kplr010874614-2009259160929_llc.fits
No.    Name      Ver    Type      Cards   Dimensions   Format
  0  PRIMARY       1 PrimaryHDU      58   ()
  1  LIGHTCURVE    1 BinTableHDU    161   4354R x 20C   [D, E, J, E, E, E, E, E, E, J, D, E, D, E, D, E, D, E, E, E]
  2  APERTURE      1 ImageHDU        48   (9, 7)   int32
Filename: /content/drive/MyDrive/exoplanet/rawdata/confirmedExoplanets/010874614/kplr010874614-2009350155506_llc.fits
No.    Name      Ver    Type      Cards   Dimensions   Format
  0  PRIMARY       1 PrimaryHDU      58   ()
  1  LIGHTCURVE    1 BinTableHDU    161   4370R x 20C   [D, E, J, E, E, E, E, E, E, J, D, E, D, E, D, E, D, E, E, E]
  2  APERTURE      1 ImageHDU        48   (9, 8)   int32
Filename: /content/drive/MyDrive/exoplanet/rawdata/confirmedExoplanets/010874614/kplr010874614-2011177032512_llc.fits
No.    Name      Ver    Type      Cards   Dimensions   Format
  0  PRIMARY       1 PrimaryHDU      58   ()
  1  LIGHTCURVE    1 BinTableHDU    161   4768R x 20C   [D, E, J, E, E, E, E, E, E, J, D, E, D, E, D, E, D, E, E, E]
  2  APERTURE      1 ImageHDU        48   (8, 8)   int32
Filename: /content/drive/MyDrive/exoplanet/rawdata/confirmedExoplanets/010874614/kplr010874614-2011073133259_llc.fits
No.    Name      Ver    Type      Cards   Dimensions   Format
  0  PRIMARY       1 PrimaryHDU      58   ()
  1  LIGHTCURVE    1 BinTableHDU    161   3279R x 20C   [D, E, J, E, E, E, E, E, E, J, D, E, D, E, D, E, D, E, E, E]
  2  APERTURE      1 ImageHDU        48   (8, 8)   int32
Filename: /content/drive/MyDrive/exoplanet/rawdata/confirmedExoplanets/010874614/kplr010874614-2009166043257_llc.fits
No.    Name      Ver    Type      Cards   Dimensions   Format
  0  PRIMARY       1 PrimaryHDU      58   ()
  1  LIGHTCURVE    1 BinTableHDU    155   1639R x 20C   [D, E, J, E, E, E, E, E, E, J, D, E, D, E, D, E, D, E, E, E]
  2  APERTURE      1 ImageHDU        48   (8, 8)   int32
Filename: /content/drive/MyDrive/exoplanet/rawdata/confirmedExoplanets/010874614/kplr010874614-2009131105131_llc.fits
No.    Name      Ver    Type      Cards   Dimensions   Format
  0  PRIMARY       1 PrimaryHDU      58   ()
  1  LIGHTCURVE    1 BinTableHDU    155   476R x 20C   [D, E, J, E, E, E, E, E, E, J, D, E, D, E, D, E, D, E, E, E]
  2  APERTURE      1 ImageHDU        48   (8, 9)   int32
```

```
for file in fpExoFits:
    hdul = fits.open(file)
    hdul.info()
```

```
  0  PRIMARY       1 PrimaryHDU      58   ()
  1  LIGHTCURVE    1 BinTableHDU    161   4780R x 20C   [D, E, J, E, E, E, E, E, E, J, D, E, D, E, D, E, D, E, E, E]
```

```
No.    Name      Ver    Type      Cards   Dimensions   Format
  0  PRIMARY       1 PrimaryHDU      58   ()
  1  LIGHTCURVE    1 BinTableHDU    161   4375R x 20C   [D, E, J, E, E, E, E, E, E, J, D, E, D, E, D, E, D, E, E, E]
  2  APERTURE      1 ImageHDU        48   (6, 4)   int32
Filename: /content/drive/MyDrive/exoplanet/rawdata/falsePositiveExoplanets/000892772/kplr000892772-2011073133259_llc.fits
No.    Name      Ver    Type      Cards   Dimensions   Format
  0  PRIMARY       1 PrimaryHDU      58   ()
  1  LIGHTCURVE    1 BinTableHDU    161   3279R x 20C   [D, E, J, E, E, E, E, E, E, J, D, E, D, E, D, E, D, E, E, E]
  2  APERTURE      1 ImageHDU        48   (5, 4)   int32
Filename: /content/drive/MyDrive/exoplanet/rawdata/falsePositiveExoplanets/000892772/kplr000892772-2012179063303_llc.fits
No.    Name      Ver    Type      Cards   Dimensions   Format
  0  PRIMARY       1 PrimaryHDU      58   ()
  1  LIGHTCURVE    1 BinTableHDU    161   4421R x 20C   [D, E, J, E, E, E, E, E, E, J, D, E, D, E, D, E, D, E, E, E]
  2  APERTURE      1 ImageHDU        48   (5, 4)   int32
Filename: /content/drive/MyDrive/exoplanet/rawdata/falsePositiveExoplanets/000892772/kplr000892772-2012088054726_llc.fits
No.    Name      Ver    Type      Cards   Dimensions   Format
  0  PRIMARY       1 PrimaryHDU      58   ()
  1  LIGHTCURVE    1 BinTableHDU    161   4044R x 20C   [D, E, J, E, E, E, E, E, E, J, D, E, D, E, D, E, D, E, E, E]
  2  APERTURE      1 ImageHDU        48   (5, 4)   int32
Filename: /content/drive/MyDrive/exoplanet/rawdata/falsePositiveExoplanets/000892772/kplr000892772-2011177032512_llc.fits
No.    Name      Ver    Type      Cards   Dimensions   Format
  0  PRIMARY       1 PrimaryHDU      58   ()
  1  LIGHTCURVE    1 BinTableHDU    161   4768R x 20C   [D, E, J, E, E, E, E, E, E, J, D, E, D, E, D, E, D, E, E, E]
  2  APERTURE      1 ImageHDU        48   (5, 4)   int32
Filename: /content/drive/MyDrive/exoplanet/rawdata/falsePositiveExoplanets/000892772/kplr000892772-2010265121752_llc.fits
No.    Name      Ver    Type      Cards   Dimensions   Format
  0  PRIMARY       1 PrimaryHDU      58   ()
  1  LIGHTCURVE    1 BinTableHDU    161   4398R x 20C   [D, E, J, E, E, E, E, E, E, J, D, E, D, E, D, E, D, E, E, E]
  2  APERTURE      1 ImageHDU        48   (5, 4)   int32
Filename: /content/drive/MyDrive/exoplanet/rawdata/falsePositiveExoplanets/000892772/kplr000892772-2010078095331_llc.fits
No.    Name      Ver    Type      Cards   Dimensions   Format
  0  PRIMARY       1 PrimaryHDU      58   ()
  1  LIGHTCURVE    1 BinTableHDU    161   4397R x 20C   [D, E, J, E, E, E, E, E, E, J, D, E, D, E, D, E, D, E, E, E]
  2  APERTURE      1 ImageHDU        48   (5, 4)   int32
Filename: /content/drive/MyDrive/exoplanet/rawdata/falsePositiveExoplanets/000892772/kplr000892772-2012004120508_llc.fits
No.    Name      Ver    Type      Cards   Dimensions   Format
  0  PRIMARY       1 PrimaryHDU      58   ()
  1  LIGHTCURVE    1 BinTableHDU    161   4754R x 20C   [D, E, J, E, E, E, E, E, E, J, D, E, D, E, D, E, D, E, E, E]
  2  APERTURE      1 ImageHDU        48   (6, 4)   int32
Filename: /content/drive/MyDrive/exoplanet/rawdata/falsePositiveExoplanets/000892772/kplr000892772-2010174085026_llc.fits
No.    Name      Ver    Type      Cards   Dimensions   Format
  0  PRIMARY       1 PrimaryHDU      58   ()
  1  LIGHTCURVE    1 BinTableHDU    161   4634R x 20C   [D, E, J, E, E, E, E, E, E, J, D, E, D, E, D, E, D, E, E, E]
  2  APERTURE      1 ImageHDU        48   (5, 4)   int32
```

## Data Augmentation

```
gen = ImageDataGenerator(rescale=1./255)
train = gen.flow_from_directory(directory=TRAIN_DIR, target_size=(224,224))
```

```
Found 29 images belonging to 2 classes.
```

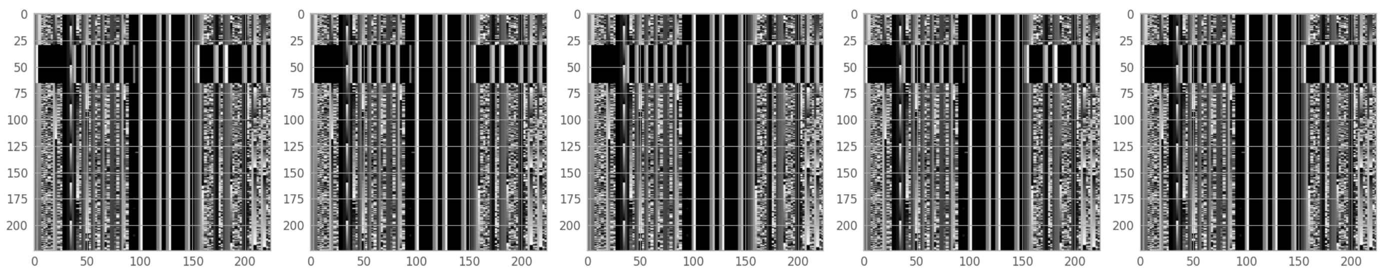
```
# This function will plot images in the form of a grid with 1 row and 5 columns where images are placed in each column.
def plotImages(images_arr):
    fig, axes = plt.subplots(1, 5, figsize=(20,20))
    axes = axes.flatten()
    for img, ax in zip( images_arr, axes):
        ax.imshow(img)
    plt.tight_layout()
    plt.show()


augmented_images = [train[0][0][0] for i in range(5)]
plotImages(augmented_images)
```

```
gen = ImageDataGenerator(rescale=1./255)
test = gen.flow_from_directory(directory=TEST_DIR, target_size=(224,224))
```

```
Found 7 images belonging to 2 classes.
```

```
# This function will plot images in the form of a grid with 1 row and 5 columns where images are placed in each column.
def plotImages(images_arr):
    fig, axes = plt.subplots(1, 5, figsize=(20,20))
    axes = axes.flatten()
    for img, ax in zip( images_arr, axes):
        ax.imshow(img)
    plt.tight_layout()
    plt.show()
```

```
augmented_images = [test[0][0][0] for i in range(5)]
plotImages(augmented_images)
```



```
gen = ImageDataGenerator()
train = gen.flow_from_directory(directory=TRAIN_DIR, target_size=(224,224))
test = gen.flow_from_directory(directory=TEST_DIR, target_size=(224,224))
```

```
Found 29 images belonging to 2 classes.
Found 7 images belonging to 2 classes.
```

```
modelScores=pd.DataFrame()
```

## CNN

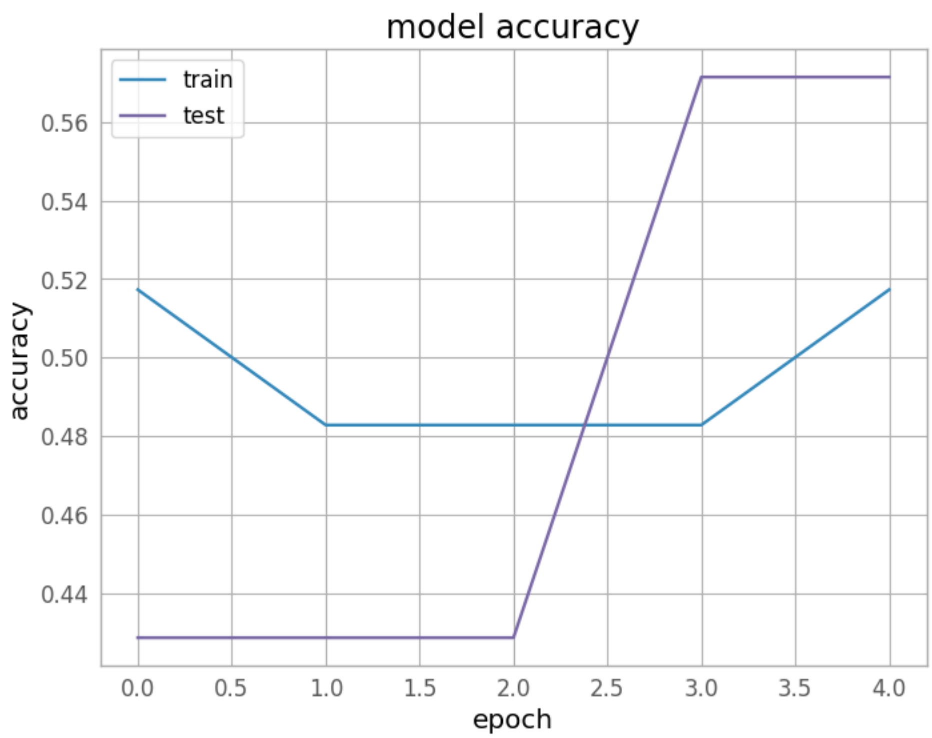
```
cnnmodel = Sequential()
cnnmodel.add(Conv2D(input_shape = (224, 224, 3), filters = 64, kernel_size = (3,3), padding = "same", activation = "relu"))
cnnmodel.add(Conv2D(filters = 64, kernel_size = (3,3), padding = "same", activation = "relu"))
cnnmodel.add(MaxPool2D(pool_size = (2,2), strides = (2,2)))
cnnmodel.add(Flatten())
cnnmodel.add(Dense(units = 2, activation = "sigmoid"))
opt = Adam(learning_rate = 0.001)
cnnmodel.compile(optimizer = opt, loss= keras.losses.categorical_crossentropy,   metrics = ['accuracy'])
cnnmodel.summary()
```

```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #   
=================================================================
 conv2d_2 (Conv2D)           (None, 224, 224, 64)      1792      
                                                                 
 conv2d_3 (Conv2D)           (None, 224, 224, 64)      36928     
                                                                 
 max_pooling2d_1 (MaxPooling  (None, 112, 112, 64)     0         
 2D)                                                             
                                                                 
 flatten_1 (Flatten)         (None, 802816)            0         
                                                                 
 dense_1 (Dense)             (None, 2)                 1605634   
                                                                 
=================================================================
Total params: 1,644,354
Trainable params: 1,644,354
Non-trainable params: 0
_________________________________________________________________
```

```
history = cnnmodel.fit(train, validation_data= test, epochs=5)
```

```
Epoch 1/5
1/1 [==============================] - 15s 15s/step - loss: 22.2313 - accuracy: 0.5172 - val_loss: 14218.1934 - val_accuracy: 0.4286
Epoch 2/5
1/1 [==============================] - 13s 13s/step - loss: 13002.6494 - accuracy: 0.4828 - val_loss: 8641.7715 - val_accuracy: 0.4286
Epoch 3/5
1/1 [==============================] - 13s 13s/step - loss: 7887.5063 - accuracy: 0.4828 - val_loss: 2775.3220 - val_accuracy: 0.4286
Epoch 4/5
1/1 [==============================] - 12s 12s/step - loss: 2517.2876 - accuracy: 0.4828 - val_loss: 191.4025 - val_accuracy: 0.5714
Epoch 5/5
1/1 [==============================] - 13s 13s/step - loss: 223.7636 - accuracy: 0.5172 - val_loss: 684.1805 - val_accuracy: 0.5714
```

```
# summarize history for accuracy
plt.plot(history.history['accuracy'])
plt.plot(history.history['val_accuracy'])
plt.title('model accuracy')
plt.ylabel('accuracy')
plt.xlabel('epoch')
plt.legend(['train', 'test'], loc='upper left')
plt.show()
```




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


```
loss, accuracy = cnnmodel.evaluate(test, verbose=2)
print("Loss = ", loss)
accuracy= accuracy*100
print("Accuracy = ", accuracy)
```

```
1/1 - 1s - loss: 684.1805 - accuracy: 0.5714 - 743ms/epoch - 743ms/step
Loss =  684.1805419921875
Accuracy =  57.14285969734192
```

```
modelScores=modelScores.append([['CNN Image', accuracy]])
modelScores
```

| | 0 | 1 |
|---|---|---|
| 0 | CNN Image | 57.14286 |

## VGG16

```
model = VGG16(include_top=False,  weights='imagenet',  input_shape=(224, 224, 3))
print(model.summary())
```

```
Downloading data from https://storage.googleapis.com/tensorflow/keras-applications/vgg16/vgg16_weights_tf_dim_ordering_tf_kernels_notop
58889256/58889256 [==============================] - 2s 0us/step
Model: "vgg16"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input_1 (InputLayer)        [(None, 224, 224, 3)]     0

 block1_conv1 (Conv2D)       (None, 224, 224, 64)      1792

 block1_conv2 (Conv2D)       (None, 224, 224, 64)      36928

 block1_pool (MaxPooling2D)  (None, 112, 112, 64)      0

 block2_conv1 (Conv2D)       (None, 112, 112, 128)     73856

 block2_conv2 (Conv2D)       (None, 112, 112, 128)     147584

 block2_pool (MaxPooling2D)  (None, 56, 56, 128)       0

 block3_conv1 (Conv2D)       (None, 56, 56, 256)       295168

 block3_conv2 (Conv2D)       (None, 56, 56, 256)       590080

 block3_conv3 (Conv2D)       (None, 56, 56, 256)       590080

 block3_pool (MaxPooling2D)  (None, 28, 28, 256)       0

 block4_conv1 (Conv2D)       (None, 28, 28, 512)       1180160

 block4_conv2 (Conv2D)       (None, 28, 28, 512)       2359808

 block4_conv3 (Conv2D)       (None, 28, 28, 512)       2359808

 block4_pool (MaxPooling2D)  (None, 14, 14, 512)       0

 block5_conv1 (Conv2D)       (None, 14, 14, 512)       2359808
```

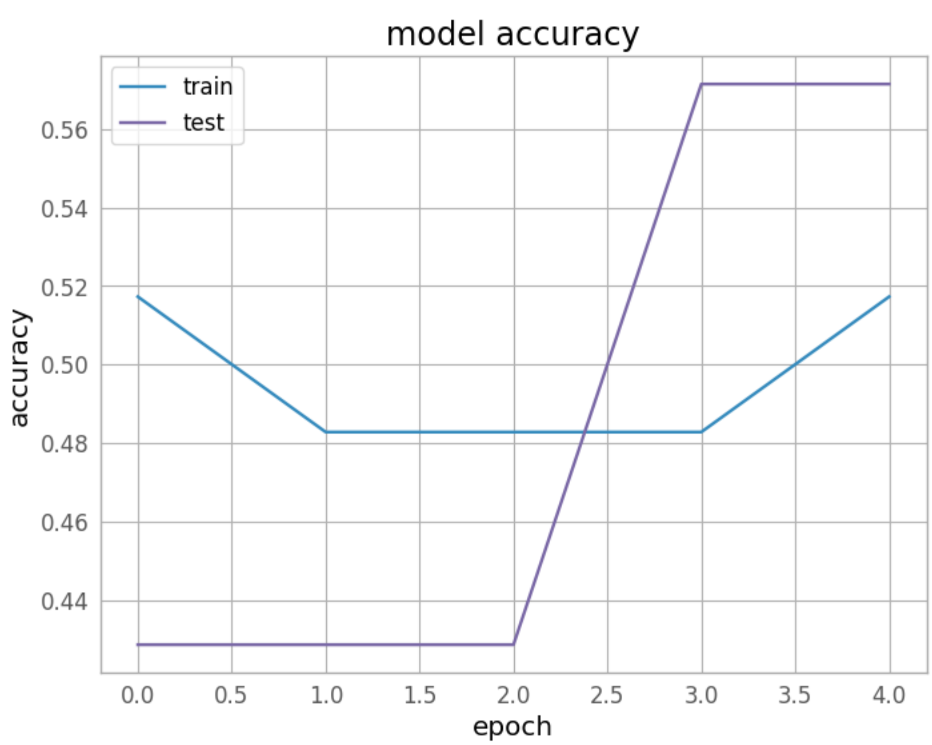
```
 block5_conv2 (Conv2D)       (None, 14, 14, 512)       2359808

 block5_conv3 (Conv2D)       (None, 14, 14, 512)       2359808

 block5_pool (MaxPooling2D)  (None, 7, 7, 512)         0

=================================================================
Total params: 14,714,688
Trainable params: 14,714,688
Non-trainable params: 0
_________________________________________________________________
None
```

```
vgg16 = model.output
vgg16 = Flatten()(vgg16)
vgg16 = Dense(4096, activation='relu')(vgg16)
vgg16 = Dense(1072, activation='relu')(vgg16)
vgg16 = Dropout(0.2)(vgg16)
output_layer = Dense(2, activation='softmax')(vgg16)
model = Model(inputs=model.input, outputs=output_layer)
opt = Adam(learning_rate = 0.001)

model.compile(optimizer = opt, loss= keras.losses.categorical_crossentropy,   metrics = ['accuracy'])
```

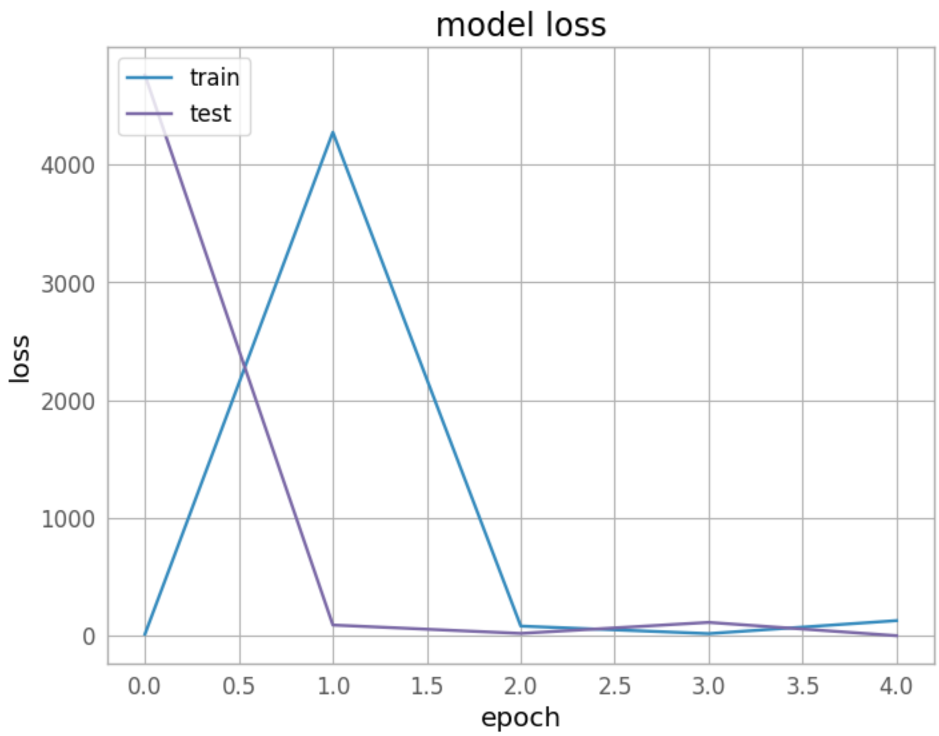
```
history = model.fit(train, validation_data= test, epochs=5)
```

```
Epoch 1/5
1/1 [==============================] - 10s 10s/step - loss: 13.8509 - accuracy: 0.5172 - val_loss: 4759.2471 - val_accuracy: 0.4286
Epoch 2/5
1/1 [==============================] - 7s 7s/step - loss: 4274.1831 - accuracy: 0.4828 - val_loss: 91.2506 - val_accuracy: 0.4286
Epoch 3/5
1/1 [==============================] - 7s 7s/step - loss: 82.3660 - accuracy: 0.4828 - val_loss: 20.1702 - val_accuracy: 0.4286
Epoch 4/5
1/1 [==============================] - 7s 7s/step - loss: 18.4339 - accuracy: 0.4828 - val_loss: 113.7924 - val_accuracy: 0.5714
Epoch 5/5
1/1 [==============================] - 7s 7s/step - loss: 128.8833 - accuracy: 0.5172 - val_loss: 0.8257 - val_accuracy: 0.5714
```

```
# summarize history for accuracy
plt.plot(history.history['accuracy'])
plt.plot(history.history['val_accuracy'])
plt.title('model accuracy')
plt.ylabel('accuracy')
plt.xlabel('epoch')
plt.legend(['train', 'test'], loc='upper left')
plt.show()
```



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




```
loss, accuracy = model.evaluate(test, verbose=2)
print("Loss = ", loss)
accuracy= accuracy*100
print("Accuracy = ", accuracy)
```

```
1/1 - 0s - loss: 0.8257 - accuracy: 0.5714 - 337ms/epoch - 337ms/step
Loss =  0.8256927132606506
Accuracy =  57.14285969734192
```

```
modelScores=modelScores.append([['VGG16 Image', accuracy]])
modelScores
```

| | 0 | 1 |
|---|---|---|
| 0 | CNN Image | 85.714287 |
| 0 | VGG16 Image | 57.142860 |

## VGG19

```
model = VGG19(include_top=False,  weights='imagenet',  input_shape=(224, 224, 3))
print(model.summary())
```

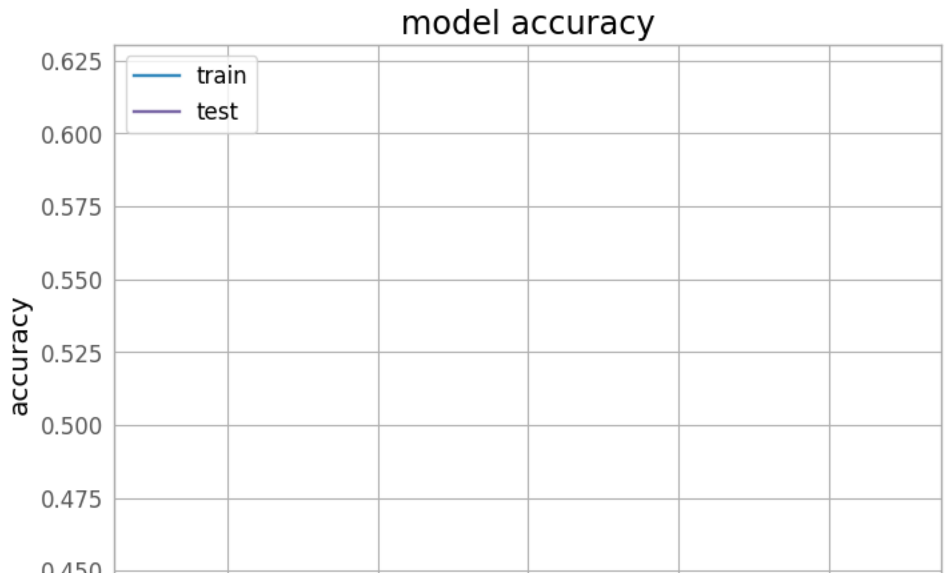
```
Model: "vgg19"
_________________________________________________________________
 Layer (type)                Output Shape              Param #   
=================================================================
 input_2 (InputLayer)        [(None, 224, 224, 3)]     0         
                                                                 
 block1_conv1 (Conv2D)       (None, 224, 224, 64)      1792      
                                                                 
 block1_conv2 (Conv2D)       (None, 224, 224, 64)      36928     
```

```
 block1_pool (MaxPooling2D)  (None, 112, 112, 64)      0

 block2_conv1 (Conv2D)       (None, 112, 112, 128)     73856

 block2_conv2 (Conv2D)       (None, 112, 112, 128)     147584

 block2_pool (MaxPooling2D)  (None, 56, 56, 128)       0

 block3_conv1 (Conv2D)       (None, 56, 56, 256)       295168

 block3_conv2 (Conv2D)       (None, 56, 56, 256)       590080

 block3_conv3 (Conv2D)       (None, 56, 56, 256)       590080

 block3_conv4 (Conv2D)       (None, 56, 56, 256)       590080

 block3_pool (MaxPooling2D)  (None, 28, 28, 256)       0

 block4_conv1 (Conv2D)       (None, 28, 28, 512)       1180160

 block4_conv2 (Conv2D)       (None, 28, 28, 512)       2359808

 block4_conv3 (Conv2D)       (None, 28, 28, 512)       2359808

 block4_conv4 (Conv2D)       (None, 28, 28, 512)       2359808

 block4_pool (MaxPooling2D)  (None, 14, 14, 512)       0

 block5_conv1 (Conv2D)       (None, 14, 14, 512)       2359808

 block5_conv2 (Conv2D)       (None, 14, 14, 512)       2359808

 block5_conv3 (Conv2D)       (None, 14, 14, 512)       2359808

 block5_conv4 (Conv2D)       (None, 14, 14, 512)       2359808

 block5_pool (MaxPooling2D)  (None, 7, 7, 512)         0

=================================================================
Total params: 20,024,384
Trainable params: 20,024,384
Non-trainable params: 0
_________________________________________________________________
None
```

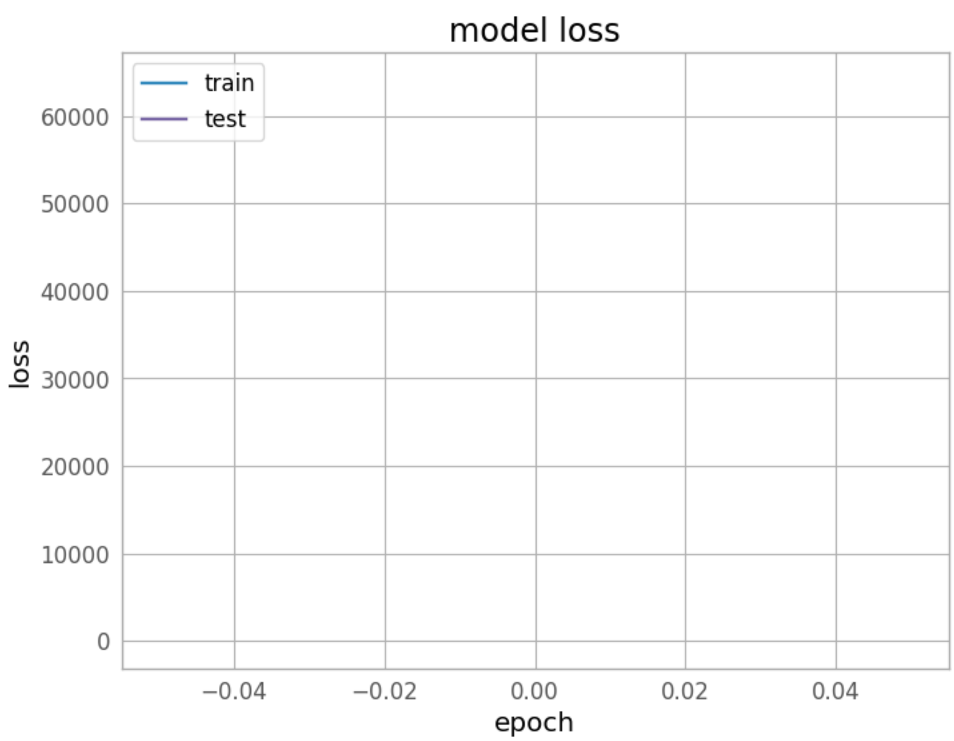
```
vgg19 = model.output
vgg19 = Flatten()(vgg19)
vgg19 = Dense(1032, activation='relu')(vgg19)
vgg19 = Dropout(0.2)(vgg19)
output_layer = Dense(2, activation='softmax')(vgg19)
model = Model(inputs=model.input, outputs=output_layer)
opt = Adam(learning_rate = 0.001)

model.compile(optimizer = opt, loss= keras.losses.categorical_crossentropy,   metrics = ['accuracy'])


history = model.fit(train,steps_per_epoch=10, validation_data= test, validation_steps=5, epochs=5)
```

```
Epoch 1/5
 1/10 [==>...........................] - ETA: 1:32 - loss: 6.6169 - accuracy: 0.6207WARNING:tensorflow:Your input ran out of data; inte
WARNING:tensorflow:Your input ran out of data; interrupting training. Make sure that your dataset or generator can generate at least `s
10/10 [==============================] - 11s 89ms/step - loss: 6.6169 - accuracy: 0.6207 - val_loss: 64033.2930 - val_accuracy: 0.4286
```

```
# summarize history for accuracy
plt.plot(history.history['accuracy'])
plt.plot(history.history['val_accuracy'])
plt.title('model accuracy')
plt.ylabel('accuracy')
plt.xlabel('epoch')
plt.legend(['train', 'test'], loc='upper left')
plt.show()
```




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




```
loss, accuracy = model.evaluate(test, verbose=2)
print("Loss = ", loss)
accuracy= accuracy*100
print("Accuracy = ", accuracy)
```

```
1/1 - 0s - loss: 64033.2891 - accuracy: 0.4286 - 387ms/epoch - 387ms/step
Loss =  64033.2890625
Accuracy =  42.85714328289032
```

```
modelScores=modelScores.append([['VGG19 Image', accuracy]])
modelScores
```

| | 0 | 1 |
|---|---|---|
| 0 | CNN Image | 85.714287 |
| 0 | VGG16 Image | 57.142860 |
| 0 | VGG19 Image | 42.857143 |

## SVM

```
svm = SVC(kernel='rbf', C=0.1, gamma=5)
```

```
svm.fit(X_train, y_train)
```

```
SVC
SVC(C=0.1, gamma=5)
```

```
pred = svm.predict(X_test)
```

```
accuracy= np.round(accuracy_score(y_test, pred)*100,0)
print("Accuracy = ", accuracy)
```

```
Accuracy =  57.0
```

```
modelScores=modelScores.append([['SVM Image', accuracy]])
modelScores
```

| | 0 | 1 |
|---|---|---|
| 0 | CNN Image | 85.714287 |
| 0 | VGG16 Image | 57.142860 |
| 0 | VGG19 Image | 42.857143 |
| 0 | SVM Image | 57.000000 |

## KNN

```
knn = KNeighborsClassifier(n_neighbors=2)
```

```
knn.fit(X_train, y_train)
```

```
KNeighborsClassifier
KNeighborsClassifier(n_neighbors=2)
```

```
pred = knn.predict(X_test)
```

```
accuracy= np.round(accuracy_score(y_test, pred)*100,0)
print("Accuracy = ", accuracy)
```

```
Accuracy =  71.0
```

```
modelScores=modelScores.append([['KNN Image', accuracy]])
modelScores
```

| | 0 | 1 |
|---|---|---|
| 0 | CNN Image | 85.714287 |
| 0 | VGG16 Image | 57.142860 |
| 0 | VGG19 Image | 42.857143 |
| 0 | SVM Image | 57.000000 |
| 0 | KNN Image | 71.000000 |

# Decision Trees

```
dt = DecisionTreeClassifier(criterion='entropy', min_samples_split=100)
```

```
dt.fit(X_train, y_train)
```

```
DecisionTreeClassifier
DecisionTreeClassifier(criterion='entropy', min_samples_split=100)
```

```
pred = dt.predict(X_test)
```

```
accuracy= np.round(accuracy_score(y_test, pred)*100,0)
print("Accuracy = ", accuracy)
```

```
Accuracy =  57.0
```

```
modelScores=modelScores.append([['Decision Trees Image', accuracy]])
modelScores
```

| | 0 | 1 |
|---|---|---|
| **0** | CNN Image | 85.714287 |
| **0** | VGG16 Image | 57.142860 |
| **0** | VGG19 Image | 42.857143 |
| **0** | SVM Image | 57.000000 |
| **0** | KNN Image | 71.000000 |
| **0** | Decision Trees Image | 57.000000 |

# Kepler Data

```
X_train, X_test, y_train, y_test = train_test_split(keplerX, keplerY, test_size=0.3, random_state=42)
X_train.shape, X_test.shape, y_train.shape, y_test.shape
```

```
((8660, 24), (3712, 24), (8660,), (3712,))
```

# **CNN**

```
xtrain = X_train.reshape((X_train.shape[0],X_train.shape[1],1))
xtest = X_test.reshape((X_test.shape[0],X_test.shape[1],1))
y_train = np.array(y_train)
y_test = np.array(y_test)
xtrain.shape, xtest.shape, y_train.shape, y_test.shape
```

```
((8660, 24, 1), (3712, 24, 1), (8660,), (3712,))
```

```
cnnmodel = Sequential()
cnnmodel.add(Conv1D(input_shape = (xtrain.shape[1],1), filters = 256, kernel_size = 3, padding = "same", activation = "tanh"))
cnnmodel.add(Flatten())
cnnmodel.add(Dense(units = 256, activation = "tanh"))
cnnmodel.add(Dense(units = 128, activation = "tanh"))
cnnmodel.add(Dense(units = 4, activation = "sigmoid"))
cnnmodel.compile(optimizer = 'adam', loss= keras.losses.sparse_categorical_crossentropy, metrics = ['accuracy'])
cnnmodel.summary()
```

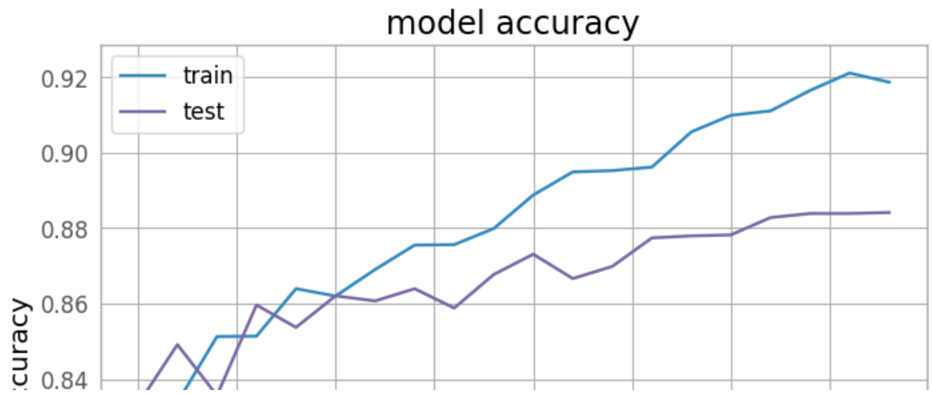
```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #   
=================================================================
 conv1d (Conv1D)             (None, 24, 256)           1024      
```

```
 flatten_3 (Flatten)         (None, 6144)              0

 dense_6 (Dense)             (None, 256)               1573120

 dense_7 (Dense)             (None, 128)               32896

 dense_8 (Dense)             (None, 4)                 516

=================================================================
Total params: 1,607,556
Trainable params: 1,607,556
Non-trainable params: 0
_________________________________________________________________
```

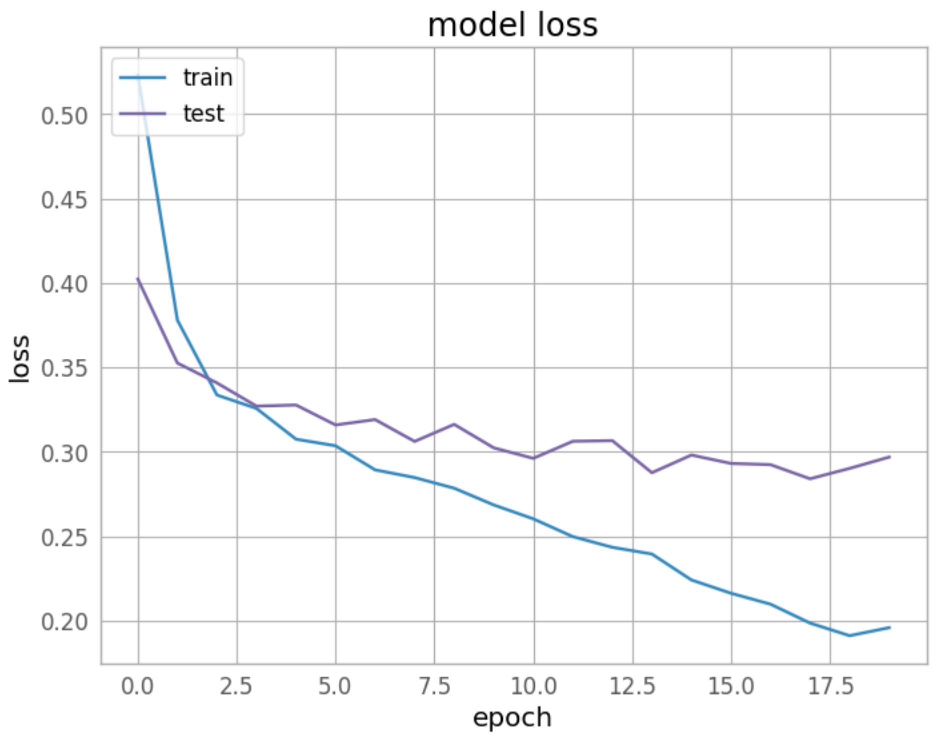
```
history = cnnmodel.fit(xtrain, y_train, validation_data= (xtest, y_test), epochs=20)
```

```
Epoch 1/20
271/271 [==============================] - 3s 9ms/step - loss: 0.5231 - accuracy: 0.7726 - val_loss: 0.4023 - val_accuracy: 0.8332
Epoch 2/20
271/271 [==============================] - 2s 7ms/step - loss: 0.3780 - accuracy: 0.8338 - val_loss: 0.3526 - val_accuracy: 0.8491
Epoch 3/20
271/271 [==============================] - 2s 7ms/step - loss: 0.3336 - accuracy: 0.8513 - val_loss: 0.3408 - val_accuracy: 0.8357
Epoch 4/20
271/271 [==============================] - 2s 7ms/step - loss: 0.3257 - accuracy: 0.8514 - val_loss: 0.3271 - val_accuracy: 0.8596
Epoch 5/20
271/271 [==============================] - 2s 7ms/step - loss: 0.3075 - accuracy: 0.8640 - val_loss: 0.3277 - val_accuracy: 0.8537
Epoch 6/20
271/271 [==============================] - 2s 7ms/step - loss: 0.3036 - accuracy: 0.8620 - val_loss: 0.3159 - val_accuracy: 0.8621
Epoch 7/20
271/271 [==============================] - 2s 7ms/step - loss: 0.2893 - accuracy: 0.8691 - val_loss: 0.3191 - val_accuracy: 0.8607
Epoch 8/20
271/271 [==============================] - 2s 7ms/step - loss: 0.2848 - accuracy: 0.8755 - val_loss: 0.3062 - val_accuracy: 0.8640
Epoch 9/20
271/271 [==============================] - 2s 7ms/step - loss: 0.2785 - accuracy: 0.8756 - val_loss: 0.3163 - val_accuracy: 0.8588
Epoch 10/20
271/271 [==============================] - 2s 7ms/step - loss: 0.2685 - accuracy: 0.8799 - val_loss: 0.3024 - val_accuracy: 0.8677
Epoch 11/20
271/271 [==============================] - 2s 7ms/step - loss: 0.2603 - accuracy: 0.8888 - val_loss: 0.2961 - val_accuracy: 0.8731
Epoch 12/20
271/271 [==============================] - 2s 7ms/step - loss: 0.2498 - accuracy: 0.8949 - val_loss: 0.3062 - val_accuracy: 0.8666
Epoch 13/20
271/271 [==============================] - 2s 7ms/step - loss: 0.2435 - accuracy: 0.8953 - val_loss: 0.3066 - val_accuracy: 0.8699
Epoch 14/20
271/271 [==============================] - 2s 7ms/step - loss: 0.2395 - accuracy: 0.8962 - val_loss: 0.2876 - val_accuracy: 0.8774
Epoch 15/20
271/271 [==============================] - 2s 7ms/step - loss: 0.2241 - accuracy: 0.9055 - val_loss: 0.2981 - val_accuracy: 0.8780
Epoch 16/20
271/271 [==============================] - 2s 7ms/step - loss: 0.2163 - accuracy: 0.9099 - val_loss: 0.2931 - val_accuracy: 0.8782
Epoch 17/20
271/271 [==============================] - 2s 7ms/step - loss: 0.2098 - accuracy: 0.9111 - val_loss: 0.2924 - val_accuracy: 0.8828
Epoch 18/20
271/271 [==============================] - 2s 7ms/step - loss: 0.1986 - accuracy: 0.9165 - val_loss: 0.2840 - val_accuracy: 0.8839
Epoch 19/20
271/271 [==============================] - 2s 7ms/step - loss: 0.1911 - accuracy: 0.9211 - val_loss: 0.2902 - val_accuracy: 0.8839
Epoch 20/20
271/271 [==============================] - 2s 7ms/step - loss: 0.1959 - accuracy: 0.9187 - val_loss: 0.2969 - val_accuracy: 0.8842
```

```
# summarize history for accuracy
plt.plot(history.history['accuracy'])
plt.plot(history.history['val_accuracy'])
plt.title('model accuracy')
plt.ylabel('accuracy')
plt.xlabel('epoch')
plt.legend(['train', 'test'], loc='upper left')
plt.show()
```




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




```
loss, accuracy = cnnmodel.evaluate(xtest, y_test, verbose=2)
print("Loss = ", loss)
accuracy= accuracy*100
print("Accuracy = ", accuracy)
```

```
116/116 - 0s - loss: 0.2969 - accuracy: 0.8842 - 241ms/epoch - 2ms/step
Loss =  0.2968772351741791
Accuracy =  88.4159505367279
```

```
modelScores=modelScores.append([['CNN Kepler', accuracy]])
modelScores
```

| | 0 | 1 |
|---|---|---|
| **0** | CNN Image | 85.714287 |
| **0** | VGG16 Image | 57.142860 |
| **0** | CNN Kepler | 88.415951 |

# SVM

```
svm = SVC(kernel='rbf', C=0.1, gamma=5)
```

```
svm.fit(X_train, y_train)
```

```
▾ SVC
SVC(C=0.1, gamma=5)
```

```
pred = svm.predict(X_test)
```

```
accuracy= np.round(accuracy_score(y_test, pred)*100,0)
print("Accuracy = ", accuracy)
```

```
Accuracy =  33.0
```

```
modelScores=modelScores.append([['SVM Kepler', accuracy]])
modelScores
```

| | 0 | 1 |
|---|---|---|
| **0** | CNN Image | 85.714287 |
| **0** | VGG16 Image | 57.142860 |
| **0** | VGG19 Image | 42.857143 |
| **0** | SVM Image | 57.000000 |
| **0** | KNN Image | 71.000000 |
| **0** | Decision Trees Image | 57.000000 |
| **0** | CNN Kepler | 88.415951 |
| **0** | SVM Kepler | 33.000000 |

# KNN

```
knn = KNeighborsClassifier(n_neighbors=2)
```

```
knn.fit(X_train, y_train)
```

```
▾ KNeighborsClassifier
KNeighborsClassifier(n_neighbors=2)
```

```
pred = knn.predict(X_test)
```

```
accuracy= np.round(accuracy_score(y_test, pred)*100,0)
print("Accuracy = ", accuracy)
```

```
Accuracy =  85.0
```

```
modelScores=modelScores.append([['KNN Kepler', accuracy]])
modelScores
```

| | 0 | 1 |
|---|---|---|
| **0** | CNN Image | 85.714287 |
| **0** | VGG16 Image | 57.142860 |
| **0** | VGG19 Image | 42.857143 |
| **0** | SVM Image | 57.000000 |
| **0** | KNN Image | 71.000000 |
| **0** | Decision Trees Image | 57.000000 |
| **0** | CNN Kepler | 88.415951 |

## Decision Trees

```
dt = DecisionTreeClassifier(criterion='entropy', min_samples_split=100)
```

```
dt.fit(X_train, y_train)
```

```
DecisionTreeClassifier
DecisionTreeClassifier(criterion='entropy', min_samples_split=100)
```

```
pred = dt.predict(X_test)
```

```
accuracy= np.round(accuracy_score(y_test, pred)*100,0)
print("Accuracy = ", accuracy)
```

```
Accuracy =  86.0
```

```
modelScores=modelScores.append([['Decision Trees Kepler', accuracy]])
modelScores
```

| | 0 | 1 |
|---|---|---|
| **0** | CNN Image | 85.714287 |
| **0** | VGG16 Image | 57.142860 |
| **0** | VGG19 Image | 42.857143 |
| **0** | SVM Image | 57.000000 |
| **0** | KNN Image | 71.000000 |
| **0** | Decision Trees Image | 57.000000 |
| **0** | CNN Kepler | 88.415951 |
| **0** | SVM Kepler | 33.000000 |
| **0** | KNN Kepler | 85.000000 |
| **0** | Decision Trees Kepler | 86.000000 |

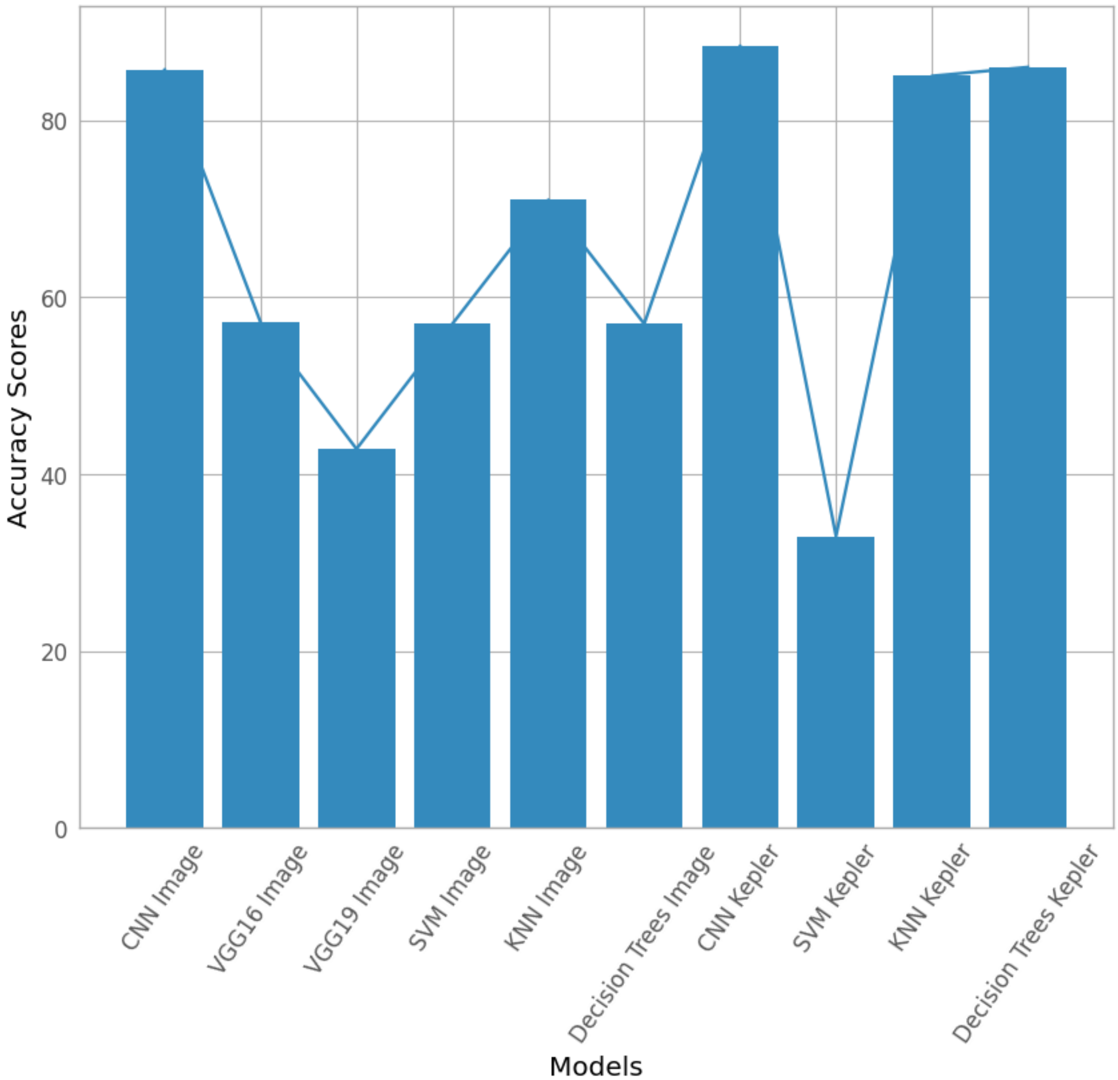
## Comparitive Analysis

```
modelScores.columns = ['Model', 'Accuracy']
modelScores.sort_values(by= 'Accuracy', ascending=False)
```

| | Model | Accuracy |
|---|---|---|
| 0 | CNN Kepler | 88.415951 |
| 0 | Decision Trees Kepler | 86.000000 |
| 0 | CNN Image | 85.714287 |
| 0 | KNN Kepler | 85.000000 |

```
plt.figure(figsize=(10,8))
plt.plot(modelScores['Model'], modelScores['Accuracy'])
plt.bar(modelScores['Model'], modelScores['Accuracy'])
plt.xticks(rotation=55)
plt.xlabel('Models')
plt.ylabel('Accuracy Scores')
```

```
Text(0, 0.5, 'Accuracy Scores')
```



3s completed at 4:34 PM