APPENDIX

CODE INPUT LIBRARY

```
import numpy as np
import pandas as pd
from keras.preprocessing.image import ImageDataGenerator, load_img
from keras.utils import to_categorical
from sklearn.model_selection import train_test_split
import matplotlib.pyplot as plt
import random
import os
from keras.models import Sequential
from keras.layers import Conv2D, MaxPooling2D, Dropout, Flatten, Dense,
Activation, BatchNormalization
```

CODE FOR INPUT DATASET

```
filenames = os.listdir('./train')

categories = []
for filename in filenames:
category = filename.split('.')[0]
if category == 'green':
categories.append(0)  #The categories are set as labels
elif category == 'ripe':
categories.append(1)
elif category == 'overripe':
categories.append(2)

df = pd.DataFrame({
'filename': filenames,
'category': categories,
})
```

CODE for CNN

```
model2 = Sequential()

model2.add(Conv2D(32, (3, 3), activation='relu', input_shape=(200,200,
3)))
```

```
model2.add(MaxPooling2D(pool_size=(2, 2)))
model2.add(Dropout(0.25))

model2.add(Conv2D(64, (3, 3), activation='relu'))
model2.add(MaxPooling2D(pool_size=(2, 2)))
model2.add(Dropout(0.25))

model2.add(Flatten())
model2.add(Dense(96, activation='relu'))
model2.add(Dropout(0.5))
model2.add(Dense(3, activation='softmax'))

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

CODE REPLACE ID TO NAME

```
df["category"] = df["category"].replace({ 0: 'green', 1: 'ripe', 2:
'overripe' })
```

CODE FOR VALIDATE

```
train_df, validate_df = train_test_split(df, test_size=0.30,
random_state=0)
train_df = train_df.reset_index(drop=True)
validate_df = validate_df.reset_index(drop=True)
validate_df['category'].value_counts().plot.bar()
```

CODE FOR MULTYPLY DATA

```
train_datagen = ImageDataGenerator(
rotation_range=15,
brightness_range=[1.0,2.0],
rescale=1./255,
shear_range=0.1,
zoom_range=0.2,
horizontal_flip=True,
width_shift_range=0.1,
height_shift_range=0.1
)

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

```
train_generator = train_datagen.flow_from_dataframe(
train_df,
"./train/",
x_col='filename',
y_col='category',
target_size=(200,200),
class_mode='categorical',
batch_size=32
)

validation_generator = validation_datagen.flow_from_dataframe(
validate_df,
"./train/",
x_col='filename',
y_col='category',
target_size=(200,200),
class_mode='categorical',
batch_size=32
)
```

CODE FOR TRAINING PROCESS

```
epochs=75
batch_size=32

history = model2.fit_generator(
train_generator,
epochs=epochs,
validation_data=validation_generator,
validation_steps=total_validate/batch_size,
steps_per_epoch=total_train/batch_size,
)
```

CODE FOR SAVE RESULT FROM TRAINING

```
model2.save('banana.hdf5')
```

CODE FOR LOAD MODEL FROM TRAINING RESULT

```
from keras import models
model = models.load_model('banana.hdf5', compile = False)
```

CODE FOR INPUT TESTING DATA

```
test_filenames = os.listdir("./tes/")
test_df = pd.DataFrame({
'filename': test_filenames
})
```

CODE FOR PREDICT DATA TESTING

```
predict = model.predict_generator(test_generator,
steps=np.ceil(total_test/batch_size))
test_df['category'] = np.argmax(predict, axis=1)
```

CODE FOR REPLACE ID TO NAME

```
# Map labels
label_map = dict((v,k) for k,v in
train_generator.class_indices.items())
test_df['category'] = test_df['category'].replace(label_map)
print(label_map, "\n")

# See the output of mapped labels
print(test_df['category'][0:4], "\n")
```

FOR SHOW RESULT PREDICT AND CALCULATE TP,TN,FN,FP

```
sample_test = test_df.head(80)
sample_test.head()
plt.figure(figsize=(12, 24))

TPgreen = 0
TNgreen = 0
FPgreen = 0
FNgreen = 0

TPoverripe = 0
TNoverripe = 0
FPoverripe = 0
FNoverripe = 0

TPripe = 0
TNripe = 0
FPripe = 0
```

```
FNripe = 0

counter = 0
for index, row in sample test.iterrows():
counter+=1
filename = row['filename']
category = row['category']
a = filename.find('.')
b = filename[0:a]
if b == 'g':
if category == 'green':
hasil = 'true';
TPgreen+=1;
TNoverripe+=1;
TNripe+=1;
else :
hasil = 'false';
FNgreen+=1;
if category == 'overripe' :
FNoverripe+=1;
TNripe+=1;
else :
FNripe+=1;
TNoverripe+=1;

elif b == 'or':
if category == 'overripe':
hasil = 'true';
TPoverripe+=1;
TNgreen+=1;
TNripe+=1;
else :
hasil = 'false';
FPoverripe+=1;
if category == 'green':
FNgreen+=1;
TNripe+=1;
else :
```

```
FNripe+=1;
TNgreen+=1;
elif b == 'r':
if category == 'ripe':
hasil = 'true';
TPripe+=1;
TNgreen+=1;
TNoverripe+=1;
else :
hasil = 'false';
FPripe+=1;
if category == 'green':
FNgreen+=1;
TNoverripe+=1;
else :
FNoverripe+=1;
TNgreen+=1;

img = load_img("./tes/"+filename, target_size=(200,200))
plt.subplot(10, 8, index+1)
plt.imshow(img)
plt.xlabel(filename + "\n Predicted " + "{}".format(category))

plt.tight_layout()
plt.show()
```

CALCULATE PRESISION RECALL AND ACCURACY

```
precisiongreen = TPgreen / (TPgreen+FPgreen)
recallgreen = TPgreen/(TPgreen+FNgreen)
print('Green')
print('precission         ',precisiongreen)
print('recall             ',recallgreen)

precisionoverripe = TPoverripe / (TPoverripe+FPoverripe)
recalloverripe = TPoverripe/(TPoverripe+FNoverripe)
print('\nOverripe')
print('precission         ',precisionoverripe)
print('recall             ',recalloverripe)
```

```
        precisionripe = TPripe /  (TPripe+FPripe)
        recallripe = TPripe/(TPripe+FNripe)
        print('\nRipe')
        print('precission          ',precisionripe)
        print('recall              ',recallripe)

        print('\nACCURACY          ',(TPgreen+TPoverripe+TPripe)/counter)
        print('\nPRECISSION
',((precisiongreen+precisionoverripe+precisionripe)/3))
        print('\nRECALL            ',((recallgreen+recalloverripe+recallripe)/3))
```

PLAGIARISM CHECK.ORG

0.82% PLAGIARISM APPROXIMATELY

Report #13361827

Introduction Background Banana is a fruit that we often see everywhere, and many of us consume bananas. Banana is one of the leading agricultural product from indonesia. This fruit is very popular in the community because it is easy to get anywhere at the price offered is also very affordable and bananas are also easy to consume starting from the age level of babies to parent. Many of us don t know what the ideal level of ripeness for a banana is, As we know if a banana is yellow meaning its ripe but is many type of banana ripeness. The conditions of the banana fruit is determined by several parameters, one of which is the level of maturity seen from the color of the peel it self and the maturity divided into 3 parts starting from the green, yellow, and blackish yellow banana. Bananas are generally differentiated by manual methods, using humans to classify them, but along with the development of computer technology that has entered the agricultural sector, both from the pre-harvest to