CHAPTER 5
IMPLEMENTATION AND TESTING

5.1. Implementation

This chapter five describes implementations and testing where the implementation would discuss about the code and testing discuss about trial result of algorithm.

```
1.     model2 = Sequential()
2.     model2.add(Conv2D(32, (3, 3), activation='relu',
       input_shape=(200,200,3)))
3.     model2.add(MaxPooling2D(pool_size=(2, 2)))
4.     model2.add(Dropout(0.25))
5.     model2.add(Conv2D(64, (3, 3), activation='relu'))
6.     model2.add(MaxPooling2D(pool_size=(2, 2)))
7.     model2.add(Dropout(0.25))
8.     model2.add(Flatten())
9.     model2.add(Dense(96, activation='relu'))
10.    model2.add(Dropout(0.5))
11.    model2.add(Dense(3, activation='softmax'))
12.    model2.compile(loss='categorical_crossentropy', optimizer='SGD',
       metrics=['accuracy'])
```

As we see on the list above, first line means sequential model applied in this process of this convolution. In the second line until line seven session is a process where the cnn work, and there are several layers consisting of several layers such as convolution layer, pooling layer, and dense layer.

```
1. epochs=75
2. batch_size=32
3.
4. history = model2.fit_generator(
5. train_generator,
6. epochs=epochs,
7. validation_data=validation_generator,
8. validation_steps=total_validate/batch_size,
9. steps_per_epoch=total_train/batch_size,
10.       )
```

With using CNN we have to train the datasets that we collected and filtered like on the previously code discussion. As we can see on the top is a code to train the processed dataset, on the 1st line is epochs=75 that means 75 repetition. On the last line is a code to determine the average results it got from training process.

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

The code above is used to predict the final result from images that we want to predict.

5.2. Testing

After having such analysis and implementation, the researcer did some testing. Here some testing from different optimizer that have done by the researcher.

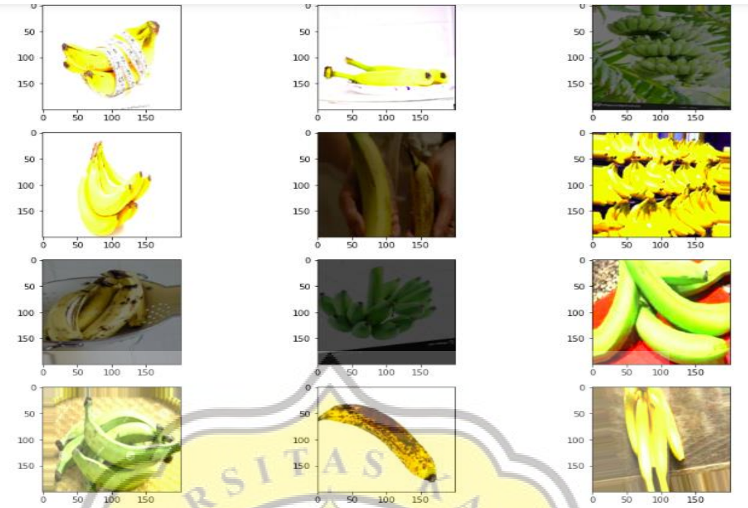
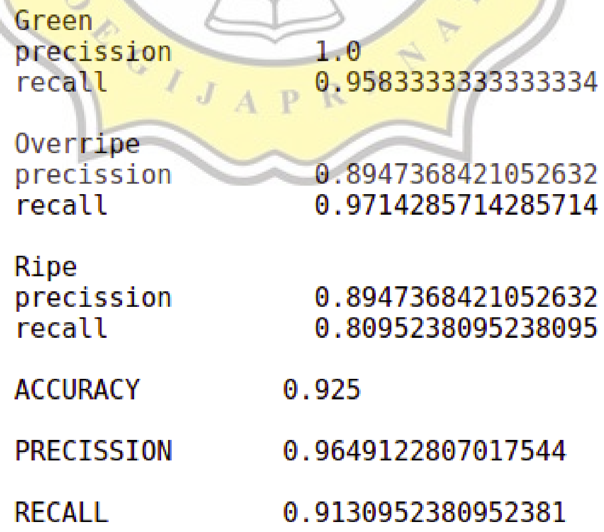


Figure 5.2 Result Brightness Adjustment

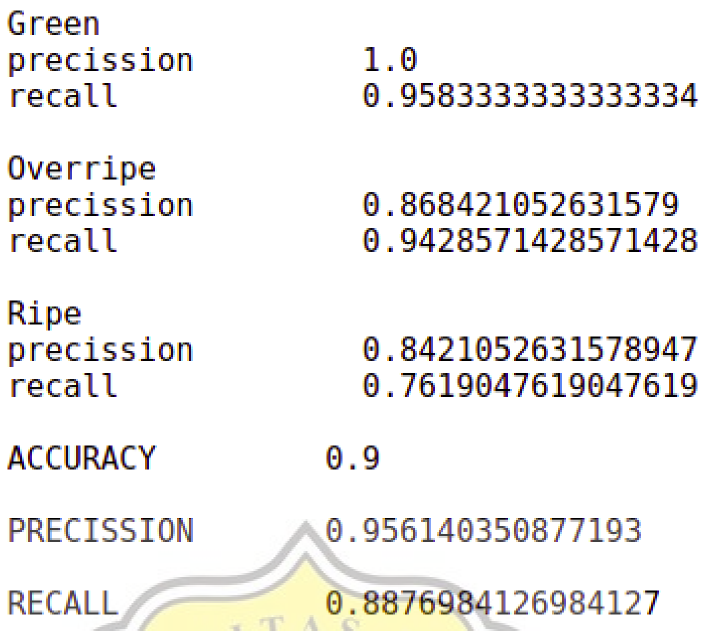
We can see from the picture above is the result of the tada set that has been set the brightness adjustment. After the brightness adjustment is complete, the next are some result from the calculation of presision, recall, and accuracy of several optimizer that are compared :



```
Green
precission          1.0
recall              0.9583333333333334

Overripe
precission          0.8947368421052632
recall              0.9714285714285714

Ripe
precission          0.8947368421052632
recall              0.8095238095238095

ACCURACY         0.925

PRECISSION       0.9649122807017544

RECALL           0.9130952380952381
```

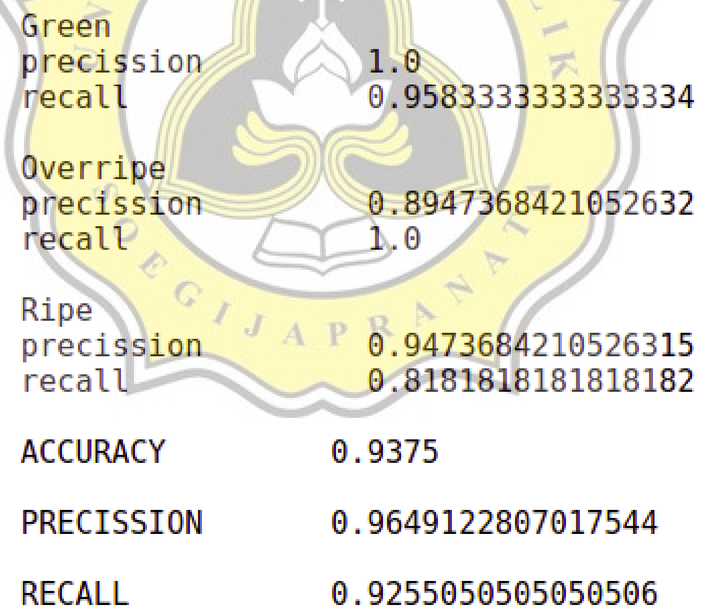
Figure 5.3 *Result Adam Optimizer*

UNIVERSITAS KATOLIK SOEGIJAPRANATA



```
Green
precission         1.0
recall             0.9583333333333334

Overripe
precission         0.868421052631579
recall             0.9428571428571428

Ripe
precission         0.8421052631578947
recall             0.7619047619047619

ACCURACY         0.9

PRECISSION       0.956140350877193

RECALL           0.8876984126984127
```

***Figure* 5.4** *Result RMSprop Optimizer*



```
Green
precission         1.0
recall             0.9583333333333334

Overripe
precission         0.8947368421052632
recall             1.0

Ripe
precission         0.9473684210526315
recall             0.8181818181818182

ACCURACY         0.9375

PRECISSION       0.9649122807017544

RECALL           0.9255050505050506
```

***Figure* 5.5** *Result SGD Optimizer*

UNIVERSITAS KATOLIK SOEGIJAPRANATA

As we can see the image above is the result of several experiments with 75 epoch using the Adam, RMSprop, and SGD optimizer. From these results, we can see that the result obtained by SGD optimizer are 93,75%, where the result the result obtained are the highest from other optimizer. Adam get 92,5% accuracy, and the last RMSprop optimizer get 90% accuracy that is not much different from Adam, because Adam is combine from RMSprop and momentum that why result from Adam and RMSprop is not so far in difference

5.3. Diagram of analysis

This sub-chapter give the result from the testing that already done, from the 80 predicted images, banana with ripeness level of green predicted as much as 23 images, ripe by 17, overripe by 35, and 5 images predicted wrong. As we can see in the diagram below this project can get results as shown in the diagram below. This diagram is for classification the ripeness of banana, grey for overripe, orange for ripe, blue for green banana, and yelow for the incorrect.

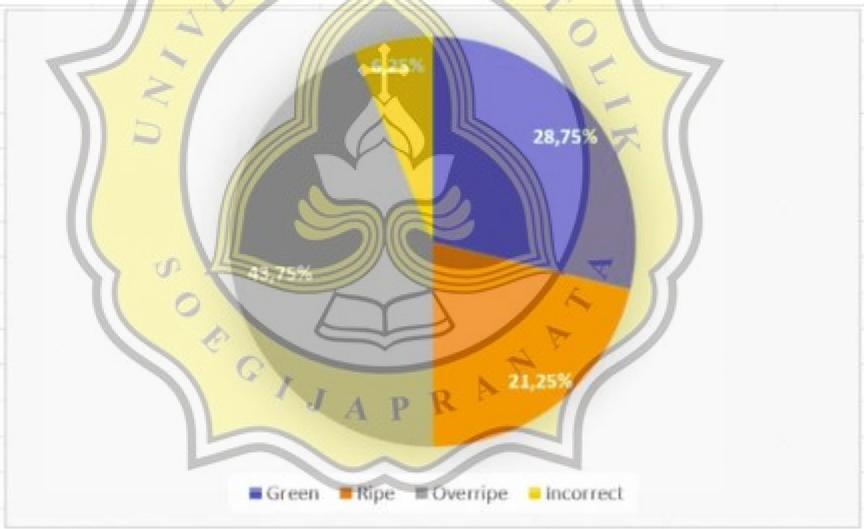


__Figure 5.6__ Diagram Analysis Predict

UNIVERSITAS KATOLIK SOEGIJAPRANATA