

SAMPLE OF SOME RECORDED OBSERVATIONS

Settings:

No of epochs: 5

Batch size: 32

Learning rate: 0.0001

Optimizer: Rmsprop

Model without Data Augmentation:

Model Performance:

```
print('Test data loss:', acc[0] )
print('Test data accuracy:', acc[1] * 100)
```

Test data loss: 1.148381689453125
Test data accuracy: 59.730000000000004

Time Taken: 1 min 7 secs

Feature Standardization:

Model Performance:

```
print('Test data loss:', acc[0] )
print('Test data accuracy:', acc[1] * 100)
```

Test data loss: 2.2379002841949465
Test data accuracy: 21.18

Time Taken: 45 mins

ZCA Whitening:

Model Performance:

```
print('Test data loss:', acc[0] )
print('Test data accuracy:', acc[1] * 100)
```

Test data loss: 8.67
Test data accuracy: 12.8

Time Taken: 5 hrs 20 mins

Random Rotation:

Rotation: 90

Model Performance:

```
print('Test data loss:', acc[0] )  
print('Test data accuracy:', acc[1] * 100)
```

Test data loss: 1.4249906951904296
Test data accuracy: 55.989999999999995

Time Taken: 90 mins

Shift

Horizontal Shift Range: 0.2

Model Performance:

```
print('Test data loss:', acc[0] )  
print('Test data accuracy:', acc[1] * 100)
```

Test data loss: 1.03478266248703
Test data accuracy: 66.62

Time Taken: 115 mins

Flips

Horizontal Flip

Model Performance:

```
print('Test data loss:', acc[0] )  
print('Test data accuracy:', acc[1] * 100)
```

Test data loss: 1.030903701210022
Test data accuracy: 66.03

Time Taken: 45 mins

Vertical Flip

Model Performance:

```
print('Test data loss:', acc[0] )  
print('Test data accuracy:', acc[1] * 100)
```

Test data loss: 1.1695779172897338
Test data accuracy: 61.24000000000001

Time Taken: 44 mins

Increasing Number of epochs

No of epochs: 10

Model without Data Augmentation:

Model Performance:

```
print('Test data loss:', acc[0] )  
print('Test data accuracy:', acc[1] * 100).
```

Test data loss: 1.0272189430236816
Test data accuracy: 64.14999999999999

Time Taken: 3 mins 4 secs

Feature Standardization:

```
print('Test data loss:', acc[0] )  
print('Test data accuracy:', acc[1] * 100)
```

Test data loss: 2.270123110961914
Test data accuracy: 23.9

Time Taken: 90 mins

Random Rotation:

Rotation: 90

Model Performance:

```
print('Test data loss:', acc[0] )  
print('Test data accuracy:', acc[1] * 100).
```

Test data loss: 1.4524426197052003
Test data accuracy: 56.56

Time Taken: 169 mins

Shift

Horizontal Shift Range: 0.1

Model Performance:

```
print('Test data loss:', acc[0] )  
print('Test data accuracy:', acc[1] * 100)
```

Test data loss: 1.066214454650879
Test data accuracy: 66.88

Time Taken: 230 mins

Vertical Shift

Model Performance:

```
print('Test data loss:', acc[0] )
print('Test data accuracy:', acc[1] * 100)
```

Test data loss: 1.153040506
Test data accuracy: 61.040000000000006

Time Taken: 90 mins 20 secs

Number of epochs: 15

```
print('Test data loss:', acc[0] )
print('Test data accuracy:', acc[1] * 100)
```

Test data loss: 1.1695779172897338
Test data accuracy: 61.240000000000001

Feature Standardization

Model Performance:

```
print('Test data loss:', acc[0] )
print('Test data accuracy:', acc[1] * 100)
```

Test data loss: 2.290641449356079
Test data accuracy: 20.48

Time Taken: 135 mins

Combination of Rotation range: 90 and Horizontal Flip

Number of epochs: 10

Model Performance:

```
print('Test data loss:', acc[0] )
print('Test data accuracy:', acc[1] * 100)
```

Test data loss: 1.45641529045105
Test data accuracy: 55.789999999999999

Time Taken: 181 mins

Observations:

- **ZCA Whitening** seems to take a lot of time to train with almost 1 hour per epoch yet doesn't provide promising results.
- **Rotation** when applied to given data, it seems to work well.
- **Feature Standardization** doesn't provide much improvement to our dataset in this case.
- **Shifts** when applied do show improvements in model performance be it height or width.
- **Flips** also tend to show increase in model performance along with decrease in Loss.
- **Image Augmentation** does bring about improvements to model, but we need to keep in mind the tradeoff between model performance and training time while building an optimal model.