CS671 Deep Learning
Even Sem 2018-19
March 31^{th}, 2019

Multi-Head Classification

Assignment - 2

Submitted By

Roll No	Name
B16066	Nikhil T R
B16021	Kaustubh Verma
B16004	Aj R Laddha

Contents

Chapter 1

Variations

	v1	v2	v3
test accuracy	0.831	0.977	0.997
epochs	10	15	15
batch size	12800	6400	3200

Table 1.1: Variations

The data set originally had 96000 images. We did a 80-10-10 split, which gave us 76800, 9600, and 9600 images in training, validation, and testing sets respectively.

Adam Optimizer was used for all the 3 variations. The loss function used was a direct sum of all the losses. This was for variation 1 and 2. This affected the learning rates of the different classification heads. After looking at their learning curves, a weighted sum with the following weights was taken as the overall loss:

1. **Length**: 5.0
2. **Width**: 2.0
3. **Colour**: 1.0
4. **Angle**: 50.0

With variation 2, MaxPooling was used along with deeper classification heads. The CNN layers' filters were increased. With variation 3, we changed the kernel and bias initializer in the classification heads from the default glorot_uniform to RandomNormal with a standard deviation of 0.01, along with introducing the loss weights. This also increased the performance.

Chapter 2

Variation 1

2.1 Architecture

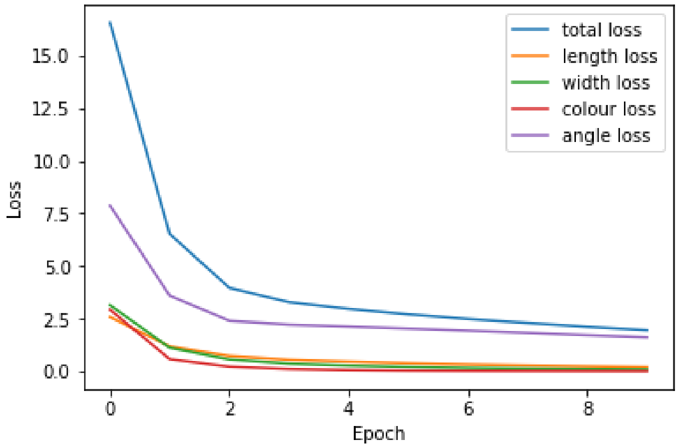
2.1.1 CNN

1. **Conv2D** 3x3, 4 filters, activation relu
2. **Conv2D** 5x5, 4 filters, activation relu
3. **Flatten**

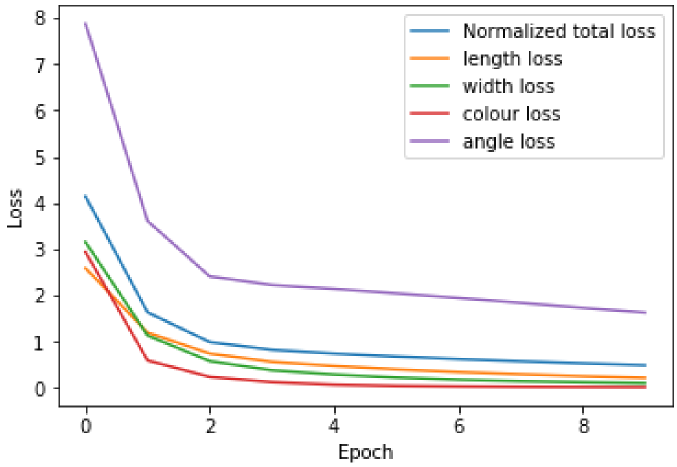
2.1.2 Classification Heads

1. **Length**
   (a) **Dense** 64, activation relu
   (b) **Dense** 1, activation sigmoid
2. **Width**
   (a) **Dense** 64, activation relu
   (b) **Dense** 1, activation sigmoid
3. **Colour**
   (a) **Dense** 64, activation relu
   (b) **Dense** 1, activation sigmoid
4. **Angle**
   (a) **Dense** 64, activation relu
   (b) **Dense** 12, activation sigmoid

2.2 Learning Curves

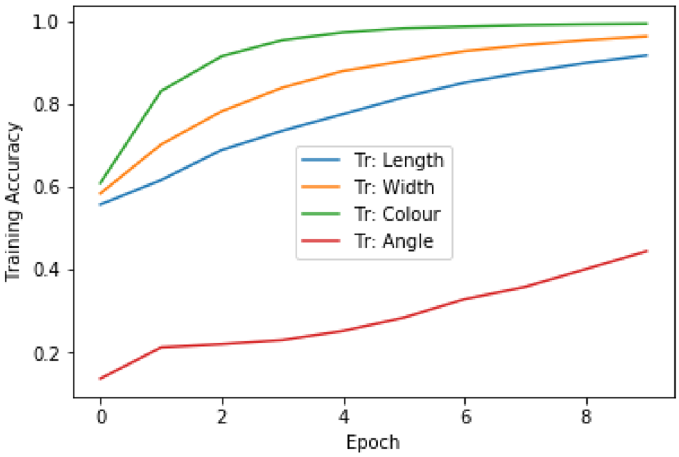


(a) Total Loss

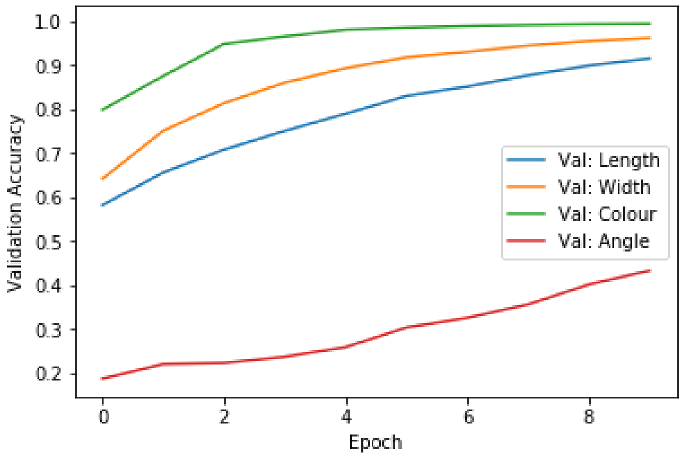


(b) Normalized Loss

Figure 2.1: Training Loss



(a) Training



(b) Validation

Figure 2.2: Accuracies

2.3 Confusion Matrices

	Short	Long
Short	4285	413
Long	359	4543

Length

	Thin	Thick
Thin	4649	146
Thick	198	4607

Width

	Blue	Red
Blue	4711	20
Red	25	4844

Colour

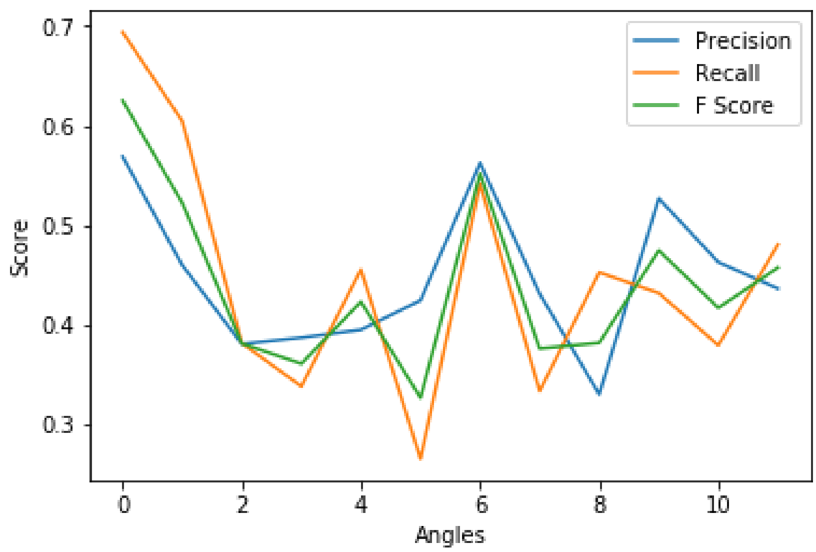
566	43	14	20	4	6	1	5	30	19	30	78
73	483	79	33	17	12	2	6	31	5	14	44
29	190	299	85	56	14	13	6	55	11	10	15
30	77	111	268	147	40	13	21	37	14	20	14
13	45	68	87	364	58	25	34	54	25	12	14
13	37	85	35	167	214	80	79	44	14	18	19
24	30	37	20	49	52	422	59	40	6	27	13
13	19	28	40	47	39	78	270	185	40	25	25
12	25	25	24	33	23	40	88	340	72	43	26
26	17	17	27	7	18	41	30	112	343	90	66
60	42	6	32	18	19	20	15	58	88	331	183
136	42	16	21	12	9	15	13	43	14	95	385

Angle

2.4 F Scores

	Length	Width	Colour
Precision	0.916	0.969	0.995
Recall	0.926	0.958	0.994
F Score	0.921	0.964	0.995

Note: The values mentioned here are average.



2.5 All 96 Classes

2.5.1 Confusion Matrix

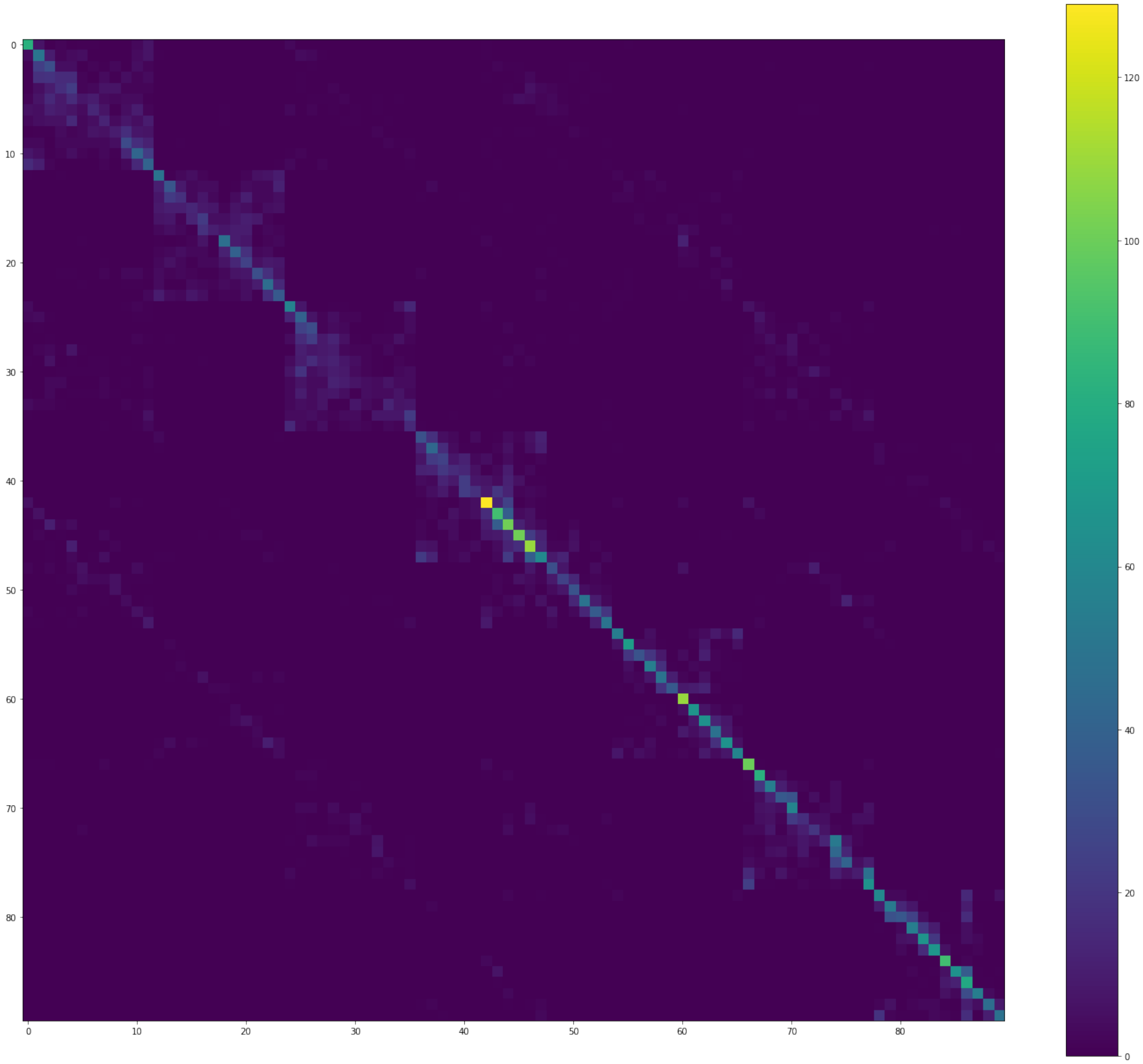


Figure 2.3: Confusion Matrix

2.5.2 Other Metrics

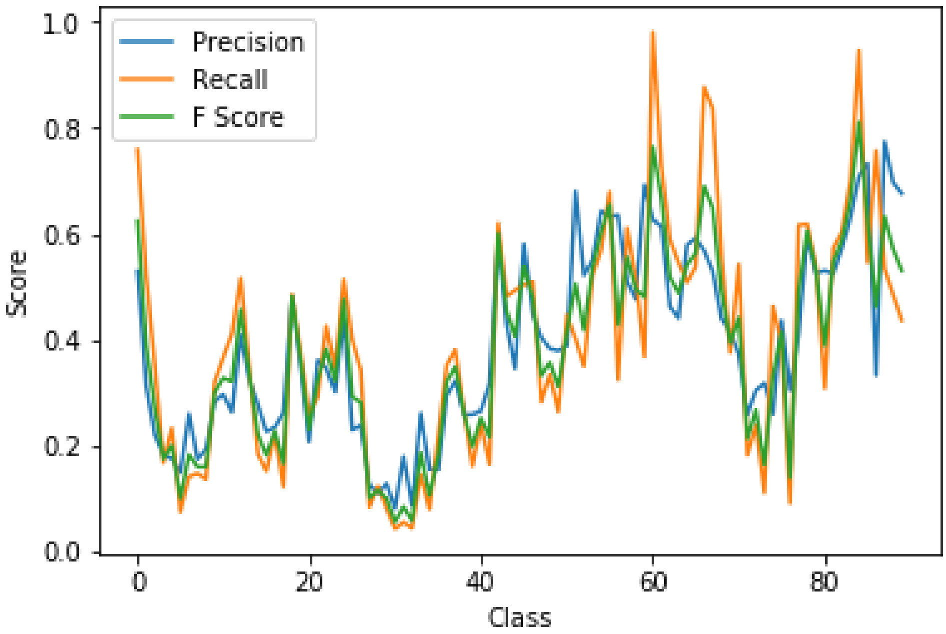


Figure 2.4: Metrics

2.6 Observations

- The model is not able to learn angles quickly. The Confusion matrix shows how there is confusion in nearby classes. They represent variations in angles.
- It is learning to identify colours well.
- The learning rates for different classification heads is different. Increasing the number of epochs would definitely make the accuracy better.

Chapter 3

Variation 2

3.1 Architecture

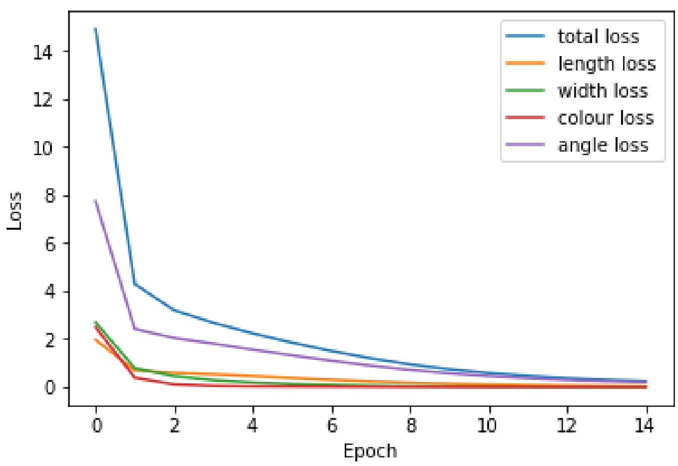
3.1.1 CNN

1. **Conv2D** 7x7, 8 filters, activation relu
2. **MaxPool2D** 2x2,strides of 2
3. **Conv2D** 5x5, 6 filters, activation relu
4. **MaxPool2D** 2x2,strides of 1
5. **Flatten**

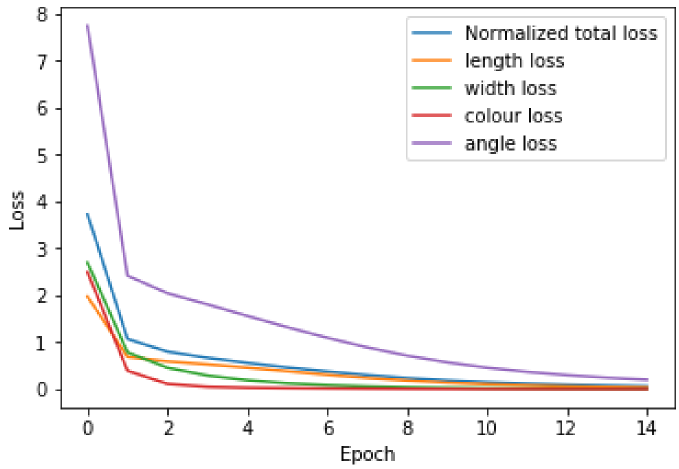
3.1.2 Classification Heads

1. **Length**
   (a) **Dense** 128, activation relu
   (b) **Dense** 32, activation relu
   (c) **Dense** 1, activation sigmoid
2. **Width**
   (a) **Dense** 64, activation relu
   (b) **Dense** 1, activation sigmoid
3. **Colour**
   (a) **Dense** 64, activation relu
   (b) **Dense** 1, activation sigmoid
4. **Angle**
   (a) **Dense** 256, activation relu
   (b) **Dense** 64, activation relu
   (c) **Dense** 12, activation softmax

3.2 Learning Curves

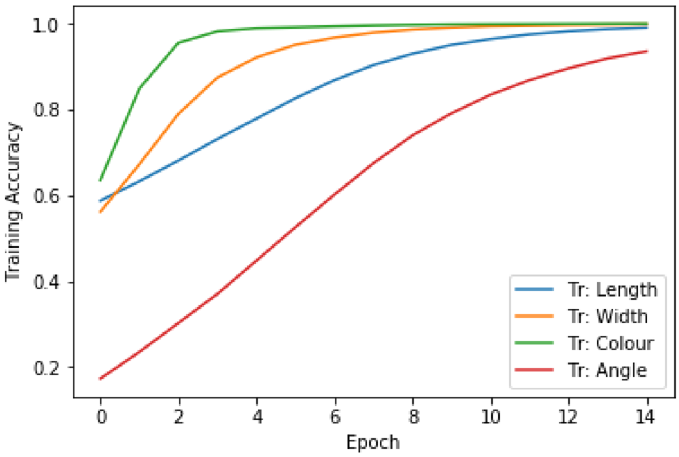


(a) Total Loss

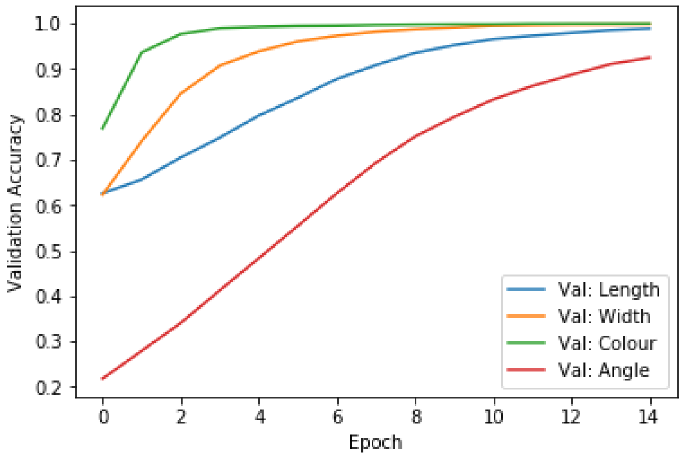


(b) Normalized Loss

Figure 3.1: Training Loss



(a) Training



(b) Validation

Figure 3.2: Training Accuracies

3.3 Confusion Matrices

	Short	Long
Short	4662	36
Long	57	4845

Length

	Thin	Thick
Thin	4788	7
Thick	6	4799

Width

	Blue	Red
Blue	4730	1
Red	3	4866

Colour

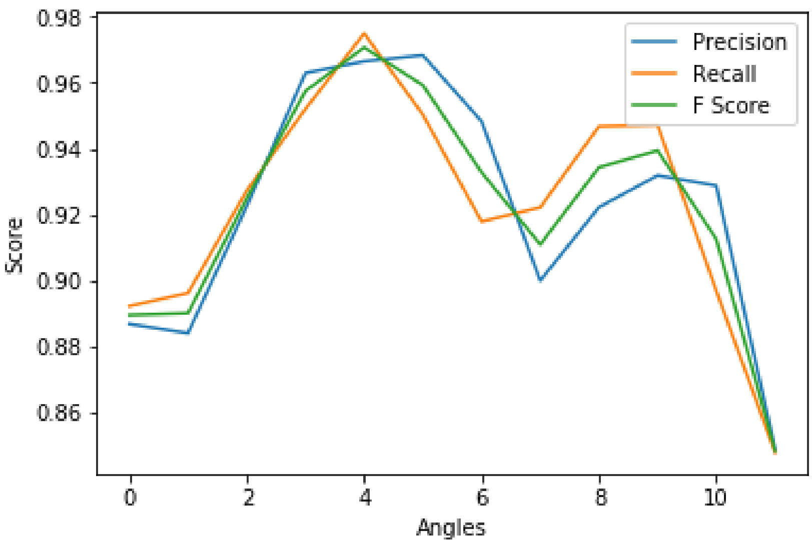
728	30	0	1	0	0	0	0	0	0	3	54
22	716	32	0	1	1	0	0	0	0	1	26
1	44	726	12	0	0	0	0	0	0	0	0
0	1	25	754	10	1	0	0	0	1	0	0
0	0	0	12	779	8	0	0	0	0	0	0
1	0	0	1	15	765	14	5	3	1	0	0
0	0	0	2	1	11	715	50	0	0	0	0
0	0	0	0	0	3	22	746	34	4	0	0
0	0	0	0	0	1	2	26	711	9	0	2
0	0	0	0	0	0	1	2	17	752	20	2
7	5	2	0	0	0	0	0	6	33	782	37
62	14	2	1	0	0	0	0	0	7	36	679

Angle

3.4 F Scores

	Length	Width	Colour
Precision	0.992	0.998	0.999
Recall	0.988	0.998	0.999
F Score	0.990	0.998	0.999

Note: The values mentioned here are average.



3.5 All 96 Classes

3.5.1 Confusion Matrix

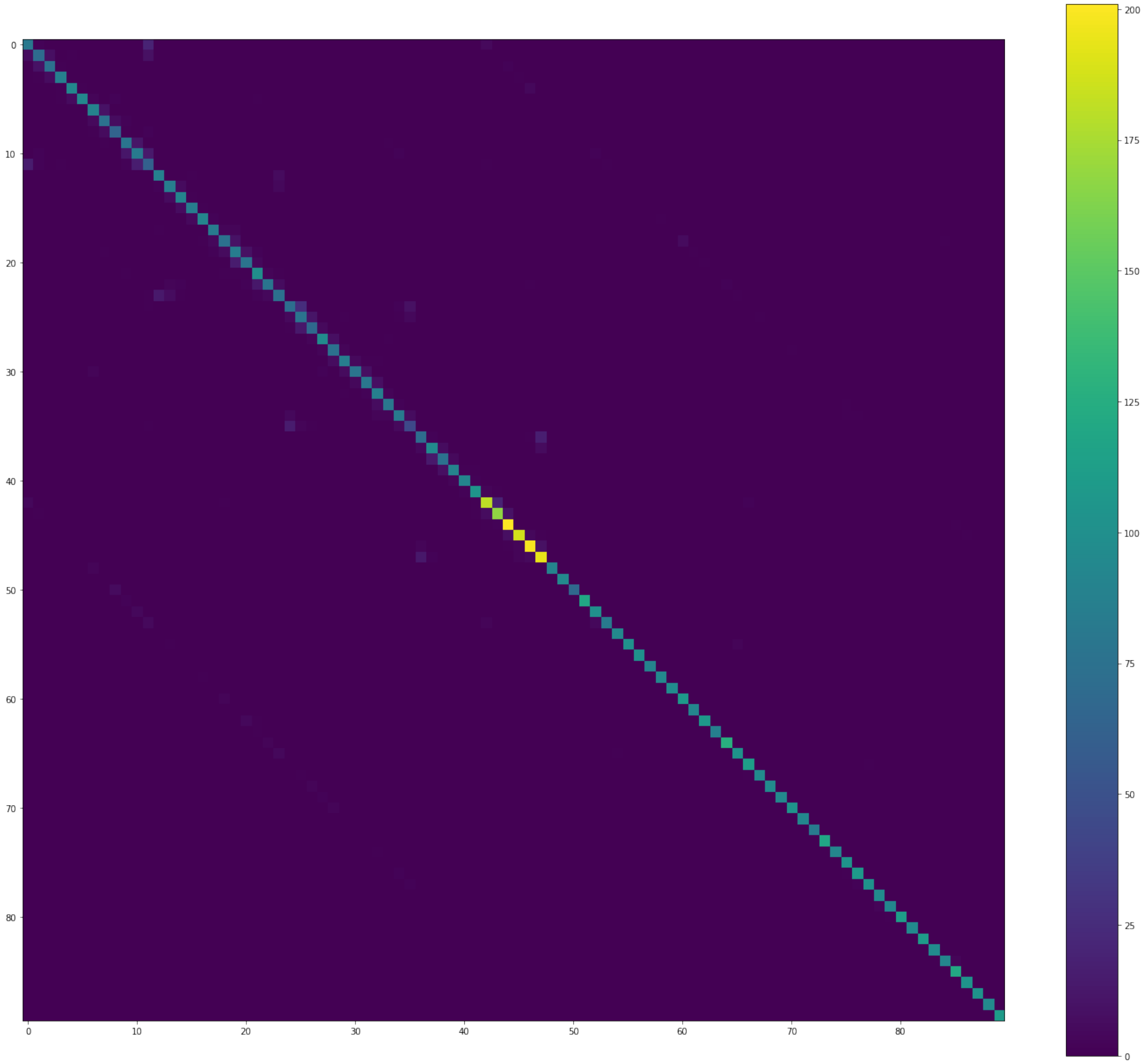


Figure 3.3: Confusion Matrix

3.5.2 Other Metrics

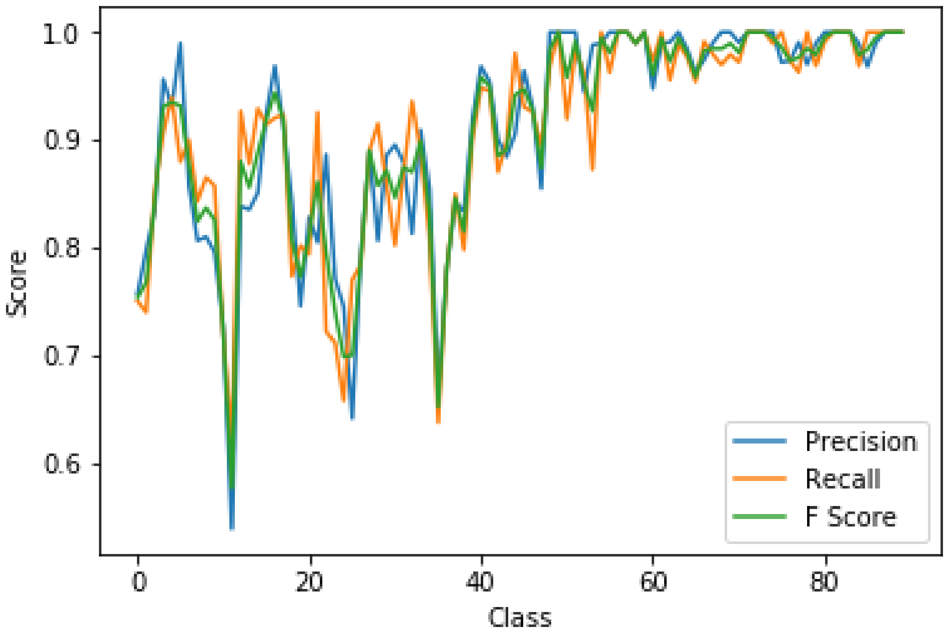


Figure 3.4: Metrics

3.6 Observations

- There is some increase in performance.
- The addition of more CNN and MaxPooling layers along with deeper dense networks in the classification heads seems to have done a difference.
- The learning, despite increasing to 15 epochs is still not satisfactory.
- The loss function is just a sum of all losses. This is not appropriate as they are different functions.

Chapter 4

Variation 3

4.1 Architecture

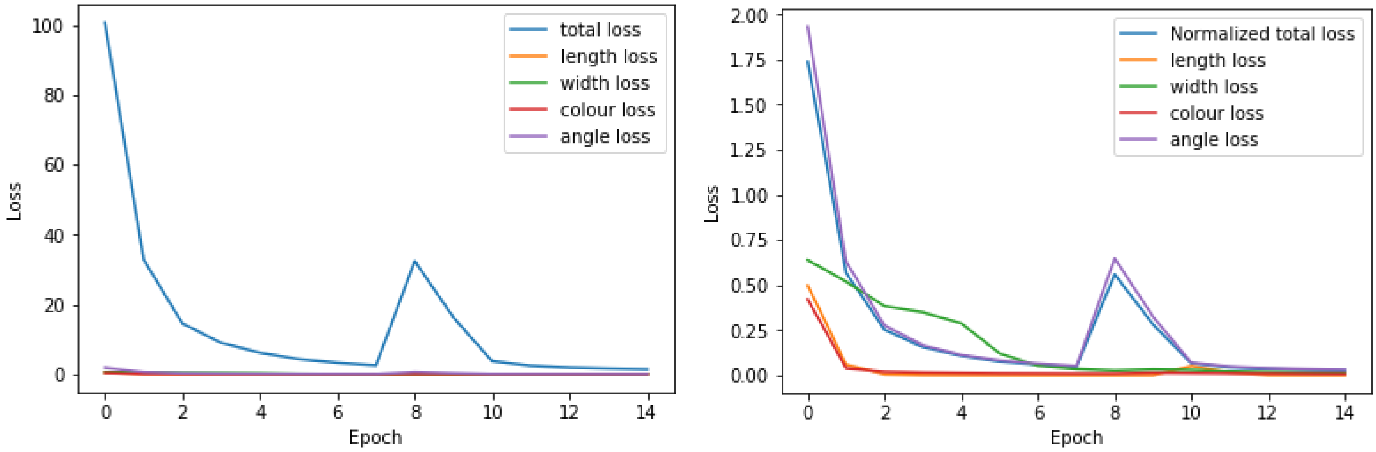
4.1.1 CNN

1. **Conv2D** 5x5, 10 filters, activation relu
2. **MaxPool2D** 2x2,strides of 2
3. **Conv2D** 3x3, 8 filters, activation relu
4. **MaxPool2D** 2x2,strides of 2
5. **Flatten**

4.1.2 Classification Heads

1. **Length**
   (a) **Dense** 128, activation relu,
   (b) **Dense** 32, activation relu
   (c) **Dense** 1, activation sigmoid
2. **Width**
   (a) **Dense** 64, activation relu
   (b) **Dense** 8, activation relu
   (c) **Dense** 1, activation sigmoid
3. **Colour**
   (a) **Dense** 64, activation relu
   (b) **Dense** 8, activation relu
   (c) **Dense** 1, activation sigmoid
4. **Angle**
   (a) **Dense** 128, activation relu
   (b) **Dense** 64, activation relu
   (c) **Dense** 12, activation softmax

4.2 Learning Curves



(a) Total loss (b) Normalized loss

Figure 4.1: Training Loss

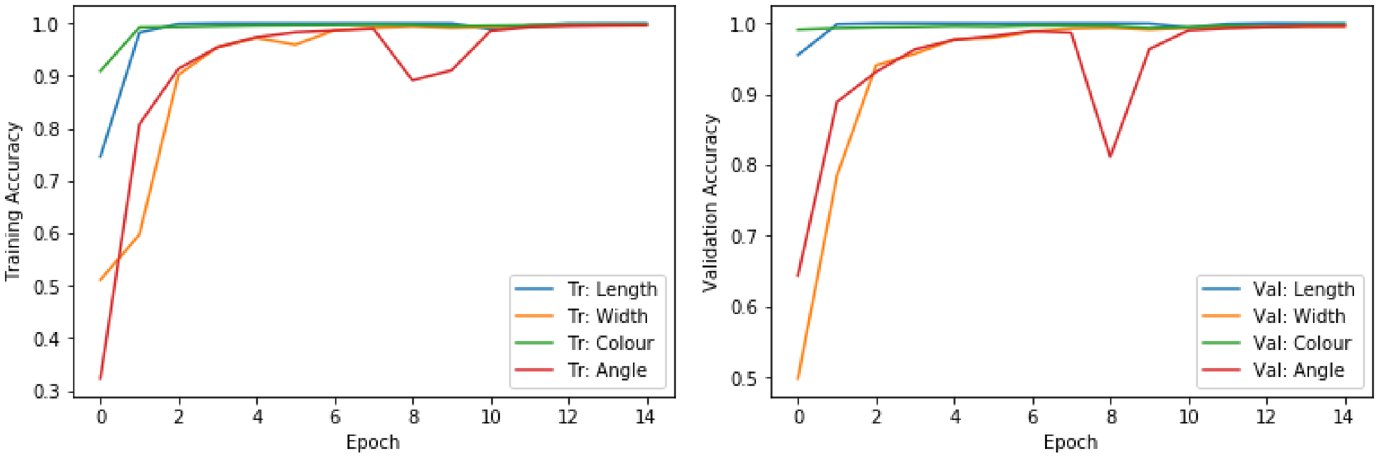


Figure 4.2: Training Accuracies

4.3 Confusion Matrices

	Short	Long
Short	4698	0
Long	0	4902

Length

	Thin	Thick
Thin	4794	1
Thick	25	4780

Width

	Blue	Red
Blue	4724	7
Red	24	4845

Colour

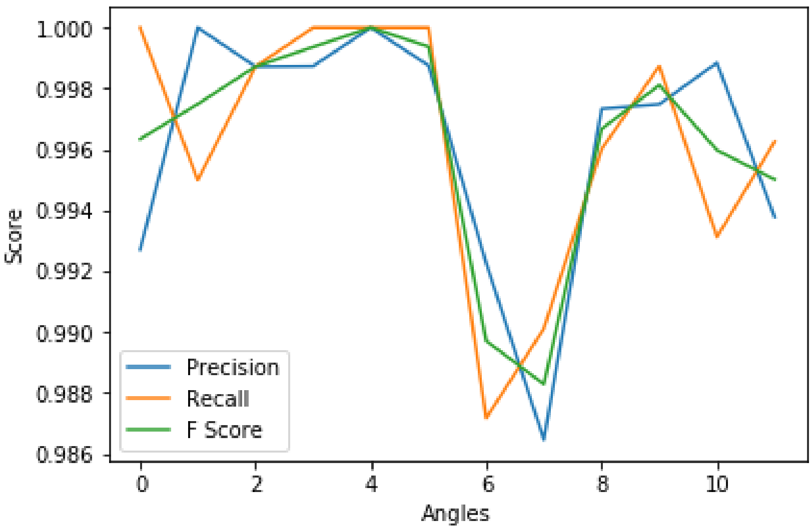
728	30	0	1	0	0	0	0	0	0	3	54
22	716	32	0	1	1	0	0	0	0	1	26
1	44	726	12	0	0	0	0	0	0	0	0
0	1	25	754	10	1	0	0	0	1	0	0
0	0	0	12	779	8	0	0	0	0	0	0
1	0	0	1	15	765	14	5	3	1	0	0
0	0	0	2	1	11	715	50	0	0	0	0
0	0	0	0	0	3	22	746	34	4	0	0
0	0	0	0	0	1	2	26	711	9	0	2
0	0	0	0	0	0	1	2	17	752	20	2
7	5	2	0	0	0	0	0	6	33	782	37
62	14	2	1	0	0	0	0	0	7	36	679

Angle

4.4 F Scores

	Length	Width	Colour
Precision	1.0	0.999	0.998
Recall	1.0	0.994	0.995
F Score	1.0	0.997	0.996

Note: The values mentioned here are average.



4.5 All 96 Classes

4.5.1 Confusion Matrix

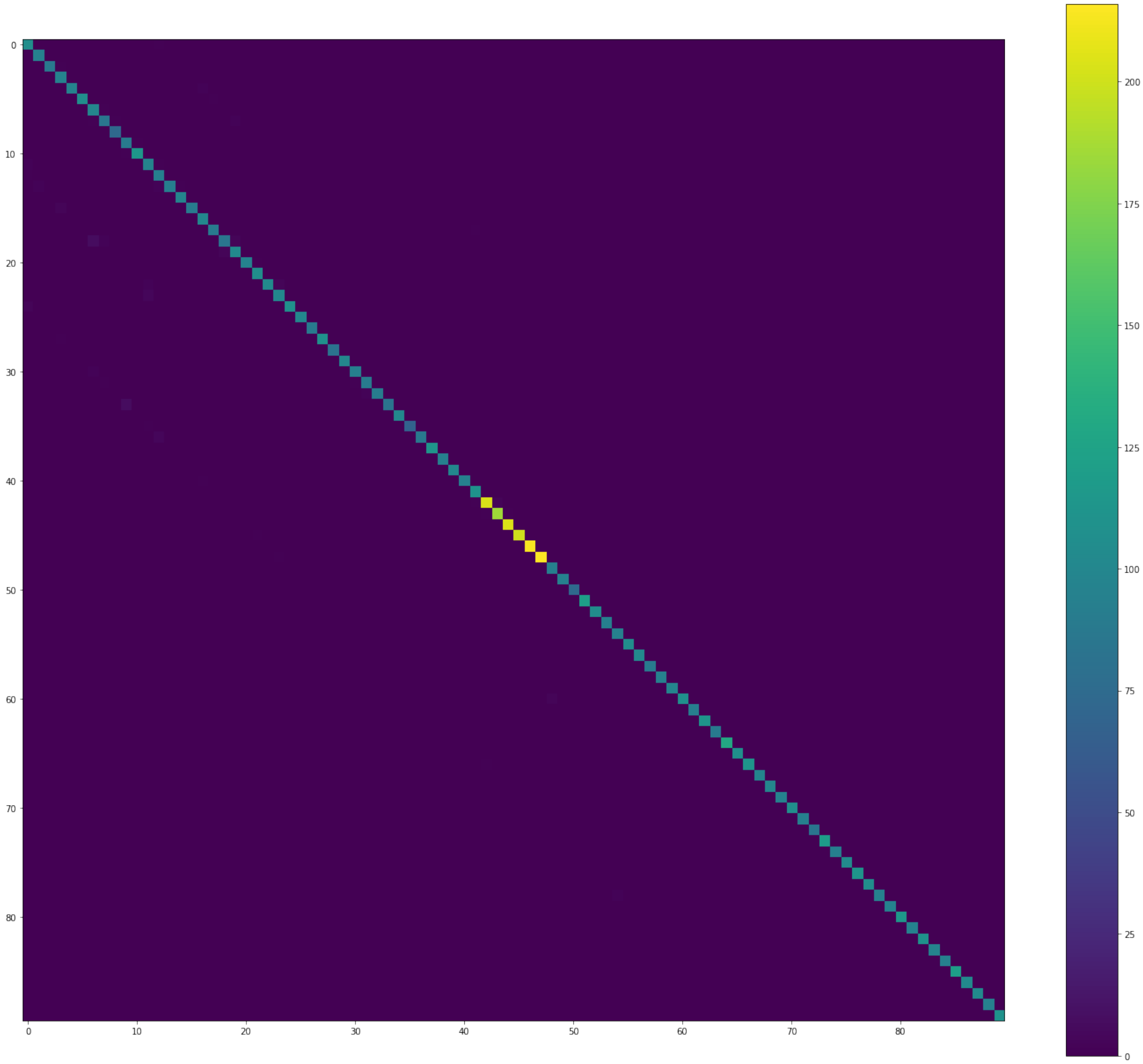


Figure 4.3: Confusion Matrix

4.5.2 Other Metrics

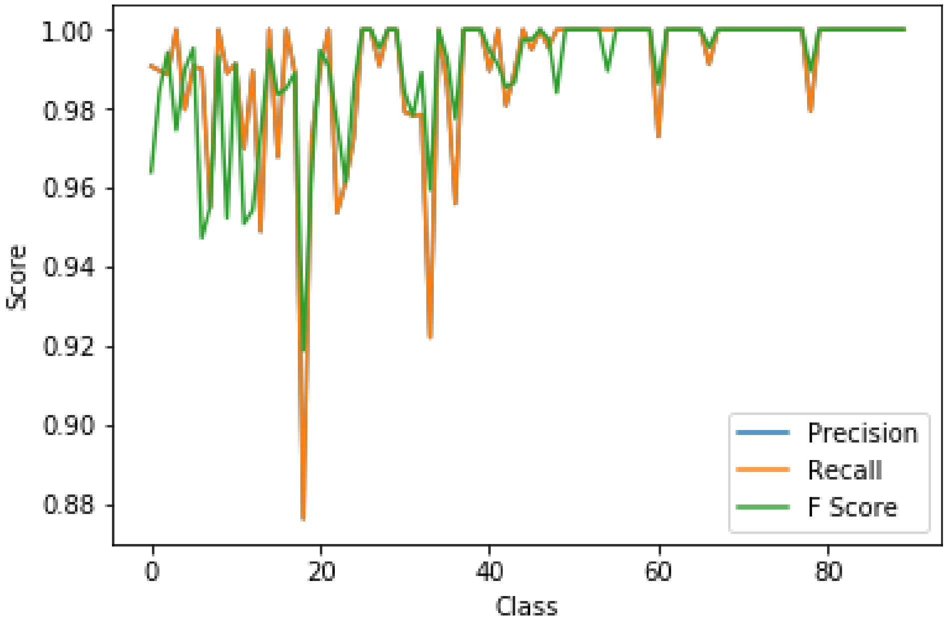


Figure 4.4: Metrics

4.6 Observations

- The model is not able to learn angles quickly.
- It is learning to identify colours well.
- The loss function is just a sum of all losses. This is not appropriate as they are different functions.

Chapter 5

Inferences

- Increasing the complexity doesn't always bring better performance. Sometimes it may just add onto the time required to train.
- Loss weights can be applied to normalize the learning rates of different classification heads.
- For dense layers, initialization with random numbers from a normal distribution with very low standard deviation seems to make a huge impact on performance. We are yet to see a case where it can hinder the same.
- After training the model, it is important to save the model. It can be reused later.