MLP Classification on Image Features

Computer Vision

Assignment 1

Varun Edachali (2022101029)

January 2025

1 Introduction

We perform some simple classification tasks using the data available here.

1.1 Dataset Analysis and Preprocessing

As a part of the pre-processing, we randomly collect 10% of the train set and store it as the validation set, thus splitting the raw train set into train and validation.

In addition, for each image in the (new) train set, we process them in order to convert them to images, and store them locally for some qualitative analysis. Based on the images present, I believe the classes are as follows:

- **0** informal wear / half sleeves + sleeveless shirts

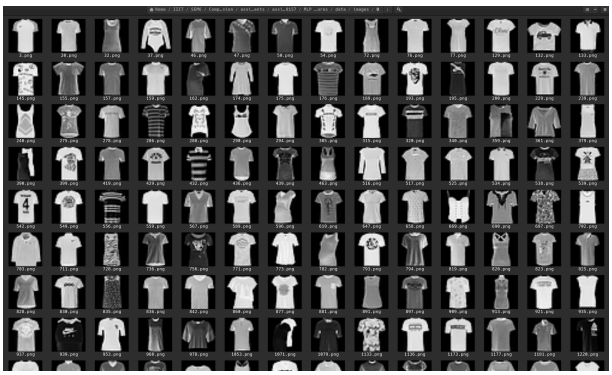

Figure 1: class 0

- **1** pants

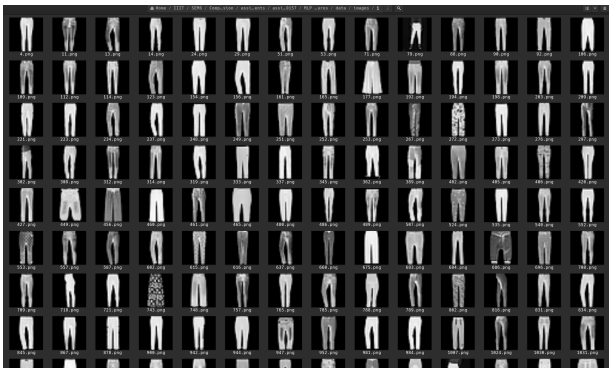

Figure 2: class 1

- **2** hoodies and sweaters

Figure 3: class 2

- **3** dresses and party wear

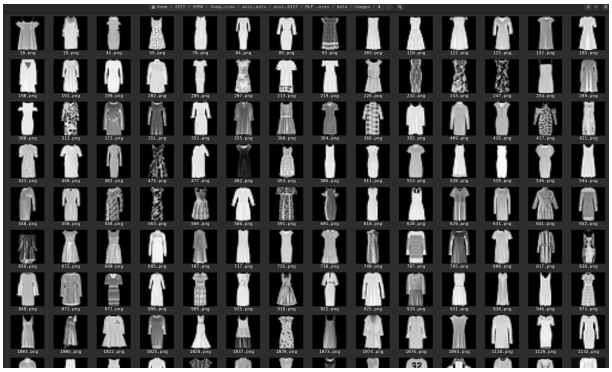

Figure 4: class 3

- **4** jackets
- **5** sandals and slippers
- **6** t shirts
- **7** sneakers
- **8** bags
- **9** formal shoes

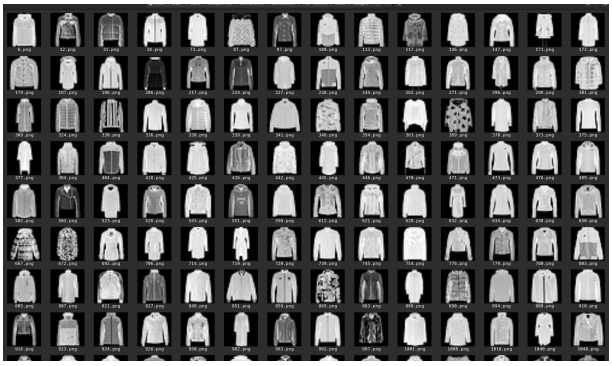

Figure 5: class 4

Figure 6: class 5

Figure 7: class 6

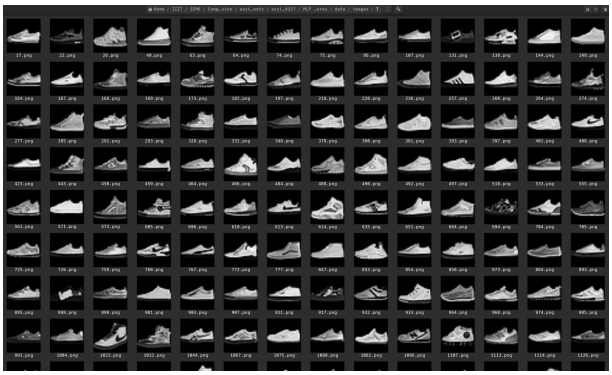

Figure 8: class 7

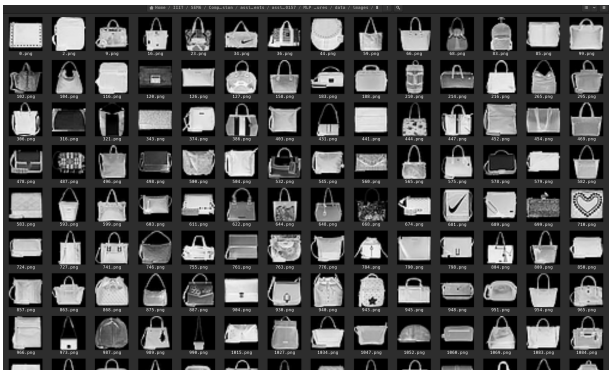

Figure 9: class 8

Figure 10: class 9

1.2 Plots and Performance Visualisations

The default values of the parameters chosen are a *learning rate* of $3.5e-5$ and a *dropout rate* of 0.1. These were the parameters that I chose to tune and analyse, as well. The batch size was 16384 and the models were trained for 100 epochs.

The variants of the models are as follows:

1. **Model 1**: *flattened raw images* without additional pre-processing
2. **Model 2**: *edge detection features* using `Canny` for edge detection
3. **Model 3**: *gaussian blurred and histogram equalised* versions of raw images

In addition, I also implemented *hog feature extraction* but did not analyse it heavily.

1.2.1 Learning Rate

The final performance scores were as below:

3e-6

Learning Rate : 3e-6					
Model	test loss	accuracy	recall	precision	f1-score
1	0.9381	0.7102	0.7102	0.7109	0.7000
2	1.3273	0.5790	0.5790	0.5801	0.5679
3	1.1204	0.6838	0.6537	0.6069	0.6537

The confusion matrices and scores can also be visualised as below.

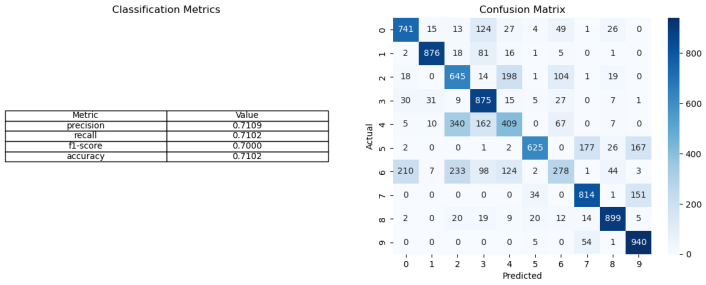

Figure 11: raw features with lr 3e-6

The loss plots on `wandb` for the same are as below.

These plots teach us that such a low learning rate is not sufficient for convergence. The validation loss keeps decreasing, and the train loss even more so significantly, throughout the 100 epochs. Also, we get early glimpses to the in-feasibility of canny edge features for this task, as they do not seem to learn as well. The decrease in losses of models 1 and 3 seem similar, with model 3 having a worse initialisation.

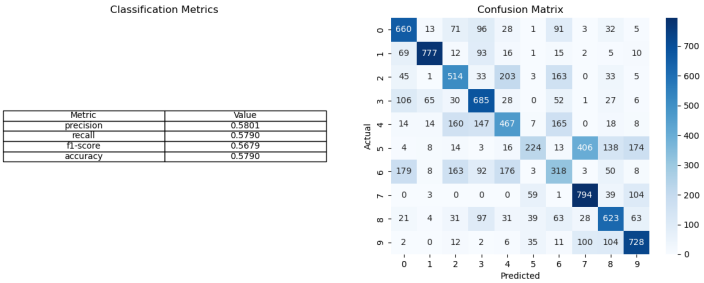

Figure 12: edge detected features with lr 3e-6

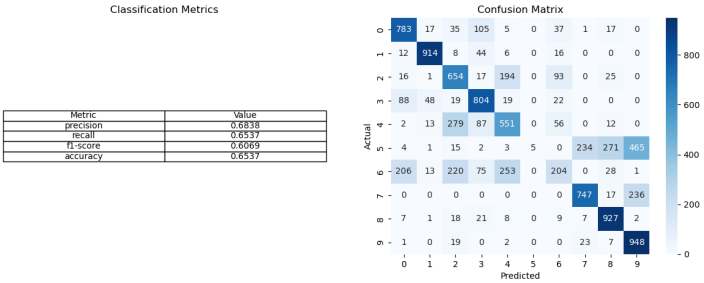

Figure 13: blurred and equalised with lr 3e-6

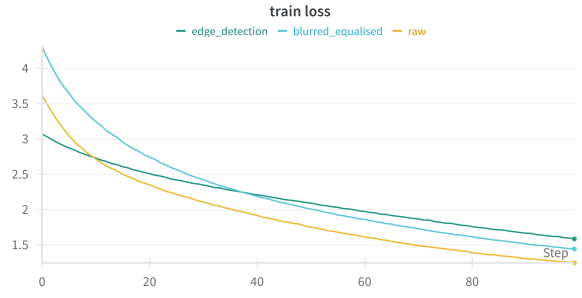

Figure 14: training loss with lr 3e-6

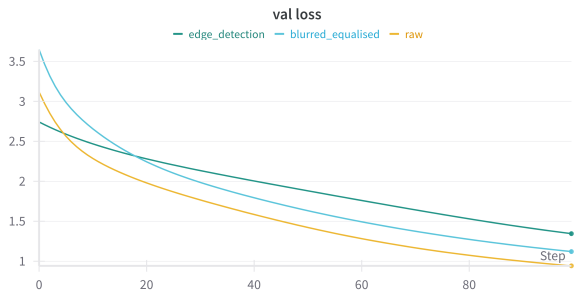

Figure 15: validation loss with lr 3e-6

3.5e-5

Learning Rate : 3.5e-5					
Model	test loss	accuracy	recall	precision	f1-score
1	0.37902	0.8699	0.8699	0.8692	0.8683
2	0.47952	0.8284	0.8283	0.8268	0.8264
3	0.37785	0.8643	0.8643	0.8633	0.8628
4	0.842769	0.6995	0.6995	0.7003	0.6924

The confusion matrices and scores can also be visualised as below.

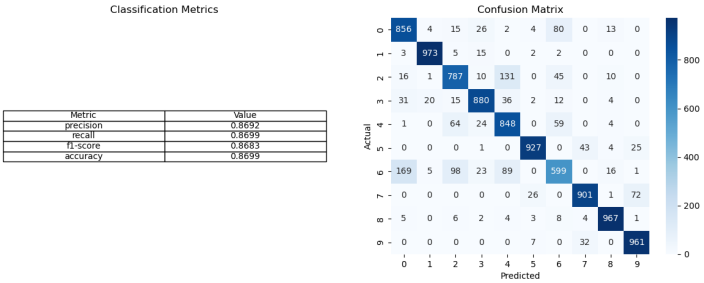

Figure 16: raw features with lr 3.5e-5

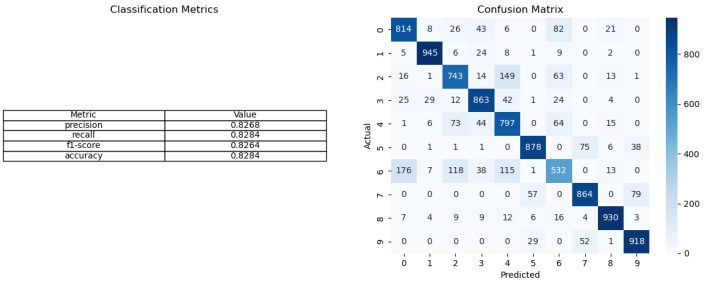

Figure 17: edge detected features with lr 3.5e-5

The loss plots on `wandb` for the same are as below:

These plots suggest we are arriving close to a good learning rate for the models given raw or blurred input features as the loss seemingly stabilises without much overfitting. Edge detection features are not sufficient for good convergence, seemingly, as their learning drastically slows at a loss higher than that of models 1 and 3. Hog features are not good at all (atleast with the current parameters) and do not yield good performance.

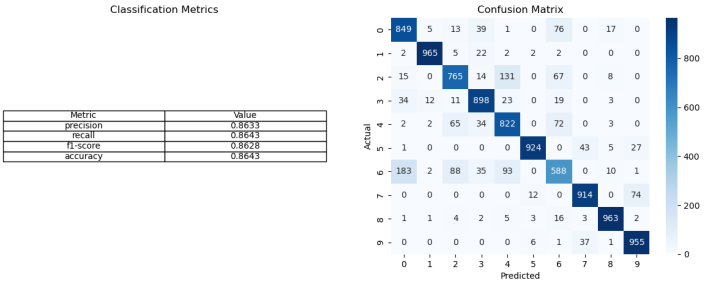

Figure 18: blurred and equalised with lr 3.5e-5

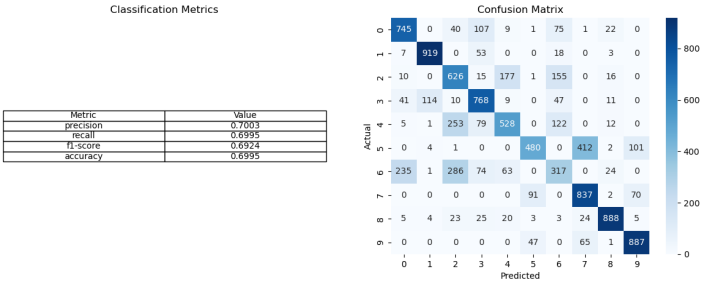

Figure 19: hog features with lr 3.5e-5

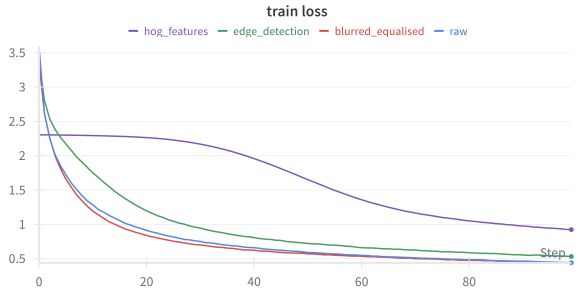

Figure 20: training loss with lr 3.5e-5

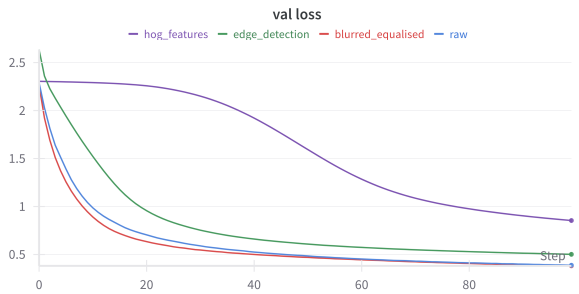

Figure 21: validation loss with lr 3.5e-5

5e-4

Learning Rate : 5e-4					
Model	test loss	accuracy	recall	precision	f1-score
1	0.29441	0.8960	0.8960	0.8957	0.8954
2	0.402709	0.8559	0.8559	0.8580	0.8565
3	0.283528	0.8965	0.8965	0.8985	0.8972
4	0.372340	0.8649	0.8649	0.8637	0.8633

The confusion matrices and scores can also be visualised as below.

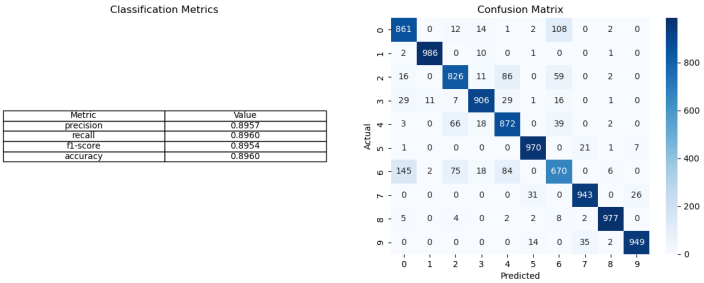

Figure 22: raw features with lr 5e-4

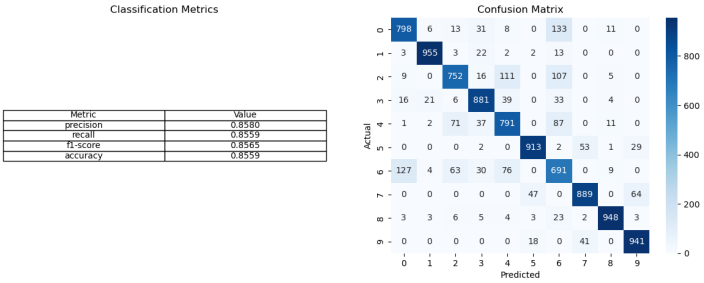

Figure 23: edge detected features with lr 5e-4

The loss plots on `wandb` for the same are as below:

We see very good performance of models 1 and 3, but the model begins to overfit on the canny edge detected features. The HOG features perform better than model 2, but train slower than the raw and blurred features and they come with a tradeoff of being considerably slower than the other three, and are not a viable option as a result.

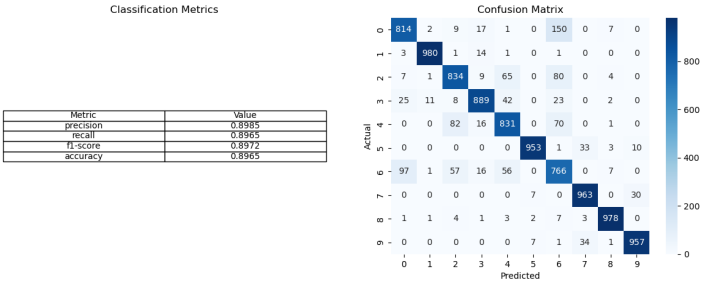

Figure 24: blurred and equalised with lr 5e-4

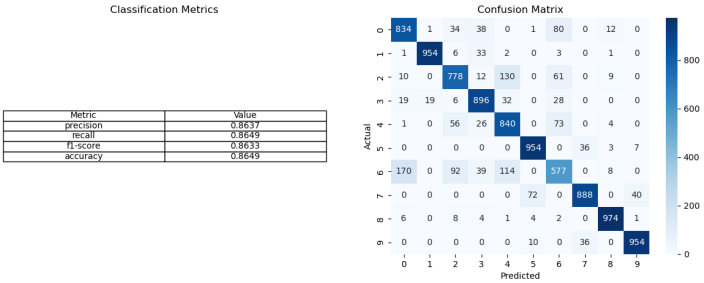

Figure 25: hog features with lr 5e-4

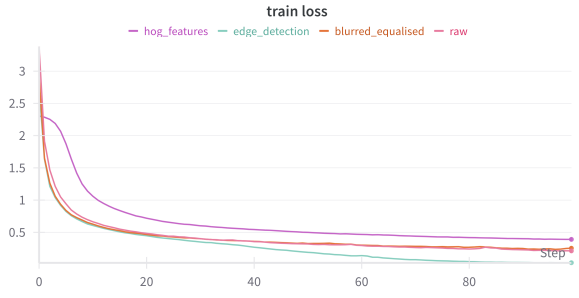

Figure 26: training loss with lr 5e-4

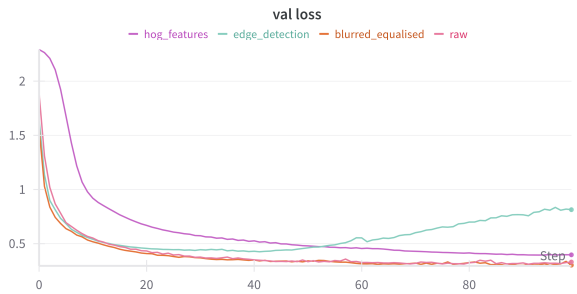

Figure 27: validation loss with lr 5e-4

Final Conclusion: if we seek to avoid overfitting of all three models with the same learning rate, then something of the order of 10^{-5} is best. To get the best out of the raw and blurred features however, we can increase the learning rate to the order of 10^{-4}.

Class 6 is difficult for the model to classify, but raw and blurred features represent it best, and the latter gets a decent score on it when we increase the learning rate.

1.2.2 dropout

Dropout of 0.1 with learning rate of $5e-4$ has been represented above. Here, I try to vary the dropout while maintaining this learning rate.

0.0

dropout: 0.0					
Model	test loss	accuracy	recall	precision	f1-score
1	0.30393	0.8904	0.8904	0.8900	0.8889
2	0.41635	0.8498	0.8498	0.8499	0.8491
3	0.29835	0.8919	0.8919	0.8936	0.8917

The confusion matrices and scores can also be visualised as below.

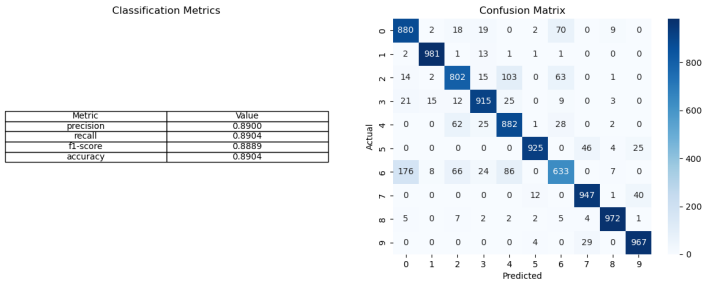

Figure 28: raw features with dropout 0.0

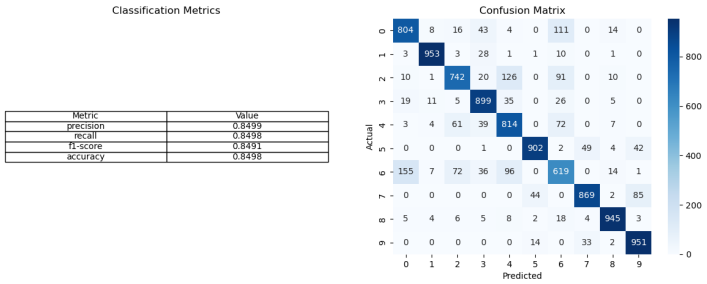

Figure 29: edge detected features with dropout 0.0

The loss plots on `wandb` for the same are as below.

The loss plots show us that introducing some dropout is much better than none. Canny edge features overfit very quickly, and even blurred and raw images begin to have highly fluctuating validation and train loss scores, suggesting difficulty to generalise.

Also, the performance on the most difficult class (6) has degraded as per the confusion matrix.

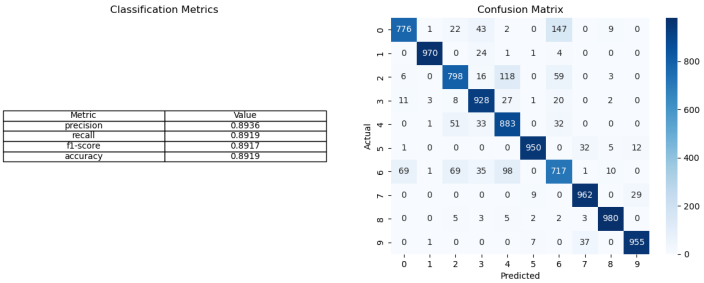

Figure 30: blurred and equalised with dropout 0.0

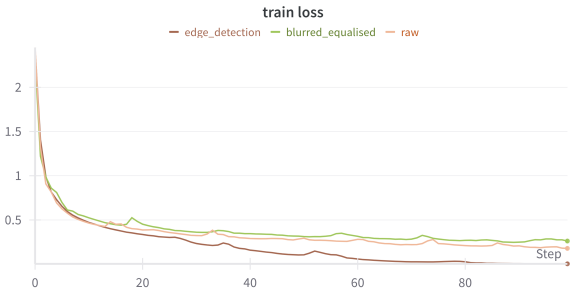

Figure 31: training loss with dropout 0.0

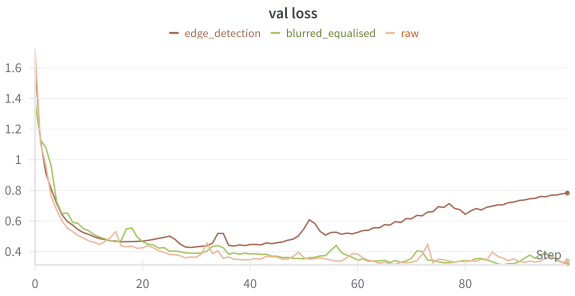

Figure 32: validation loss with dropout 0.0

0.2

dropout: 0.2					
Model	test loss	accuracy	recall	precision	f1-score
1	0.31268	0.8930	0.8930	0.8925	0.8920
2	0.41869	0.8525	0.8525	0.8551	0.8532
3	0.28859	0.8959	0.8959	0.8965	0.8961

The confusion matrices and scores can also be visualised as below.

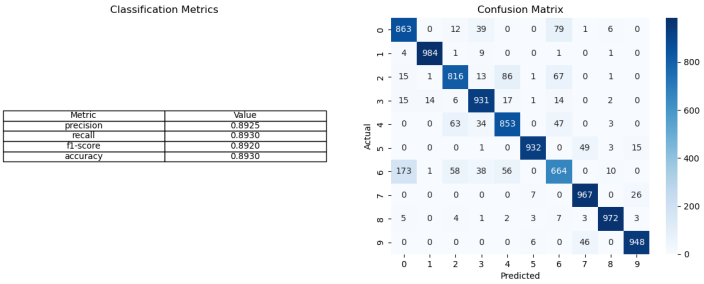

Figure 33: raw features with dropout 0.2

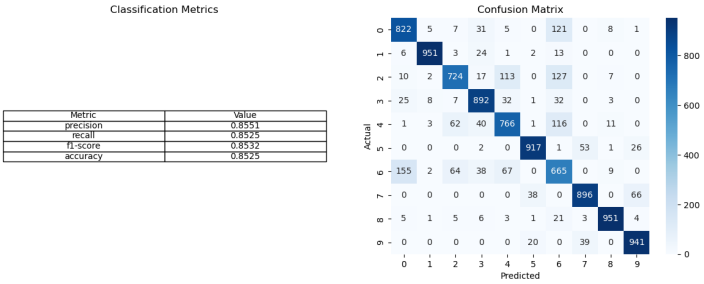

Figure 34: edge detected features with dropout 0.2

The loss plots on `wandb` for the same are as below.

The loss curves show that the overfitting of the edge detected features has slowed considerably, with a general trend of decrease of validation loss for models 1 and 3 throughout. Thus, I believe the ideal dropout (for these models and this task) is somewhere closer to 0.2 than 0.1, likely slightly above it. This suggests that the model is aided by increased generalisability, and that there is good variation between train and validation sets. The increased test scores also affirm this conclusion.

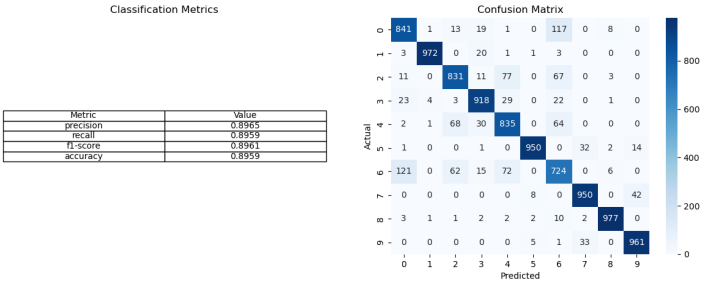

Figure 35: blurred and equalised with dropout 0.2

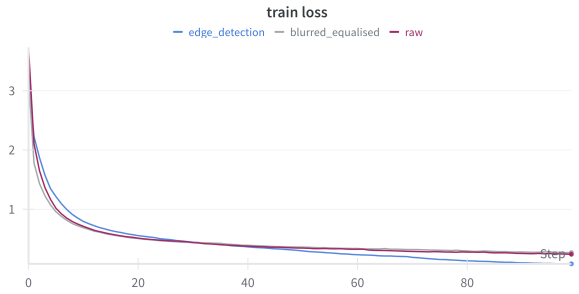

Figure 36: training loss with dropout 0.2

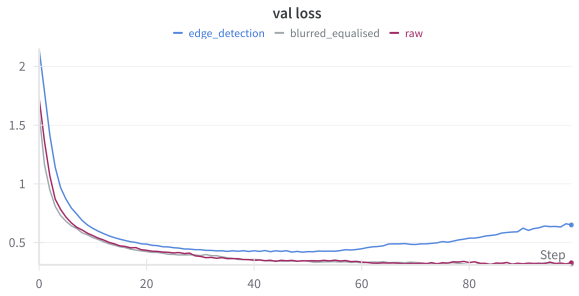

Figure 37: validation loss with dropout 0.2

1.3 Analysis and Logical Comparison

In general, canny features and hog features do not have the capacity to accurately encode the information about each image such that the model can accurately decode the class it belongs to. Raw and blurred images maintain a lot more information about the input, which is important for a classification task such as this one where a number of different classes may have a similar structure but differ, for example: classes 2, 4 and 6. In general, for this classification task, the "body" of the items (the part within the boundaries) seems to matter just as much as the shape, leading to feature extraction methods such as canny and hog losing key information and context.

Hog features take a lot of time to extract and bring the training time all the way from about 2 to 3 minutes (for raw images) to about 90 minutes. Thus, the fact that they do not yield better results puts them out of consideration for the best features for this task. Hog features seem to be good for object detection, which is not a key requirement for our classification.

Canny features overfit quickly since they cannot adequately represent each input class distinctly, and begin to attempt to fit the training set instead of generalising by learning the structure of the input images. Raw images and blurred images have minimal loss of detail, and seem to continue learning even over 100 epochs - they generalise well with good, and comparable, scores on the test set. Blurring and equalising performs very slightly better, as a matter of fact, likely because of the slightly decreased noise and 'brighter' representation of images.

Raw and blurred features perform best when every part of the input image is important for the classification task at hand, for example, where classes are quite similar. Canny features may be useful if the overall structure (eg: just the outline drawing) of classes were consistently different.