MPhys Lab Book Semester 2

James Harbon

Project Partner: Joshua Heaton

**Week 1 (9/2/21 - 16/2/21)**

**1.1 Timeline/Plan for the Semester**

**Diagram

Description automatically generated**

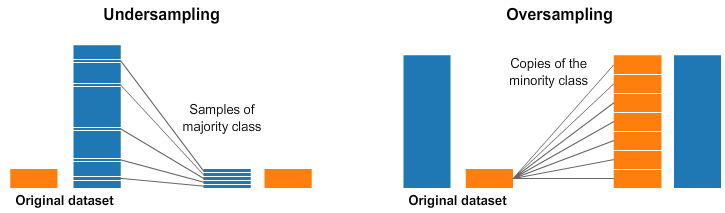
**1.2 Systematic Approach**

We will attempt to produce results with a more systematic approach this semester. This is because we would like to minimise any ambiguity in our methods and results. We (myself and Josh) will share the approach for any independent work we conduct during the semester since we can then share all our results with each other at the end and hence improve productivity during the semester.

The same dataset will be used by both of us for a given classification task. Also, we will use the same training configurations as each other for each given task and we will potentially use the same configurations for any task.

**1.3 Methods for Dealing with Unbalanced Datasets**

Last semester we observed that the datasets used for the classification of different groups of smectic phases suffered from class imbalance. There were significant imbalances for the smectic 3 phases and smectic 6 phases tasks. In the former task, the best mean test accuracy achieved was 86%, however this was inflated due to the same class imbalances in the test set. In the latter task, the best accuracy was 54%. Despite the fact that we will expand our dataset this semester, we expect that some class imbalances will still exist. Clearly, we need to find methods for dealing with the imbalance problem. Also, since a general test set won’t necessarily have the same class distribution as our test sets, we need to find metrics which give a better representation of how our models will perform on novel data.

Two simple methods for dealing with class imbalances are under-sampling and over-sampling. The former involves reducing the number of samples used from the overrepresented classes to improve the balance and was already used last semester when a new dataset for each of the smectic 3 and 6 phases tasks was created. The mean test accuracy for every model was observed to decrease, which could have been due to an inadequate amount of data to learn all the meaningful features in the overrepresented classes. Over-sampling involves creating duplicates of the samples in the underrepresented classes to improve the balance. This method has not been used yet and could be viable since the overrepresented classes were observed to contain a lot of images which look similar to each other. The duplicates could be created as simple copies or we could potentially generate images, using a GAN, which look similar to those in the underrepresented classes. Over-sampling is a lot more promising than under-sampling, since with the former we can both retain all of the data in the overrepresented classes and improve the balance between all classes.

The specific iteration of a model chosen after training for some number of epochs is usually the one which corresponds to the lowest validation loss score. This is because the loss is indicative of the confidence of a model’s predictions. The loss score for an epoch is computed using the samples in the validation set, and hence overrepresented classes have a greater weight on the loss score. Due to the logarithmic relation in the cross-entropy formula, lower confidence in predictions is penalised disproportionately compared to higher confidence. This can give greater weight to a poor performance on underrepresented classes, however a large enough class imbalance will override this effect. One method of dealing with this problem is to manually alter the weights given to each class during the computation of the loss function. The weights of the underrepresented and overrepresented classes can be respectively increased and decreased to effectively balance the classes in the eyes of the loss function. Giving greater weight to the underrepresented classes will force the model to improve its performance on those classes and/or give us a fairer representation of how confident the model is for making predictions on novel data.

Manually altering the weights of different classes for the loss function can be tricky to get right. We can instead use a different loss function, called focal loss, which can automate the process of choosing weights for sample images. This loss function “down-weights” the samples which were correctly classified with a high confidence, since the performance of the model on these samples is already very good and hence there isn’t much room for improvement left. The attention is turned towards the samples which were either incorrectly classified or correctly classified with a confidence close to 0.5, which is helpful since the model will be forced to improve performance on these samples to further reduce the loss value. Training emphasis is effectively placed on the samples which are harder to correctly classify.

Diagram

Description automatically generated

**Week 2 (16/2/21 - 23/2/21)**

**2.1 New Data and Extension of Smectic 3 Phases Dataset**

Our supervisor Ingo sent us a collection of videos which contained LC phases and transitions. There were 19 videos in total which contained smectic A, smectic C, smectic F, smectic I and cholesteric phases. We extracted the frames from the videos and saved them as images via VLC’s scene filter function. Since the video file names contained information about the sequence of phases which occurred, we could identify the phases in the images based on where the phase transition occurred in the video. The images were sorted into corresponding folders for different phases and then they were each split into six sub-images before they were resized to a resolution of 256x256.

The first dataset we decided to augment with the new data was the smectic 3 phases dataset. The new dataset is shown below.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Phase | Total | Train | Valid | Test |
| Fluid | 3204 | 2047 | 576 | 581 |
| Hexatic | 2226 | 1338 | 474 | 414 |
| Soft Crystal | 840 | 600 | 144 | 96 |

Chart, bar chart

Description automatically generatedBelow is a visualisation which compares our new dataset to the corresponding one from last semester.

We can see that the number of fluid images increased, the number of hexatic images greatly increased, the number of soft crystal (SC) images remained constant and the balance between fluid and hexatic significantly improved.

**2.2 First Results**

**Chart, treemap chart

Description automatically generated**Four different architectures were trained and tested on the new dataset. The averages and their uncertainty values were computed over three repeats.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 3 Layers | 4 Layers | Inception 3M | ResNet50 |
| Validation Acc R1/% | 56 | 51 | 60 | 68 |
| Validation Acc R2/% | 61 | 51 | 57 | 55 |
| Validation Acc R3/% | 60 | 50 | 62 | 64 |
| Validation Acc Mean/% | 59 | 51 | 60 | 62 |
| Validation Acc Uncertainty/% | 2.5 | 0.5 | 2.5 | 6.5 |
| Test Acc R1/% | 63 | 67 | 62 | 70 |
| Test Acc R2/% | 81 | 53 | 60 | 59 |
| Test Acc R3/% | 83 | 78 | 69 | 71 |
| Test Acc Mean/% | 76 | 66 | 64 | 67 |
| Test Acc Uncertainty/% | 10 | 12.5 | 4.5 | 6 |

**Chart, treemap chart

Description automatically generated**

All three architectures performed well on the hexatic class. “3 Layers” also performed well on the fluid class, but mediocrely on the soft crystal (SC) class. ResNet50 and Inception had a performance on fluid and SC which was opposite to that of “3 Layers”.

**Week 3 (23/2/21 - 2/3/21)**

**3.1 Tuning Model Hyperparameters**

The “No Free Lunch Theorem” (NFL) states that, within certain constraints, the average performance of each optimisation algorithm over all possible problems is the same. Due to the close link between ML algorithms and their corresponding optimisation techniques, this implies that there is no ML algorithm which is the best for any problem. We hence must experiment through trial and error (systematic methods exist to carry this out) to find the best model for a given problem. For neural networks, one of the most common methods for tweaking the architecture (for a given problem) is hyperparameter tuning. Hyperparameters can define an architecture and how it learns. Examples are number of layers, number of units in the layers, learning rate and batch size. For now, due to time constraints, we will tune only the batch size and learning rate values.

**3.2 “3 Layers” Model Tuning**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 3 Layers, Batch Size = 8 | | | | |
| Learning Rate | Val Accuracy R1/% | Val Accuracy R2/% | Val Accuracy R3/% | Mean Val Accuracy/% |
| 0.1 | 48 | 48 | 48 | 48 |
| 0.01 | 68 | 66 | 67 | 67 |
| 0.001 | 66 | 38 | 45 | 50 |
| 0.0001 | 66 | 57 | 63 | 62 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 3 Layers, Batch Size = 16 | | | | |
| Learning Rate | Validation Accuracy R1/% | Validation Accuracy R2/% | Validation Accuracy R3/% | Mean Validation Accuracy/% |
| 0.1 | 48 | 48 | 48 | 48 |
| 0.01 | 59 | 61 | 51 | 57 |
| 0.001 | 73 | 75 | 62 | 70 |
| 0.0001 | 59 | 63 | 64 | 62 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 3 Layers, Batch Size = 32 | | | | |
| Learning Rate | Validation Accuracy R1/% | Validation Accuracy R2/% | Validation Accuracy R3/% | Mean Validation Accuracy/% |
| 0.1 | 48 | 48 | 48 | 48 |
| 0.01 | 41 | 48 | 65 | 51 |
| 0.001 | 49 | 53 | 45 | 49 |
| 0.0001 | 64 | 60 | 65 | 63 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 3 Layers, Batch Size = 64 | | | | |
| Learning Rate | Validation Accuracy R1/% | Validation Accuracy R2/% | Validation Accuracy R3/% | Mean Validation Accuracy/% |
| 0.1 | 48 | 48 | 48 | 48 |
| 0.01 | 40 | 37 | 65 | 47 |
| 0.001 | 58 | 42 | 48 | 49 |
| 0.0001 | 48 | 48 | 48 | 48 |

A batch size of 16 and a learning rate of 0.001 produced the optimal performance for the “3 Layers” architecture.

**3.2 ResNet50 Tuning**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ResNet50, Batch Size = 8 | | | | |
| Learning Rate | Val Accuracy R1/% | Val Accuracy R2/% | Val Accuracy R3/% | Mean Val Accuracy/% |
| 0.1 | 48 | 48 | 48 | 48 |
| 0.01 | 59 | 67 | 64 | 63 |
| 0.001 | 54 | 66 | 59 | 60 |
| 0.0001 | 51 | 57 | 61 | 56 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ResNet50, Batch Size = 16 | | | | |
| Learning Rate | Validation Accuracy R1/% | Validation Accuracy R2/% | Validation Accuracy R3/% | Mean Validation Accuracy/% |
| 0.1 | 48 | 48 | 48 | 48 |
| 0.01 | 48 | 58 | 64 | 57 |
| 0.005 | 56 | 63 | 51 | 57 |
| 0.001 | 67 | 66 | 64 | 66 |
| 0.0005 | 50 | 50 | 53 | 51 |
| 0.0001 | 61 | 63 | 63 | 62 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ResNet50, Batch Size = 32 | | | | |
| Learning Rate | Val Accuracy R1/% | Val Accuracy R2/% | Val Accuracy R3/% | Mean Val Accuracy/% |
| 0.1 | 48 | 48 | 48 | 48 |
| 0.01 | 60 | 60 | 49 | 56 |
| 0.001 | 39 | 63 | 59 | 54 |
| 0.0001 | 51 | 54 | 53 | 53 |

Batch size of 64 was left out since the performance appeared to only worsen on average when the batch size was larger than 16.

A batch size of 16 and learning rate of 0.001 produced the optimal performance for the ResNet50 architecture.

**3.3 Inception 3M Tuning**

The full inception architecture was considered as a possible variation we could use. For this reason, I henceforth referred to the variation with only 3 Inception modules as “Inception 3M”.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Inception 3M, Batch Size = 8 | | | | |
| Learning Rate | Val Accuracy R1/% | Val Accuracy R2/% | Val Accuracy R3/% | Mean Val Accuracy/% |
| 0.1 | 48 | 48 | 48 | 48 |
| 0.01 | 48 | 48 | 48 | 48 |
| 0.001 | 45 | 48 | 48 | 47 |
| 0.0001 | 60 | 72 | 54 | 62 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Inception 3M, Batch Size = 16 | | | | |
| Learning Rate | Val Accuracy R1/% | Val Accuracy R2/% | Val Accuracy R3/% | Mean Val Accuracy/% |
| 0.1 | 48 | 48 | 48 | 48 |
| 0.01 | 48 | 48 | 48 | 48 |
| 0.001 | 48 | 48 | 48 | 48 |
| 0.0001 | 63 | 60 | 60 | 61 |
| 0.00005 | 59 | 53 | 61 | 60 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Inception 3M, Batch Size = 32 | | | | |
| Learning Rate | Val Accuracy R1/% | Val Accuracy R2/% | Val Accuracy R3/% | Mean Val Accuracy/% |
| 0.1 | 48 | 48 | 48 | 48 |
| 0.01 | 48 | 48 | 48 | 48 |
| 0.001 | 48 | 48 | 48 | 48 |
| 0.0001 | 60 | 55 | 56 | 57 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Inception 3M, Batch Size = 64 | | | | |
| Learning Rate | Val Accuracy R1/% | Val Accuracy R2/% | Val Accuracy R3/% | Mean Val Accuracy/% |
| 0.1 | 48 | 48 | 48 | 48 |
| 0.01 | 48 | 48 | 48 | 48 |
| 0.001 | 48 | 48 | 48 | 48 |
| 0.0001 | 59 | 62 | 60 | 60 |

The highest mean accuracy of 62% was achieved by a batch size of 8 and a learning rate of 0.0001. However this configuration produced a much higher variance than the configuration which achieved the second highest of 61%. The latter configuration consisted of a batch size of 16 and a learning rate of 0.0001. This configuration was selected for future training of Inception 3M.

**Week 4 (2/3/21 - 9/3/21)**

|  |  |  |  |
| --- | --- | --- | --- |
|  | 3 Layers | ResNet50 | Inception 3M |
| Val Accuracy R1/% | 60 | 64 | 62 |
| Val Accuracy R2/% | 59 | 66 | 64 |
| Val Accuracy R3/% | 65 | 65 | 59 |
| Val Accuracy R4/% | 72 | 62 | 66 |
| Val Accuracy R5/% | 66 | 61 | 65 |
| Val Accuracy R6/% | 71 | 71 | 61 |
| Val Accuracy R7/% | 57 | 70 | 64 |
| Val Accuracy R8/% | 75 | 68 | 62 |
| Val Accuracy R9/% | 65 | 61 | 60 |
| Val Accuracy R10/% | 59 | 67 | 60 |
| Mean Val Acc/% | 65 | 66 | 62 |
| Val Accuracy Uncertainty/% | 6.2 | 3.6 | 2.4 |
| Test Accuracy R1/% | 67 | 77 | 59 |
| Test Accuracy R2/% | 73 | 66 | 55 |
| Test Accuracy R3/% | 77 | 69 | 50 |
| Test Accuracy R4/% | 79 | 71 | 64 |
| Test Accuracy R5/% | 67 | 69 | 58 |
| Test Accuracy R6/% | 88 | 71 | 67 |
| Test Accuracy R7/% | 67 | 79 | 51 |
| Test Accuracy R8/% | 66 | 77 | 53 |
| Test Accuracy R9/% | 76 | 71 | 64 |
| Test Accuracy R10/% | 84 | 79 | 59 |
| Mean Test Accuracy/% | 74 | 73 | 58 |
| Test Accuracy Uncertainty/% | 7.8 | 4.7 | 5.8 |

Due to the variance-accuracy trade off which has been discussed above in the context of model selection, ResNet50 with its newly tuned hyperparameters was identified as the best model for this task.

We created a script to compute the individual class accuracy and uncertainty values and then display them in two confusion matrices.

Chart, line chart

Description automatically generated

**Chart, waterfall chart

Description automatically generated**

**Chart, waterfall chart

Description automatically generated**

The two architectures with the best test accuracy averages, 3 Layers and ResNet50, both had a mediocre performance (65-70%) on the fluid class and a very good performance (85-95%) on the hexatic class. There was a difference of 54% between their SC class accuracy values. The greatest error for both models originated from a misclassification of SC as fluid. As can be seen in the uncertainty confusion matrices, the greatest uncertainty for both models was contained in these two elements of the matrices.

After a discussion with our supervisor, we decided to remove the SC class from future classifiers involving smectic classes. This is due to a small amount of SC data relative to other classes in our overall dataset and also because SC generally isn’t an interesting class in research settings.

**Week 5 (9/3/21 - 16/3/21)**

We decided to investigate three binary classification problems: Smectic A and C, smectic F and I and cholesteric and mixed smectic. Also, a cholesteric-fluid smectic-hexatic smectic classifier was investigated too. We picked these problems because we have a lot of data for these phases and we were interested to see if our networks could achieve very high test accuracy values for these datasets.

The models again needed to be tuned, however this time we kept the batch size either constant or in a narrower range since we found a batch size of 16 as optimal for all of the networks in the previous task. Also, we decided to carry out 10 repeats for each learning rate for a given batch size to more reliably determine the best hyperparameters. Due to the higher number of repeats, the uncertainty was computed using the sample standard deviation instead of half of the spread.

**5.1 Smectic A and C**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Phase | Total | Train | Valid | Test |
| Smectic A | 1106 | 722 | 180 | 204 |
| Smectic C | 2000 | 1388 | 301 | 311 |

| ResNet50, Batch Size = 16 | | | | |
| --- | --- | --- | --- | --- |
| Learning Rate | Mean Val Accuracy/% | Val Accuracy STD/% | Mean Test Accuracy/% | Test Accuracy STD/% |
| 0.0001 | 92 | 3.6 | 92 | 6.9 |
| 0.00005 | 92 | 2.4 | 92 | 5.8 |
| 0.00001 | 90 | 1.0 | 91 | 2.2 |

**5.2 Smectic F and I V1**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Phase | Total | Train | Valid | Test |
| Smectic F | 900 | 570 | 162 | 168 |
| Smectic I | 1326 | 918 | 210 | 198 |

**Chart, bar chart

Description automatically generated**

| 3 Layers, Start Filters = 32, Batch Size = 16 | | | | |
| --- | --- | --- | --- | --- |
| Learning Rate | Mean Val Accuracy/% | Val Accuracy STD/% | Mean Test Accuracy/% | Test Accuracy STD/% |
| 0.005 | 80 | 5.9 | 39 | 8.1 |
| 0.001 | 70 | 5.3 | 57 | 11 |
| 0.0005 | 77 | 6.5 | 58 | 12 |
| 0.0001 | 88 | 6.3 | 51 | 6.9 |

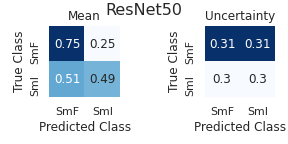
Interestingly, there was a significant difference between the mean validation and mean test accuracy values for the 3 Layers network. The uncertainties in the test accuracies were also large.

| ResNet50, Batch Size = 16 | | | | |
| --- | --- | --- | --- | --- |
| Learning Rate | Mean Val Accuracy/% | Val Accuracy STD/% | Mean Test Accuracy/% | Test Accuracy STD/% |
| 0.005 | 54 | 10.8 | 61 | 10 |
| 0.001 | 57 | 8.2 | 57 | 14 |
| 0.0005 | 59 | 6.8 | 53 | 5.6 |
| 0.0001 | 53 | 11 | 51 | 4.3 |

| Inception, Batch Size = 16 | | | | |
| --- | --- | --- | --- | --- |
| Learning Rate | Mean Val Accuracy/% | Val Accuracy STD/% | Mean Test Accuracy/% | Test Accuracy STD/% |
| 0.001 | 56 | 0.5 | 54 | 0.5 |
| 0.0005 | 56 | 1 | 55 | 1 |
| 0.0001 | 57 | 1 | 54 | 1 |
| 0.00005 | 56 | 1 | 56 | 1 |

Chart

Description automatically generated with medium confidence



**Week 6 (16/3/21 - 23/3/21)**

**6.1 Cholesteric and Smectic**

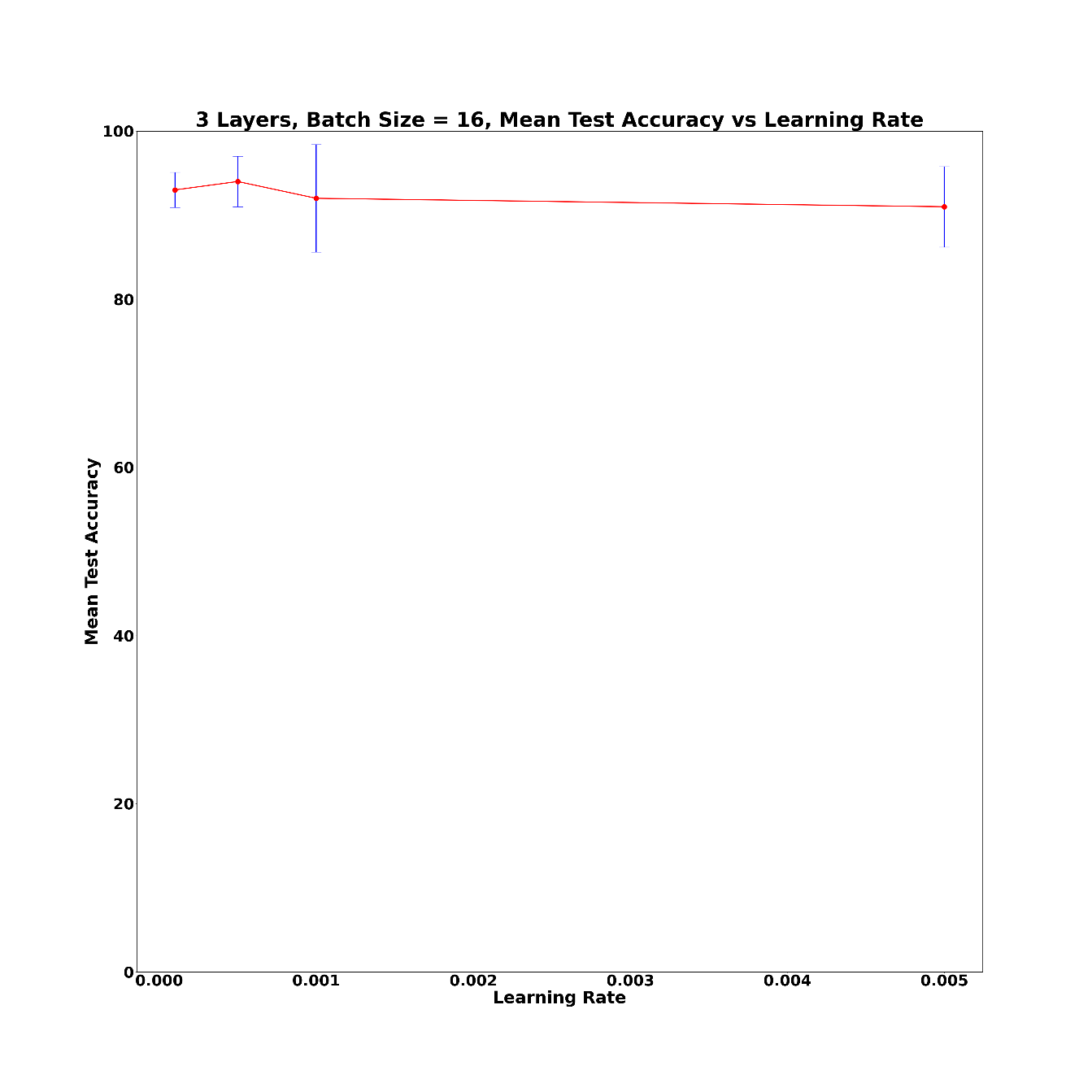
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Phase | Total | Train | Valid | Test |
| Cholesteric | 1646 | 1148 | 245 | 253 |
| Smectic | 5332 | 3598 | 853 | 881 |

**Chart, bar chart

Description automatically generated**

| 3 Layers, Start Filters = 32, Batch Size = 16 | | | | |
| --- | --- | --- | --- | --- |
| Learning Rate | Mean Val Accuracy/% | Val Accuracy STD/% | Mean Test Accuracy/% | Test Accuracy STD/% |
| 0.005 | 89 | 2.6 | 91 | 4.8 |
| 0.001 | 89 | 3.1 | 92 | 6.4 |
| 0.0005 | 91 | 3.6 | 94 | 3.0 |
| 0.0001 | 91 | 2.1 | 93 | 2.1 |

The model trained with a learning rate of 0.0001 was identified as the most successful due to achieving the lowest uncertainties on the validation (2.1%) and test (2.1%) sets, joint highest mean accuracy (91%) for the validation set and a mean test accuracy (93%) only slightly behind the highest of (94%).

****

**Chart, waterfall chart

Description automatically generated**

| 3 Layers, Start Filters = 64, Batch Size = 16 | | | | |
| --- | --- | --- | --- | --- |
| Learning Rate | Mean Val Accuracy/% | Val Accuracy STD/% | Mean Test Accuracy/% | Test Accuracy STD/% |
| 0.0005 | 91 | 3.1 | 96 | 3.7 |
| 0.0001 | 92 | 2.5 | 95 | 1.3 |
| 0.00005 | 93 | 1.4 | 96 | 2.8 |

| ResNet50, Batch Size = 16 | | | | |
| --- | --- | --- | --- | --- |
| Learning Rate | Mean Val Accuracy/% | Val Accuracy STD/% | Mean Test Accuracy/% | Test Accuracy STD/% |
| 0.0005 | 89 | 3.7 | 86 | 3.9 |
| 0.0001 | 91 | 1.6 | 93 | 3.0 |
| 0.00005 | 89 | 2.8 | 92 | 5.0 |
| 0.00001 | 84 | 2.1 | 81 | 2.2 |

**6.2 Cholesteric, Fluid and Hexatic**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Phase | Total | Train | Valid | Test |
| Cholesteric | 1646 | 1148 | 245 | 253 |
| Fluid Smectic | 3106 | 2110 | 481 | 515 |
| Hexatic Smectic | 2226 | 1488 | 372 | 366 |

**Chart, bar chart

Description automatically generated**

| 3 Layers, Start Filters = 32, Batch Size = 16 | | | | |
| --- | --- | --- | --- | --- |
| Learning Rate | Mean Val Accuracy/% | Val Accuracy STD/% | Mean Test Accuracy/% | Test Accuracy STD/% |
| 0.005 | 76 | 7.0 | 77 | 7.9 |
| 0.001 | 78 | 5.6 | 81 | 6.5 |
| 0.0005 | 78 | 5.5 | 80 | 7.3 |
| 0.0001 | 80 | 1.2 | 83 | 2.0 |
| 0.00005 | 81 | 1.0 | 85 | 1.7 |

Chart, waterfall chart

Description automatically generatedChart

Description automatically generated with low confidence

**Week 7 (23/3/21 - 30/3/21)**

**7.1 More Results for Cholesteric, Fluid and Hexatic**

| 3 Layers, Start Filters = 64, Batch Size = 16 | | | | |
| --- | --- | --- | --- | --- |
| Learning Rate | Mean Val Accuracy/% | Val Accuracy STD/% | Mean Test Accuracy/% | Test Accuracy STD/% |
| 0.005 | 71 | 12 | 74 | 13 |
| 0.001 | 79 | 7.9 | 79 | 7.9 |
| 0.0005 | 78 | 3.4 | 84 | 4.4 |
| 0.0001 | 81 | 2.8 | 84 | 3.0 |
| 0.00005 | 82 | 1.7 | 85 | 1.6 |

| 3 Layers, Start Filters = 128, Batch Size = 16 | | | | |
| --- | --- | --- | --- | --- |
| Learning Rate | Mean Val Accuracy/% | Val Accuracy STD/% | Mean Test Accuracy/% | Test Accuracy STD/% |
| 0.005 | 74 | 6.7 | 76 | 7.2 |
| 0.001 | 74 | 5.7 | 75 | 9.8 |
| 0.0005 | 77 | 5.4 | 80 | 5.8 |
| 0.0001 | 83 | 2.8 | 85 | 2.3 |
| 0.00005 | 81 | 2.4 | 84 | 2.9 |

| ResNet50, Batch Size = 16 | | | | |
| --- | --- | --- | --- | --- |
| Learning Rate | Mean Val Accuracy/% | Val Accuracy STD/% | Mean Test Accuracy/% | Test Accuracy STD/% |
| 0.0005 | 68 | 7.1 | 69 | 7.1 |
| 0.0001 | 66 | 6.4 | 71 | 5.6 |
| 0.00005 | 65 | 3.3 | 65 | 3.7 |
| 0.00001 | 67 | 2.5 | 72 | 2.4 |

**Easter Break (30/3/21 – 13/4/21)**

**Week 8 (13/4/21 - 20/4/21)**

**8.1 Smectic F and I V2**

| 3 Layers, Start Filters = 16, Batch Size = 16 | | | | |
| --- | --- | --- | --- | --- |
| Learning Rate | Mean Val Accuracy/% | Val Accuracy STD/% | Mean Test Accuracy/% | Test Accuracy STD/% |
| 0.005 | 55 | 15 | 51 | 3.4 |
| 0.001 | 65 | 6 | 53 | 3.2 |
| 0.0005 | 62 | 8.5 | 54 | 3.6 |
| 0.0001 | 67 | 0.5 | 53 | 0.8 |
| 0.00005 | 67 | 0.1 | 55 | 0.4 |

| 3 Layers, Start Filters = 32, Batch Size = 16 | | | | |
| --- | --- | --- | --- | --- |
| Learning Rate | Mean Val Accuracy/% | Val Accuracy STD/% | Mean Test Accuracy/% | Test Accuracy STD/% |
| 0.005 | 67 | 1.1 | 62 | 6 |
| 0.001 | 67 | 0.5 | 54 | 0.9 |
| 0.0005 | 63 | 0.3 | 53 | 0.7 |
| 0.0001 | 67 | 0.4 | 53 | 0.5 |
| 0.00005 | 67 | 0.3 | 53 | 0.4 |

| 3 Layers, Start Filters = 64, Batch Size = 16 | | | | |
| --- | --- | --- | --- | --- |
| Learning Rate | Mean Val Accuracy/% | Val Accuracy STD/% | Mean Test Accuracy/% | Test Accuracy STD/% |
| 0.005 | 61 | 12 | 59 | 6.3 |
| 0.001 | 60 | 12 | 52 | 2.2 |
| 0.0005 | 65 | 5.8 | 53 | 0.8 |
| 0.0001 | 64 | 4.7 | 54 | 1.1 |
| 0.00005 | 66 | 3.2 | 53 | 0.9 |

**Week 9 (20/4/21 - 27/4/21)**

**9.1 Smectic F and I V3**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Phase | Total | Train | Valid | Test |
| Smectic F | 630 | 420 | 90 | 120 |
| Smectic I | 762 | 534 | 108 | 120 |

| 3 Layers, Start Filters = 16, Batch Size = 16 | | | | |
| --- | --- | --- | --- | --- |
| Learning Rate | Mean Val Accuracy/% | Val Accuracy STD/% | Mean Test Accuracy/% | Test Accuracy STD/% |
| 0.005 | 84 | 15 | 53 | 25 |
| 0.001 | 90 | 5 | 60 | 22 |
| 0.0005 | 92 | 6 | 63 | 23 |
| 0.0001 | 73 | 12 | 86 | 18 |
| 0.00005 | 55 | 10 | 52 | 12 |

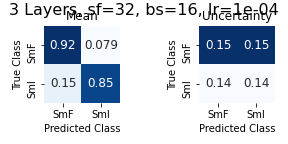
| 3 Layers, Start Filters = 32, Batch Size = 16 | | | | |
| --- | --- | --- | --- | --- |
| Learning Rate | Mean Val Accuracy/% | Val Accuracy STD/% | Mean Test Accuracy/% | Test Accuracy STD/% |
| 0.005 | 85 | 12 | 52 | 30 |
| 0.001 | 90 | 7 | 63 | 29 |
| 0.0005 | 92 | 6.6 | 77 | 19 |
| 0.0001 | 90 | 8.7 | 89 | 8.7 |
| 0.00005 | 66 | 13 | 85 | 15 |

| 3 Layers, Start Filters = 32, Batch Size = 64 | | | | |
| --- | --- | --- | --- | --- |
| Learning Rate | Mean Val Accuracy/% | Val Accuracy STD/% | Mean Test Accuracy/% | Test Accuracy STD/% |
| 0.001 | 57 | 11 | 43 | 17 |
| 0.0005 | 73 | 16 | 41 | 23 |
| 0.0001 | 55 | 5.5 | 50 | 11 |
| 0.00005 | 56 | 6.0 | 49 | 13 |

| 3 Layers, Start Filters = 128, Batch Size = 16 | | | | |
| --- | --- | --- | --- | --- |
| Learning Rate | Mean Val Accuracy/% | Val Accuracy STD/% | Mean Test Accuracy/% | Test Accuracy STD/% |
| 0.001 | 91 | 5.3 | 72 | 25 |
| 0.0005 | 92 | 5.2 | 68 | 24 |
| 0.0001 | 84 | 12 | 93 | 5.6 |

| 3 Layers, Start Filters = 64, Batch Size = 16 | | | | |
| --- | --- | --- | --- | --- |
| Learning Rate | Mean Val Accuracy/% | Val Accuracy STD/% | Mean Test Accuracy/% | Test Accuracy STD/% |
| 0.005 | 85 | 12 | 30 | 25 |
| 0.001 | 87 | 7.3 | 56 | 12 |
| 0.0005 | 88 | 13 | 73 | 17 |
| 0.0001 | 93 | 3.8 | 90 | 11 |
| 0.00005 | 79 | 18 | 76 | 19 |

3 Layers architecture with start filters = 32, batch size = 16, learning rate = 1e-04 achieved a mean test accuracy of (89 +/- 8.7)%

****

| ResNet50, Batch Size = 16 | | | | |
| --- | --- | --- | --- | --- |
| Learning Rate | Mean Val Accuracy/% | Val Accuracy STD/% | Mean Test Accuracy/% | Test Accuracy STD/% |
| 0.001 | 60 | 15 | 48 | 14 |
| 0.0005 | 65 | 12 | 42 | 19 |
| 0.0001 | 62 | 15 | 53 | 5.9 |
| 0.00005 | 62 | 13 | 59 | 15 |
| 0.00001 | 68 | 7.5 | 66 | 12 |

**9.2 Cholesteric and Smectic A, C, F and I**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Phase | Total | Train | Validation | Test |
| Cholesteric | 1646 | 1148 | 245 | 253 |
| Smectic A | 1106 | 722 | 180 | 204 |
| Smectic C | 2000 | 1388 | 301 | 311 |
| Smectic F | 630 | 420 | 90 | 120 |
| Smectic I | 762 | 534 | 108 | 120 |

| 3 Layers, Start Filters = 16, Batch Size = 16 | | | | |
| --- | --- | --- | --- | --- |
| Learning Rate | Mean Val Accuracy/% | Val Accuracy STD/% | Mean Test Accuracy/% | Test Accuracy STD/% |
| 0.0005 | 66 | 3.7 | 77 | 7.4 |
| 0.0001 | 66 | 3.4 | 83 | 8.0 |
| 0.00005 | 65 | 4.3 | 81 | 7.3 |

| 3 Layers, Start Filters = 32, Batch Size = 16 | | | | |
| --- | --- | --- | --- | --- |
| Learning Rate | Mean Val Accuracy/% | Val Accuracy STD/% | Mean Test Accuracy/% | Test Accuracy STD/% |
| 0.0005 | 64 | 6.0 | 74 | 4.6 |
| 0.0001 | 66 | 3.4 | 81 | 5.5 |
| 0.00005 | 65 | 3.5 | 82 | 6.8 |

| 3 Layers, Start Filters = 64, Batch Size = 16 | | | | |
| --- | --- | --- | --- | --- |
| Learning Rate | Mean Val Accuracy/% | Val Accuracy STD/% | Mean Test Accuracy/% | Test Accuracy STD/% |
| 0.0005 | 64 | 5.7 | 75 | 9.0 |
| 0.0001 | 66 | 6.2 | 82 | 4.6 |
| 0.00005 | 65 | 5.3 | 85 | 3.8 |
| 0.00001 | 65 | 3.9 | 81 | 9.4 |

The best configuration was 3 Layers with starting filters = 64, batch size = 16 and learning rate = 5e-05. The mean test accuracy was (85 +/- 3.8)%**Chart

Description automatically generated**

**Graphical user interface, chart

Description automatically generated**The second best configuration was 3 Layers with starting filters = 64, batch size = 16 and learning rate = 1e-04. The mean test accuracy was (82 +/- 4.6)%

**Chart, line chart

Description automatically generated**

| 3 Layers, Start Filters = 128, Batch Size = 16 | | | | |
| --- | --- | --- | --- | --- |
| Learning Rate | Mean Val Accuracy/% | Val Accuracy STD/% | Mean Test Accuracy/% | Test Accuracy STD/% |
| 0.0001 | 67 | 6.1 | 80 | 7.2 |
| 0.00005 | 66 | 5.3 | 83 | 4.8 |
| 0.00001 | 65 | 1.8 | 87 | 3.4 |

| 3 Layers, Start Filters = 128, Batch Size = 64 | | | | |
| --- | --- | --- | --- | --- |
| Learning Rate | Mean Val Accuracy/% | Val Accuracy STD/% | Mean Test Accuracy/% | Test Accuracy STD/% |
| 0.0005 | 68 | 5.7 | 72 | 4.3 |
| 0.0001 | 68 | 3.7 | 83 | 2.7 |
| 0.00005 | 69 | 2.7 | 86 | 3.2 |
| 0.00001 | 66 | 2.2 | 82 | 4.2 |

| ResNet50, Batch Size = 16 | | | | |
| --- | --- | --- | --- | --- |
| Learning Rate | Mean Val Accuracy/% | Val Accuracy STD/% | Mean Test Accuracy/% | Test Accuracy STD/% |
| 0.0005 | 60 | 4.6 | 66 | 6.8 |
| 0.0001 | 58 | 3.9 | 67 | 8.1 |
| 0.00005 | 59 | 3.1 | 67 | 7.9 |

**Weeks 10 - 12 (27/4/21 - 18/5/21)**

The final three weeks were spent finalising the presentation of results (confusion matrices, summary graphs etc…), creating figures for the report and writing the report.

In the report we concluded that we had expanded upon and consolidated the results we produced in the previous semester. Due to the accuracy saturation in the ChSm2 and ChSm4 tasks, and the relatively greater variance in the SmIF task, we believe that the greatest limitation for these tasks is the size of our overall dataset. The creation of an open-source database, where labelled liquid crystal texture images could be contributed by researchers and technicians, would likely further improve the performance of the networks and also provide the possibility for even more ambitious tasks, such as the addition of smectic B and E to our final dataset.