Skin Disease Image Classification - Milestone 3

Manjiri Bhandarwar MB8070@NYU.EDU

Hannah Phung HMP3554@NYU.EDU

Yen Le YTL2008@NYU.EDU

1. Methodology

1.1 Ensemble model: averaging (DenseNet201 and ResNet50)

To improve the performance, we tried creating an ensemble of the DenseNet201 and ResNet50 with the best macro average F1 scores (from milestone 2). The classifiers were trained on the same training data, and their versions with highest validation accuracies were saved using checkpointing. Then they were reloaded and combined into an ensemble. The ensemble's output is the average of their individual models' softmax predictions for each class and the final prediction was made accordingly. Finally, the ensemble was compiled, retrained and tested.

1.2 Hierarchical ensemble (DenseNet201, Siamese Network CNNs)

Since the ensemble with averaging outputs still confused some classes for others (e.g. class 1 as 0), we tried a stacking a hierarchical ensemble. To address the imbalanced dataset problem (refer to appendix A), we partition our model into smaller subproblems, assigning each neural network the responsibility of handling classes with comparable sizes. Most classes were getting confused for class 0 because it has the largest sample size thus the model was overfitting to it. So the first densenet in our hierarchy's job was to distinguish between class 0 and the rest. Also, building upon insights from the milestone, where confusion existed between class 1 and 2, as well as between class 3 and 6, we have chosen to further segment our models into distinct classifiers [see appendix B]. We wanted to see if we can further improve the network's distinguishing power by replacing the bottom most densenets with Siamese network and CNNs to classify for two classes that were still getting confused (i.e. 1 & 2, 3 & 6).

1.3 Multimodal with metadata of age, sex, and localization

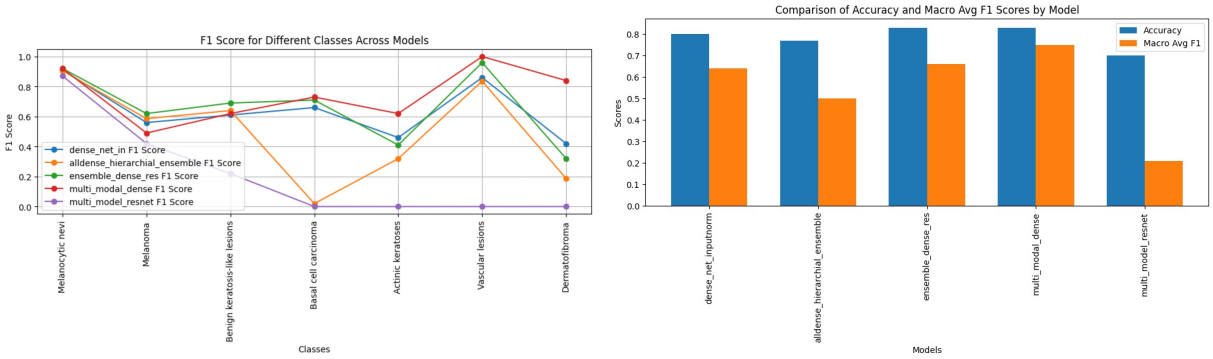
In our study, we pre-processed image data and its corresponding metadata for our model. We chose this approach since some skin diseases could be more prevalent in certain demographics of people (i.e. different ages and sexes). Also, the same disease could appear differently depending on the area on the body (i.e. localization field). Hence, adding metadata would enhance the model's prediction by incorporating the context of images. We shuffled the dataframe to create training and testing sets and reshaped the image data into a three-dimensional numpy array with dimension (125,100,3). We focused on preparing categorical data, particularly scaling the 'age' feature and one-hot encoding 'dx_type' and 'localization', resulting in 20 feature columns for each image. We trained a DenseNet201 model with a Concatenate layer for processing categorical data.To optimize the model, we implemented an Adam optimizer with an ExponentialDecay learning rate schedule. This schedule starts with an initial learning rate of 0.0005, decaying by a factor of 0.9 every 10,000 steps. The model is compiled with the categorical cross entropy loss function, focusing on optimizing accuracy, a common choice for multi-class classification tasks. To enhance robustness and performance, we also incorporate data augmentation techniques. We then plot the training and validation loss along with its respective accuracy, and confusion matrix.

1.4 Testing on secondary dataset

In order to test our model generalizability, we test our model on a secondary dataset: ISIC-2019 skin lesion dataset. This data includes 1616 images with 15 features. Among those features, we only select

images with features that map to features in our HAM10000 dataset, which are “age”,”localization”, “dx_type” and “dx”. Since not all the data from ISIC-2019 match the data in HAM10000, we try our best to match categorical features to each other to best fit the input dimension and value of our model. Since there’s a mismatch in one-hot-encoded features value between the two datasets, we include dummy_coded columns in our categorical data features to match the input size of our trained model.

2. Results



Our results showed significant improvements in model performance with the multi-modal method. The DenseNet201 multi-modal is the best as it achieved an accuracy of 0.83, macro average F1 score is 0.75.

3. Analysis

Hierarchical ensemble could have done better with more fine tuning at individual stages. Since each stage is dependent on the previous one, it is essential to not only tune the last layer to improve the final performance but also the initial layers. However, due to time constraints, we could not try that many approaches. Also, we could have tried a different structure of the pipeline (e.g. have a classifier that only deals with class 3 and 6 due to frequency of misclassification). Before incorporating into the pipeline, we tried out the Siamese model with class 1 and 2 of the training set and used the validation dataset to find the best threshold. However, the best accuracy and f1 score for the validation set was not good (0.50 and 0.67). Hence, we decided to not include this as classifier 3 in the final layer (to classify class 1 and 2). Similar to our approach with the Siamese model, we also tested a basic CNN structure to classify class 3 and 6 before incorporating it into the pipeline. However, the precision of this model for class 6 did not improve compared to our previous models, as the classifier was still mistaken class 6 as 3 (see Appendix K).

For the average ensemble, the accuracy on the test set was 0.83 and the macro average F1 score was 0.66. Both metrics are slightly higher than those of individual DenseNet201 (accuracy: 0.80, macro average F1: 0.64) and ResNet50 (accuracy: 0.81, macro average F1: 0.61). This improvement is consistent with the ability of ensemble methods to outperform individual models due to their ability to combine multiple models to produce a stronger, more robust prediction.

Testing multimodel approach on a secondary dataset two main issues affecting model accuracy. Firstly, the approximate matching of features like disease localization and type between datasets (ISIC-2019 and HAM10000) leads to performance degradation. This mismatch arises because the features do not align perfectly. So by approximately matching features values, we introduce errors in our data. Secondly, to accommodate the model’s input dimensions, dummy columns filled with zeros are introduced for categorical features. This inclusion of potentially irrelevant data (the ’nonsense information’) in the model further compromises its performance by diluting the meaningful information in the dataset.

4. Conclusions

In our project, we employed and refined pretrained models like DenseNet201 and ResNet50, aligning with our initial plan outlined in milestone 1. Nevertheless, in retrospect, we acknowledge that we should have scrutinized the paper more thoroughly to locate the code corresponding to the described architecture. Due to the problem of missing code for the neural network architecture described in our paper, our initial approach of combining model structures was overly ambitious. Unfortunately, due to time constraints, we were unable to construct models from the ground up. We took another approach and followed the advice of building an ensemble model instead of incorporating multiple structures into a single model. This idea showed some promising room for improvements and also helped with the problem of imbalance dataset.

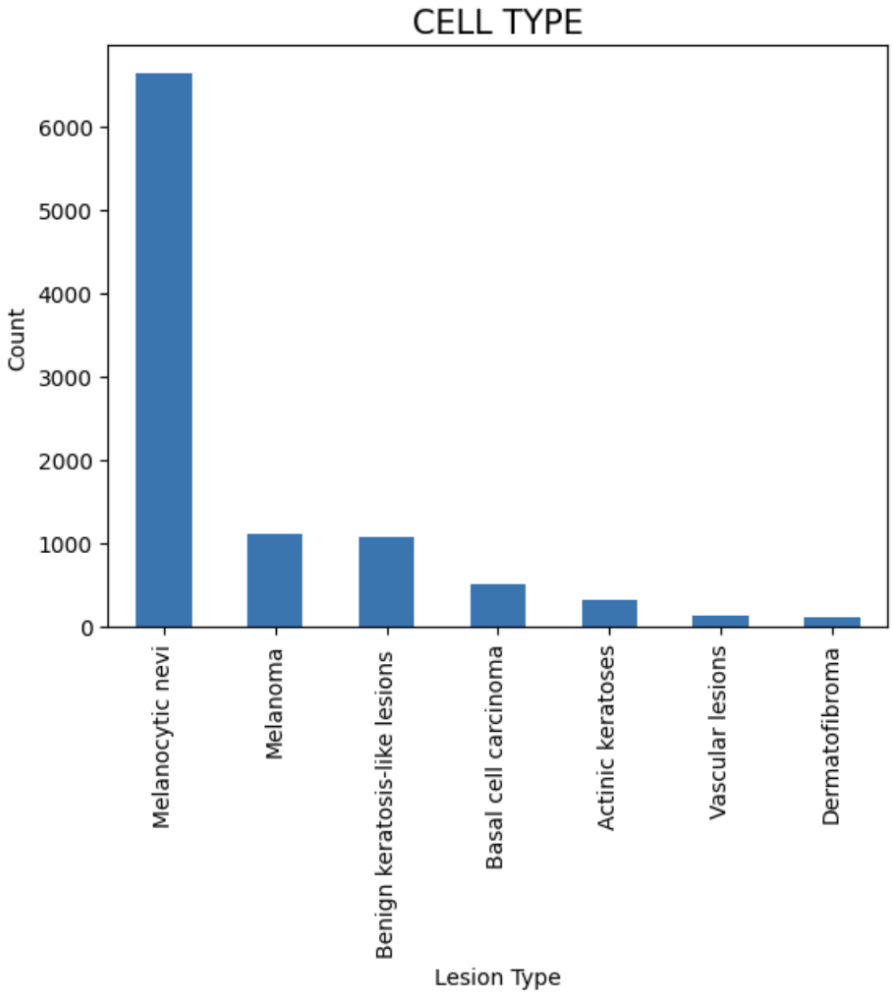
Since the models such as DenseNet201 performed well with binary classification of the majority class in the case of classifying 0 versus the remaining classes, one of the reasons might be the size of our data for some of the classes. For future improvements, we could figure out ways to obtain more data, including getting data from the ISIC portal and requesting access to private datasets such as the one mentioned in one of our papers. However, according to the paper Structural Analysis and Optimization of Convolutional Neural Networks with a Small Sample Size, "optimal network structure was mostly determined by the data nature (photographic, calligraphic, or medical images), and less affected by the sample size." This means that finding a larger training dataset might not suffice to improve our model. In our future work, we could take other approaches on image preprocessing to increase the distinctiveness of each class, such as adjusting saturation and contrast, or find a hair removal software like our initial plan.

Our analysis and results for different models indicates that the model tested perform well for the majority classes but are less effective for the minority classes due to insufficient training data. For this reason, we conclude that the model we introduced can be used to identify the majority classes. In order to improve classification of the minority classes, we suggest collecting more images of the dataset for the minority class and possibly employing more robust data augmentation techniques. This could help the model learn the characteristics of the minority classes better and improve overall classification performance.

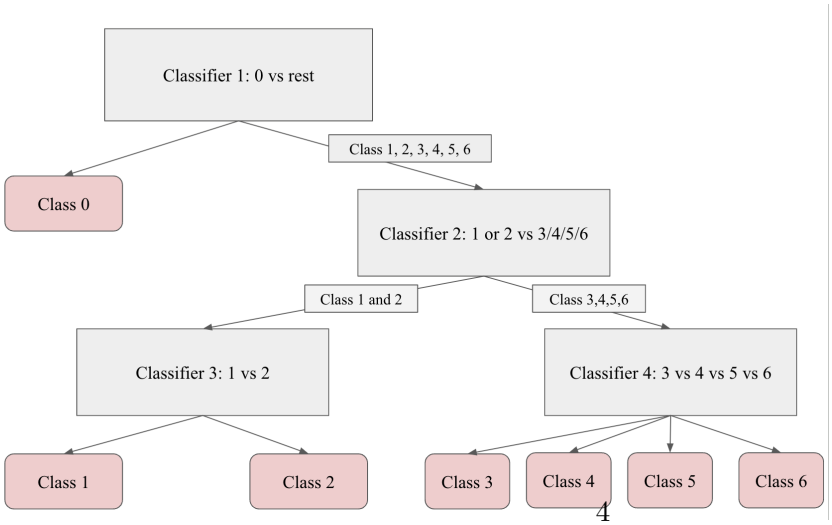
5. Workflow

	Manjiri	Hannah	Yen
Milestone 1	Each read and summarize 2 research paper Came together to compare our findings, then identified common weaknesses and tailored goal and methodology accordingly Find datasets (primary and secondary)		
	Condense finding in report	Set up github and write report	Preliminary data analysis
Milestone 2	Try out different models of DenseNet and ResNet, adding normalization, batch normalization, and drop out		
	Data preprocessing and augmentation	Hyperparameter sweeps	
Milestone 3	Ensemble model: bagging Replacing models in Hannah's pipeline with CNN	Hierarchical ensemble: stacking models into a pipeline using best DenseNet model Try out Siamese for classes that are usually mistaken	Multimodal with metadata of age, sex, and localization Testing with secondary dataset

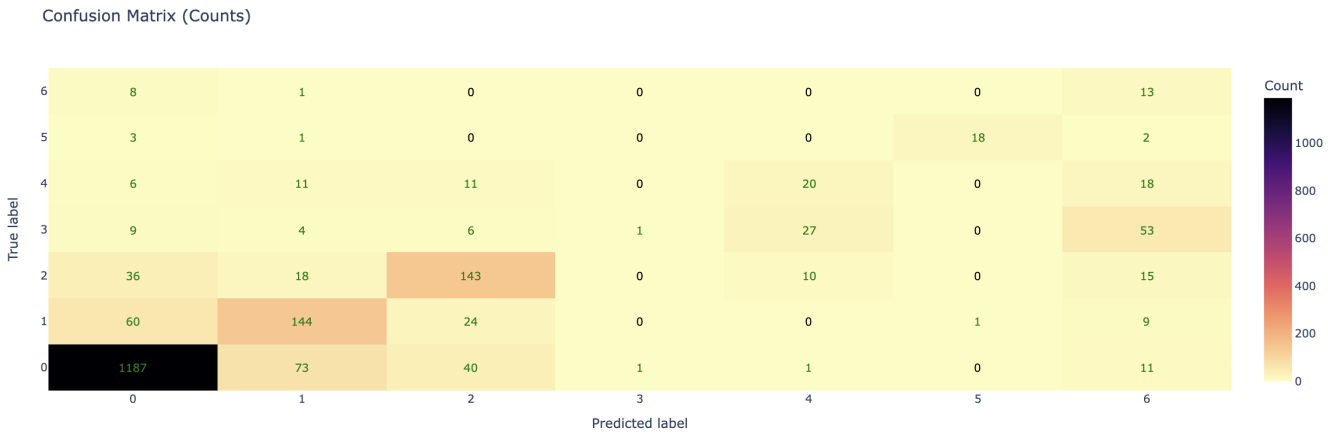
Appendix A: Distribution of Classes



Appendix B: Structure of hierarchical ensemble



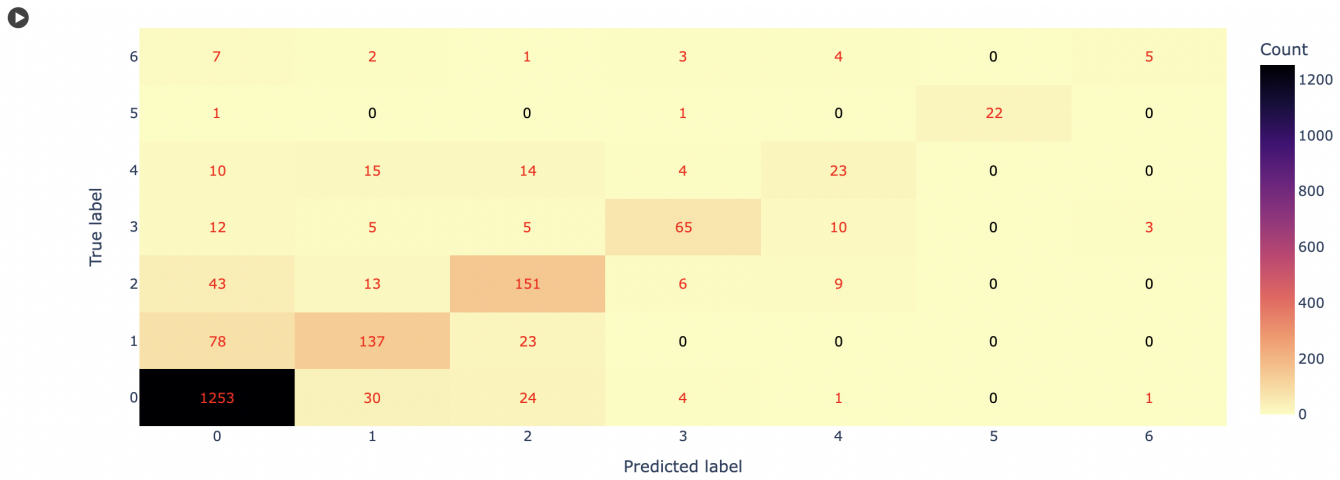
Appendix C: Detailed results of hierarchical ensemble



Class	Precision	Recall	F1
0	0.91	0.9	0.905
1	0.57	0.61	0.586
2	0.64	0.64	0.64
3	0.5	0.01	0.0198
4	0.34	0.3	0.3192
5	0.95	0.75	0.8361
6	0.11	0.59	0.1875

Table 1: Precision, Recall, and F1 values for each class - All-DenseNet201 hierarchical ensemble

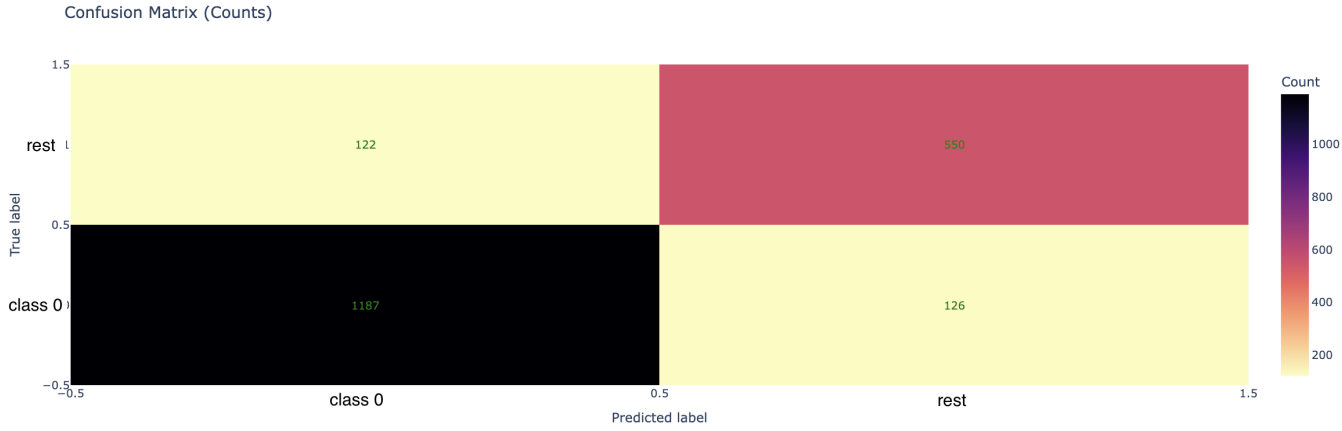
Appendix D: Confusion Matrix Results for a Average Ensemble



Classification Report for <<Ensemble (ResNet & DenseNet)>>:

	precision	recall	f1-score	support
0	0.89	0.95	0.92	1313
1	0.68	0.58	0.62	238
2	0.69	0.68	0.69	222
3	0.78	0.65	0.71	100
4	0.49	0.35	0.41	66
5	1.00	0.92	0.96	24
6	0.56	0.23	0.32	22
accuracy			0.83	1985
macro avg	0.73	0.62	0.66	1985
weighted avg	0.82	0.83	0.83	1985

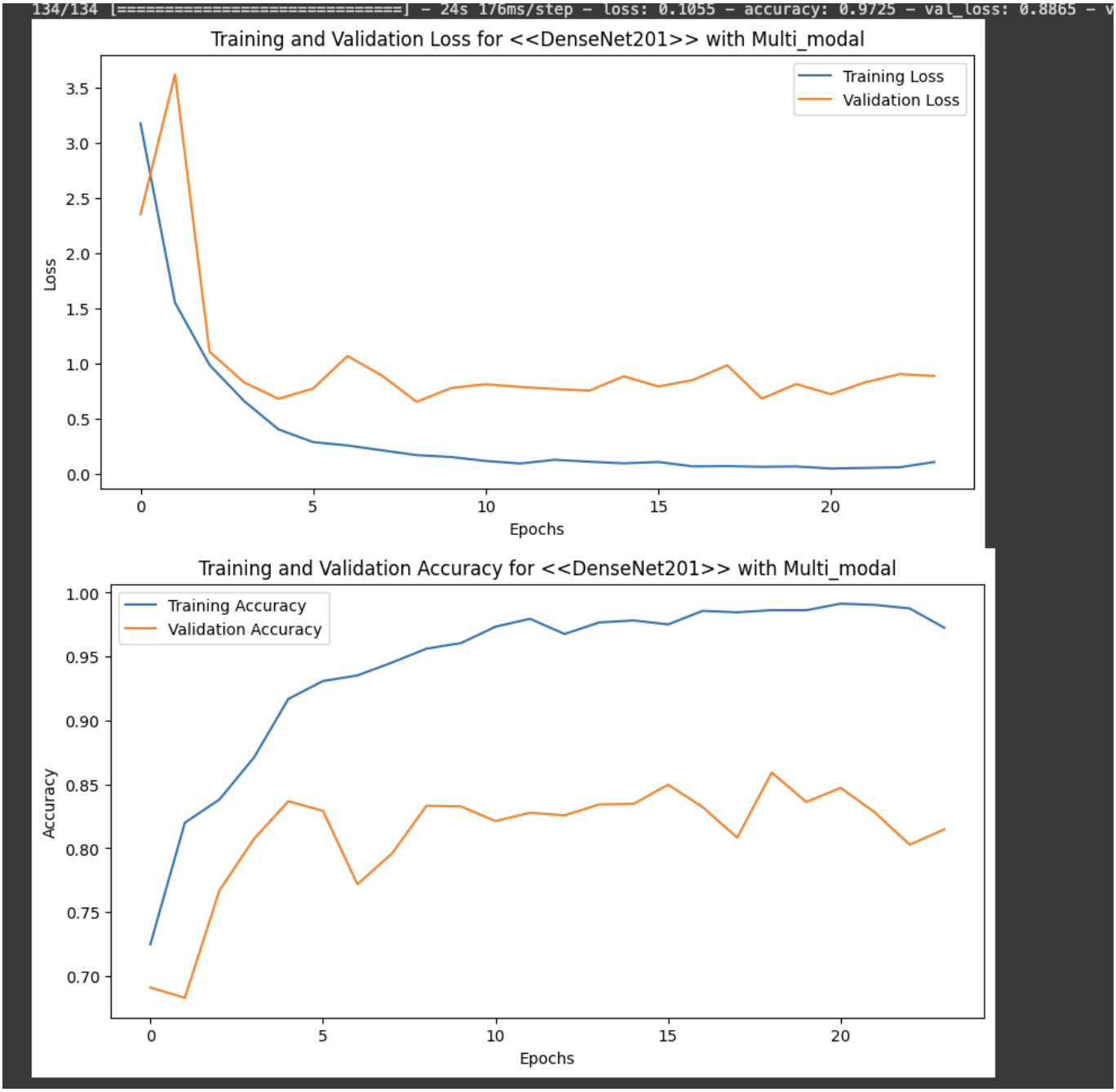
Appendix E: Result of initial classifier 1 in hierarchical ensemble (class 0 vs. remaining 6 classes)



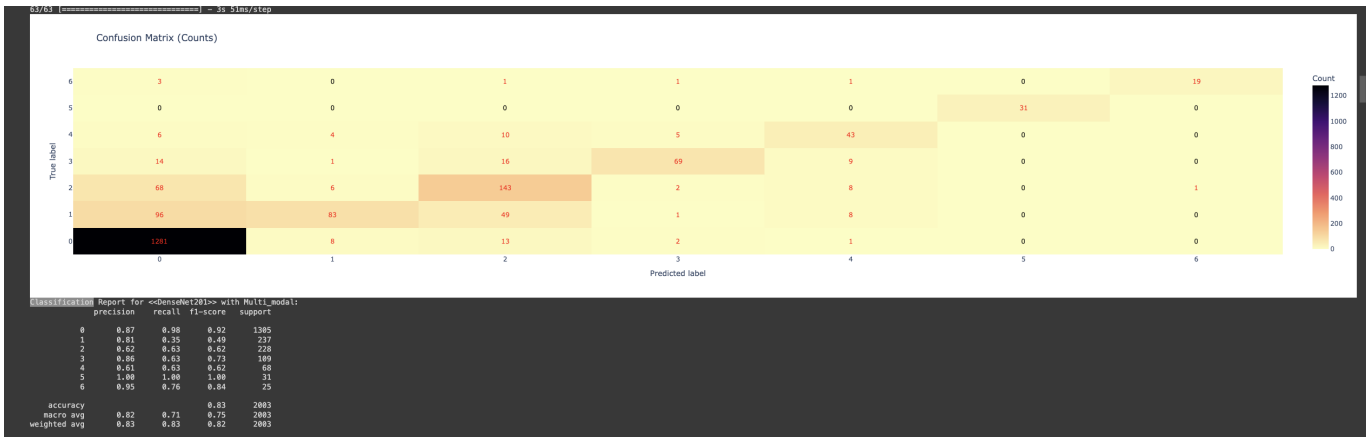
Classification Report for <<DenseNet201>> :

	precision	recall	f1-score	support
0	0.91	0.90	0.91	1313
1	0.81	0.82	0.82	672
accuracy			0.88	1985
macro avg	0.86	0.86	0.86	1985
weighted avg	0.88	0.88	0.88	1985

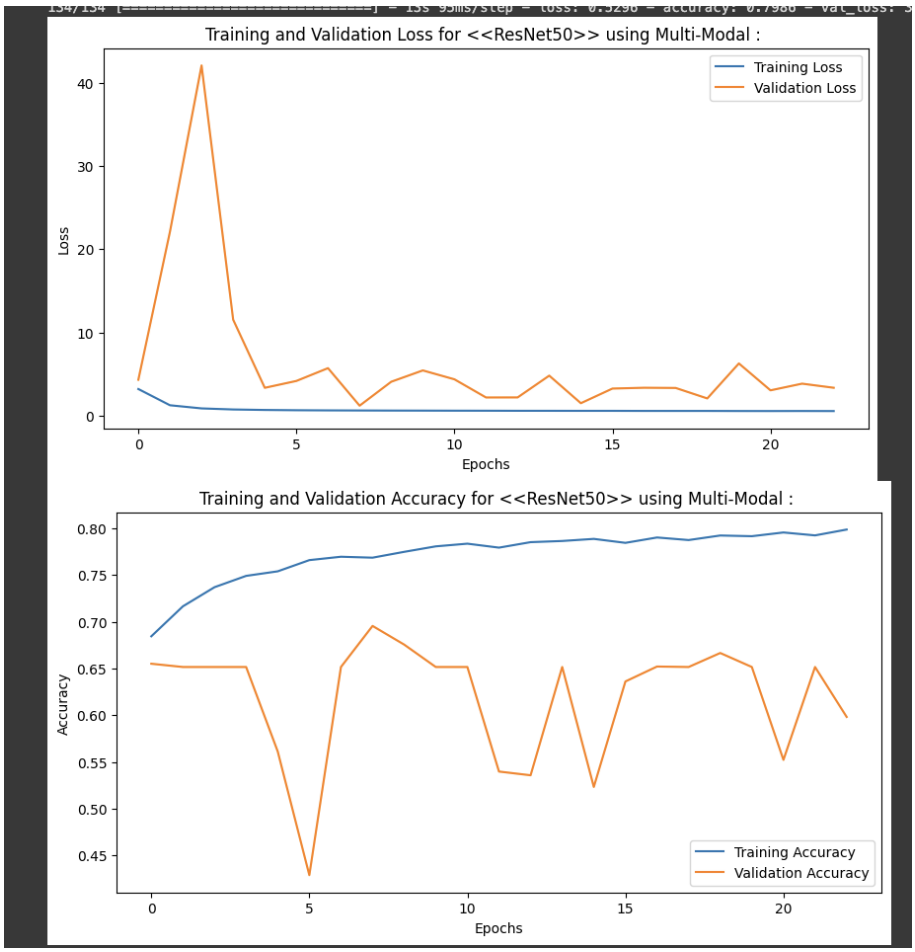
Appendix F: Training & Validation Loss and Accuracy Results for a DenseNet201 model with Multi-modal Approach



Appendix G: Confusion Matrix Results for a DenseNet201 model with Multi-modal Approach



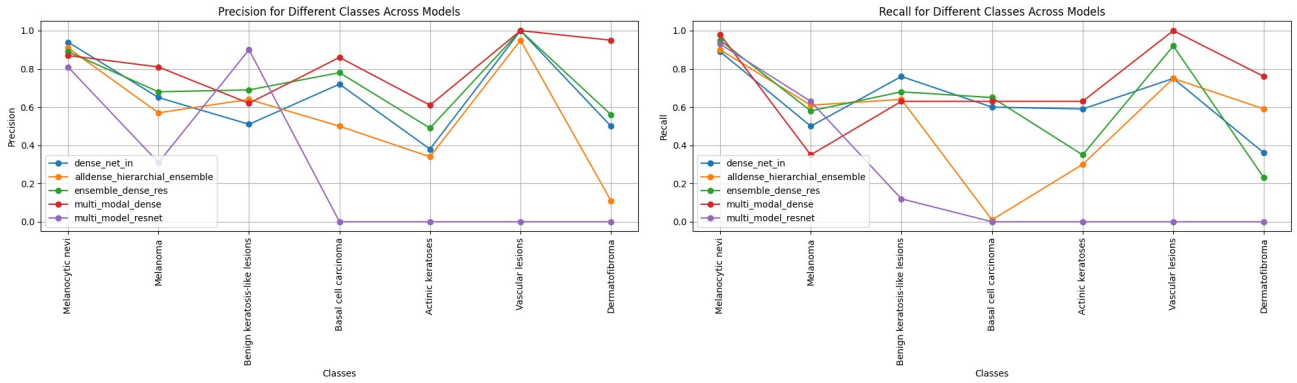
Appendix H: Training & Validation Loss and Accuracy Results for a ResNet50 model with Multi-modal Approach



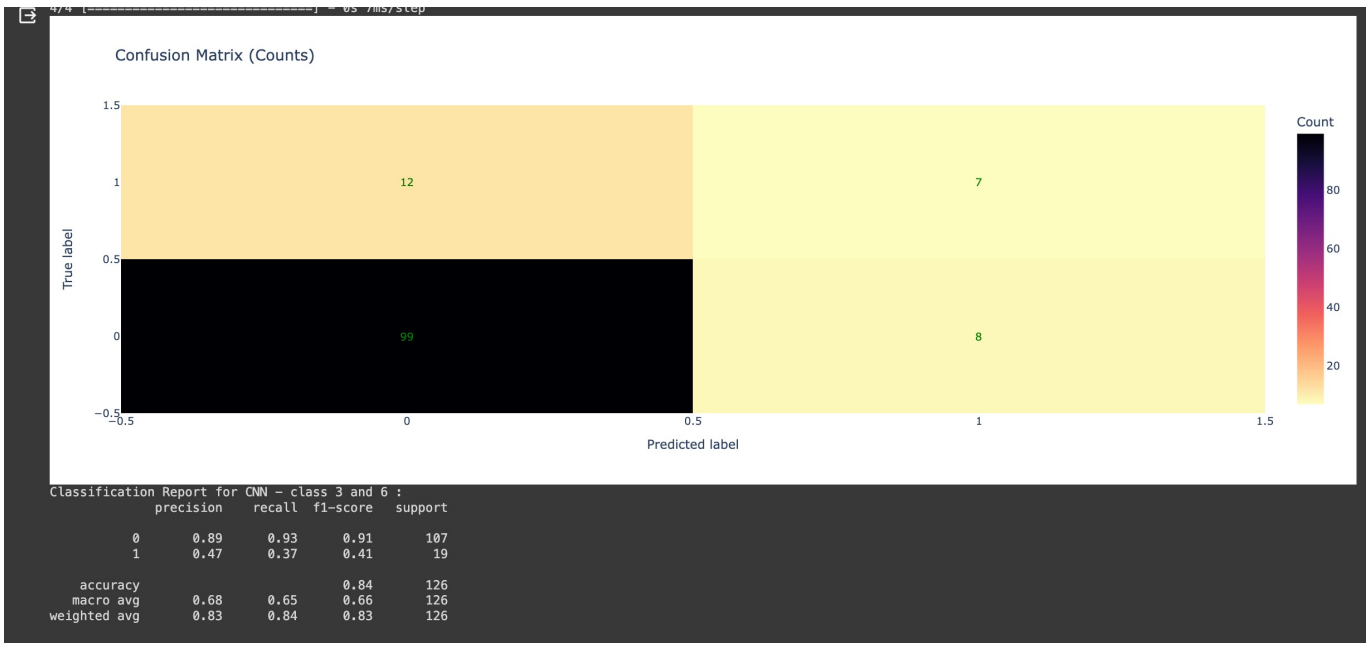
Appendix I: Confusion Matrix Results for a ResNet50 model with Multi-modal Approach


```
Classification Report for <<ResNet50>> using Multi-Modal :
              precision    recall  f1-score   support

           0       0.81      0.93      0.87      1305
           1       0.31      0.63      0.42       237
           2       0.90      0.12      0.22       228
           3       0.00      0.00      0.00       109
           4       0.00      0.00      0.00        68
           5       0.00      0.00      0.00        31
           6       0.00      0.00      0.00        25

    accuracy                           0.70      2003
   macro avg       0.29      0.24      0.21      2003
weighted avg       0.67      0.70      0.64      2003
```

Appendix J: Precision and Recall



Appendix K: CNN for 3 and 6




```
Classification Report for CNN - class 3 and 6 :
              precision    recall  f1-score   support

           0       0.89      0.93      0.91       107
           1       0.47      0.37      0.41        19

    accuracy                           0.84       126
   macro avg       0.68      0.65      0.66       126
weighted avg       0.83      0.84      0.83       126
```

References

Bhandari, A. "Feature engineering: Scaling, normalization, and standardization (Updated 2023)." Analytics Vidhya, 7 July 2023, www.analyticsvidhya.com/blog/2020/04/feature-scaling-machine-learning-normalization-standardization/.

Chen, Guangyong, et al. "Rethinking the Usage of Batch Normalization and Dropout in the Training of Deep Neural Networks." ArXiv.org E-Print Archive, 2019, arxiv.org/pdf/1905.05928.pdf.

Farshadjafari. "Skincancer Detection-Multiple Models :83%Accuracy." Kaggle, Kaggle, 30 July 2023, www.kaggle.com/code/farshadjafari97/skincancer-detection-multiple-models-83-accuracy.

Mumuni, A., and F. Mumuni. "Data augmentation: A comprehensive survey of modern approaches." Science Direct, Dec. 2022, www.sciencedirect.com/science/article/pii/S2590005622000911.

Pramoditha, Rukshan. "How to Mitigate Overfitting by Creating Ensembles." Medium, Towards Data Science, 5 Oct. 2021, towardsdatascience.com/how-to-mitigate-overfitting-by-creating-ensembles-77e9299b9ad0.

Shirsat, Mithilesh. "Ensemble Learning: Combining Multiple Models for Better Predictions." LinkedIn, 15 Apr. 2023, www.linkedin.com/pulse/ensemble-learning-combining-multiple-models-better-mithilesh-shirsat/.

Wei, Mingjun, et al. A Skin Disease Classification Model Based on DenseNet and ConvNeXt Fusion. MDPI, 14 Jan. 2023. www.mdpi.com/2079-9292/12/2/438.

Zhang, Chaoning et al. ResNet or DenseNet? Introducing Dense Shortcuts to ResNet. 2020, October 23. https://doi.org/10.48550/arXiv.2010.12496