Group project report

Skin cancer classification using neural networks

1. Introduction

Deep learning neural networks present themselves as one of the cutting-edge image analysis techniques that perform an auxiliary function to the classification and diagnosis of diseases. The objective of this paper is to build a model that classifies skin lesions and diagnoses skin cancer when given a picture of the lesion, and the paper hopes to achieve a model that classifies lesions correctly far above the level achieved by pure chance.

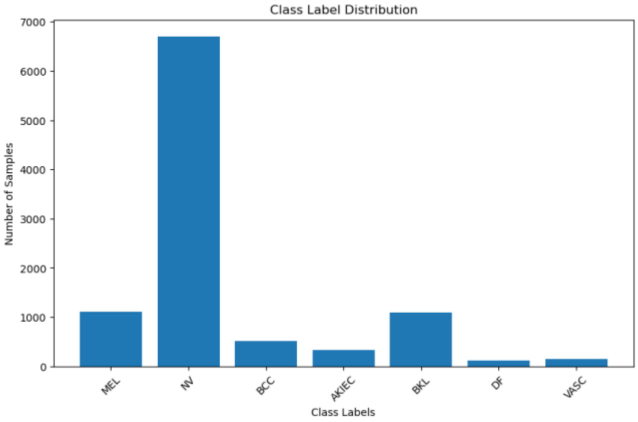


Figure 1. Dataset class distribution

2. Data Set

The data set being used is taken from Skin Cancer Dataset [1] and the Training Ground Truth is used as the entire data set for training, validation and testing purposes. The data includes 6409 sample pictures of 7 classes of lesions. The largest class is Melanocytic nevi (class 1), which has 4293 pictures, followed by Benign keratosis-like lesions with 714 cases. The smallest two classes are vascular lesions (90 cases) and dermatofibromas (75 cases). It shows that the dataset suffers from severe data imbalance (see Figure 1.) that skews in favor of Class 1.

The data set includes a data file and a label file, which are used as X (features) and y (outcomes) in the model training respectively. They are both in .npy format, containing arrays, and the image arrays are image tensors that can be used to regenerate the picture. The y (label) file is the one-hot coding of the 7 classes. The project randomly selected 15 image tensors to make sure they represent realistic lesion pictures.

Before the model is trained, the data and label go through stratified train-test splits with ratios of 0.2 for both training and validation data, with a random seed of 42. Thus, the training set makes up 64% of the original data, the validation set 20% and the test set 16%. Then all feature sets go through normalization so that the scale is limited between 0 and 1, and the y sets are converted into floats.

3. Baseline model

The baseline model is firstly made up of two consecutive convolutional layers with Relu activation and padding, equipped with 64 and 32 filters respectively, and the two layers are followed by a max pooling layer of the size 2x2. The set of convolutional and max pooling layers is repeated once before the flatten layer. Then there are two dense layers of size 32 with Relu activation, followed by an output layer with Adam as optimizer, categorical cross entropy as the loss function and accuracy as metrics. The model is then fit onto the data using the epoch time equal to 10 and batch size equal to 32.

While the model history shows that overfitting begins from the 6th epoch when validation stopped rising while training kept increasing, the baseline model achieves a fairly good result at lesion classification. Te test accuracy of the baseline model is 0.681. For all classes, the ROC-AUC is significantly larger than 0.5, which means the predictions are not correct by pure chance and guessing.

However, the baseline model shows the tendency to predict most test images as the majority class (Class 1), as can be seen from the confusion matrix. This is further corroborated by the class-specific F1 scores. Class 0's F1 score is 0.228, while the F1 scores for Class 2, 3, 4 and 5 are all close to 0, which means the baseline model almost categorized no image in these classes correctly. It is predicted because the dataset is overwhelmingly dominated by Class 1, the majority class drives the loss function the model tries to minimize, while it has not learned enough features of the smaller classes. This affects the model's overall effectiveness in a multi-class classification setting.

4. Enhanced models

In the attempt to improve the models, the paper tuned various hyper-parameters and applied data augmentation techniques. The project chose to set epoch times to 20 to examine the longer training trajectory of each model for further analysis. The applied hyper-parameter tuning includes: adjusting the number of epochs, changing the number of layers, changing the type of optimizer, changing the loss function, adding data augmentation layers to increase diversity of the training set. In addition, the project attempts to address the data imbalance problem by creating new images using keras data Image Generator and tests the efficacy of this approach.

The indices selected to evaluate the models are as follows: test accuracy, unweighted mean of test F1 score, weighted mean of test F1 score, micro AUC-ROC and macro AUC-ROC. Test accuracy measures the proportion of real positives in the entire number of judgements. It is complemented by some other metrics. Both the weighted and unweighted mean of F1 score considers both precision and recall, while the unweighted one better reflects the F1 scores of the minority classes, in contrast to the weighted one that considers then majority class more important. Similarly, macro-average ROC-AUC better reflects the categorization ability of the minority classes given class imbalance, while micro-average ROC-AUC considers the holistic capability of the model to distinguish, thereby giving more importance to the majority class.

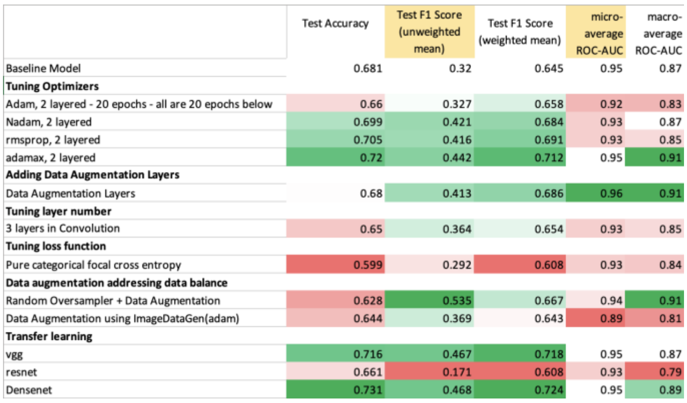
Among the five metrics, the unweighted mean of test F1 score and micro-average ROC-AUC are less swayed by the imbalanced nature of the dataset and reflects the performances of the model on smaller classes, while accuracy, weighted mean of test F1 score and macro-average ROC-AUC report how the model performs on the entire dataset, and they may be more useful if it is decided that the majority class is more medically important for their size. Therefore, the report decides to report the five metrics despite the deficiencies of some of the metrics, in order to enable the medical experts' comprehensive interpretation for further discussion.

The authors first changed the epoch time from 10 to 20 and examined the performance. Based on this, various optimizers are tried, including Adamax, Nadam, RMSprop, along with Adam. The optimizers were selected because of their ability to process image data. Then besides categorical cross entropy, categorical focal cross entropy a the loss function was tried due to the supposed ability to handle imbalanced data. The model was also tried with 3 sets of convolutional and max pooling layers as well as keras data augmentation layers, which are added before the first input layer. The aforementioned changes are possible thanks to how the model building function isn written (see codes).

Lastly, the authors attempted to address the problem of data imbalance by creating new images. In doing do, two approaches were implemented: 1) Random oversampler of imblearn library is used so that every class's image count reaches the majority classes through replication. There is the risk that the machine over-learns the features of the minority classes that are excessively replicated. Thus, it is combined with keras data augmentation layers which augment the images before becoming input. The generated images are reasonable. 2) Image Data Generator of keras is used so that new images of minority classes are generated through sifting, rotating, shearing, among other techniques. It is combined with undersampling so that the frequency of every class becomes 1000. The generated images are less realistic with blurs (as seen in the code) but the data of the model trained on this are worth looking at.

Results

The authors collected the metrics of each model in an excel table (as seen in the Figure 2), and every column is colors against the baseline model's corresponding metric, where then seen color means better performance while red means underperformance. We set the threshold value to be 0.5. The two metrics suitable for imbalanced data are marked yellow. As we see, most models saw a significant improvement in the unweighted mean of Test F1 score, meaning they were likelier to categorize smaller classes better. Doubling the epoch time seemed to worse the performance. The F1 scores saw very slight improvements at the cost of lower accuracy, micro and macro-average ROC-AUC. It happened possibly because the machine over-learned the pattern of the test images. Thus, it cannot predict the unseen images well.



	Test Accuracy	Test F1 Score (unweighted mean)	Test F1 Score (weighted mean)	micro-average ROC-AUC	macro-average ROC-AUC
Baseline Model	0.681	0.32	0.645	0.95	0.87
Tuning Optimizers					
Adam, 2 layered - 20 epochs - all are 20 epochs below	0.66	0.327	0.658	0.92	0.83
Nadam, 2 layered	0.699	0.421	0.684	0.93	0.87
rmsprop, 2 layered	0.705	0.416	0.691	0.93	0.85
adamax, 2 layered	0.72	0.442	0.712	0.95	0.91
Adding Data Augmentation Layers					
Data Augmentation Layers	0.68	0.413	0.686	0.96	0.91
Tuning layer number					
3 layers in Convolution	0.65	0.364	0.654	0.93	0.85
Tuning loss function					
Pure categorical focal cross entropy	0.599	0.292	0.608	0.93	0.84
Data augmentation addressing data balance					
Random Oversampler + Data Augmentation	0.628	0.535	0.667	0.94	0.91
Data Augmentation using ImageDataGen(adam)	0.644	0.369	0.643	0.89	0.81
Transfer learning					
vgg	0.716	0.467	0.718	0.95	0.87
resnet	0.661	0.171	0.608	0.93	0.79
Densenet	0.731	0.468	0.724	0.95	0.89

Figure 2. Model performances

Changing the optimizers based on the baseline model architecture proved more fruitful, with almost overall increases in the metrics. Adamax achieves the best performance across the board, raising test accuracy from 0.68

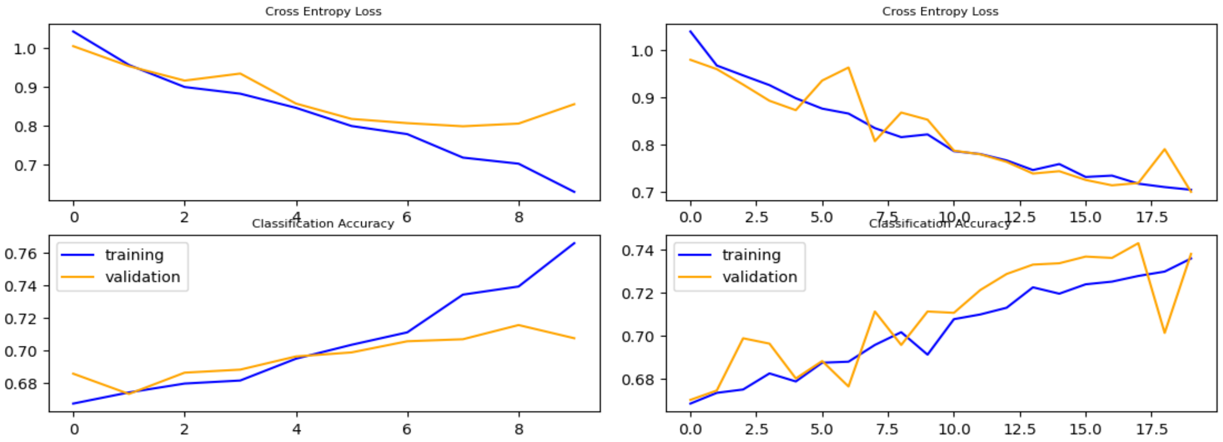


Figure 3. Baseline model history versus Data Augmentation layers model history

to 0.72, while increasing unweighted F1 by 0.12. Meanwhile, no metric saw a decrease. Both Nadam and RMSprop are slightly less outstanding but they still resulted in a substantial increase in test accuracy, unweighted and weighted mean of Test F1 scores, meaning that the optimizers helped to find a better minimum for the loss function.

Adding keras data augmentation layers achieved similarly good results as Adamax - overall improvement of model performances; in addition, the problem of overfitting is largely solved (Figure 3). It might result from the way data augmentation layers function, which helps the model to learn features unique to every class which are however invariant to these distortions and thus to become more robust, while avoiding the model seeing similar images twice. In this way, the model avoids over-learning the features of the majority class. The suppression of overfitting is clear to be seen in Figure. From the lack of divergence between trajectories of training and validation accuracy on the right.

Adding one more set of layers in the convolutional part of the model did not improve the overall performances, while choosing "categorical cross focal entropy" deteriorated all the metrics, as shown by the red color across the row. They imply that are complicated model architecture may be too complex for the current training set with

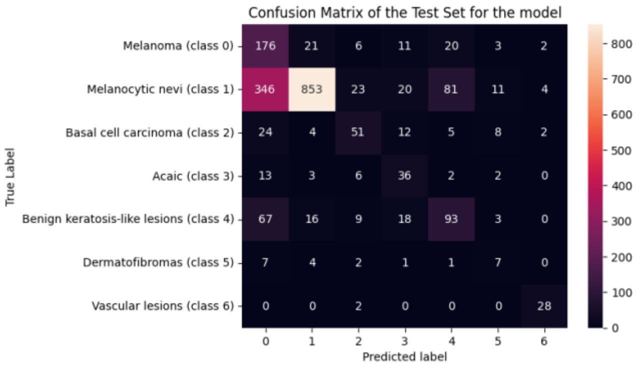


Figure 4.Confusion matrix of random oversampler model

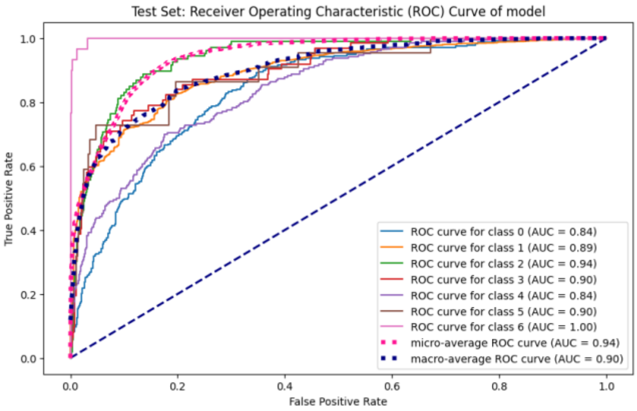
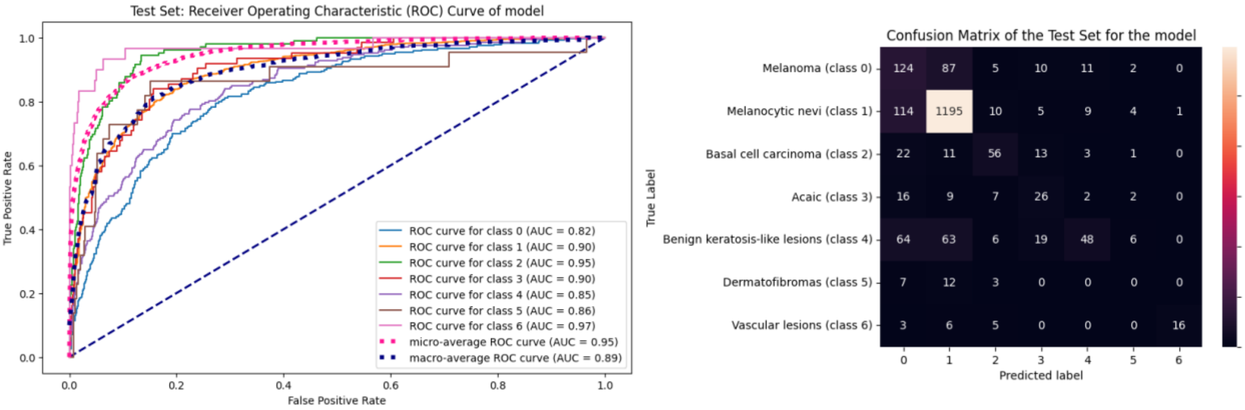


Figure 5. Test ROC curves of random oversampler model

roughly 4000 pictures and thus it is counterproductive. Also,
the loss function does not suit the data set and model architectures although it may work better with other hyper-parameters and architectures.

Comparing the two methods to address data imbalance, both models outperform the baseline in classifying the minority classes, but the overall performance of random oversampler is better. Despite the decrease in the overalls



	0	1	2	3	4	5	6
Melanoma (class 0)	124	87	5	10	11	2	0
Melanocytic nevi (class 1)	114	1195	10	5	9	4	1
Basal cell carcinoma (class 2)	22	11	56	13	3	1	0
Acaic (class 3)	16	9	7	26	2	2	0
Benign keratosis-like lesions (class 4)	64	63	6	19	48	6	0
Dermatofibromas (class 5)	7	12	3	0	0	0	0
Vascular lesions (class 6)	3	6	5	0	0	0	16

Figure 6. Performances of Densenet

accuracy by 0.2, the unweighted mean of F1 is 0.21 higher than the baseline and is the highest, and it is also the only model that has a significancy non-zero F1 score for Class 5. We see that the ability to predict small classes well comes at the expense of classifying Class 1 (the majority class) as Class 0 from Figure 4; the medical implications of it in practice should be discussed with experts and is beyond the scope of the report. In contrast, using image data generator only produced a slight improvement in micro-average F1 score at the expense of all other metrics, especially its micro-average ROC-AUC which is the lowest, possibly because the undersampling process got rid of useful information in the majority class, while oversampling created many unrealistic images as can be seen in the codes. The attempt to address data imbalance was not the best way to achieve the best performances, and the root cause is the imbalanced nature of the dataset. Any generation of images would mean the same features of the limited classes are somehow replicated and the model over-learns them, or they are so distorted that the model no more learns the correct features. Either way, the model can be improved through having more diversified images that are balanced between classes.

Lastly, the authors used transfer learning, which is to integrate a pre-trained model as the feature extractor and add dense layers tailored for the dataset. By leveraging the pretrained models' knowledge, the authors expect to achieve better performances. Densenet turned out to be the top performer in terms of all metrics, surpassing the baseline's accuracy by 0.05 while significantly increasing F1 scores by 0.15 and 0.08 with slightly increased ROC-AUC. Thus, it becomes the best model tried in the project and best predicts the minority classes. In contrast, resnet performs surprisingly bad across the board, while VGG's performance was as good as other models discussed in the report.

Conclusion

As mentioned above, the fundamental solution would be to collect more image data in its absolute number and in terms of data balance, which are subsequently fed into transfer learning models [2]. The collected images should be of higher quality with similar lighting, focus and angle for better learning. In addition, other authors have shown that the application of AlexNet coupled with K-nearest neighbor using cosine distance metrics can produce more satisfactory performances, and more advanced data augmentation technique such as Gaussian low pass filter, histogram equalization, noise addition, guided filtering, JPEG compression and motion blur may be applied too [3]. Attention mechanism in image recognition, too, can locate the maximum response points and crop images which the model can learn better [4]. Overall, Densenet was the best model among all above, and ll the models achieved an accuracy around 0.70 and could non-randomly classify images, while underperforming for minority classes. However, further improvements are still to be done with cutting-edge model architectures and higher computational capabilities, in cooperation with medical experts for decisions around crucial image data quality, labelling and threshold value.

References

[1] Skin Cancer Dataset: https://challenge.isic-archive.com/data/#2018, accessed on 01/10/23.

[2] Brinker TJ, Hekler A, Utikal JS, Grabe N, Schadendorf D, Klode J, Berking C, Steeb T, Enk AH, von Kalle C. Skin Cancer Classification Using Convolutional Neural Networks: Systematic Review. J Med Internet Res. 2018 Oct 17;20(10):e11936. doi: 10.2196/11936. PMID: 30333097; PMCID: PMC6231861.

[3] V. Pomponiu, H. Nejati and N. . -M. Cheung, "Deepmole: Deep neural networks for skin mole lesion classification," *2016 IEEE International Conference on Image Processing (ICIP)*, Phoenix, AZ, USA, 2016, pp. 2623-2627, doi: 10.1109/ICIP.2016.7532834.

[4] Liu, Y.-P., et al.: "Multiscale ensemble of convolutional neural networks for skin lesion classification" *IET Image Process.* 15, 2309–2318 (2021). https://doi.org/10.1049/ipr2.12214