Programming Assignment 5: Dermatological image classification

Chew Ching Hian
Chalmers University of Technology
chingh@chalmers.se

Poh Shi Qian
Chalmers University of Technology
shiqia@chalmers.se

Zhang Xiaoyang
Chalmers University of Technology
xiazhan@chalmers.se

Abstract

This report details the implementation of a dermatological image classifier that discerns melanomas from melanocytic nevus skin lesions. This classifier is created for the purposes of a supervised machine learning assignment. The training data and the testing data are sourced from the 2018 ISIC challenge dataset.

1 Introduction

Skin cancer is one of the most prevalent malignancies worldwide, with melanoma being especially dangerous because of its short survival time if not caught early. In order to differentiate between melanoma and melanocytic nevus, a benign birthmark, we investigate the application and assessment of machine learning methods for the classification of skin lesions in this paper. Melanomas can be visually discerned from nevus skin lesions as they are generally larger, and have a wider range of shapes and colors.

2 Technical Solutions

2.1 Data Augmentation

In this section, we detail the technical solutions implemented for preprocessing the training and validation datasets provided to us. This preprocessing is done with the use of PyTorch's torchvision.transforms module to define a series of transformations for the training and validation datasets. The input images undergo these preprocessing before being fed into the model. In this case, we have also decided to perform separate transformations on the training and validation datasets in order to introduce randomness and variability in training.

2.2 Data Loading

Firstly, the image data is loaded from the specified directory.

In order to efficiently manage the memory usage and improve computation efficiency, we have separated the loading process into batches of 4. This process will also be completed in a random order with shuffling at each epoch to ensure overfitting is being reduced.

2.3 Model Preparation, Loss Function and Optimizer

We used the ResNet model. It utilizes batch normalization, allowing for greater stability in the model's learning. Furthermore, by using the ResNet model, we are able to use the features learnt by that model on our data, which reduces the amount of training needed, and thus increasing efficiency.

ResNet was a suitable model to use as it had been trained on ImageNet, a huge database of images. As such, it would

have picked out features that are commonly used to differentiate between various categories.

3 Experiment

3.1 Training, Validation Function and Training Execution

To ensure the robustness and generalizability of our image classification model, we implemented a thorough training regime in addition to a rigorous validation process. Model training is done in the function 'train_model', designed to iterate over epochs, managing both training and validation phases within each cycle. This function schedules learning rate modifications, as well as to coordinate the forward pass, loss computation, backward pass, and parameter updates.

Core of our model training leverages Pytorch. We also used CrossEntropyLoss for its suitability in classification tasks and an SGD optimizer for efficient learning. A step-based learning rate scheduler reduced the learning rate periodically to refine the training process.

Training is done over 25 epochs, balancing improvement in validation accuracy against risk of overfitting. Data augmentation applied to the training set enhances model robustness whereas for the validation set, it was processed with basic resizing and normalization, providing a consistent benchmark for performance evaluation.

3.2 Test Data Evaluation

Following the comprehensive training and validation process, the model was evaluated on an unseen test dataset.

It was processed using the same preprocessing pipeline to match input format of the model, including resizing, center cropping, tensor conversion, and normalization. The test dataset was loaded using Pytorch's ImageFolder utility, ensuring that images were automatically labeled based on folder structure. We also applied transformations to resize and normalize the images, aligning them with the model's expected input format.

Model was set to evaluation mode to disable training-specific operations like dropout, to ensure predictions solely relied on learned parameters.

After gathering predictions and true labels, we calculated the model's accuracy on the test set by comparing predicted labels against true labels.

4 Results

4.1 Training and Evaluation

Our model's training process over 25 epochs demonstrated consistent improvement in both training accuracy and validation accuracy, along with reduction in loss for both phases. Starting with initial training accuracy of 70.31% and validation accuracy of 82.67%, there was steady improvement over more epochs. By epoch 4, validation accuracy increased notably to 86.98%, at the same time training accuracy reached 86.93% by final epoch.

Furthermore, the training loss decreased from an initial 0.6686 to 0.2846 by the end of the training process, indicating robust learning from training data. Validation loss showed a downward trend as well, starting at 0.3567 and finishing at 0.2466.

The peak performance of the model was observed at epoch 24 with the highest validation score of 89.616%.

4.2 Evaluation on Test Set

The evaluation yielded an accuracy of 88.5% on the test dataset. This result indicates the model's strong generalization capabilities as it is much higher than 50%. This means that the model is not merely guessing whether an image is of a melanoma or a melanocytic nevus skin lesion,

but has instead identified features to distinguish them.

The accuracy of the evaluation on the test set is approximately 1% lower than that of the evaluation on the highest validation score. This slight drop could be due to the fact that the test set was unseen by the model. Nevertheless, the difference in accuracy is very slight and thus shows that the model has learnt to identify the distinguishing features accurately.

4.3 Comparison of Results With and Without Data Augmentation

We hypothesized that data augmentation was a key contributor to the accuracy of the model. To test this hypothesis, we trained a separate model without carrying out data augmentation.

The best accuracy of this new model on the validation data is 52.8754%, and the accuracy of it on the test data is 50.66%, which shows that the model is likely guessing the categories rather than predicting categories due to learnt features.

As this is a significant difference of our previously-achieved accuracy of 88.5% on the test data set, we can conclude that our hypothesis was proven.

5 Conclusion

To conclude, our model was successful in distinguishing between melanoma and melanocytic skin lesions. This success can be attributed to a few reasons. Firstly, the data augmentation process introduced randomness in the training to decrease the likelihood of overfitting. Secondly, the usage of the pre-trained model ResNet, which had pre-trained weights from a large dataset that would have increased the accuracy of the model, considering that our training data size is significantly smaller.

6 Limitations

Despite the promising results, our study has several limitations. Firstly, the dataset size is limited, which may affect the generalizability of the models. Hence, this may limit the model's effectiveness in the real world context.

In addition, the classification task is simplified to merely distinguishing between melanoma vs. nevus. As a result, this may not capture the full complexity of skin cancer diagnosis given that several other types exist in the real world.

7 Ethical Considerations

When developing and implementing machine learning solutions for medical diagnostics, it is crucial to consider the ethical issues surrounding such sensitive personal data. Careful consideration must be given to issues including data privacy, bias in training data, and the possible effects of misclassifications on patient outcomes.

7.1 Data privacy

Sensitive patient data, such as genetic information, medical histories and images of patients, are usually utilized to train machine learning models for such medical diagnoses. As such, effective data governance is essential. This entails the use of anonymization techniques to prevent these data from being linked to a particular patient. Safe storage with access controls to only authorized personnel is also essential to ensure such sensitive data are restricted. Explicit patient permission policies outlining the intended use of their data should also be considered and made known to patients before such data are being stored and used for research purposes. To guarantee compliance, regulatory frameworks unique to healthcare AI are also required.

7.2 Bias in Training Data

When training data is biased, algorithms may misdiagnose particular demographic groups disproportionately. For instance,

an algorithm that was mostly trained on pictures of people with lighter skin tones may not detect skin cancer in people with deeper skin tones. In order to lessen such misdiagnosis, varied datasets that reflect the general public are required.

7.3 Possibility of misclassifications on patient outcomes

In addition, despite the efficiency brought about by machine learning in aiding such diagnosis and classification of skin cancer types, it is important to research and have an in-depth understanding of such automated diagnosis systems before leveraging it for making healthcare decisions. This is due to the possibility of misclassification of patient outcomes, which could occur as the system's accuracy is unlikely to be 100%. As such, the use of automated technologies should not take the place of human competence but rather enhance the diagnosis process.