Skin Cancer Classification

Background

The skin cancer classification task is a supervised learning process of machine learning, recognizing images and correctly classifying the class of skin cancer to the given labels.

In this skin cancer classification project, it should train and learn from eight types of labeled skin cancer images through a deep learning computer vision model, and correctly identify and classify the class of skin cancer in the test set. I need to consider what kind of the computer vision model, data processing, and the implementation of the training process. And also adjust the current process based on the feedback from the training results to achieve better training outcomes. Moreover, according to the observation of image data, images have different influences like exposure, so I can adopt measures such as image enhancement, image preprocessing and to improve the robustness and recognition ability of the model, and also the image numbers of each class is different, I can apply augmentations to solve the imbalance datasets problem.

Method

I adopt a pretrained convolutional neural network based on ResNet-50 as the base model. ResNet-50 is a classic deep residual network that introduces residual connections to solve the gradient vanishing problem during the training process, and it also have strong feature extraction capabilities and no need to do extra feature extraction for each images. I use transfer learning on the base model ResNet-50, fine-tuned the last layer and limited the output results to 0-8 categories to adapt to the skin cancer classification task.

To train the neural network model, I define the cross-entropy loss function as the loss function, and used the stochastic gradient descent optimizer with momentum to optimize the model parameters.

For data preprocessing, I apply different data augmentation operations on the training set, such as random rotation, random horizontal flip, random vertical flip, color jitter, Gaussian blur, random sharpening, etc., to increase the diversity of data and improve the generalization ability of the model. For the validation set, I only performed normalization, scaling the image pixel values.

Experiments

Version 1 - Fine-tune the pre-training base model

I firstly implemented the pre-trained resnet50 model using the PyTorch and torchvision libraries, and added a fully connected layer to achieve the final output of 8 categories.

At the same time, the data was only normalized, and the folder name was used as the label for 20 epochs of training.

The training results and test results are as follows:

Epoch	Train Loss	Train Acc	Val Loss	Val Acc
1	0.9188	0.6742	1.7847	0.4163
2	0.6950	0.7477	1.7732	0.4188
3	0.5317	0.8075	1.6185	0.4863
4	0.3901	0.8580	1.5073	0.5537
5	0.2836	0.8968	1.7151	0.5600
6	0.2135	0.9240	1.7440	0.5487
7	0.1588	0.9445	1.8616	0.5725
8	0.1310	0.9531	2.0668	0.5775
9	0.1091	0.9627	2.0553	0.5763
10	0.0839	0.9709	2.1126	0.5850
11	0.0684	0.9770	2.3744	0.5763
12	0.0524	0.9821	2.2084	0.5875
13	0.0491	0.9844	2.5336	0.5475
14	0.0350	0.9882	2.3404	0.5763
15	0.0333	0.9882	2.4685	0.5863
16	0.0335	0.9893	2.5175	0.5875
17	0.0261	0.9915	2.4610	0.5850
18	0.0221	0.9933	2.3486	0.6088
19	0.0269	0.9913	2.6706	0.5637
20	0.0231	0.9926	2.6141	0.5713

```
Test Accuracy: 0.5713
Test F1-Score: 0.5618
Class Report:
```

	precision	recall	f1-score	support
AK	0.76	0.42	0.54	100
BCC	0.45	0.84	0.59	100
BKL	0.51	0.50	0.51	100
DF	0.96	0.26	0.41	100
MEL	0.61	0.69	0.64	100
NV	0.43	0.90	0.58	100
SCC	0.74	0.35	0.48	100
VASC	0.97	0.61	0.75	100
accuracy			0.57	800
macro avg	0.68	0.57	0.56	800
weighted avg	0.68	0.57	0.56	800

Based on the training results and test results, I obtain the following information for optimization:

During the training process, the accuracy is flat around 7-10 epochs, so I can test different process based on 10 around epochs.

At present, the precision of VCC, BKL, NV, and MEL needs to be improved. The recall of AK, BKLDF, SCC, and VASC needs to be improved.

Version 2 - Image Enhancement and Handling Class Imbalance

By analyzing the data, I found: The number of images in the training set is different, there a few of images for AK, DF, NV, SCC, and VASC. To reduce the impact caused by the number of training images of each class, I handle class imbalance by adjusting class weights. Also, noticing that images are affected to varying degrees, I further enhanced the data.

The training and test results after 20 epochs are as follows:

Epoch	Train Loss	Train Acc	Val Loss	Val Acc
1	1.5557	0.4696	1.5604	0.5225
2	1.3243	0.5377	0.8805	0.5800
3	1.1891	0.5775	1.3843	0.5950
4	1.1146	0.5913	1.1598	0.5775
5	1.0298	0.6116	1.0005	0.6162
6	0.9518	0.6372	0.7748	0.6438
7	0.9043	0.6498	0.8734	0.6700
8	0.8640	0.6563	1.0077	0.6412
9	0.8093	0.6659	1.0495	0.6600
10	0.7699	0.6833	0.9524	0.6650
11	0.7462	0.6860	1.1406	0.6613
12	0.8110	0.6618	0.9706	0.6763
13	0.7148	0.6941	1.0749	0.6475
14	0.6626	0.7094	1.1160	0.6562
15	0.6230	0.7268	1.0574	0.6688
16	0.5724	0.7362	1.2965	0.6663
17	0.5681	0.7411	1.0650	0.6737
18	0.5752	0.7334	0.9233	0.7013
19	0.5397	0.7453	1.1458	0.7025
20	0.5088	0.7576	1.2723	0.6987

```
Test Accuracy: 0.6987
Test F1-Score: 0.7015
Class Report:
              precision    recall  f1-score   support

          AK       0.64      0.75      0.69       100
         BCC       0.55      0.71      0.62       100
         BKL       0.70      0.55      0.61       100
          DF       0.88      0.61      0.72       100
         MEL       0.55      0.67      0.61       100
          NV       0.70      0.74      0.72       100
         SCC       0.78      0.67      0.72       100
        VASC       0.94      0.89      0.91       100

    accuracy                           0.70       800
   macro avg       0.72      0.70      0.70       800
weighted avg       0.72      0.70      0.70       800
```

As shown in the result images, after enhancing the images and handling class imbalance, the accuracy and various class accuracy have been improved, the accuracy of the test set has increased from 0.57 to 0.69.

version 3 - More Image Enhancement + Sampling

This time, I adopt more image enhancements such as color jitter, Gaussian blur, and random sharpening, and also oversampled the data.

The training and test results after 20 epochs are as follows:

Epoch	Train Loss	Train Acc	Val Loss	Val Acc
1	1.0498	0.6271	2.1129	0.3362
2	0.9337	0.6663	1.9741	0.3788
3	0.8844	0.6804	1.9736	0.3663
4	0.8469	0.6943	1.6627	0.4238
5	0.8153	0.7038	1.7253	0.4338
6	0.7870	0.7124	1.6556	0.4438
7	0.7697	0.7174	1.5208	0.4800
8	0.7472	0.7283	1.5259	0.4763
9	0.7211	0.7351	1.5829	0.4550
10	0.6984	0.7437	1.4601	0.5038
11	0.6797	0.7505	1.6136	0.4763
12	0.6649	0.7535	1.4944	0.5075
13	0.6373	0.7642	1.5169	0.5012
14	0.6196	0.7731	1.3752	0.5537
15	0.6035	0.7749	1.4893	0.5262
16	0.5866	0.7839	1.3490	0.5700
17	0.5702	0.7908	1.4692	0.5388
18	0.5503	0.7980	1.4235	0.5413
19	0.5334	0.8013	1.5753	0.5250
20	0.5247	0.8066	1.3751	0.5938

```
Test Accuracy: 0.5938
Test F1-Score: 0.5894
Class Report:
              precision    recall  f1-score   support

          AK       0.75      0.41      0.53       100
         BCC       0.41      0.85      0.55       100
         BKL       0.53      0.55      0.54       100
          DF       0.91      0.41      0.57       100
         MEL       0.58      0.68      0.62       100
          NV       0.53      0.88      0.66       100
         SCC       0.86      0.31      0.46       100
        VASC       0.99      0.66      0.79       100

    accuracy                           0.59       800
   macro avg       0.69      0.59      0.59       800
weighted avg       0.69      0.59      0.59       800
```

The effect has become worse, and after repeated tests, the more image enhancements, the worse the effect, the accuracy of the test set has increased from 0.69 to 0.59.

Next, I will continuously adjust image enhancement and handle class imbalance issues.

version 4 - Final Version

After repeated attempts, I got the best result:

Epoch	Train Loss	Train Acc	Val Loss	Val Acc
1	1.1073	0.5845	1.0802	0.6050
2	0.7618	0.7157	0.9992	0.6488
3	0.6029	0.7781	1.0582	0.6688
4	0.5356	0.8002	1.0577	0.6663
5	0.4563	0.8300	1.2680	0.6613
6	0.4147	0.8472	1.1490	0.6900
7	0.3759	0.8598	1.2469	0.6638
8	0.3452	0.8718	1.1884	0.6875
9	0.3056	0.8863	1.1368	0.7113
10	0.2950	0.8902	1.2041	0.7050
11	0.2746	0.8970	1.2032	0.6875
12	0.2651	0.9004	1.1717	0.7087
13	0.2379	0.9102	1.3138	0.6913
14	0.2257	0.9135	1.1231	0.7125
15	0.2126	0.9197	1.1741	0.7050
16	0.1982	0.9257	1.3501	0.6875
17	0.1895	0.9287	1.1440	0.7400
18	0.1788	0.9315	1.2277	0.7288
19	0.1756	0.9330	1.3946	0.6950
20	0.1720	0.9358	1.2966	0.7150

```
Test Accuracy: 0.7150
Test F1-Score: 0.7189
Class Report:
              precision    recall  f1-score   support

          AK       0.78      0.60      0.68       100
         BCC       0.59      0.86      0.70       100
         BKL       0.64      0.68      0.66       100
          DF       0.98      0.60      0.75       100
         MEL       0.62      0.78      0.69       100
          NV       0.63      0.78      0.70       100
         SCC       0.79      0.67      0.72       100
        VASC       1.00      0.75      0.86       100

    accuracy                           0.71       800
   macro avg       0.75      0.72      0.72       800
weighted avg       0.75      0.71      0.72       800
```

The test accuracy is 0.715 which is the highest result. This version selectively retains part of image enhancement and using oversampling to reduce class imbalance, get the best test set results.