

## Classification and Research of Skin Lesions Based on Machine Learning

Jian Liu<sup>1</sup>, Wantao Wang<sup>1</sup>, Jie Chen<sup>2,\*</sup>, Guozhong Sun<sup>3</sup> and Alan Yang<sup>4</sup>

**Abstract:** Classification of skin lesions is a complex identification challenge. Due to the wide variety of skin lesions, doctors need to spend a lot of time and effort to judge the lesion image which zoomed through the dermatoscopy. The diagnosis which the algorithm of identifying pathological images assists doctors gets more and more attention. With the development of deep learning, the field of image recognition has made long-term progress. The effect of recognizing images through convolutional neural network models is better than traditional image recognition technology. In this work, we try to classify seven kinds of lesion images by various models and methods of deep learning, common models of convolutional neural network in the field of image classification include ResNet, DenseNet and SENet, etc. We use a fine-tuning model with a multi-layer perceptron, by training the skin lesion model, in the validation set and test set we use data expansion based on multiple cropping, and use five models' ensemble as the final results. The experimental results show that the program has good results in improving the sensitivity of skin lesion diagnosis.

**Keywords:** Skin lesions, deep learning, data expansion, ensemble.

### 1 Introduction

Skin lesions are a major threat to people's health, more than 5 million cases occur every year. There are many kinds of skin lesions, common skin lesions include psoriasis, eczema, vitiligo, melanoma, some of these lesions are particularly fatal, such as melanoma [Pathan, Prabhu and Siddalingaswamy (2018)]. If the doctor can detect skin lesion in advance which the patient is suffering from, whether it is benign or malignant, the process of follow-up treatment will become extremely beneficial.

In the early stage of some skin diseases, the lesions are small and therefore difficult to be observed by doctors. With the development of medical technology, there is a visual inspection technique that is easy for doctors to observe: the dermatoscopy, the advantages of dermatoscopy are not only to enlarge the lesion size of the skin [Tschandl and Wiesner

---

<sup>1</sup> School of Computer & Communication Engineering, University of Science and Technology Beijing, Beijing, 100083, China.

<sup>2</sup> Department of Breast Surgery, West China Hospital, Sichuan University, Chengdu, 610041, China.

<sup>3</sup> Dawning Information Industry Chengdu Co., Ltd., Chengdu, 610213, China.

<sup>4</sup> Amphenol AssembleTech, Houston, TX 77070, US.

\* Corresponding Author: Jie Chen. Email: chenjiadoctor@126.com.

(2018)], but also it can eliminate the interference of some light. For doctors, the probability of misdiagnosis is reduced, and the accuracy of diagnosing the types of skin lesions has also been improved to some extent. In the actual diagnosis, the dermatoscopy images are often identified by multiple doctors to determine the type of skin lesions, and the doctors spend a lot of time and energy on diagnosis. The program algorithm to assist doctors is also beginning to appear [Rebouças Filho, Peixoto, da Nóbrega et al. (2018)].

### ***1.1 Related works***

Traditional image classification often uses image preprocessing, segmentation, feature extraction and general process of image classification for image recognition. Manerkar et al. [Manerkar, Snehalatha, Harsh et al. (2016)] use C-means and watershed algorithm to segment skin lesion images, and then uses gray level co-occurrence matrix algorithm to extract data features, finally he selects Support Vector Machine as the classifier. Nezhadian et al. [Nezhadian and Rashidi (2017)] try to extract texture features for melanoma lesion images, combined with Support Vector Machine as classifier for benign and malignant melanoma classification. George et al. [George, Aldeen and Garnavi (2017)] have carried out five color space transformations on psoriasis lesion data, and selected the voting result as the final diagnosis. To achieve a good effect, Albay et al. [Albay and Kamaşak (2015)] extract the Fourier properties of the lesion boundary after segmentation of the skin lesion and serves as a classification feature.

The traditional feature extraction method has achieved good results. However, it still cannot exceed the level of human recognition. In recent years, with the hot development of deep learning, more and more new models have emerged, breaking one record after another. The usage of deep learning models to train image data has become the first choice for many research methods. Rajesh [Rajesh (2017)] use forward feedback neural network combined with ABCD rules to classify melanoma and benign skin lesions. Islam et al. [Islam, Gallardo-Alvarado, Abu et al. (2017)] want to identify eczema, impetigo and psoriasis, after using data preprocessing, feature extraction, finally the artificial neural network ANN is used to classify the data, which has achieved faster diagnosis and recognition than human doctors. Ge et al. [Ge, Demyanov, Bozorgtabar et al. (2017)] use different depth neural networks' models for skin data, then the bilinear pooling technique is adopted for some models, finally using the Support Vector Machine as classifier achieves better results.

### ***1.2 Organization***

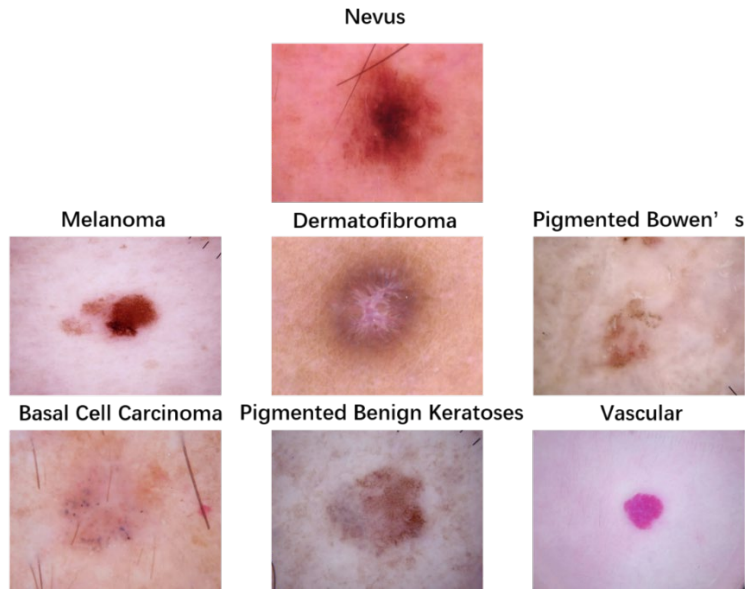
This paper is organized as follows: In Section 2.1, we discuss the skin lesion datasets and the various categories of data. In Section 2.2, we introduce the conditions of the device and implement our classification goals in three steps including choosing models, multiple cropping, ensemble. In Section 2.3, we show our experimental results. In Section 3, we have summarized the full text and make an outlook.

## **2 Experiments**

### ***2.1 Prepared***

In this article, our study is based on ISIC 2018: The Great Challenge Dataset for Skin

Lesion Analysis to Melanoma Detection [Codella, Gutman, Celebi et al. (2018); Tschandl, Rosendahl and Kittler (2018)], which includes seven types of skin lesions. Fig. 1 shows seven different skin lesions. There is a total of 10015 images in the entire data set, image size is uniform to 450×600, 3-channel color image. Tab. 1 includes the seven types in the data set and the corresponding number of pictures.



**Figure 1:** The pictures of Seven types skin lesion

**Table 1:** ISIC 2018 skin lesion dataset

Skin lesion class	Number per class
Melanoma	1113
Melanocytic Nevus	6705
Basal Cell Carcinoma	514
Actinic Keratosis/Bowen's disease	327
Benign keratosis	1099
Dermatofibroma	115
Vascular lesion	142

Due to the extremely unbalanced data volume of these kinds of classes, it brings certain challenges to the classification. The class with the most data is 58 times the class with the least amount of data. Huge data differences can have a big impact on model training, and this is one of the problems which we want to solve.

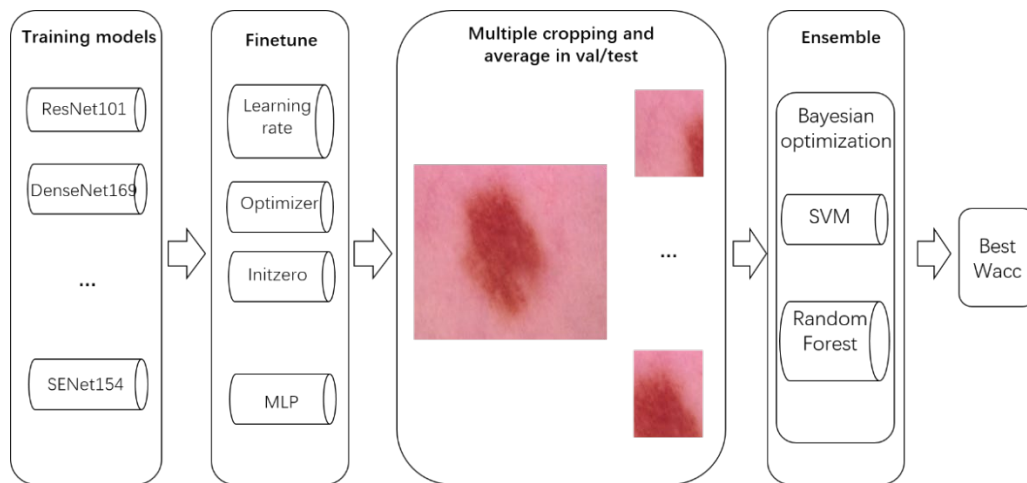
**2.2 Methods**

We randomly divide the data set into train set, validation set, test set. The number ratio is 8:1:1. In the original data set image size is 450×600, our network model is a pre-trained

classification model on ImageNet, the model has a batch size of 20 on 4 TITAN X. In the data enhancement section, we use random horizontal/vertical flipping, normalization, and randomly cut out 224×224 size from the original image size of 450×600. Our learning rate is 0.0001, the whole training contains 150 epochs. Considering the huge difference between the 7 types of pictures in the dataset, we choose the weighted loss optimizer with unbalanced class, Eq. (1) represents this method.

$$\omega_i = \frac{\sum x_i}{x_i}, \quad i \in [1, \dots, 7] \quad (1)$$

where  $\omega_i$  represents the optimization weight of each class,  $x_i$  represents the number of each class, and then we use the softmax for classification. In the process of training, we save the model parameters of the best sensitivity, and we hope to get higher sensitivity by loading the validation set and test set. Fig. 2 shows the whole process of the algorithm.



**Figure 2:** Algorithm process diagram

In the course of the research, we use the following three steps:

**Step 1:** Model Selection and Finetune.

Deep learning uses a convolutional neural network (CNN) model to identify images. With the development of deep learning, many excellent models have emerged. We will briefly introduce several models and the reasons why we choose them.

Firstly, the ResNet [He, Zhang, Ren et al. (2016)] model is proposed when image recognition in deep learning field is caught in a bottleneck, it proposes a concept such as shortcut connection and practices it into the model. The proposed method solves the problem that the accuracy rate decreases as the number of network layers deepens.

Secondly, the DenseNet [Huang, Liu, Van Der Maaten et al. (2017)] model is proposed in the concept of ResNet, the difference between ResNet and DenseNet is that DenseNet connects each layer to the dimensions of all the previous layers by using shortcut connection, the optimization of the structure reduces many parameters and calculations,

making the anti-over-fitting effect better.

Next, the SENet [Hu, Shen and Sun (2017)] model proposes a new block structure named SE block, using squeeze to compress each feature layer in the model structure and using excitation to capture feature channel dependencies, the SE block can be combined with many existing models.

And the ResNeXt [Xie, Girshick, Dollár et al. (2017)] model does not gradually deepen the network like other models, it optimizes the hyperparameters and divides the same large filter into a corresponding number of small filters to reduce the number of hyperparameters and to improve the accuracy, in this paper, we use the ResNeXt model combined with SE block.

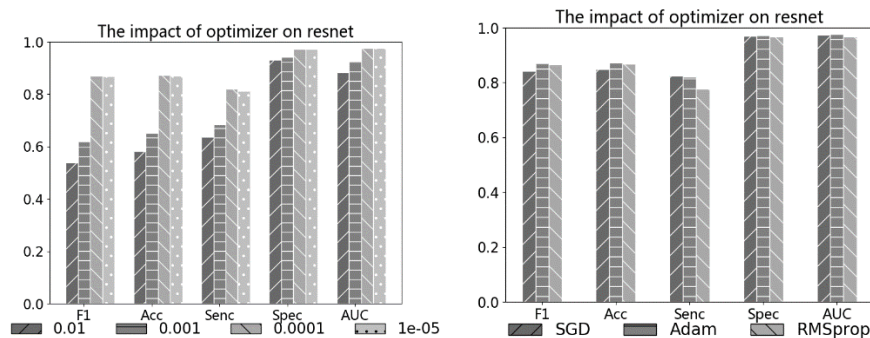
Finally, the DPN [Chen, Li, Xiao et al. (2017)] model combines the characteristics of ResNext and DenseNet models, and the improvement in accuracy is not particularly large, but it is mainly optimized in parameter optimization and computational overhead.

We select the above five models that have performed well in the classification field [Cui, McIntosh and Sun (2018)] in recent years, the reason why we choose these models is that they have large depths and enough parameters, so the image features are more fully learned. In addition, some models can reduce the influence of over-fitting and under-fitting, and some models can drop partially unimportant parameters to reduce the amount of calculation. Tab. 2 shows the performance of these models on ImageNet.

**Table 2:** Pretrained models on ImageNet valset

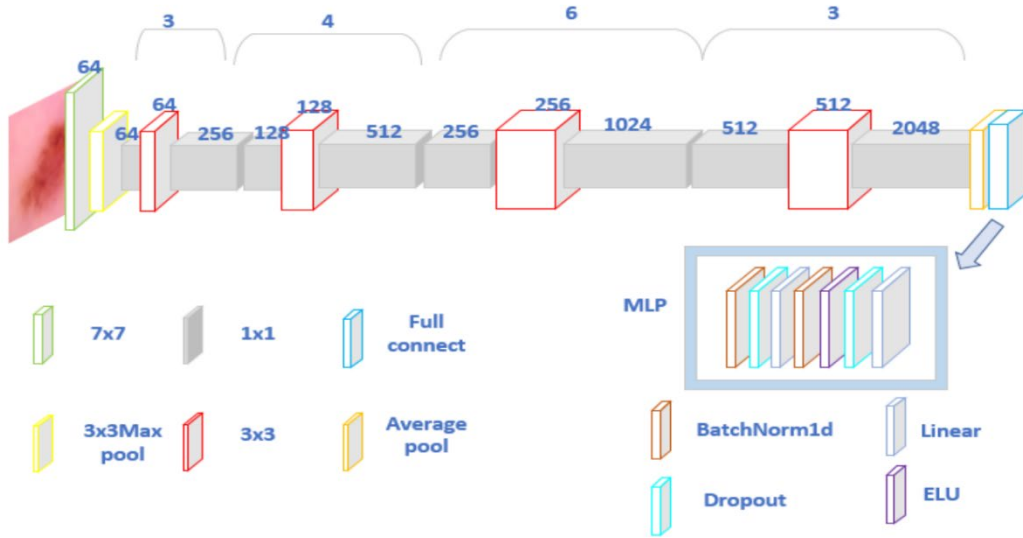
models	ACC(Top1)	ACC(Top5)
DenseNet169	0.760	0.930
ResNet101	0.774	0.937
SENet154	0.813	0.955
SE-Resnext101	0.802	0.950
DPN68b	0.770	0.936

In the fine-tuning, we take the resnet101 model as an example, respectively for learning rate, optimizer as Fig. 3 shows. We can get the following graphs. we finally choose a learning rate of 0.0001 and Adam’s optimizer.



**Figure 3:** Fine-tuning Resnet101 (learning rate and optimizer)

In addition, we use the zero-initialize method for the resnet model, zero-initialize is used for the last BatchNorm layer in each residual branch, the accuracy and sensitivity of the model have increased slightly. We also made a small change to the model structure, Fig. 4 is a resnet50 structure commonly used in classification, at the same time, it shows that the fully connected layer is changed to a multilayer perceptron structure (MLP) by fine tuning.



**Figure 4:** Resnet50 model structure (add multilayer perceptron)

**Step 2:** Multiple Cropping for Validation and Test set.

In the validation set and test set, we use the data expansion method, we use multiple cropping method to obtain  $N$  images of different positions for validation, and average  $N$  results, we use the following formula to cut  $N$  copies around the original image center.

$$\begin{cases} P_{[k,0]} = \frac{x}{2} + (m-x) * \frac{i}{\sqrt{N}-1}, \\ P_{[k,1]} = \frac{y}{2} + (n-y) * \frac{j}{\sqrt{N}-1}, \end{cases} \quad k \in [0, \dots, N-1], i, j \in [0, \dots, \sqrt{N}-1]. \quad (2)$$

where  $(P_{[k,0]}, P_{[k,1]})$  coordinate value represents the center coordinates of the crop,  $N$  represents the number of crops,  $(x, y)$  represents the size you want to crop,  $(m, n)$  represents the size of the original image, so the cropped image position for validation and test set is:

$$BOX_k = (P_{[k,0]} - \frac{x}{2}, P_{[k,1]} - \frac{y}{2}, P_{[k,0]} + \frac{x}{2}, P_{[k,1]} + \frac{y}{2}), k \in [0, \dots, N-1] \quad (3)$$

We average the cropped  $N$  pictures as the result, by using the following formula:

$$\overline{pred} = \frac{\sum_{k=0}^{N-1} pred[k]}{N}, \quad k \in [0, \dots, N-1] \tag{4}$$

See Fig. 5, we compare N to none, 16, 36 and 64 cases, where the value of 36 is the best. using this method, the performance is improved a lot compared to the various indicators obtained by using a single random cut. By cutting the validation set and the test set around the center into multiple copies, 36 images of different positions are obtained. The randomness obtained by multiple cropping is reduced, and the lesion image recognition is more accurate for some image lesions.

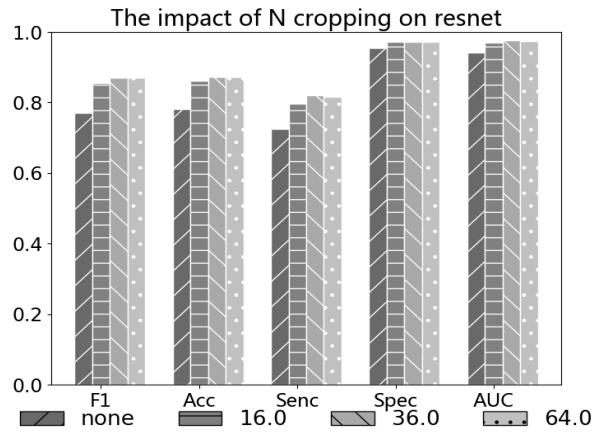


Figure 5: Different N cropping for Resnet101

According to this method, we obtain various metrics for different models on the validation set and test set, and calculate the mean. The results of DenseNet121, DenseNet169, DenseNet201, ResNet50, ResNet101, ResNet152, SENet154, SE-Resnext101, DPN68b and ResNet101MLP are obtained in turn, as Tab. 3 shows.

Table 3: Validation/Test set pass models' mean metrics

Models	SEN	ACC	F1	SPE	AUC
DenseNet121	0.823/0.812	0.822/0.810	0.820/0.813	0.967/0.964	0.970/0.971
DenseNet169	0.804/0.801	0.862/0.872	0.859/0.871	0.970/0.970	0.976/ <b>0.976</b>
DenseNet201	0.828/0.817	0.868/0.862	0.862/0.861	0.972/0.969	0.976/0.974
ResNet50	0.821/0.810	0.872/0.868	0.865/0.869	0.973/0.965	0.970/0.973
ResNet101	0.825/0.819	0.862/0.860	0.859/0.860	0.971/0.968	0.967/0.971
ResNet152	0.819/0.801	0.870/0.865	0.869/0.870	0.972/0.969	0.971/0.972
SENet154	0.855/0.800	<b>0.886/0.881</b>	<b>0.884/0.880</b>	<b>0.976/0.971</b>	<b>0.980/0.970</b>
SE-Resnext101	0.844/0.790	0.881/0.872	0.881/0.874	0.972/0.967	0.976/0.970
DPN68b	0.828/0.801	0.861/0.852	0.860/0.851	0.968/0.964	0.973/0.970
ResNet101MLP	<b>0.859/0.831</b>	0.814/0.805	0.808/0.791	0.968/0.962	0.972/0.975

From the results of these models, the best overall model is SENet154, the model with the best sensitivity is the resnet101 model with MLP structure. However, we hope to get higher sensitivity results at the same time. we use model ensemble to combine the advantages of each model to get better results.

**Step 3:** Grid Search and Support Vector Machine classifier.

For the results of validation set, we adopt an approach of ensemble combined with search strategy of Bayesian optimization. From the above model, we select the following five models for the ensemble, including DenseNet201, SENet154, SE-Resnext101, DPN68b and ResNet101MLP. The vectors are extracted before passing the softmax module, we connect all the vectors from the validation set through these five models, and train these vectors with Support Vector Machine or RandomForest as a classifier. In the field of hyperparameter automatic search, there are common grid search, random search and bayesian optimization search, but the first two searches are not efficient, so we adopt bayesian optimization search, we will introduce the implementation process of Bayesian optimization Algorithm 1.

The pseudo algorithm using Bayesian optimization is as follows:

---

**Algorithm 1** Sequential Model-Based Optimization

---

**Input:**  $f, X, S, M$

- 1:  $D \leftarrow \text{InitSamples}(f, X)$
  - 2: **for**  $i \leftarrow |D|$  to  $T$  **do**
  - 3:      $p(y|x, D) \leftarrow \text{FitModel}(M, D)$
  - 4:      $x_i \leftarrow \text{argmax}_{x \in X} S(x, p(y|x, D))$
  - 5:      $y_i \leftarrow f(x_i)$
  - 6:      $D \leftarrow D \cup (x_i, y_i)$
  - 7: **end for**
- 

where  $f$  is the unknown function relationship,  $X$  is the input data,  $S$  is acquisition Function,  $M$  is model based on input data hypothesis, firstly we get the initialized data set based on the input data, then make a loop to select  $T$  times parameters. The model we chose is based on Gaussian distribution ,the mean  $\mu$  and covariance  $K(x, x^*)$  of a Gaussian function are fixed, formulated as follows:  $f \sim GP(\mu, K)$ , when the Gaussian process is used as a priori for Bayesian inference, the posterior function can be used to predict new data, we suppose  $y$  is a function value known by training data,  $y_*$  is the function value of the test set input  $x_*$ ,  $\bar{\mu}$  is the mean of training set,  $\bar{\mu}_*$  is the mean of test set,  $\Sigma_*$  is the covariance of the training set,  $\Sigma_{**}$  is the covariance of the test set. Eq. (5) shows the relationship.



$$\begin{bmatrix} f \\ f_* \end{bmatrix} \sim N\left(\begin{bmatrix} \bar{\mu} \\ \bar{\mu}_* \end{bmatrix}, \begin{bmatrix} \Sigma & \Sigma_* \\ \Sigma_*^T & \Sigma_{**} \end{bmatrix}\right) \tag{5}$$

Gaussian process extends multivariate Gaussian distribution to infinite dimension, a training set  $y$  can be represented as a sample taken from a multivariate Gaussian distribution:  $y = [y_1, y_2, \dots, y_n]^T$ . We set the mean of the Gaussian process to 0 and the most common choice for covariance is the squared exponential, See Eq. (6):

$$k(x, x') = \sigma_f^2 \exp\left(\frac{-(x - x')^2}{2l^2}\right) \tag{6}$$

Due to the existence of noise, we express the formula as Eqs. (7), (8):

$$y = f(x) + N(0, \sigma_n^2) \tag{7}$$

$$k(x, x') = \sigma_f^2 \exp\left(\frac{-(x - x')^2}{2l^2}\right) + \sigma_n^2 \sigma(x, x') \tag{8}$$

where  $\sigma(x, x')$  is the KroneckerDelta function, in addition to calculating the covariance of the training set  $K$ , see Eq. (9), we also need to calculate the covariance between the new independent variable and the training set independent variable  $K_*$  Eq. (10) and the covariance of the new independent variable  $K_{**}$  Eq. (11). Then Eq. (12) shows the relationship.

$$K = \begin{bmatrix} k(x_1, x_1) & k(x_1, x_2) & \dots & k(x_1, x_n) \\ k(x_2, x_1) & k(x_2, x_2) & \dots & k(x_2, x_n) \\ \vdots & \vdots & \ddots & \vdots \\ k(x_n, x_1) & k(x_n, x_2) & \dots & k(x_n, x_n) \end{bmatrix} \tag{9}$$

$$K_* = [k(x_*, x_1), k(x_*, x_2), \dots, k(x_*, x_n)] \tag{10}$$

$$K_{**} = k(x_*, x_*) \tag{11}$$

$$\begin{bmatrix} y \\ y_* \end{bmatrix} \sim N\left(0, \begin{bmatrix} K & K_*^T \\ K_* & K_{**} \end{bmatrix}\right) \tag{12}$$

The training set obeys a multidimensional normal distribution, according to  $K$ , we can know that the posterior probability of the test set  $y_*$  is  $p(y_* | y \sim N(\bar{\mu}, \hat{\sigma}^2))$ , the mean  $\bar{\mu}$  and variance  $\hat{\sigma}^2$  of  $y_*$  are expressed as follows Eq. (13), Eq. (14):

$$\bar{\mu} = K_* K^{-1} y \tag{13}$$

$$\hat{\sigma}^2 = K_{**} - K_* K^{-1} K_*^T \tag{14}$$

The hyperparameter to be determined is  $\theta = [\sigma_f^2, l]$ , since the training set obeys a multidimensional normal distribution, the likelihood function is Eq. (15):

$$\begin{aligned} L &= \log p(y | x, \theta) \\ &= -\frac{1}{2} \log |K| - \frac{1}{2} (y - \mu)^T K^{-1} (y - \mu) - n * \log(2\pi) / 2 \end{aligned} \quad (15)$$

Bayesian optimization maps  $x$  to the real space  $R$  through the Acquisition function, indicating the probability that the objective function value of the point can be larger than the current optimal value. the two main types of acquisition functions are commonly used, the first is probability of improvement, see Eq. (16).

$$\begin{aligned} POI(X) &= P(f(X) \geq f(X^+) + \xi) \\ &= \Phi\left(\frac{\mu(x) - f(X^+) - \xi}{\sigma(x)}\right) \end{aligned} \quad (16)$$

where  $f(X)$  is the value of the  $X$  objective function,  $f(X^+)$  is the optimal  $X$  objective function value so far,  $\mu(x)$ ,  $\sigma(x)$  are the mean and variance of the objective function obtained by the Gaussian process, respectively,  $\xi$  is the trade-off factor which adjust to select the points around  $X^+$ . In general, we use MonteCarlo simulation method to find  $X$  so that  $POI(X)$  is the largest.

The second is expected improvement. The POI is a probability function, so only the probability that  $f(x)$  is larger than  $f(x^+)$  is considered, and expected improvement is a desired function, so it is considered how much  $f(x)$  is larger than  $f(x^+)$ . We get  $x$  by the following Eq. (17).

$$x = \arg \max_x E(\max\{0, f_{t+1}(x) - f(X^+)\} | D_t) \quad (17)$$

where  $D_t$  is the first  $t$  samples, under the premise of normal distribution, we can get the following Eq. (18):  $\text{expression}(Z = \frac{\mu(x) - f(x^+)}{\sigma(x)})$ :

$$EI(x) = \begin{cases} (\mu(x) - f(x^+))\Phi(Z) + \sigma(x)\phi(Z), & \text{if } \sigma(x) > 0 \\ 0, & \text{if } \sigma(x) = 0 \end{cases} \quad (18)$$

For the classifier model, we use SVM and RandomForest, SVM configuration is as follows, we use 10-fold cross-validation. For unbalanced data sets, we also use class balance weighted, we use sensitivity as the main evaluation indicator. Search parameters include  $C$  and kernel, the best performance parameters are the following values:  $C=[0.1, 1000]$ , kernel=['linear', 'poly', 'rbf', 'sigmoid']. The parameter control for the SVM classifier is mainly from the following formula:

According to validation set vectors  $x_i \in R^p, i=1, \dots, n$ , and a vector  $y \in \{1, -1\}^n$ , the SVM formula can be expressed as Eq. (19):

$$\begin{cases} tar = \min_{\omega, b, \zeta} \frac{1}{2} \omega^T \omega + C \sum_{i=1}^n \zeta_i, & \zeta_i \geq 0, j = 1, \dots, n \\ y_i(\omega^T \phi(x_i) + b) \geq 1 - \zeta_i, \end{cases} \quad (19)$$

Due to the duality, this formula is equivalent to the following Eq. (20):

$$\begin{cases} tar = \min_{\alpha} \frac{1}{2} \alpha^T Q \alpha - e^T \alpha, & a \leq \alpha_i \leq C, j = 1, \dots, n \\ \text{subject to } y^T \alpha = 0, \end{cases} \quad (20)$$

where  $e$  is the vector of all ones,  $Q$  is a  $n$  by  $n$  positive semidefinite matrix.  $Q_{ij} = y_i y_j K(x_i, x_j)$ . Validation set vectors are implicitly mapped into a higher dimensional space by the function  $\phi$ . The decision function is Eq. (21):

$$dec = \text{sgn}\left(\sum_{i=1}^n y_i \alpha_i K(x_i, x) + \rho\right) \quad (21)$$

where  $K(x_i, x_j) = \phi(x_i)^T \phi(x_j)$  is the kernel.  $C > 0$  is the upper bound, we change the value to get best results.

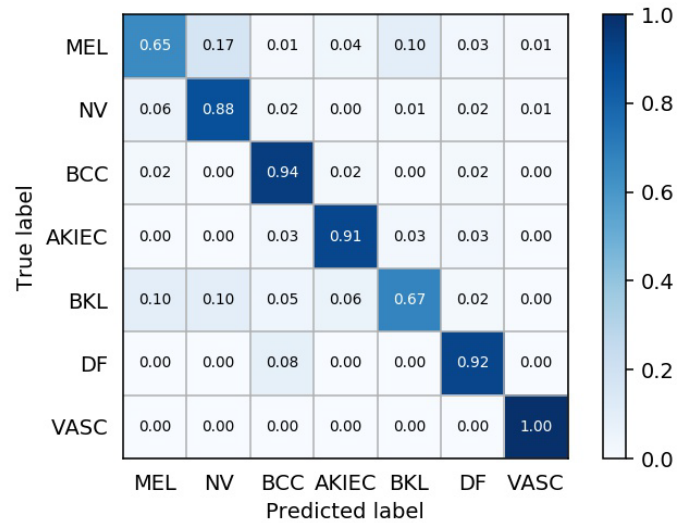
Next, we consider the classification model. The random forest model is a combination of the bagging model plus decision trees which creates multiple subtrees by splitting features. The difference is that decision trees usually generate nodes and rules by calculating the information gain and the Gini index. In contrast, random forests are random. Deeper decision trees tend to have over-fitting problems, while random forests can prevent most situations by creating random subsets of features and using them to build smaller trees, which then form subtrees, this method can prevent overfitting in most cases.

In this experiment, we select the hyperparameter range to include the following values:  $N$ (the number of trees in the forest)=[10, 500],  $MinSamplesSplit$  (the minimum number of samples required to split an internal node)=[2, 100],  $MaxFeatures$  (the number of features to consider when looking for the best split)=[0.1, 0.999],  $MaxDepth$  (the maximum depth of the tree)=[5, 80], to find the best sensitivity value by Bayesian optimization combined with random forest model.

### 2.3 Results

According to the results of Bayesian optimization, we use SVM classifier to search for hyperparameters  $C$  and kernel, the best performing set of coefficients is  $C=10$  and kernel='rbf', the result of the test set after passing the best set of SVM classifiers is  $SEN=0.834$ ,  $ACC=0.842$ ,  $F1=0.837$ ,  $SPE=0.966$ ,  $AUC=0.900$ . For random forest classifier, the best performing set of coefficients is  $N=255$ ,  $MaxFeatures=0.1385$ ,  $MaxDepth=75$  and  $MinSamplesSplit=100$ , the result of the test set after passing the best set of random forest classifier is  $SEN=0.846$ ,  $ACC=0.840$ ,  $F1=0.833$ ,  $SPE=0.968$ ,  $AUC=0.907$ . By the comparison of the two classifiers, the random forest classifier with better sensitivity is selected as the final result.

Confusion matrix as shown below:



**Figure 6:** Confusion Matrix of test set

### 3 Discuss and conclusion

In this work, we have classified the unbalanced data types of skin lesions. We finally choose a method for ensemble of multiple models. We use the weighted loss optimization for unbalanced data during training. The correct classification of the model is greatly promoted. In the validation and test set, the method of multiple cropping is used to verify that the image with the image size of  $450 \times 600$  is cropped into multiple copies of  $224 \times 224$  for average around the center, and the evaluation results of various metrics are obviously improved. Finally, we connect the vectors which the validation set passed multiple models, and train a best-performing of sensitivity classifier by using the Bayesian optimization search hyperparameter method.

The research on the classification of skin lesions is worthy of more trials, there are still many shortcomings in our work. In the future, we will do further research and make a progress on classification methods, and we are committed to helping doctors reduce the fatigue caused by diagnosis.

**Acknowledgement:** This work is supported by Intelligent Manufacturing Standardization Program of Ministry of Industry and Information Technology (No. 2016ZXFB01001).

**Conflicts of Interest:** The authors declare that they have no conflicts of interest to report regarding the present study.

### References

- Albay, E.; Kamaşak, M. (2015): Skin lesion classification using fourier descriptors of lesion borders. *Medical Technologies National Conference*, pp. 1-4.
- Chen, Y.; Li, J.; Xiao, H.; Jin, X.; Yan, S. et al. (2017): Dual path networks. *Advances*

*in Neural Information Processing Systems*, pp. 4467-4475.

**Codella, N. C.; Gutman, D.; Celebi, M. E.; Helba, B.; Marchetti, M. A. et al.** (2018): Skin lesion analysis toward melanoma detection: a challenge at the 2017 international symposium on biomedical imaging (ISBI), hosted by the international skin imaging collaboration (ISIC). *IEEE 15th International Symposium on Biomedical Imaging*, pp. 168-172.

**Cui, Q.; McIntosh, S.; Sun, H.** (2018): Identifying materials of photographic images and photorealistic computer generated graphics based on deep CNNs. *Computers, Materials & Continua*, vol. 55, no. 2, pp. 229-241.

**Ge, Z.; Demyanov, S.; Bozorgtabar, B.; Abedini, M.; Chakravorty, R. et al.** (2017): Exploiting local and generic features for accurate skin lesions classification using clinical and dermoscopy imaging. *IEEE 14th International Symposium on Biomedical Imaging*, pp. 986-990.

**George, Y.; Aldeen, M.; Garnavi, R.** (2017): A pixel-based skin segmentation in psoriasis images using committee of machine learning classifiers. *International Conference on Techniques and Applications*, pp. 1-8.

**He, K.; Zhang, X.; Ren, S.; Sun, J.** (2016): Deep residual learning for image recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770-778.

**Hu, J.; Shen, L.; Sun, G.** (2017): Squeeze-and-excitation networks. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, vol. 7, pp. 7132-7141.

**Huang, G.; Liu, Z.; Van Der Maaten, L.; Weinberger, K. Q.** (2017): Densely connected convolutional networks. *CVPR*, vol. 1, pp. 4700-4708.

**Islam, M. N.; Gallardo-Alvarado, J.; Abu, M.; Salman, N. A.; Rengan, S. P. et al.** (2017): Skin disease recognition using texture analysis. *Control and System Graduate Research Colloquium*, pp. 144-148.

**Manerkar, M. S.; Snekhalatha, U.; Harsh, S.; Saxena, J.; Sarma, S. P. et al.** (2016): Automated skin disease segmentation and classification using multi-class SVM Classifier. *Electrical, Electronics, Engineering Trends, Communication, Optimization and Sciences, 3rd International Conference*, pp. 1-6.

**Nezhadian, F. K.; Rashidi, S.** (2017): Melanoma skin cancer detection using color and new texture features. *Artificial Intelligence and Signal Processing Conference*, pp. 1-5.

**Pathan, S.; Prabhu, K. G.; Siddalingaswamy, P.** (2018): Techniques and algorithms for computer aided diagnosis of pigmented skin lesions-a review. *Biomedical Signal Processing and Control*, vol. 39, pp. 237-262.

**Rajesh, A.** (2017): Classification of malignant melanoma and benign skin lesion by using back propagation neural network and ABCD rule. *IEEE International Conference on Electrical, Instrumentation and Communication Engineering*, vol. 6, pp. 538-548.

**Rebouças Filho, P. P.; Peixoto, S. A.; da Nóbrega, R. V. M.; Hemanth, D. J.; Medeiros, A. G. et al.** (2018): Automatic histologically-closer classification of skin lesions. *Computerized Medical Imaging and Graphics*, vol. 68, pp. 40-54.

**Tschandl, P.; Rosendahl, C.; Kittler, H.** (2018): The ham10000 dataset: a large

collection of multi-source dermatoscopic images of common pigmented skin lesions. *Scientific Data*, vol. 5.

**Tschandl, P.; Wiesner, T.** (2018): Advances in the diagnosis of pigmented skin lesions. *British Journal of Dermatology*, vol. 178, no. 1, pp. 9-11.

**Xie, S.; Girshick, R.; Dollár, P.; Tu, Z.; He, K.** (2017): Aggregated residual transformations for deep neural networks. *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5987-5995.