Report of Project IMA205 Challenge 2019: Classify images as either melanoma or benign nevus

Fengli LIN

I. INTRODUCTION

A skin lesion is defined as a superficial growth or patch of the skin that is visually different and/or has a different texture than its surrounding area. Skin lesions, such as moles or birthmarks, can degenerate and become melanoma, one of the deadliest skin cancer. Its incidence has been increasing during the last decades, especially in the areas mostly populated by white people.

The most effective treatment is an early detection followed by surgical excision. This is why several approaches for melanoma detection have been proposed in the last years (non-invasive computer-aided diagnosis (CAD)).

In this project, I build an automated classification of skin lesions using images. The final classification model is an ensemble result using the stacking technique. The detailed models can be mainly divided into two parts: 1) Features-based classical machine learning methods and 2) Deep Convolutional Neural Network(DCNN) Based methods.

For the first part of models, I extract a 73 dimension feature vector for each input image, which is used as the model input for various classifiers including SVM, Logistic Regression, Random Forest, Extra Tree Classifier, AdaBoost and XGBoost. These feature vectors embed the Shape, Color, Histogram and Texture information, which are the most common features used in human diagnosis.

For the second part of the models, I use the technique of transfer learning based on DCNN models. Deep convolutional neural networks (CNNs) show potential for general and highly variable tasks across many fine-grained object categories. CNN can make classification of skin lesions by training end-to-end from images directly, using only pixels and disease labels as inputs, which can avoid complex feature extraction process. I utilize a GoogleNet Inception v3 CNN architecture that was pre-trained on approximately 1.28 million images (1,000 object categories) from the 2014 ImageNet Large Scale Visual Recognition Challenge, and train it on our dataset. Figure 1 shows the working system. The CNN is trained and validated using 700 binary labeled images.

The final result shows that the combining of classical models with DCNN models can effective improve the testing score, which reaches 0.44131 using the Matthews correlation coefficient.

II. CLASSICAL FEATURE-BASED MACHINE LEARNING METHODS

A. *Feature extraction*

Inspired by Asymmetry, Border, Colour and Diameter characteristics of the lesion under study(ABCD rules) [4] [3], I extract following 4 categories: Shape, Color, Histogram and Texture.

Since the datasets already provide us the segmentation results, the feature extraction process is based on segmented image(i.e. image_mul_mask).

*1) **Shape features**:* Shape information is very important in judging a skin lesion. Therefore, I calculate following 5 features to embed shape feature.

Since the shape features are irrelevant with colors, we use gray value image to extract these features. Besides, I use *cv2.findContours* to get the contour points.

- **Area ratio between fitting ellipse and contour:** Most of the lesions match ellipse shape so I made a best fit ellipse on the lesion and then compared its deviation from our best fit ellipse and judged it on a factor determined by taking the ratio.
- **Length ratio between longest axis and its perpendicular axis:** This feature is used to measure the asymmetry information.
- **Area ratio between between fitting rectangle and contour**
- **Area ratio between between fitting convex hull and contour**
- **Ratio between perimeter and contour area:** This feature measures the border irregularity. Because with the same area size, the more irregular border has longer perimeter.

*2) **Color features**:* Color feature is also very important in this classification task. Hence, I use 12 features to represent the color information.

I calculate following features on RGB channels respectively.

- **minimum intensity**
- **maximum intensity**
- **intensity mean**
- **intensity standard deviation**

*3) **Histogram features**:* In addition to above color features, I also use histogram features to capture the distribution information of color intensity. There is 32 dimension in histogram features.

I calculate following features on RGB and gray value channels respectively.

- **Calculated histogram representing by a 8-dimension vector**

*4) **Texture features**:* Here I attempt to capture local spatial information in the skin lesions. I crop the segmented lesion using above fitting rectangle to reduce redundant zero regions. For each rectangle, a feature vector containing spatial descriptors that consist of image textures is computed. I use textures in an attempt to discriminate between some of the anatomical structures that dermatologists consider (e.g., the D part of the ABCD rule corresponds to the presence of up to five structural features: network, structureless (or homogeneous) areas, branched streaks, dots, and globules). Texture should at a minimum be invariant to rotation, and not very sensitive to acquisition issues. We focus on the use of uniform rotation invariant Gray-Level Co-Occurrence Matrix (GLCM), computed from the gray-scale version of the images.

Next, I Compute a 24-dimension feature of a grey level co-occurrence matrix to serve as a compact summary of the matrix. The properties are computed as follows:

- **contrast**

$$\sum_{i,j=0}^{levels-1} P_{i,j}(i-j)^2$$

- **dissimilarity**

$$\sum_{i,j=0}^{levels-1} P_{i,j}|i-j|$$

- **homogeneity**

$$\sum_{i,j=0}^{levels-1} \frac{P_{i,j}}{1+(i-j)^2}$$

- **energy**

$$\sum_{i,j=0}^{levels-1} P_{i,j}^2$$

- **correlation**

$$\sqrt{ASM}$$

- **ASM**

$$\sum_{i,j=0}^{levels-1} P_{i,j}\left[\frac{(i-\mu_i)(j-\mu_j)}{\sqrt{(\sigma_i^2)(\sigma_j^2)}}\right]$$

B. Feature processing

*1) **Scale**:* The first feature processing step to scale the data such that each feature has average equal to 0 and unit variance, which is necessary for some processing(i.e. PCA) and it could help in speeding up the calculations in following algorithm.

*2) **PCA**:* PCA is a technique used to derive low number of features from a set of high dimensional features. Lets say we have a data set with m independent variables. PCA projects these m variables into n new dimensions (n¡m), such that the n variables explain most of the variability in the original data set.

PCA can be used to extract a low number of uncorrelated features from a set of high number of features, while capturing most of the information from the original data set.

I try the PCA algorithm to reduce the dimension of obtained feature and compare the result with original feature based on classification score.

*3) **Forward feature selection**:* I also try the Forward feature selection algorithm to select features and compare the result with original feature based on classification score.

Forward feature selection is one of the simplest sub-optimal step-wise search strategies. It iteratively identifies the best feature subset obtained by adding to the current feature subset one feature at a time. The search depth (maximum number of features selected) was empirically set to 10 features in our application.

C. Models

I train 7 basic classifier respectively using the GridSearchCV(CV=5) method in Sklearn to obtain the best parameter setting. The best parameters are shown in the Jupyter notebook code. To be notice that, due to the unbalance of training data(benign:melanoma=418:282), I explicitly set *class_weight* as 'balanced' in most models.

After obtaining those basic model, I also use the stacking method to build an ensemble classifier using the VotingClassifier method in Sklearn.

*1) **SVM classifier**:* SVM is a supervised machine learning algorithm which can be used for classification or regression problems. It uses a technique called the kernel trick to transform our data and then based on these transformations it finds an optimal boundary between the possible outputs.

The major advantage of the SVM methodology is that it can be paired with the kernel trick. So, theoretically, by exploring and fine tuning kernels we may create appropriate feature spaces, where the linear classification is able to classify data created by non-linear phenomena.

*2) **K Nearest Neighbors classifier**:* KNN is a non-parametric, lazy learning algorithm. Its purpose is to use a database in which the data points are separated into several classes to predict the classification of a new sample point.

I mainly use KNN to enrich the stacking models and reduce its variance.

*3) **Logistic regression classifier**:* Logistic regression measures the relationship between the categorical dependent variable and one or more independent variables by estimating probabilities using a logistic function, which is the cumulative logistic distribution.

*4) **Random forest classifier& Extra Tree classifier**:* Random forest builds multiple decision trees and merges them together to get a more accurate and stable prediction. The Extra-Tree method (standing for extremely randomized trees) has the main objective of further randomizing tree building in the context of numerical input features, where the choice of the optimal cut-point is responsible for a large proportion of the variance of the induced tree.

It turns out these two model has the best performance among single classical models because of their ability to reduce variance and avoid overfitting.

*5) **AdaBoost classifier**:* Ada-boost or Adaptive Boosting is one of ensemble boosting classifier proposed by Yoav Freund and Robert Schapire in 1996. It combines multiple classifiers to increase the accuracy of classifiers. AdaBoost is an iterative ensemble method. AdaBoost classifier builds a strong classifier by combining multiple poorly performing classifiers so that we will get high accuracy strong classifier.

However, due to the limits of data size and features, AdaBoost doesn't show very impressing performance in this task.

*6) **XGBoost classifier**:* XGBoost is an optimized distributed gradient boosting library designed to be highly efficient, flexible and portable. It implements machine learning algorithms under the Gradient Boosting framework. XGBoost provides a parallel tree boosting (also known as GBDT, GBM) that solve many data science problems in a fast and accurate way.

XGBoost is one of the most popular machine learning methods in Kaggle contests. However, due to its numerous parameters and limited time, I may not find the best parameter setting and the testing result is also not so impressing.

*7) **Stacking classifier**:* Ensemble learning helps improve machine learning results by combining several models. This approach allows the production of better predictive performance compared to a single model. That is why ensemble methods placed first in many prestigious machine learning competitions, such as the Netflix Competition, KDD 2009, and Kaggle.

After trying the stacking method, I obtain big improvement compared to single model.

III. Deep Convolutional Neural Networks(DCNN)

In this part, I attempt to fine-tune deep convolution neural networks that have succeeded with ImageNet dataset for classifying 2 types of skin lesion using Keras [1] and 700 dermatoscopic images. [2] Fine-tuning the top layers was performed with Inception V3.

A. Baseline CNN

Before fine-tuning DCNNs, I build a small CNN to estimate the difficulty of classifying skin lesions. The architecture of the CNN is as follow:

1) A convolutional layer with 16 kernels, each of size 3 and padding such that the size of the image is maintained.
2) A max pooling layer with window 2x2. The output is feature maps with spatial activation size reduced by a factor of 2
3) A convolutional layer with 32 kernels, each of size 3 and padding to maintain the same size.
4) A max pooling layer with window 2x2. The output is feature maps with spatial activation size reduced by a factor of 2.
5) A convolutional layer with 64 kernels, each of size 3 and padding to maintain the same size.
6) A max pooling layer with window 2x2. The output is feature maps with spatial activation size reduced by a factor of 2.

The architecture of this model is heuristically based. I follow the convention in famous DCNNs: using the smallest (3x3) convolutional layers; and double the number of filters in the output whenever the spatial activation size is halved to maintain roughly constant hidden dimensions. To train this model, data augmentation is employed. The intuition of this method is to transform the training dataset a bit in each epoch to produce variation and to guarantee that the model will never see the same image twice. Learning rate is initialized at 0.01 and Adam optimizer is used. Learning rate decay is also used so that the learning rate will halve whenever the validation accuracy plateaus for 3 epochs. Baseline model is trained for a total of 40 epochs.

B. InterceptionV3

Inception V3 was the top performers on ImageNet with 0.937 accuracy for top-5 and 0.779 for top-1. The namesake of Inception v3 is the Inception modules it uses, which are basically mini models inside the bigger model. The inspiration comes from the idea that you need to make a decision as to what type of convolution you want to make at each layer: Do you want a 33? Or a 55? The idea is that you dont need to know ahead of time if it was better to do, for example, a 33 then a 55. Instead, just do all the convolutions and let the model pick whats best. Additionally, this architecture allows the model to recover both local feature via smaller convolutions and high abstracted features with larger convolutions. The larger convolutions are more computationally expensive, so [4] suggests first doing a 11 convolution reducing the dimensionality of its feature map, passing the resulting feature map through a ReLU, and then doing the larger convolution (in this case, 55 or 33). The 11 convolution is key because it will be used to reduce the dimensionality of its feature map.

To fine-tune InterceptionV3, the top fully-connected layers are removed, and new fully-connected layers (consisting of: one global max pooling layers, one fully connected layer with 1024 units, one dropout layer with 0.2 rate, one sigmoid activation layer for classification of skin lesions) for our classification tasks are added. First, freeze all convolutional layers in InterceptionV3, and perform feature extraction for the newly added FC layers so that the weights for these layers arent completely random and the gradient wouldnt be too large when we start fine-tuning. After 10 epochs of feature extraction, we unfreeze the final convolutional block of convolutional and start fine-tune the model for 20 epochs. Throughout the training process, learning rate of 0.0001 and

SGD optimizer are used. I also train 2 network using two different input images(image,image_mul_mask).

IV. Experiment results

A. Classical model

TABLE I
Result of single model on original feature(73-dimension)

Model	precision	recall	f1-score
SVM	0.76	0.76	0.76
LR	0.67	0.68	0.68
KNN	0.68	0.69	0.67
ETC	0.71	0.71	0.71
RF	0.71	0.71	0.71
AdaBoost	0.75	0.74	0.73
XGB	0.70	0.70	0.70
Ensemble	0.75	0.75	0.75

TABLE II
Result of single model on PCA feature(20-dimension)

Model	precision	recall	f1-score
SVM	0.64	0.65	0.65
LR	0.70	0.70	0.68
KNN	0.69	0.69	0.67
ETC	0.72	0.72	0.72
RF	0.73	0.73	0.73
AdaBoost	0.66	0.67	0.66
XGB	0.67	0.67	0.67
Ensemble	0.66	0.67	0.66

TABLE III
Result of single model on forward feature selected feature(10-dimension)

Model	precision	recall	f1-score
SVM	0.70	0.70	0.70
LR	0.71	0.71	0.71
KNN	0.72	0.71	0.70
ETC	0.72	0.72	0.72
RF	0.64	0.64	0.64
AdaBoost	0.66	0.67	0.66
XGB	0.69	0.70	0.69
Ensemble	0.69	0.70	0.69

B. CNN model

TABLE IV
Result of Baseline CNN on original image

Class	precision	recall	f1-score
0	0.58	1.00	0.73
1	0.00	0.00	0.00
average	0.34	0.58	0.43

TABLE V
Result of Interception V3 on original image

Class	precision	recall	f1-score
0	0.63	0.98	0.76
1	0.89	0.18	0.30
average	0.74	0.65	0.57

TABLE VI
Result of Interception V3 on segmented image

Class	precision	recall	f1-score
0	0.68	0.85	0.76
1	0.69	0.45	0.55
average	0.69	0.69	0.67

C. Analysis

First of all, for the classical model part, we can see that

- ExtraTree classifier is the most stable model on different dimension features.
- PCA and Forward feature selection can improve the performance of LR and KNN, while decrease the score of SVM, AdaBoost and XGB
- In general, those models trained on original feature have better performance
- According to the Kaggle score, despite that Ensemble result validation score may be a little lower than best single classifier, it increases the Kaggle performance a lot in practice. I think this is because ensemble method effectively reduce the overfitting phenomenon and reduce the variance.

Next, for the CNN model part, we can see that

- Due to the unbalance of data set, CNN models are all shown bias problem, especially for Baseline CNN model.
- However, the Interception V3 model, which is trained on original images, has a relatively high(0.89) precision for ”melanoma” class. I think this is because DCNN can capture some hard to describe features.

Therefore, in order to use the advantage of Interception V3 model, I decide to merge the ensemble result with Interception V3 model result. In details, I assign high weight for ”melanoma” class in Interception V3 model and low weight for ”benign” class. The result on Kaggle score shows this strategy works well.

V. Conclusion

In this project, I handle a specific Image classification problem. This might be the first time which I go through the whole process of solving machine learning problems. During this process, I make the knowledge and theory into practice and have a better understanding of each machine learning model theory. Besides, through papers and blogs, I also learn many new techniques, such as transfer learning. I am very glad to have the opportunity to participate this contest.

For the future improvement, due to the computation power limits, I haven't tried many other powerful DCNN networks such as DenseNet and haven't tuned the best parameters for all the model. I think I will try more state-of-art models and methods to improve the performance.

ACKNOWLEDGMENT

Thanks to professor Pietro Gori and Nicolas Gonthier for their instructions and helps.

REFERENCES

[1] François Chollet et al. Keras, 2015.

[2] Andre Esteva, Brett Kuprel, Roberto A Novoa, Justin Ko, Susan M Swetter, Helen M Blau, and Sebastian Thrun. Dermatologist-level classification of skin cancer with deep neural networks. *Nature*, 542(7639):115, 2017.

[3] Roberta B Oliveira, Norian Marranghello, Aledir S Pereira, and João Manuel RS Tavares. A computational approach for detecting pigmented skin lesions in macroscopic images. *Expert Systems with Applications*, 61:53–63, 2016.

[4] Maciel Zortea, Thomas R Schopf, Kevin Thon, Marc Geilhufe, Kristian Hindberg, Herbert Kirchesch, Kajsa Møllersen, Jörn Schulz, Stein Olav Skrøvseth, and Fred Godtliebsen. Performance of a dermoscopy-based computer vision system for the diagnosis of pigmented skin lesions compared with visual evaluation by experienced dermatologists. *Artificial intelligence in medicine*, 60(1):13–26, 2014.