A Deep Learning Approach to Universal Skin Disease Classification

Haofu Liao
University of Rochester
Department of Computer Science
haofu.liao@rochester.edu

Abstract

Skin diseases are very common in people’s daily life. Each year, millions of people in American are affected by all kinds of skin disorders. Diagnosis of skin diseases sometimes requires a high-level of expertise due to the variety of their visual aspects. As human judgment are often subjective and hardly reproducible, to achieve a more objective and reliable diagnosis, a computer aided diagnostic system should be considered. In this paper, we investigate the feasibility of constructing a universal skin disease diagnosis system using deep convolutional neural network (CNN). We train the CNN architecture using the 23,000 skin disease images from the Dermnet dataset and test its performance with both the Dermnet and OLE, another skin disease dataset, images. Our system can achieve as high as 73.1% Top-1 accuracy and 91.0% Top-5 accuracy when testing on the Dermnet dataset. For the test on the OLE dataset, Top-1 and Top-5 accuracies are 31.1% and 69.5%. We show that these accuracies can be further improved if more training images are used.

1. Introduction

Skin diseases are one of the most commonly seen infections among people. Due to the disfigurement and associated hardships, skin disorders cause lots of trouble to the sufferers [13]. Speaking of skin cancer, the facts and figures become more serious. In United States, skin cancer is the most common form of cancer. According to a 2012 statistics study, over 5.4 million cases of nonmelanoma skin cancer, including basal cell carcinoma and squamous cell carcinoma, are treated among more than 3.3 million people in America [20]. In each year, the number of new cases of skin cancer is more than the number of the new incidence of cancers of the breast, prostate, lung and colon in combined [24]. Research also shows that in the course of a lifetime, one-fifth of Americans will develop a skin cancer [19].

However, the diagnosis of skin disease is challenging. To diagnose a skin disease, a variety of visual clues may be used such as the individual lesional morphology, the body site distribution, color, scaling and arrangement of lesions. When the individual components are analyzed separately, the recognition process can be quite complex [6] [15]. For example, the well studied skin cancer, melanoma, has four major clinical diagnosis methods: ABCD rules, pattern analysis, Menzies method and 7-Point Checklist. To use these methods and achieve a good diagnostic accuracy, a high level of expertise is required as the differentiation of skin lesions need a great deal of experience [30].

Unlike the diagnosis by human experts which depends a lot on subjective judgment and is hardly reproducible [16], a computer aided diagnostic system is more objective and reliable. By using well-crafted feature extraction algorithms and combining with some popular classifiers (e.g. SVM and ANN), current state of art computer aided diagnostic systems [2, 31, 11, 22, 3] can achieve very good performance on certain skin cancers such as melanoma. But they are unable to perform diagnosis over broader classes of skin diseases.

Human engineered feature extraction is not suitable for an universal skin disease classification system. On one hand, hand-crafted features are usually dedicated for one or limited number of skin diseases. They can hardly be applied to other classes and datasets. One the other hand, due to the diversity nature of skin diseases [6], human engineering for every skin disease is unrealistic. One way to solve this problem is to use feature learning [4] which eliminates the need for feature engineering and lets the machine to decide which feature to use. Many feature learning based classification systems have been proposed in past few years [5, 8, 7, 28, 29, 1]. However, they have been mostly restricted to dermoscopy or histopathology images. And they mainly focus on the detection of mitosis, an indicator of cancer [28].

In recent years, deep convolutional neural networks (CNN) become very popular in feature learning and object classification. The use of high performance GPU makes it possible to train a network on a large-scale dataset so as to yield a better performance. Many researches
from the ImageNet Large Scale Visual-Recognition Challenge (ILSVRC) [21] show that the state-of-art CNN architectures are able to surpass human in object classification. Recently, Esteva et al. [10] proposed a CNN-based universal skin disease classification system. They train their network by fine-tuning the VGG16 and VGG19 architecture [25]. Their network achieved 60.0% Top-1 classification and 80.3% Top-3 classification which significantly outperformed the human specialists in their experiment.

Inspired by [10], we conducted a research in a similar path and achieved a better accuracy. We use the skin images from two different sources. First, we collect images from Dermnet (www.dermnet.com), a publicly available dataset of more than 23000 dermatologist-curated skin disease images. Second, we also obtain 1300 skin images from New York State Department of Health. We call this dataset OLE. We fine tune our CNNs with the VGG16, VGG19 and GoogleNet [27] models and test them on the two datasets. The result shows that our CNNs can achieve 73.1% Top-1 classification and 91.0% Top-5 classification on the Dermnet dataset and 31.1% Top-1 classification and 69.5% Top-5 classification on the OLE dataset.

The rest of this report is organized as follows. Section 2 introduces the dataset and CNN models we use to construct the CNN architecture. Section 3 investigates the performance of the CNNs using different training and test data settings. Conclusions and future works of this project is given in Section 4.

2. Method

2.1. Dataset

We build our skin disease dataset from two different sources: Dermnet and OLE. Dermnet is one of the largest photo dermatology source that available publicly. It has more than 23,000 skin disease images on a wide variety of skin conditions. Dermnet organizes the skin diseases biologically in a two-level taxonomy. The bottom-level contains more than 600 skin diseases in a fine-grained granularity. The top-level contains 23 skin disease classes as shown in Table 1. Each of the top-level skin disease class contains a subcollection of the bottom-level skin diseases. We adapt the skin disease taxonomy from Dermnet for our classification system and use the 23 top-level skin disease classes to label all the skin disease images. OLE dataset contains more than 1300 skin disease images from New York State Department of Health. It contains 19 skin diseases each can map to one of the bottom-level skin diseases from the Dermnet taxonomy. Hence, we further label these 19 skin diseases with their top-level counterparts in the Dermnet taxonomy. The labeling results are shown in Table 2.

To prepare the dataset, we need to download the 23,000 skin disease images from Dermnet. Since there are no direct link or API for these images, we download these images by parsing their address and send HTTP request to the web server. The downloaded images are not well labeled and contain watermarks. Their naming format are not consistent. Therefore, multiple name analysis strategies are used to extract class information from image names. Further, a two-level hierarchical map is established using the extracted class information and the Dermnet taxonomy. All the images from Dermnet are originally labeled using the bottom-level classes. Since we only consider the 23 top-level classes, we merge each of these bottom-level classes according to the two-level hierarchical map and label each image with the 23 top-level classes. During the construction of the dataset, if the names of some images contain no class information or the number of the images of some bottom-level class is small, we will discard those images. Figure 1 (a) shows the numbers of images in each of the 23 top-level classes. (b) The number of the OLE images in each of the skin disease in Table 2. The x-axis denotes the label of the classes. Each of the label is associated with a skin disease class in Table 1. The bars in red are the classes we will delete in Section 3.2.

![Figure 1](image1.png)

(a) Dermnet

(b) OLE

Figure 1: (a) The number of the Dermnet images in each of the 23 top-level classes. (b) The number of the OLE images in each of the skin disease in Table 2. The x-axis denotes the label of the classes. Each of the label is associated with a skin disease class in Table 1. The bars in red are the classes we will delete in Section 3.2.

2.2. CNN Architecture

It has been shown that in many cases transfer learning can be used to efficiently train a deep CNN [18] [32]. In transfer learning, instead of training the network from randomly initialized parameters, people takes a pretrained net-
Table 1: The 23 top-level categories of the Dermnet taxonomy. We use these top-level categories to label images from Dermnet and OLE. Each of the category is assigned to a numeric label ranges from 0 to 22. Due to the layout limitation, long category names are shortened.

<table>
<thead>
<tr>
<th>Subclass Names</th>
<th>Labels</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rosacea</td>
<td>0</td>
</tr>
<tr>
<td>Actinic Keratosisis</td>
<td>1</td>
</tr>
<tr>
<td>Basal Cell Carcinoma</td>
<td>2</td>
</tr>
<tr>
<td>Atopic Dermatitis</td>
<td>3</td>
</tr>
<tr>
<td>Sebaceous Keratosis</td>
<td>4</td>
</tr>
<tr>
<td>Tinea Corporis</td>
<td>5</td>
</tr>
<tr>
<td>Tinea Versicolor</td>
<td>6</td>
</tr>
<tr>
<td>Urticaria</td>
<td>7</td>
</tr>
<tr>
<td>Herpes</td>
<td>8</td>
</tr>
<tr>
<td>Acne and Rosacea</td>
<td>9</td>
</tr>
<tr>
<td>Malignant Lesions</td>
<td>10</td>
</tr>
<tr>
<td>Atopic Dermatitis</td>
<td>11</td>
</tr>
<tr>
<td>Bullous Disease</td>
<td>12</td>
</tr>
<tr>
<td>Bacterial Infections</td>
<td>13</td>
</tr>
<tr>
<td>Eczema</td>
<td>14</td>
</tr>
<tr>
<td>Exanthems &amp; Drug Eruptions</td>
<td>15</td>
</tr>
<tr>
<td>Hair Diseases</td>
<td>16</td>
</tr>
<tr>
<td>Pigmentation Disorders</td>
<td>17</td>
</tr>
<tr>
<td>Connective Tissue diseases</td>
<td>18</td>
</tr>
<tr>
<td>Contact Dermatitis</td>
<td>19</td>
</tr>
<tr>
<td>Pigmentation Disorders</td>
<td>20</td>
</tr>
<tr>
<td>Psoriasis &amp; Lichen Planus</td>
<td>21</td>
</tr>
<tr>
<td>Infestations &amp; Bites</td>
<td>22</td>
</tr>
<tr>
<td>Benign Tumors</td>
<td>23</td>
</tr>
</tbody>
</table>

In our experiments, we fine-tune these models with our skin disease datasets and conduct all the trainings and tests on an NVIDIA Titan Black GPU to accelerate the computation.

Fine-tuning with Caffe requires modifying the network definition (the deploy protocol or the train validation protocol) of the pretrained models. First, since we need to train the network using our skin disease datasets, the data layer of the pretrained network should be reinitialized. The following is an example of the data layer we used for our datasets.

```
layer {
  name: "data"
  type: "ImageData"
  top: "data"
  top: "labels"
  include {
    phase: TRAIN
  }
  transform_param {
    mirror: true
    crop_size: 224
    mean_file: "imagenet_mean.binaryproto"
  }
  image_data_param {
    source: "train.txt"
    batch_size: 32
    new_height: 256
    new_width: 256
  }
}
```

In the data layer, we specify the input dataset using the "source" parameter which is the name of a text file with each line giving an image filename and a label. We set the input image size and the number of images to process at a time using the "new_height", "new_width", and "batch_size" parameters. We also define the preprocessing to the input images using "mirror", "crop_size" and "mean_file" parameters. They are set the same values as the pretrained models so that the input to the fine-tuned network is at the same scale as the pretrained network. For the last fully connected layer (the one that outputs scores of the skin disease classes), we change the output number to 23 for Dermnet’s 23 top-level classes. Hence, it will output scores to indicate the classification of the input skin disease image. We want this layer to train from scratch so that its weights can fit our dataset instead of the pretrained model.
So, we change its name to randomize its weights and train it with a higher learning rate to get a faster convergence. For deploy protocols, we also replace the softmax layer by the softmaxwithloss layer so that the loss function can be applied to the training.

### 3. Experiments

#### 3.1. Test on Dermnet Dataset

In our first experiment, we train and test the CNNs using the Dermnet dataset only. We label all the Dermnet images using the labeling strategies introduced in Section 2.1. After construction, we get a set of 17630 labeled images. We randomly pick 16630 of them as the training set and 1000 of them as the test set. Then, we fine-tune three CNNs using the three ImageNet pretrained models (VGG16, VGG19 and GoogleNet) respectively. The Top-1 and Top-5 accuracies of the networks are given in Table 3. We can see that all of the three networks achieved good classification results. VGG19 achieved relatively better performance on Top-1 accuracy. It is probably because that it has more layers (19) than other models.

<table>
<thead>
<tr>
<th>CNN Model</th>
<th>Top-1 Accuracy</th>
<th>Top-5 Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>VGG 16</td>
<td>72.7%</td>
<td>91.0%</td>
</tr>
<tr>
<td>VGG 19</td>
<td>73.1%</td>
<td>90.9%</td>
</tr>
<tr>
<td>GoogleNet</td>
<td>71.8%</td>
<td>90.7%</td>
</tr>
</tbody>
</table>

Table 3: Top-1 and Top-5 accuracies of the CNNs using different ImageNet pretrained models. All the CNNs are trained using the Dermnet images only.

Some example images along with their predictions are given in Figure 2. The ground truth is given at the top of each image and the Top-5 predictions together with the probabilities are given below. We can see melanoma, psoriasis, basal cell carcinoma, and systemic disease are predicted correctly with high confidence. The bullous disease is misclassified with divergent predictions. Viral infections, and fungal infections only hit Top-5 predictions with very low confidence. Benign tumors even gets misclassified with high probability prediction on cellulitis. We will further analyze the the misclassified situations Section 3.2.

The confusion matrices of the three networks are given in Figure 3. The index of each row in the confusion matrix corresponds to a true label and the indices of the columns denote the predicted labels. The color of each cell represents the probability of the prediction and the number appears on each cell gives the occurrences of the prediction. We can see from the three confusion matrices that the high confidence predictions are located along the diagonal. This indicates that they don’t get high error rate on some particular classes and In most of the times, the predictions are correct.

<table>
<thead>
<tr>
<th>CNN Model</th>
<th>Top-1 Accuracy</th>
<th>Top-5 Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>VGG19</td>
<td>24.8%</td>
<td>61.7%</td>
</tr>
<tr>
<td>VGG19 Improved</td>
<td>31.1%</td>
<td>69.5%</td>
</tr>
</tbody>
</table>

Table 4: Top-1 and Top-5 accuracies of the CNNs using the VGG19 model only. All the CNNs are tested on the OLE images. For VGG19 Improved, the refined dataset is used.

#### 3.2. Test on OLE Dataset

We then investigate the performance of the CNN on the OLE dataset. Our training set is the 17630 skin disease images from Dermnet. But for the test set, we choose the OLE images. Note that in this case, our training set has 1000 more Dermnet images than the last experiment. We need to retrain the CNN. Since all the three models show a similar performance and the VGG19 model performs slightly better, we fine-tune the CNN with the VGG19 model only. The Top-1 and Top-5 accuracies are shown in Table 4. The new CNN only yields 24.8% Top-1 accuracy and 61.7% Top-5 accuracy on OLE dataset. This is reasonable as the OLE dataset may have some skin disease images that the CNN can’t learn from the Dermnet dataset. Figure 4(a) shows the confusion matrix of the CNN on the OLE dataset. As we’ve shown in Table 2, the OLE dataset only contains a subset of the 23 top-level classes. Hence, lots of the rows are marked with zeros. Consistent with the accuracy, diagonal elements in the confusion matrix show little confidence.

Based on the observation above, we further analyze the CNN by selecting some test images from different skin disease classes and retrieving their nearest neighbors in the training set. We choose the output of the “fc7” layer as the feature vector. The reason we choose this layer against other layers is because the “fc7” is the last layer before the final output layer (the layer that outputs class scores) and should contain more specific details of the classes in the Dermnet dataset. Besides, the dimension of the feature vector is 4096 which can hold a great deal of information of the input image. To get the nearest neighbors, we first build a feature database for all the skin disease images in the training set. Then, for a given test image, we obtain its feature vector from the “fc7” layer and calculate the Euclidean distance between this feature vector and all the other features in the feature database. The five training images with the top-5 smallest distances to the input test image are chosen to be the nearest neighbors. Figure 4 and 5 are the image retrieval results of two test images from the OLE dataset. For each of the figure, the top left image is the test image and the rest are the retrieved neighbors. Figure 4 uses a tinea versicolor
Figure 2: The prediction results output by the fine-tuned GoogleNet network. The label at the top of each image is the ground truth. The Top-5 predictions and the corresponding probabilities are given at the bottom of each images.

Figure 3: The confusion matrices of the VGG16, VGG19 and GoogleNet based CNNs. All the three CNNs are trained and tested using the Dermnet images only.

To verify our assumption, we design another experiment. As the misclassification is probably due to the lack of training images on certain diseases, we refine both the training set and the test set by removing some skin diseases with inadequate photos. Thus, if the CNN is trained correctly, it should learn enough details from the refined training set. For any given test image with a skin disease that is included in the refined training set, the CNN should have a higher confidence to classify it correctly. The red bars in Figure 4 denote the classes we removed from the training set (Dermnet) and the test set (OLE). We remove the skin diseases that appear in both training set and test set and are low in image numbers. We then retrain the CNN using the refined training set. The Top-1 and Top-5 accuracies of the newly trained CNN on the refined test set are given in Table 5.
4. Conclusion and Future work

We have investigated the feasibility of building an universal skin disease classification system using deep CNN. We tackle this problem by fine-tuning ImageNet pretrained models (VGG16, VGG19, GoogleNet) with the Dermnet dataset. Our experiments show that the current state-of-art CNN models can achieve as high as 73.1% Top-1 accuracy (VGG19) and 91.0% Top-5 accuracy (GoogleNet) when testing on the Dermnet dataset. We further discover the performance of the CNN architecture when testing on a different dataset (OLE). We find the classification system can only achieve 24.8% Top-1 accuracy and 61.7% Top-5 accuracy due to the lack of a broader variance in the training set. We show that by increasing the variance of the training set the Top-1 and Top-5 accuracies can be improved to 31.1% and 69.5%.

In our future research, we hope we can push this work much further and get a better accuracy. There are several places we can improve our work. First, since ImageNet data are not specialized for skin data, the ImageNet pretrained models may not be the best choice for skin disease classification. Thus, we should train a CNN model from scratch and test its performance. Second, the Dermnet images are organized using a biological taxonomy which is not the best choice for computer vision applications. We will work with a dermatologist to design a visually organized taxonomy and apply it to our classifier. Third, as the experimental results suggest that more variance in the training set would lead to a better accuracy, we should increase the size of our training set. Also note that the images retrieved by the networks are closely related to the ground truth. We may need to design a hierarchical classification algorithm using the retrieved images to improve the accuracy.

References


\(^2\)This idea is inspired by [10].


