Identifying Skin Cancer Type from Image Data

Ian Hall
iahall@davidson.edu

Eric Xu
erxu@davidson.edu

Abstract

Skin Cancer is a disease which affects millions of lives each year. Providing patients with the correct treatment is crucial and modern day solutions to adequately identifying the type of skin cancer requires microscopy and general consensus from a collection of doctors, which can be costly and time consuming. As society looks into simplifying and automating many tasks, machine learning models have been looked into as a solution to identification problems. In this paper, we implement various techniques such as Support Vector Machines, Neural Networks, and Random Search Hyper-Parameter Tuning to create accurate models to identify types of skin cancer from image data. Our model can be used to correctly map a pictures of skin cancer to its correct type.

Introduction

Cancer treatments have improved over recent years, but identifying the type of cancer remains essential to providing patients with the optimal treatment. With enough training data and images, machine learning models have been created to accurately identify lots of different types of objects. In this paper we experiment with various models to predict skin cancer type from images of the skin cancer itself. We conducted many experiments to find the optimal hyperparameters for our models to give the highest accuracy. This paper will explore the background behind these models, information about the data and the data analysis performed, the experiments conducted, the final conclusions, and a section about the ethical implications of these models.

Background

In this section we will discuss the machine learning models used, and the exploration we performed on the data.

Suppot Vector Machine

A Support Vector Machine (SVM) works by dividing the data into multiple sections by creating a hyperplane between each section. We experimented with many different SVMs, including ones using a linear, polynomial and sigmoid functions to separate the data. Due to the number of features in our dataset, we believe that this model will not perform as well as others, but we explore this technique as a possible model for our problem.

Neural Networks

A Neural network is a type of architecture, modeled after how human brains work, representing a learnable function mapping some set of inputs to an output. Neural networks are comprised of layers of individual nodes, containing an input layer, one or more hidden layers, and an output layer. Each node connects to another node in the next layer and has an associated weight and threshold. Each node also has a constant value, known as the bias, added to the weighted sum of the inputs to the given node. If the output of any individual node is above the specified threshold value, that node is activated, sending data to the next layer of the network. Otherwise, no data is passed along to the next layer of the network. Through an the combination of gradient descent and backpropogation, Neural networks adjust the weights and biases in order to minimize a loss function, showing how well the Neural network fits the data. Through adequate training and proper architecture, Neural networks have been able to learn many different types of function, including the image classification problem we are tackling.

Data

Our data comes from the the HAM10000 dataset, compiled by Phillip Tschandl and others from Medical University of Vienna. The images themselves come two difference institutions: the Department of Dermatology at the Medical University of Vienna, Austria, and the skin cancer practice of Cliff Rosendahl in Queensland, Australia. The data set contains 10015 unique data points with a patient ID number, the associated image of the skin cancer in the form of 450x600 pixels, t type of Skin Cancer (7 types), how the Skin Cancer was identified, the age of Patient, the sex of Patient, and where the Skin Cancer was located. For our models, we decided to focus purely on the image data itself and used the other variables to check for potential biases in the data set.

Data Analysis

To get our data into manageable form, we created a dataframe containing the RGB, ranging from 0 to 255, pixel values and the associated output for each individual image.

Due to the size of the images and therefore a large number of features, we implemented an average-pooling technique to reduce the size of the images and therefore the number of features. We explored various sizes for our model from 28x28 images to 128x128. Furthermore, when dealing with the size of the pixels, we normalized the pixel values to a range of 0 to 1, dividing the pixel values by 255.
Additionally, we looked at the distribution of the types of skin cancer in the data set. As seen in Table 1, our data set is heavily skewed toward nv or melanocytic nevi.

Table 1: Table of type of skin cancer and the number of examples for each skin cancer type

Cancer Type	Number of Data Points
akiec	327
bcc	515
bkl	1099
df	115
mel	1113
nv	6705
vasc	142

Due to this heavily skewed data set, we looked into methods to even out the data set, so our models would not over-fit towards the nv type of cancer as most of the training data would be nv.

Experiments

In this section we discuss the models that we utilized: svm Model and Neural Network Model. We talk about what each model is and how it works, and we discuss the hyper-parameter tuning and selection methods that we used to choose hyper-parameters.

Support Vector Machine(svm) Model

A support vector machine constructs a hyper-plane or set of hyper-planes in a high or infinite dimensional space, which can be used for data training for classification. A good separation is achieved by the hyper-plane that has the largest distance to the nearest training data points of any class, because larger margin can produce lower error when we input the model with new, unknown data. In SVC, given training vectors:

$$x_i \in R^p (i = 1, ..., n), y \in \{1, -1\}^n$$

The goal is to find $w \in R^p, b \in R$ such that the prediction is made by the following function:

$$w^T \phi(x) + b$$

We implemented scikit-learn package in Python library to build the model for our dataset. Specifically, we used:

```
from sklearn import svm
clf = svm.SVC(kernel = 'poly', degree = 2, gamma = 0.007)
clf.fit(x_train, y_train)
predicted = clf.predict(x_test)
```

This estimator implements C-Support Vector Classification. The implementation is based on libsvm. In the classifier, there are four hyper-parameters, which are choice of kernel, degree of kernel if it is $poly$, regularization value(C), and gamma. Tuning these parameters will be elaborated in hyper-parameter tuning section.

Hyper-parameter Tuning

The determination of hyper-parameter of svm model is done by gridsearch on four variables: choice of kernel, degree of kernel if it is $poly$, regularization value(C), and gamma. We have split the variables and done two gridsearches due to the fact that the table would be too complex if we chose to do one gridsearch on all the variables. The result of each gridsearch is shown here.

Table 2: Gridsearch 1

kernel&C	accuracy	performance rank
linear & 1	0.63370647	14
poly & 1	0.63873632	1
linear & 5	0.60876617	5
poly & 5	0.63873632	1
linear & 9	0.60777114	6
poly & 9	0.63873632	1

From Table 2, we found that $poly$ kernel has a better performance than $linear$ kernel, and different C values do not have influence on $poly$ kernel. We then considered to do more research on degrees of $poly$ kernel since right now we are using its default degree 3.

Table 3: Gridsearch 2

degree& gamma	accuracy	performance rank
2 & 0.01	0.66465672	1
2 & 0.05	0.62677612	8
2 & 0.1	0.62677612	8
3 & 0.01	0.64472139	5
3 & 0.05	0.63873632	6
3 & 0.1	0.63873632	6
4 & 0.01	0.64772139	2
4 & 0.05	0.64772139	2
4 & 0.1	0.64772139	2

We did a gridsearch on degrees of $poly$ kernel and gamma value. The result in Table 3 shows that $poly$ kernel with degree = 2 has the best performance among other results. And gamma values do influence the performance of the model. So we did further researches to determine the best gamma value, which found out to be 0.007.

Therefore, we eventually chose kernel = $poly$, degree = 2, gamma = 0.007. Since C value does not influence the result, we chose $C = 1$ for convenience.

Validation

We did a K-fold validation for svm model using

```
K_fold = svm.SVC(kernel='poly',degree=2,gamma=0.007)
validation_result=cross_val_score(K_fold,x_val,y_val,cv=5)
```

It is found that the mean accuracy of 5-fold validation of our svm model is around 0.63, which is not good.

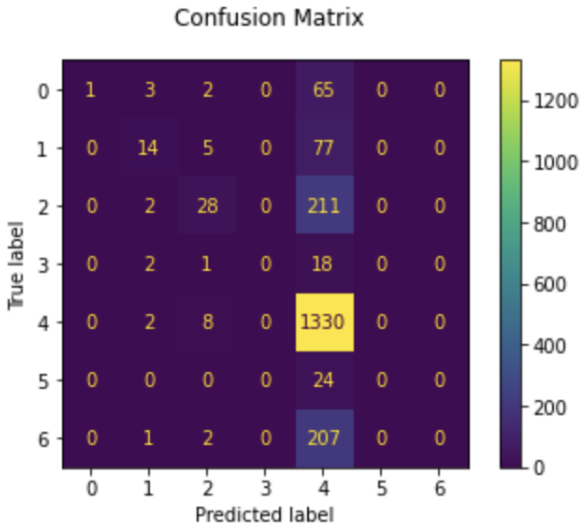

Figure 1: Confusion Matrix for svm Model

As shown in Figure 1, our svm model is good at identifying only one of the six classes, which is cancer type nv. And the model performs poorly on other five classes. We believe it is caused by the unbalanced distribution of our data set. 70 percent(6705) of the data set(10015) are nv cases. This drastic difference is probably the main reason why our models are doing so badly.

Neural Network Experiments

Due to the size of the images and therefore the need to preserve many of the pixels because of the need for accuracy in this type of scenario, we implemented pre-trained models and used transfer learning to apply them to this scenario. The pre-trained model that we implemented in our code was the MobileNetV2 architecture developed by Mark Sander and others at Google. This large Convolutional neural network has already been trained on millions of training examples. To implement this for our experiments, we downloaded the pre-trained network without the last layer of the network, which was previously set for 1000 classes. We then added our own layer to the end of the network with 7 nodes corresponding to the 7 possible outputs from our model. We then ensure that the weights and biases of the pre-trained model were set to untrainable, meaning that they would not be adjusted during the training phase. By using the larger model, we were able to use more features as the features for our model were tuples of (224,224,3) RGB values for each image. We divided our data into a 60/20/20 train, validation, and test split and then trained the model to fit our data. After are initial training, we fine-tuned the model by allowing a set number of layers in the MobileNetV2 architecture to be trainable, training the model further. We implemented various versions of this model, implementing two different loss functions, Categorical Cross Entropy and Sigmoid Focal Cross Entropy Loss. The Sigmoid Focal Cross Entropy Loss, developed by Lin et al. (2017), is supposed to account for class imbalances by "reshaping the standard cross entropy loss such that it down-weights the loss assigned to well-classified examples." We also looked adjusted the value of the learning rate using three distinct values. We report the results of our models in Table 4.

Table 4: Results of Neural Network Experiments

Loss Function	Learning Rate	Accuracy
Cross Entropy	0.01	.7435
Cross Entropy	0.001	.8091
Cross Entropy	0.001	.7956
Focal Cross Entropy	0.01	0.7652
Focal Cross Entropy	0.001	0.8334
Focal Cross Entropy	0.0001	0.8122

From the results of the table, we can see that the best performing model was using the Focal Cross Entropy Loss function and a learning rate of 0.001, which gave us an accuracy of 83.34% on the validation set.

Final Test

With the Neural Network using Focal Cross Entropy loss and a learning rate of 0.001 as our best model overall, we ran final tests using the model and ended up with an overall accuracy of 80.15%. As for svm Model, the overall accuracy for final test is 69%, which is not as good as the Neural Network model.

Conclusion

Through this paper, we have explored various methods to try and create an adequate model to identify skin cancer types from image data. We used both a variety of support vector machines and transfer learning neural network models to approach this problem. Our strongest model was a transfer learning neural network using a learning rate of 0.001 and a Focal Cross Entropy Loss function to correct a bit for the imbalance in the data set. The final accuracy on the test set using our model achieved an accuracy of 80.15%, which is not terrible, but should be much higher for addressing something as serious as skin cancer. In the future, we would like to implement more ways to adjust for the imbalance in the data set including data augmentation, such as creating additional images for each of the classes by rotating the photos. Therefore, our models will not over-fit toward the "nv" or melanocytic nevi. Overall, we have created a decent model for predicting the type of skin cancer from an image, showing that neural networks can most likely solve this problem, and further work in the matter will allow us to create stronger models that could be used in practice by doctors in the real world.

Ethics

There are many ethical concerns when addressing the problem of identifying skin cancer from images and the model's training itself. First of all, the matter of diagnosing cancer types could be potentially life-saving or life-threatening if our model incorrectly labels the type of cancer a patient has. The data for this model comes from Australia and Austria, whose demographics are strongly skewed toward Caucasian people. Therefore, depending on the background of the images or how accurate the images inputted in the model are, there could be systematic diagnose at a lower success rate for those who are not Caucasian. Furthermore, our model does not have any training data on non-cancerous data, therefore it will always assign a type of cancer to the input image even if it is not cancerous, which could be problematic if used without assurance that the image is of skin cancer.

References

Lin, T.,Goyal, P.,Girshick, R., He, K., and Dollár, P. 2017. Focal Loss for Dense Object Detection. arXiv. https://arxiv.org/abs/1708.02002

Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., and Chen, L., MobileNetV2: Inverted Residuals and Linear Bottlenecks. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018, pp. 4510-4520, doi: 10.1109/CVPR.2018.00474

Tschandl, P., Rosendahl, C. & Kittler, H. The HAM10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions. Sci Data 5, 180161 (2018). https://doi.org/10.1038/sdata.2018.161