Skin Cancer type detection using thresholding and machine approaches

Chaitanya Sunil Kulkarni†
SCAI
Arizona State University
Tempe AZ USA
ckulka11@asu.edu

ABSTRACT

Skin cancer is one the deadliest type of cancer which is bound to spread to different body parts when left untreated. Early skin cancer detection and treatment have been shown to reduce mortality risk. Machine learning and deep learning techniques have become increasingly effective with the advancement of artificial intelligence in a variety of domains, including healthcare. Convolutional Neural Networks (CNN) have emerged as an effective tool, particularly for medical imagery like pictures of skin cancer. In this project, I suggest using a Novel CNN model with thresholding techniques to classify skin cancer images into seven classes. I've also investigated the outcomes of Support Vector Machine, Logistic Regression, and XgBoost with the CNN model. The HAM10000 dataset, which includes 10015 images of several types of skin cancer, was used. The minority classes were oversampled through data augmentation. The raw images were subjected to thresholding techniques before being resized to 80*80. The CNN model performed extremely well giving us a training accuracy of 99.9 and testing accuracy of 100. Machine learning models also performed well, various metrics of all the models have been tabulated. Unexpectedly, every model failed to perform adequately with the augmented test data and incorrectly identified one class. The primary cause of this can be the similarity of the images and information loss as a result of image down shaping.

KEYWORDS

Machine learning, deep learning, computer vision, Data mining

1 Introduction

Skin cancer is one of the most common types of cancer which is the main reason making it one of the attention seeking health issues worldwide. Melanoma is a cancer that has historically been uncommon, but in the last 50 years, its prevalence has dramatically increased globally. In terms of the average number of years lost to death from cancer, it is actually one of the more common cancers.

With new technology and SOTA models emerging every single day, artificial intelligence has been booming from a few decades. Artificial Neural Networks are composed of artificial neurons that were modeled after the biological neurons in our brains. A modified feed-forward neural network type known as a convolutional neural network (CNN) is frequently employed for image categorization applications. Because CNNs are aware of translation invariance, they can identify a specific object even when it appears in several ways. This is a crucial distinction that gives CNN an edge over feed-forward neural networks, which are unable to comprehend translation invariance.

Convolutional layer, pooling layer, and fully connected layer are the three layers that make up a CNN. Convolutional layer extracts key features using the filters and pooling layer is used to simplify the model by reducing the number of parameters. Fully connected layer takes the flattened 1d vector and helps classify the input into N pre-defined classes.

Machine learning models like Logistic Regression, Support Vector Machine (SVM), and XgBoost has been used to compare the results with the CNN model.

The report is structured as follows: 1 covers the introduction to relevant topics, 2 has previous works related to this project. In 3 I discuss about the dataset and 4 has the proposed methodology. Implementation details are discussed in 5. Results are discussed in 6 and finally I conclude the paper in 7 along with future work opportunities.

2 Related Work

A system of classifying skin lesions into benign and malignant categories was proposed by Xie et al. [1]. Three phases made up the planned system's operation. The suggested model outperformed the other classifiers in terms of sensitivity by at least 7.5% with a 91.11% accuracy. A new technique for skin cancer diagnosis based on genetic algorithms (GAs) and ANN algorithms was published by Aswin et al. in [2]. Dull-Rozar, a piece of medical imaging software, was used to remove hair from the images before using the Otsu thresholding approach to determine the region of interest.

3 Dataset

The HAM10000 dataset is openly available in Kaggle and is a collection 10015 images of size 800*600 and 7 classes indicating

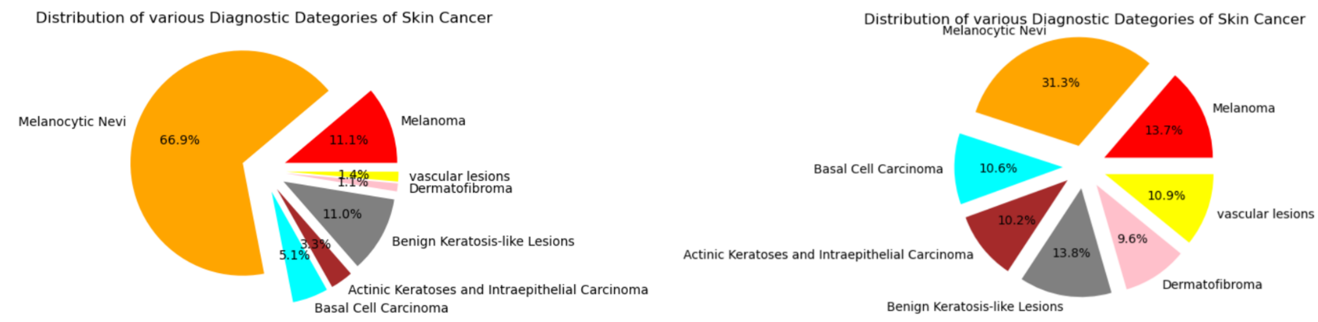

Figure 1

different types of cancer. The original data was imbalanced with a single class containing 70% of the data and other classes contributing as low as 1% to the total dataset. I used data augmentation to fix this issue and over sampled the minority class using data augmentation techniques like flipping, rotation at different angles. Figure 1 shows the distribution of different classes before the over sampling and after the over sampling. The images were down shaped to 80*80 due to computational constrains and lack of GPUs.

4 Proposed Methodology

The proposed methodology is divided into 3 stages: pre-processing, model training and testing as shown in figure3.

4.1 Pre-Processing

Pre-processing is the most important step in this project as we employ thresholding techniques to extract meaningful information from the images. The raw images are passed through Gaussian and Otsu thresholding technique and the resulting images is summed to get the Gaussian + Otsu images.

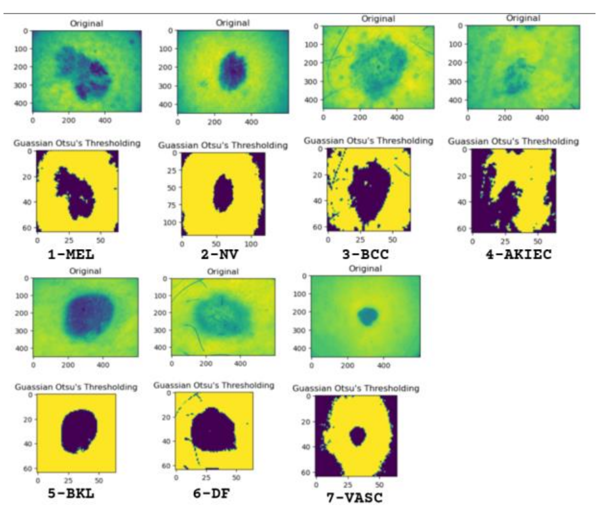

Figure 2

Figure 2 shows the images before and after applying the thresholding techniques for all the classes. The area of interest is enhanced and clearer in the processed image making it easier for the ML models to classify the images. The images are then re shaped to 80*80 to reduce the computational cost and make the models run faster.

4.2 Model-training

After pre-processing images, I employed CNN and 3 ML algorithms to test the performance on the classifying skin cancer. The data was split into training and testing in a ratio of 70:30 where 70% of the dataset was used for training and 30% of unseen data was used for testing. A 3-layer deep CNN model was employed with dropouts at every layer. Model had 3 fully connected layer before the output layer with 7 neurons.

Machine learning models like Support Vector Machine, Logistic Regression and Boosting techniques like XgBoost was also employed to compare the results. The details of implementations are discussed in the later sections.

4.3 Testing

The 30% of the unseen data is used for testing the performance of the model. Furthermore, I created 1000 augmented data using the pre-processed data to test the robustness of the model. The data was flipped and rotated to create new image and the ratio of samples in the testing data created was close to the original dataset. Therefore, the model was tested on 2 test data's, one which was taken from the dataset and other which was created using data augmentation techniques. The results of both the test data have been tabulated in the further sections.

5 Implementation details

CNN model had 3 pairs of convolutional and pooling layer with kernels 64, 256, 512 respectively. I employed max Pooling for this architecture and there were dropout layers after every layer ranging from 04.-0.8 to avoid overfitting. The model had 3 fully connected layer with 512, 254 and 64 neurons respectively before the final output layer with 7 neurons. "Relu" activation

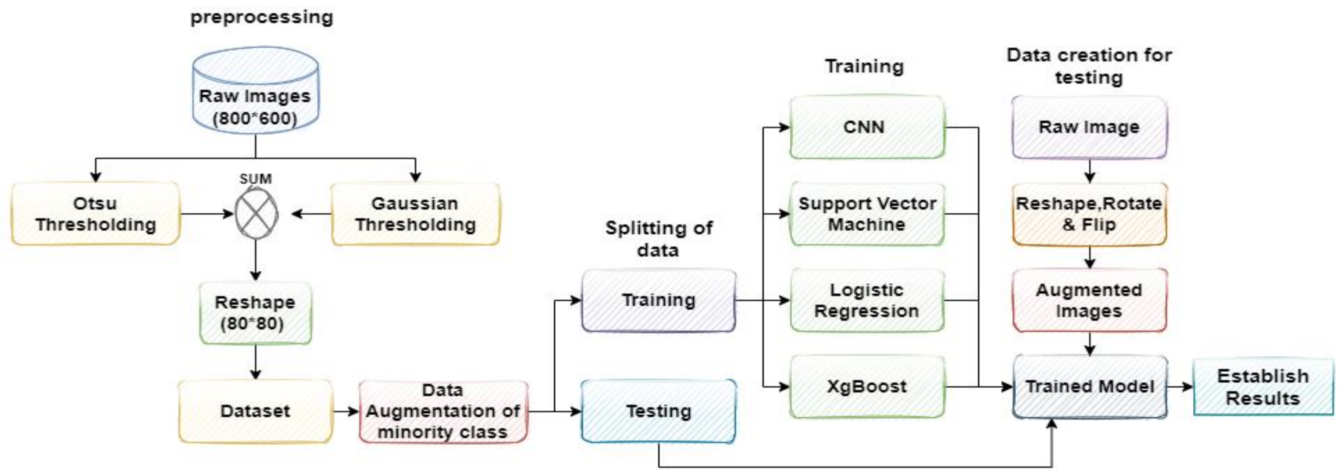

Figure 3

was used for all the layers except for the output layer where "SoftMax" was used. A kernel size of (3,3) with (2,2) strides was maintained for all the layers. The CNN model was trained for 10 epochs with a batch size of 32. "Adam" optimizer was used to optimize the "sparse categorical cross entropy" loss function. The plot of loss and accuracy for training and validation data is shown in figure 4. The training data was further split to 80:20 using the inbuilt function of keras. The model was trained on intel core I7 10th gen processor and each epoch took 19-20 sec to train.

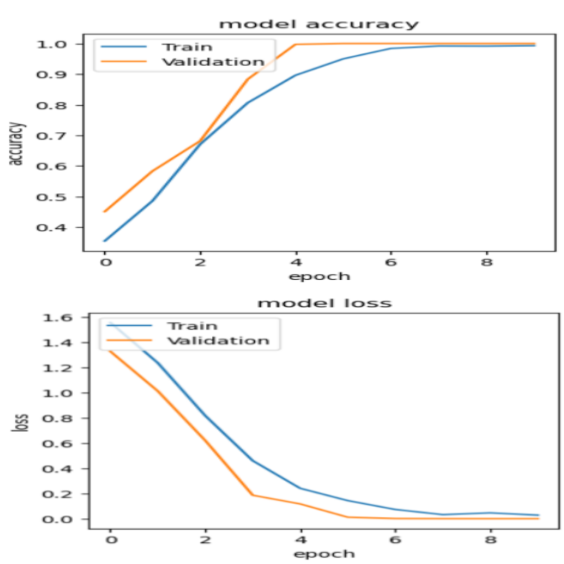

Figure 4

The SVM model was trained on "rbf" kernel with a penalty of 1 with gamma value as scale. The SVC model took 5.11 sec to train and 1.5 sec to predict for the test data. The XgBoost algorithm with 100 estimators and a learning rate of 0.3 and max_depth of 6 took 45.9 sec to train and predicted the classes in 306 ms. Logistic regression model was implemented with "l2" penalty and c equal to 1. Logistic Regression model took 8.71 sec to train and predicted the classes in 78.4 ms.

6 Results and Discussion

The CNN model produced exceptional results both on training and testing data. It produced a training accuracy of 99.3 % and testing accuracy of 100%. Along with the accuracy the F1score, recall and precision were 100% as well. But when I tested the model for augmented test data which was created by me it performed poorly for 1 class which reduced its accuracy from 100% to 70%. The f1 score, accuracy, precision and recall were 100% for all the classes except for 2 classes. My model confused class "NV" as class "VASC" and "BCC" which can be observed in figure 2. One reasonable reason for this could be the similarity between those 2 class images and as we are down-shaping the images from 600*800 to 80*80 there might have been some loss of information essential to classify these classes.

Machine learning models also performed similarly and gave a training and testing accuracy of 100% along with 100% f1score, precision and recall. The accuracy on augmented test data was similar to CNN model and ML models also faced a similar problem of classifying class "NV" as class "VASC" and "BCC". Table 1 shows the summary of all the models along with their training and predicting time. I have only included the performance of the model on augmented test data as the model gave out perfect results on test data. The Precision, recall and F1score reported are the macro values taken from the classification report generated using Sklearn inbuilt function.

Table 1

Model	*Training Time*	*Prediction time*	*Accuracy*	*Precision*	*Recall*	*F1score*
CNN	195 sec	847 msec	70 %	71%	86%	75%
XgBoost	45.9 sec	306 msec	70%	86%	71%	75%
SVM	5.11 sec	1.5 sec	70%	86%	76%	78%
LR	8.71 sec	78.4 msec	72%	87%	90%	79%

In this project I try to classify the skin cancer images which makes this a medical domain problem therefore, **recall** becomes my primary metric in comparing the results as we always try to lower the number of false negatives i.e., when someone has a particular type of skin cancer and model classifies it as some other type. Therefore, **logistic regression with thresholding techniques** gives the best results of 90% recall with the augmented test data and also the prediction time is very low for the LR model which makes it the fastest. LR took very less time to train as well when compared with CNN and XgBoost model.

All my models got a nearly 100% training and testing accuracy on the original data due to the novel thresholding approach. As seen in the figure 2 thresholding enhanced the image information and made it easy for classifying the images for the model.

7 Conclusion

Skin cancer being one of the most common and deadliest type of cancer around the world has been spreading faster in past 5 decades. It is very important to build a fast and reliable model to classify between different types of skin cancer as early treatment of the disease has often resulted in a cure. In this project I propose a ML model along with thresholding techniques to implement a model to classify between different types of cancer. I used the HAM10000 dataset to implement and used augmentation techniques along with oversampling of data to fix the data imbalance problem. The Otsu + gaussian thresholding technique enhanced the features of the image making it easier to classify an image. I trained 3 Machine learning models including XgBoost, Logistic regression and support vector machine. I also implemented a CNN network with dropout layers to avoid overfitting of the model. All my models performed extremely well for the training and testing data, therefore I created a augmented test data using the original data to check the robustness of my model. Surprisingly, all the models performed poorly on the augmented test data and all the models misclassified a single class. Potential reason of this could be the similarity in those classes and loss of information due to down shaping the original image from 600*800 to 80*80.

For future scope I would implement a binary classification model to first classify the images for those 2 particular classes or use the image with higher shapes and pixels which would require more computational cost and training time.

In conclusion, Logistic regression model was seen to produce the best results of 90% recall and 87% precision. Recall being my main metric in this problem. LR model also took very less time for training and the lowest time for predicting the results on test data.

REFERENCES

[1]- Xie F., Fan H., Li Y., Jiang Z., Meng R., Bovik A. Melanoma Classification on Dermoscopy Images Using a Neural Network Ensemble Model. IEEE Trans. Med. Imaging. 2017;36:849–858. doi: 10.1109/TMI.2016.2633551

[2]- Aswin R.B., Jaleel J.A., Salim S. Hybrid Genetic Algorithm: Artificial Neural Network Classifier for Skin Cancer Detection; Proceedings of the 2014 International Conference on Control, Instrumentation, Communication and Computational Technologies (ICCICCT); Kanyakumari, India. 10–11 July 2014; pp. 1304–1309.

[3]- 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

[4]- Tschandl, P., Rinner, C., Apalla, Z. et al. Human–computer collaboration for skin cancer recognition. Nat Med 26, 1229–1234 (2020). https://doi.org/10.1038/s41591-020-0942-