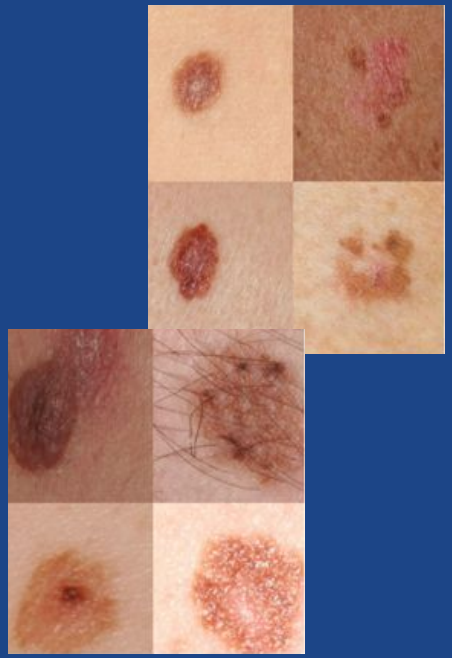
Skin Cancer MNIST: HAM10000

Image Recognition Deep Learning Project



Neil Cabrera

23-Jan 2019

Project Background

Dermatoscopic images from different populations were acquired and stored by different modalities.

The final dataset consists of 10015 dermatoscopic images which can serve as a training set for academic machine learnin purposes.

Cases include a representative collection of all important diagnostic categories: Bowen's disease (**akiec**), basal cell carcinoma (**bcc**), benign keratosis-like lesions (**bkl**), dermatofibroma (**df**), melanoma (**mel**), melanocytic nevi (**nv**) and vascular lesions (**vasc**).

Project Objectives

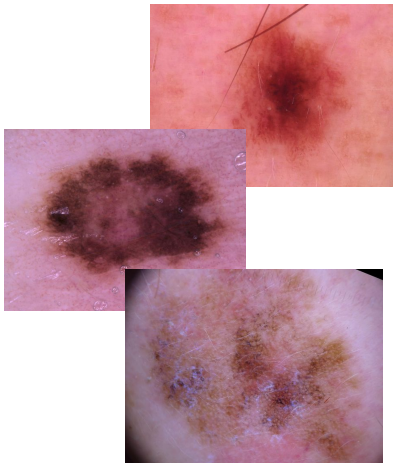
Implement Image Classification Model using Deep Learning to identify the different types of skin cancer based on the image.

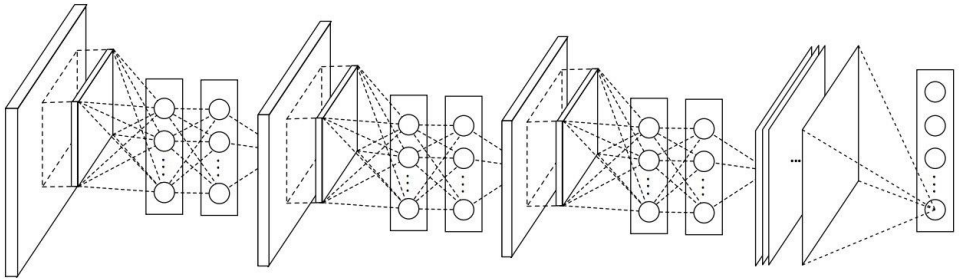
Project will be implemented using Python and Keras-GPU.

Project Strategy

EDA	Prelim Models	Evaluate	Final Model	Interpret
Load the Data and Perform EDA	**Perform Experimentations and Create DL Models**	**Compare the results of different models**	**Create the Final Model**	**Interpret the results and present the project**
Check data consistency, perform cleaning, visualize sample data and distribution of features	Experiment to find the most efficient way to process the data and the best model approach	Compare and update the prelim models until the best	Compile the codes from EDA, File Processing, Modeling to generate the final notebook	

Project Approach





Skin Cancer Classes:
'nv'
'mel'
'bkl'
'bcc'
'akiec'
'vasc'
'df'

In -> (Conv2D(relu)->MaxPool2D->DropOut)*4->Flatten->(Dense->DropOut)*2->Out

Project Approach (cont.)

- Images were resized by 70%
  - Preliminary testing used 50% but consumed too much memory (RAM) and had difficulties fitting in the GPU
  - Images were normalized as they are loaded

- Train/Test Split Ratio is 80/20

- MaxPool2D and DropOut was used in the CNN to perform regularization and avoid overfitting

- Model was evaluated to have 0.7204 accuracy (there is class imbalance in the dataset with one class close to 67% of the entire data)

Project Approach (cont.)

- More than 20 experimentations in modeling were performed to find the best ratio for image size vs speed vs metrics in the modeling

- Final Model was selected by examining the precision and recall for each class as well as the time performed to train the model

- Model can be improved further if given enough time (can experiment on model stacking and by simplifying the classes into 2 for the first model and then pushing the minority classes into the subsequent models)
    - Image Augmentation was also explored but for this case did not help much and in some cases worsened the results

Project Approach (cont.)

- Refer to the Jupyter notebook for the actual code and results

Appendix

I. Project Environment

- Ubuntu 18.04.01 LTS x84
- Anaconda, Jupyter, Python 3.6, Keras-GPU
- Intel Core i5-8400 (6 cores, 2.8Ghz)
- 16 GB RAM with SSD + HDD
- nVidia GTX 1070 8GB
    - used to run gpu-accelerated models

II. References

Dataset and Project Background:
https://dataverse.harvard.edu/dataset.xhtml?persistentId=doi:10.7910/DVN/DBW86T