Computational Methods II: Final Project

Joseph Hostyk and Sarah Yam

2019-3-29

1 Summary

Our analysis will work on recognizing and classifying skin lesion type based on images of skin (with and without lesions) to diagnose skin disease. Automated diagnosis of pigmented skin lesions is constrained by the lack in sample size and diversity of dermatoscopic images, making it difficult to train the necessary neural networks. By using dermatoscopic images from persons of a wide variety of backgrounds, an automated diagnosis of skin lesions may be possible. This could improve the accuracy and efficiency in lesion-based skin disease diagnostics.

Our model will both have to recognize whether or not the image contains a lesion, and if so, the disease phenotype to which the lesion belongs.

2 Dataset to be Analyzed

Our input will be images, with information on both whether a lesion exists in the image and the disease classification of such a lesion.

1. Skin Lesion Images classified under the disease 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).
   (a) NIH images.
   (b) HAM10000 images .
2. Neutral images
   (a) Derm101.
   (b) Potentially could use DeepLesion non-skin images as well to classify whether an image is a skin pigment lesion.

3 Methods

Using convolutional neural networks, we will learn features present in images of skin that contain lesions. By training on a given set of skin lesions, the model will learn to detect images that contain skin lesions of this type. Training a separate model on images of the specific disease classes as stated above, we will determine how to classify images of skin lesions based on disease type. In order to determine accuracy, we can validate the detection of an image containing a skin lesion using a combined set of images with and without skin lesions. Furthermore, we can use a subset of the disease-type classified skin lesion images as a validation set to determine how well the classification model is working. This will involve manually curating the layers involved in the model, learning parameters via a gradient (SGD or Adam), and comparing validation error with testing error. For tweaking the model more efficiently, we may need to modify our model and implement a dynamic convolutional network.

4 Baseline methods

For a more basic model, we will implement the following PyTorch functions: Linear regression, Sigmoidal Functions, Loss Functions (potentially cross entropy loss).

With a more intricate model: We propose implementing CNN layers, regularization through ReLU, and average pooling. Our loss function would likely still be cross entropy loss.

5 Contributions

Data acquisition and scraping: Sarah.
Data preparation and initial analysis: Joe.
Model creation and implementation: Sarah and Joe.