LESION CLASSIFICATION*

Ethan Sargent

December 11, 2019

*This work was not supported by any organization. Ethan Sargent graduated from Harvey Mudd College in 2019 with a B.S. in Mathematics. Contact: `esargent at hmc dot edu`

1 Problem Statement

The International Skin Imaging Collaboration (ISIC) archive contains 23k images of classified skin lesions, along with patient and biopsy result metadata. The main goal of this project is designing a classifier which accurately predicts malignance. We deliver two models - a binary classifier, which predicts with AUC=.96 whether an lesion image is benign or in one of 18 malignant lesion classes, and a multiclassifer, which classifies a lesion image as benign or as one of 8 malignant image classes - our multiclassifier has AUC>.96 for each of these classes. Both classifiers are pretrained deep neural networks with the Densenet-161 architecture, and were fine-tuned on the ISIC archive using PyTorch.

An effective skin lesion classifier is of interest to clients in the medical field for obvious reasons. The client might be an insurance company, a doctor, a patient, or a clinical trial director.

2 Table of Contents

Contents

3 Exploratory Data Analysis

3.1 Summary

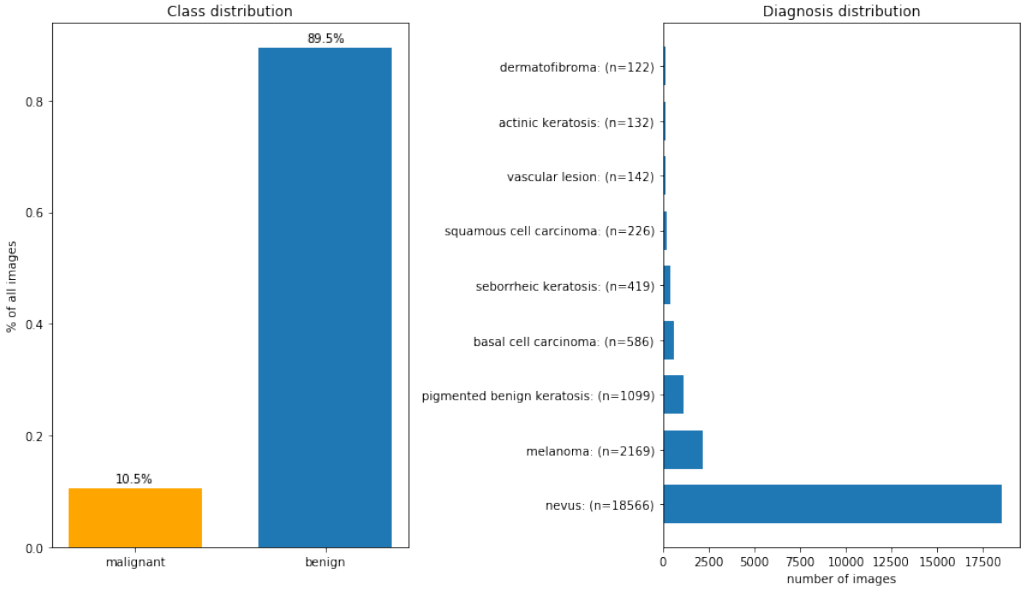
The ISIC archive, available at [3], contains 23,906 images of skin lesions and associated clinical metadata, as of December 11, 2019. The images are available to the public through an API. For purposes of this project, we are interested in distinguishing between malignant and benign lesions, and in a multiclassifier which can predict the specific lesion diagnosis subclass. Of the 23,906 images, 21,693 have a label of malignant or benign, so we restrict our binary classifier to these data. 23,461 lesion images have an associated diagnosis subclassification - a diagnosis classifies a lesion as one of

1. nevus (n=18566)
2. melanoma (n=2169)
3. pigmented benign keratosis (n=1099)
4. basal cell carcinoma (n=586)
5. seborrheic keratosis (n=419)
6. squamous cell carcinoma (n=226)
7. vascular lesion (n=142)
8. actinic keratosis (n=132)
9. dermatofibroma (n=122)
10. lentigo NOS (n=71)
11. solar lentigo (n=57)
12. lentigo simplex (n=27)
13. angioma (n=15
14. atypical melanocytic proliferation (n=13)
15. other (n=10)
16. angiofibroma or fibrous papule (n=1)
17. lichenoid keratosis (n=1
18. scar (n=1)

3.2 Class Imbalance

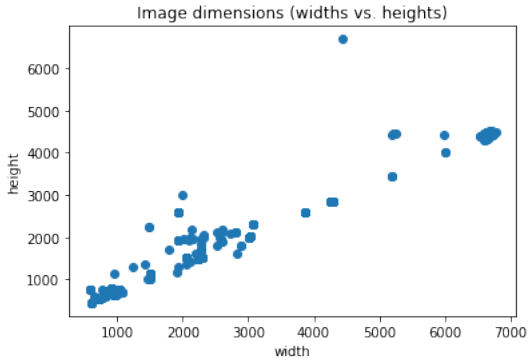
Of the 21,693 relevant images, only 2,286 are malignant. This presents challenges to a potential binary classifier. In particular, a naive classifier (one that only guesses the negative class) has an accuracy of 89.5% on this dataset, implying we will need to use more robust metrics. These will include receiver-operating-characteristic (ROC) and precision-recall curves, the area under these curves, and $F-1$ score.

The vast majority of images in the archive are classified as benign nevi - per [4], nevus is the medical term for mole. The most common malignant class is melanoma, with 2,169 examples. The multiclassifier is trained on 9 lesion diagnosis classes, including the benign class. There are 19 diagnosis classes in total - we train only on those with more than 100 examples. The 9 relevant classes and their distributions are shown below.



3.3 Image Shapes

Below is a scatter plot of image dimensions. The image sizes will be standardized before they are fed into our models, so we desire some notion of an average shape. It appears almost all images are slightly wider than they are tall. Concretely, we compute the median aspect ratio (0.66) as the quotient of the median height, with the median width and resize all images to shape $(216, \lfloor 216/0.66 \rfloor, 3)$.



3.4 Preprocessing

3.4.1 Binary Classifier

The image shapes are standardized by the procedure above and then loaded into large PyTorch Tensors. Prior to being fed into the model, the images are batch-normalized along

each channel (red, green, and blue). To compensate for the class imbalance, we first employed a weighted cross-entropy loss function - concretely, misclassifications in the positive class were weighted by some $\gamma > 1$. In practice $\gamma \in [5, 10]$ were used with mixed results; the neural network typically did not converge, or converged to a uniform model (i.e one that always predicted either the positive or negative class).

We had more success oversampling the underrepresented classes. Concretely, images were selected during batch formation with probability inversely proportional to their class frequency. That is, an image is selected, with replacement, for inclusion in a batch, with relative probability

$$w_i = \frac{n}{n_i},$$

where n is the total number of training examples and n_i is the number of training examples in class i. This is achieved with PyTorch's `WeightedRandomSampler`. Thus the model "sees" roughly equal numbers of malignant and benign examples.

3.4.2 Multiclassifier

The class imbalance is more severe in the multiclass case. In particular, the smallest lesion class, dermatofibroma, has only 122 examples, compared to the nevus class which has 18566. To compensate, we use image augmentations.

An image augmentation is a modification of an image which leaves the image mostly intact - common modifications include rotation, horizontal flipping, and cropping. In the domain of image classification, certain augmentations can be applied at random during batch formation while training, so that the classifier sees differently augmented images in different epochs. Image augmentations can thus be viewed as a way to create synthetic data in each class, with the helpful effect of reducing overfitting in the underrepresented classes.

Using PyTorch's `ImageFolder` utility, we compose the following image augmentations pipeline.

- Training
    1. `RandomResizedCrop((300, 300)`,
    2. `RandomHorizontalFlip()`,
    3. `ToTensor()`
- Validation
    1. `Resize((300,300))`,
    2. `ToTensor()`

Note that augmentations are applied at random only during training - during validation, there is no need to create synthetic data since we are not training the network.

As with the binary classifier, we also employ oversampling of the underrepresented classes during batch formation. Rather than assign probabilities to each example based on their inverse class frequency, we weight examples by the log of their inverse class frequency. Concretely, for an image in class i, where there are n total training examples and n_i examples in class i, the image is selected with relative probability

$$w_i = \log \frac{n}{n_i}.$$

Since some classes are quite small, selecting images from the dominant class with the same frequency as images from the underrepresented classes would almost definitely result in overfitting of the underrepresented classes, so we ought not to use raw inverse class frequency as a weight. The log is motivated by a desire for a smoothed distribution of inverse class frequency.

4 Models

4.1 Overview of DenseNet Architecture

The DenseNet architecture for deep convolutional neural networks was introduced in [8] and empirically demonstrated state-of-the-art performance on various benchmark datasets. From the abstract:

> DenseNets have several compelling advantages: they alleviate the vanishing-gradient problem, strengthen feature propagation, encourage feature reuse, and substantially reduce the number of parameters.

The computational efficiency and accuracy of DenseNets arise from the notion, introduced in [8], of dense connections between layers of a neural network. Concretely, dense layer l of a dense block receives as input the outputs $[x_{l-1}, \ldots, x_0]$ of the preceding $l-1$ layers, concatenated into a single tensor. By connecting each layer to each subsequent layer, the authors introduce a principled way of addressing the vanishing gradient problem - that is, the tendency of early-layer information in deep neural networks to vanish by the time it reaches the end.

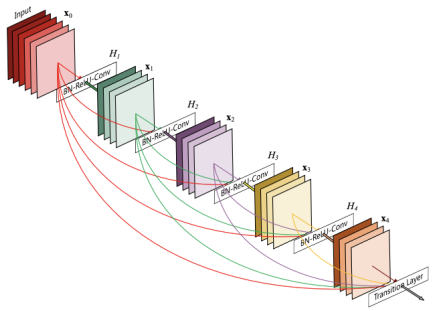


Figure 1: Graphic reprinted from [8]. While not shown, dense connections also exist between the layers of the individual dense blocks.

DenseNets owe their computational efficiency to their relatively narrow layers - compared to other deep neural network architectures, DenseNet convolutional layers have fewer filters, even though layers have high "collective knowledge" since they receive input from all preceding layers.

DenseNets are ideal for our purposes, both for their speed, and for their concrete interpretation in a transfer-learning scenario. In particular, our models are pre-trained Densenet-161 networks with the weights in the last dense block unfrozen and fine-tuned on the ISIC archive. The models were hosted on a Paperspace Cloud VM instance and trained on a single GPU.

4.2 Binary Classifier

4.2.1 Training

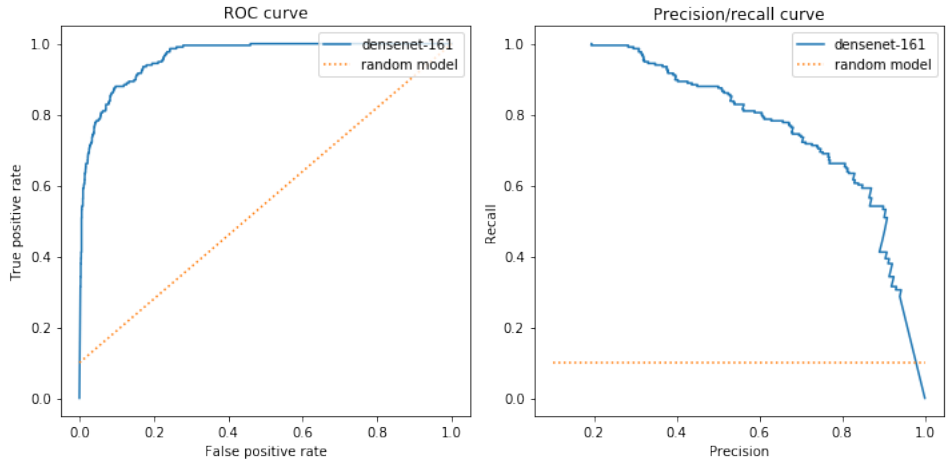
As described in in 3.4.1, we employed oversampling of the positive class using the PyTorch package `WeightedRandomSampler`. A batch size of 64 was used, the learning rate was tuned with the Adam optimizer, and the neural network was trained for 100 epochs.

4.2.2 Evaluation

The final model is a binary classifier. A table of metrics is shown below. Per [6], the Area under the ROC curve (AUC) is a preferred metric for medical diagnosis, and per [7], it has a concrete interpretation as the "the probability that a randomly chosen diseased subject is rated or ranked as more likely to be diseased than a randomly chosen nondiseased subject." A perfect model has AUC 1, while a random model has AUC .5.

Metric	Value
Accuracy	95.07%
True negatives	88.47%
False positives	1.568%
False negatives	3.366%
True positives	6.593%
F1 score	.7277
Average precision score:	.7863
Area under ROC curve	.9616

The ROC and precision/recall curves are shown below, along with the hypothetical curves of a random model for comparison.



4.3 Multiclassifier

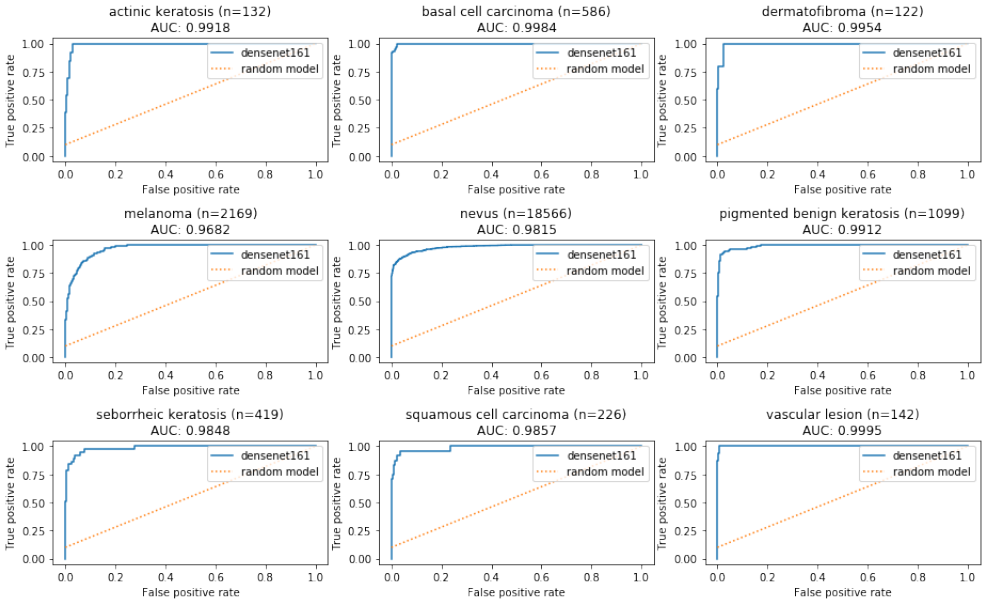
4.3.1 Training

Recall that the multiclassifier is trained on 9 lesion diagnosis classes. The 9 relevant classes and their distributions are shown in the figure in section 3.2. As with the binary classifier, we use the Adam optimizer, the cross-entropy loss function, and train on a single GPU for 100 epochs. Batches of size 64 were collated following the procedure outlined in section 3.4.2.

4.3.2 Evaluation

For each lesion class, our classifier has an AUC of over .96. This is on par with the state of the art classifier described in [9], albeit for different classes and on different datasets.

Our classifier has a balanced multiclass accuracy of 73.40%. The balanced multiclass accuracy is the arithmetic mean of the true-positive-rates for each class.



References

[1] S. G. Finlayson, J. D. Bowers, J. Ito, J. L. Zittrain, A. L. Beam, I. S. Kohane1. "Adversarial attacks on medical machine learning." Science, 22 March 2019. https://science.sciencemag.org/content/363/6433/1287

[2] "The Department of Health and Human Services And The Department of Justice Health Care Fraud and Abuse Control Program Annual Report For FY 2007." United States Department of Health and Human Services, November 2008. https://oig.hhs.gov/publications/docs/hcfac/hcfacreport2007.pdf

[3] ISIC Archive, n.d,. https://www.isic-archive.com

[4] Mayo Clinic, 19 November 2019. https://www.mayoclinic.org/diseases-conditions/moles/symptoms-causes/syc-20375200

[5] A. Ilyas, S. Santurkar, D. Tsipras, L. Engstrom, B. Tran, and A. Mądry. "Adversarial Examples Are Not Bugs, They Are Features." NIPS, 2019. https://arxiv.org/pdf/1905.02175.pdf

[6] K. Hajian-Tilaki. "Receiver Operating Characteristic (ROC) Curve Analysis for Medical Diagnostic Test Evaluation." Caspian J Intern Med 2013. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3755824/pdf/cjim-4-627.pdf

[7] J. A. Hanley, B. J. McNeil. "The Meaning and Use of the Area under a Receiver Operating Characteristic (ROC) Curve." Radiology 143, April 1982. https://pubs.rsna.org/doi/pdf/10.1148/radiology.143.1.7063747

[8] G. Huang, Z. Liu, L. van der Maaten, K. Q. Weinberger. "Densely Connected Convolutional Networks." IEEE Conference on Computer Vision and Pattern Recognition (CVPR), July 2017. https://arxiv.org/pdf/1608.06993.pdf

[9] A. Esteva, B. Kuprel, R. A. Novoa, J. Ko, S. M. Swetter, H. M. Blau, S. Thrun. "Dermatologist-level classification of skin cancer with deep neural networks." January 2017. https://www.nature.com/articles/nature21056

[10] "ISIC 2019 Leaderboards." https://challenge2019.isic-archive.com/leaderboard.html