

A decorative graphic on the left side of the slide consisting of two overlapping parallelograms. The front one is blue and the back one is a light green. They are positioned diagonally, with the blue one partially covering the green one.

Skin Lesion Classification for Cancer Detection

Ryan Slattery



The Problem

- Melanoma kills about 60,000 people a year
- Early detection has high impact on survival rate
- Professional medical diagnosis costs money/time



Dataset

- HAM10000 dataset (<https://arxiv.org/pdf/1803.10417.pdf>)
- Over 10000 labelled images of skin lesions
- 7 classes, 3 are cancerous



Lesion Classes

- Cancerous
 - mel - Melanoma
 - akiec - Aktinic keratoses or intraepithelial carcinoma
 - bcc - Basal cell carcinoma
- Benign
 - bkl - Benign keratoses
 - df - Dermatofibroma
 - nv - Melanocytic nevi
 - vasc - Vascular lesions

Lesion Class Examples

Cancerous

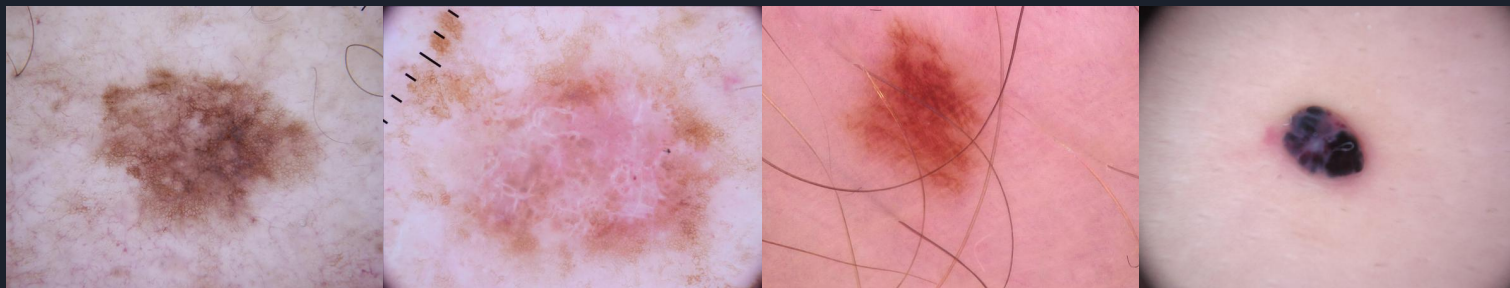


akiec

mel

bcc

Benign



bkl

df

nv

vasc



Modeling Overview

- Created 2 CNNs in PyTorch to do image classification
 - LeNet-like architecture trained from scratch
 - ResNet-50 pre trained on ImageNet and fine tuned on HAM10000

LeNet-like Details

Architecture:

Layer (type:depth-idx)	Output Shape	Param #
Conv2d: 2-1	[64, 16, 224, 224]	1,216
ReLU: 2-2	[64, 16, 224, 224]	--
MaxPool2d: 2-3	[64, 16, 112, 112]	--
Dropout: 2-4	[64, 16, 112, 112]	--
Conv2d: 2-5	[64, 16, 112, 112]	2,320
ReLU: 2-6	[64, 16, 112, 112]	--
MaxPool2d: 2-7	[64, 16, 56, 56]	--
Dropout: 2-8	[64, 16, 56, 56]	--
Conv2d: 2-9	[64, 32, 56, 56]	4,640
ReLU: 2-10	[64, 32, 56, 56]	--
MaxPool2d: 2-11	[64, 32, 28, 28]	--
Dropout: 2-12	[64, 32, 28, 28]	--
Flatten: 2-13	[64, 25088]	--
Linear: 2-14	[64, 512]	12,845,568
ReLU: 2-15	[64, 512]	--
Dropout: 2-16	[64, 512]	--
Linear: 2-17	[64, 512]	262,656
ReLU: 2-18	[64, 512]	--
Dropout: 2-19	[64, 512]	--
Linear: 2-20	[64, 7]	3,591

Hyperparameters:

```
batch size = 64
optimizer = Adam
learning rate = 1e-4
dropout = 0.2
```

ResNet-50 Details

Architecture:

See more info at <https://pytorch.org/vision/stable/modules/torchvision/models/resnet.html>

Layer (type:depth-idx)	Output Shape	Param #
Conv2d: 2-1	[64, 64, 112, 112]	(9,408)
BatchNorm2d: 2-2	[64, 64, 112, 112]	(128)
ReLU: 2-3	[64, 64, 112, 112]	--
MaxPool2d: 2-4	[64, 64, 56, 56]	--
Sequential: 2-5	[64, 256, 56, 56]	(215,808)
Sequential: 2-6	[64, 512, 28, 28]	(1,219,584)
Sequential: 2-7	[64, 1024, 14, 14]	7,098,368
Sequential: 2-8	[64, 2048, 7, 7]	14,964,736
AdaptiveAvgPool2d: 2-9	[64, 2048, 1, 1]	--
Sequential: 2-10	[64, 7]	14,343

Hyperparameters:

```
pretrained = True
freeze = True
batch size = 64
optimizer = Adam
learning rate = 1e-5
dropout = 0.5
```




Model Comparison

- LeNet-Like validation accuracy: 49%
- ResNet-50 validation accuracy: 74%
- ResNet-50 test accuracy: 76%



Recommendations

- Deploy as app that can provide tentative diagnosis
- Keep a human in the loop
- Improve further by acquiring more data



Limitations

- Our model suffers low accuracy
- Visual analysis will likely never fully replace biopsy



Future Improvement

- More data would get us better results
- Synthetic data could be generated
(<https://medium.com/abacus-ai/gans-for-data-augmentation-21a69de6c60b>)
- Data flywheel could be made in deployment



Questions?