Machine Learning Engineer Nanodegree

Skin Cancer Detection Capstone Report

Srinivasa Amirapu
Dec 19th, 2018

Definition

Project Overview

Skin cancer is the most common cancer. Each year there are more new cases of skin cancer than the combined incidence of cancers of the breast, prostate, lung and colon. Over the past three decades, more people have had skin cancer than all other cancers combined. Skin cancer is a major public health problem, with over 5 million newly diagnosed cases in the United States each year. Melanoma is the deadliest form of skin cancer, responsible for over 9,000 deaths each year. The estimated 5-year survival rate for patients whose melanoma is detected early is about 98 percent in the U.S. The survival rate falls to 62 percent when the disease reaches the lymph nodes, and 18 percent when the disease metastasizes to distant organs.

Personal Motivation:
As the computers are getting better at understanding the images due to advances in Deep Learning , We can apply Deep Learning with Transfer Learning to develop models which help doctors in early detection of skin cancer. To err is human, to forgive is divine. But to forgive an human error we need a better backup from a AI machine to validate our errors . In the current context we do not want to send sick patients home i.e., patients with malignant tumor should not be misclassified and sent home.

In this Project, I created predictive model for skin cancer detection which can help in increasing survival rate of the patients. This predictive model for skin cancer detection can help doctors to better serve patients by offering them pre-screening.

Datasets

The data is pulled from the 2017 ISIC Challenge on Skin Lesion Analysis Towards Melanoma Detection

The 2017 challenge consisted of 3 tasks: lesion segmentation, dermoscopic feature detection, and disease classification. For the capstone project we would focus on disease classification task.

The training, validation, and test datasets continue to be available for download from the following address: http://challenge2017.isic-archive.com/ .

Disease Classification Task: Participants were asked to classify images as belonging to one of 3 categories , including “melanoma” (374 training, 30 validation, 117 test), “seborrheic keratosis” (254, 42, and 90), and “benign nevi” (1372, 78, 393), with classification scores normalized between 0.0 to 1.0 for each category (and 0.5 as binary decision threshold). Lesion classification data included the original image paired with the gold standard diagnosis, as well as approximate age (5 year intervals) and gender when available. But for this capstone we would only consider images for training the model and drop age and gender.

Original image sizes are of various sizes from 1022 x 767 to 4288 x 2848. I wanted to use 1022 x 767 to preserve all patterns and features of the original images, but would have been computationally very expensive, so I reduced image size to 224 x 224.

```
def path_to_tensor(img_path):
    # loads RGB image as PIL.Image.Image type
    img = image.load_img(img_path, target_size=(224, 224))
    # convert PIL.Image.Image type to 3D tensor with shape (224, 224, 3)
    x = image.img_to_array(img)
    # convert 3D tensor to 4D tensor with shape (1, 224, 224, 3) and return 4D tensor
    return np.expand_dims(x, axis=0)
```

Problem Statement

The goal of the challenge is to develop image analysis tools to enable the automated diagnosis of melanoma from dermoscopic images. The main objective is to design an algorithm that can visually diagnose melanoma, the deadliest form of skin cancer. Our algorithm will distinguish this malignant skin tumor from two types of benign lesions (nevi and seborrheic keratoses). The data and objective are pulled from the 2017 ISIC Challenge on Skin Lesion Analysis Towards Melanoma Detection. As part of the challenge, participants were tasked to design an algorithm to diagnose skin lesion images as one of three different skin diseases (melanoma, nevus, or seborrheic keratosis).

We will create a model to generate our own predictions by training a CNN that would be able to classify skin lesion images into these 3 classes. Deep learning based techniques (CNNs) has been very popular in the last few years where they consistently outperformed traditional approaches for feature extraction to the point of winning

imagenet challenges. In this project, transfer learning along with data augmentation will be used to train a convolutional neural network to classify images of skin lesion to their respective classes. Transfer learning refers to the process of using the weights from pre-trained networks on large dataset. As the pretrained networks have already learnt how to identify lower level features such as edges, lines, curves etc with the convolutional layers which is often the most computationally time consuming parts of the process, using those weights help the network to converge to a good score faster than training from scratch. Fortunately many such networks such as RESNET, InceptionV3, VGG16 pretrained on imagenet challenge is available for use publicly.

Solution Statement

As deep learning techniques have been very effective in image classification over the years, in this project, transfer learning along with data augmentation will be used to train a convolutional neural network to classify images of skin lesions to their respective classes. Transfer learning refers to the process of using the weights from pre-trained networks on large dataset.

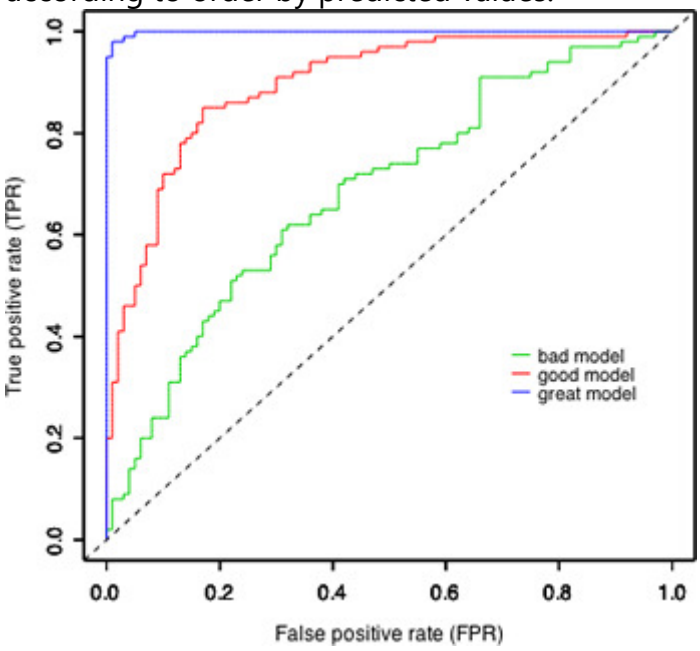
For training skin cancer detection model I will use weights from Pre-trained Keras models (VGG19,ResNet50,InceptionV3) and apply transfer learning which produces solution for this multi-class image classification problem. Finally I will select the model which yields better accuracy for predictions on test set. Residual Networks, introduced by He et al., allow you to train much deeper networks than were previously practically feasible.The skip-connections help to address the Vanishing Gradient problem.

Evaluation Metrics

ROC AUC for Melanoma Classification

The probabilistic interpretation of ROC-AUC score is that if you randomly choose a positive case and a negative case, the probability that the positive case outranks the

negative case according to the classifier is given by the AUC. Here, rank is determined according to order by predicted values.



Source of Image: UNC Lecture
Mathematically, it is calculated by area under curve of sensitivity (TPR) vs. FPR(1-specificity). Ideally, we would like to have high sensitivity & high specificity, but in real-world scenarios, there is always a tradeoff between sensitivity & specificity.

In the this category, we will gauge the ability of your CNN to distinguish between malignant melanoma and the benign skin lesions (nevus, seborrheic keratosis) by calculating the area under the receiver operating characteristic curve (ROC AUC) corresponding to this binary classification task. This is one of the evaluation metric used by the 2017 ISIC Challenge on Skin Lesion Analysis Towards Melanoma Detection to rank various teams. ROC-AUC score is independent of the threshold set for classification because it only considers the rank of each prediction and not its absolute value. The same is not true for F1 score which needs a threshold value in case of probabilities output. For this reason, I will use ROC Curves and AUC score to distinguish between malignant melanoma and the benign skin lesions (nevus, seborrheic keratosis).

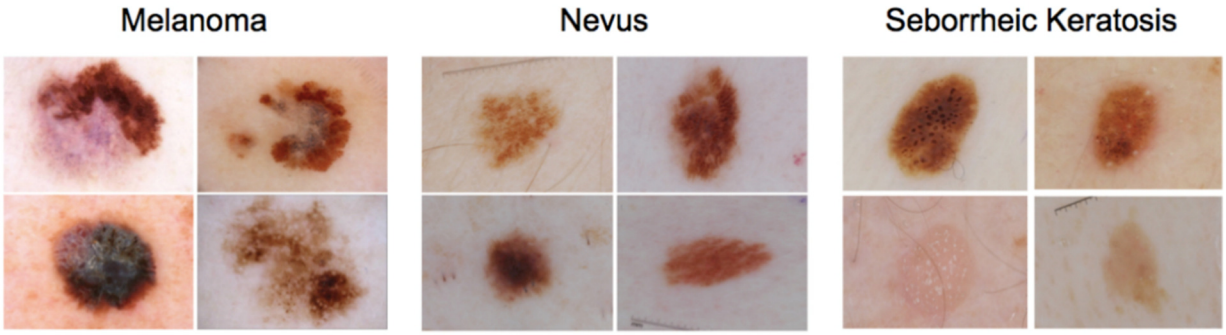
Analysis

Data Exploration

Dataset consists of images including “melanoma” (374 training, 30 validation, 117 test), “seborrheic keratosis” (254, 42, and 90), and “benign nevi” (1372, 78, 393), with

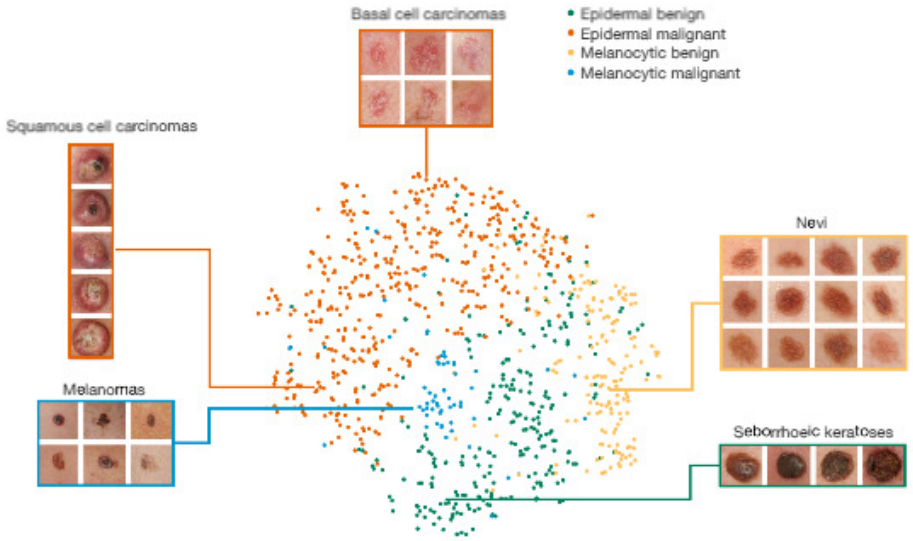
classification scores normalized between 0.0 to 1.0 for each category (and 0.5 as binary decision threshold).

Picture of various classes of skin cancer for this project:-



Exploratory Visualization(t-SNE analysis)

There are lots of images with significant variations in the color intensities as the images were taken from different cameras, different angles and at the different times of the day. There are only few thousand images available for the training and this can be a difficult task to classify these images correctly. Below is the t-SNE visualization of last hidden layer representations in the CNN. Colored point clouds represent the different disease categories, showing how the algorithm clusters the diseases.



Algorithms and Techniques

1. **Deep Learning** - Deep learning (also known as deep structured learning or hierarchical learning) is part of a broader family of machine learning methods based on learning data representations, as opposed to task-specific algorithms.

2. **Convolutional neural network** - In machine learning, a convolutional neural network (CNN or ConvNet) is a class of deep, feed-forward artificial neural networks that has successfully been applied to analyzing visual imagery. A CNN consists of an input and an output layer, as well as multiple hidden layers. The hidden layers are either convolutional, pooling or fully connected. We give CNN an input and it learns by itself that what features it has to detect. We won't specify the initial values of features or what kind of patterns it has to detect.

Various Layers:-

Convolutional - Also referred to as Conv. layer, it forms the basis of the CNN and performs the core operations of training and consequently firing the neurons of the network. It performs the convolutional operation over the input.

Pooling layers -Pooling layers reduce the spatial dimensions (Width x Height) of the input Volume for the next Convolutional Layer. It does not affect the depth dimension of the Volume.

Fully connected layer - The fully connected or Dense layer is configured exactly the way its name implies. It is fully connected with the output of the previous layer. Fully connected layers are typically used in the last stages of the CNN to connected to the output layer and construct the desired number of outputs.

Dropout layer - Dropout is a regularization technique for reducing overfitting in neural networks by preventing complex co-adaptations on training data. It is a very efficient way of performing model averaging with neural networks. The term "dropout" refers to dropping out units (both hidden and visible) in a neural network.

Flatten - Flattens the output of the convolutional layers to feed into the Dense layers.

3. **Activation Functions -** In CNN, the activation function of a node defines the output of
that node given an input or set of inputs. Some activation functions are:

Softmax - The softmax function squashes the output of each unit to be between 0 and 1, just like a sigmoid function. It also divides each output such that the total sum of the outputs is equal to 1.

ReLu - A ReLu (or rectified linear unit) has output 0 if the input is less than 0, and raw output otherwise. i.e, if the input is greater than 0, the output is equal to the input.

4. **Transfer Learning** - In transfer learning, we take the learned understanding and pass it to a new deep learning model. We take a pre-trained neural network and adapt it to a new neural network with different dataset.
For this problem we use ResNet50 network. The reason is explained in the section "*Why ResNet Model with 50 layers?*".

Benchmark Model

Algorithm Selection

A wide class of models can be used for image classification with weights trained on ImageNet:

- Xception
- VGG16
- VGG19
- ResNet50
- InceptionV3
- InceptionResNetV2
- MobileNet
- DenseNet
- NASNet
- MobileNetV2

All of these architectures are compatible with the Tensor Flow backends provided in the AWS EC2 instance.

A CNN model made from scratch is used as benchmark model for measuring the performance of transfer learning approach.

Architecture for Benchmark model:-

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_1 (Conv2D)            (None, 223, 223, 16)      208
_________________________________________________________________
dropout_1 (Dropout)          (None, 223, 223, 16)      0
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 111, 111, 16)      0
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 110, 110, 32)      2080
_________________________________________________________________
dropout_2 (Dropout)          (None, 110, 110, 32)      0
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 55, 55, 32)        0
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 54, 54, 64)        8256
_________________________________________________________________
max_pooling2d_3 (MaxPooling2 (None, 27, 27, 64)        0
_________________________________________________________________
dropout_3 (Dropout)          (None, 27, 27, 64)        0
_________________________________________________________________
flatten_1 (Flatten)          (None, 46656)             0
_________________________________________________________________
dense_1 (Dense)              (None, 500)               23328500
_________________________________________________________________
dropout_4 (Dropout)          (None, 500)               0
_________________________________________________________________
dense_2 (Dense)              (None, 3)                 1503
=================================================================
Total params: 23,340,547
Trainable params: 23,340,547
Non-trainable params: 0
_________________________________________________________________
```

Methodology

Data Preparation

These Datasets were also available in Udacity github repository and were downloaded by following below steps:-

1. Clone the repository and create a data/ folder to hold the dataset of skin images.

2. git clone https://github.com/udacity/dermatologist-ai.git

3. mkdir data; cd data

4. Create folders to hold the training, validation, and test images.

5. mkdir train; mkdir valid; mkdir test

6. Download and unzip the training data (5.3 GB).

7. Download and unzip the validation data (824.5 MB).

8. Download and unzip the test data (5.1 GB).

9. Place the training, validation, and test images in the data/ folder, at data/train/, data/valid/, and data/test/, respectively. Each folder should contain three sub-folders (melanoma/, nevus/, seborrheic_keratosis/), each containing representative images from one of the three image classes.

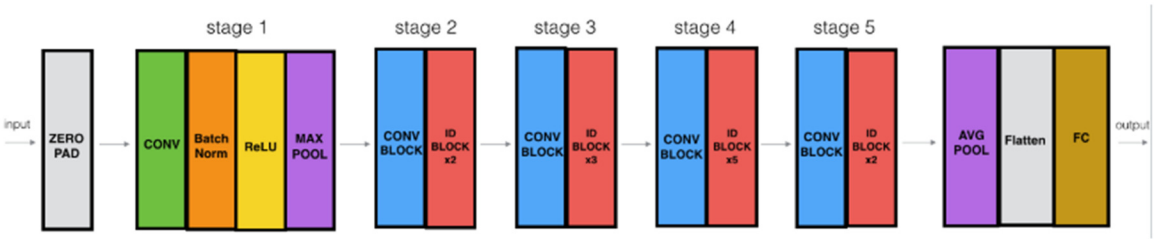
Data Preprocessing

When using TensorFlow as backend, Keras CNNs require a 4D tensor as input, with shape (nb_samples, rows, columns, channels) where nb_samples corresponds to the total number of images (or samples), and rows, columns, and channels correspond to the number of rows, columns, and channels for each image, respectively.
The path_to_tensor function in the notebook takes a string-valued file path to a color image as input and returns a 4D tensor suitable for supplying to a Keras CNN.
The function first loads the image and resizes it to a square image that is 224*224 pixels. Next, the image is converted to an array, which is then resized to a 4D tensor. In this case, since we are working with color images, each image has three channels. Likewise, since we are processing a single image (or sample), the returned tensor will always have shape(1,224,224,3).
The paths_to_tensor function takes a numpy array of string-valued image paths as input and returns a 4D tensor with shape (nb_samples,224,224,3).
Here, nb_samples is the number of samples, or number of images, in the supplied array of image paths.

Model Selection: *Why ResNet Model with 50 layers?*

To model image classifier we would need very deep network like the one used in the referenced publication -Dermatologist-level classification of skin cancer with deep neural networks* which used InceptionV3 model to represent very complex functions. Very deep "plain" networks don't work in practice because they are hard to train due to vanishing gradients.

Residual Networks, introduced by He et al., allow you to train much deeper networks than were previously practically feasible.The skip-connections help to address the Vanishing Gradient problem. They also make it easy for a ResNet block to learn an identity function. There are two main type of blocks: The identity block and the convolutional block.Very deep Residual Networks are built by stacking these blocks together.



Implementation

Create a CNN to Classify Skin Cancer (using Transfer Learning)

We used a proven technique - transfer learning to create a CNN that can classify images into melanomas, nevus, or SBK. Our CNN attained more than 70% accuracy on the test set.

We applied transfer learning to create a CNN Model using bottleneck features obtained from ResNet-50 pre-trained network trained on ImageNet data (unlike in the reference publication -Dermatologist-level classification of skin cancer with deep neural networks* which used Inception-V3 model) :

We further performed below steps to fine tune our CNN model to yield better accuracy:

1. Image augmentation: Boosted image dataset by implementing Rotations, noising, scaling
2. Transfer Learning: We added a global spatial average pooling layer to the last convolutional output of ResNet50 Model trained on ImageNet followed by fully-connected layer with 'relu' activation and a fully connected layer, where the latter contains one node for each skin cancer category(melanomas, nevus, or SBK) and is equipped with a softmax.

Refinement:

We were able to get about 65 percent accuracy by fine tuning above CNN classifier trained on initial dataset with 2000 images . After boosting image dataset by image augmentation, we were able to increase accuracy by 6 percent to get 71 percent from 60 percent of benchmark model. Model accuracy can further be improved by . 1)First way is tuning hyperparameters 1)increase the number of layers with a more gradual decrease in the number of units in each. 2)Second way could be adding more images to training set which are obtained by data augmentation 3)The third way is to try to add one more convolution and pooling layers after loading the bottleneck features

Data augmentation

We used codebox python utility (https://codebox.net/pages/image-augmentation-with-python) to scan directory containing image files, to generate images by performing a specified set of augmentation operations on each file that it finds. This process multiplies the number of training examples that can be used to significantly improve the resulting network's performance, particularly when the number of training examples is relatively small. We boosted image set with total skin images of 14307. Out of which training set included skin cancer images of 13557.

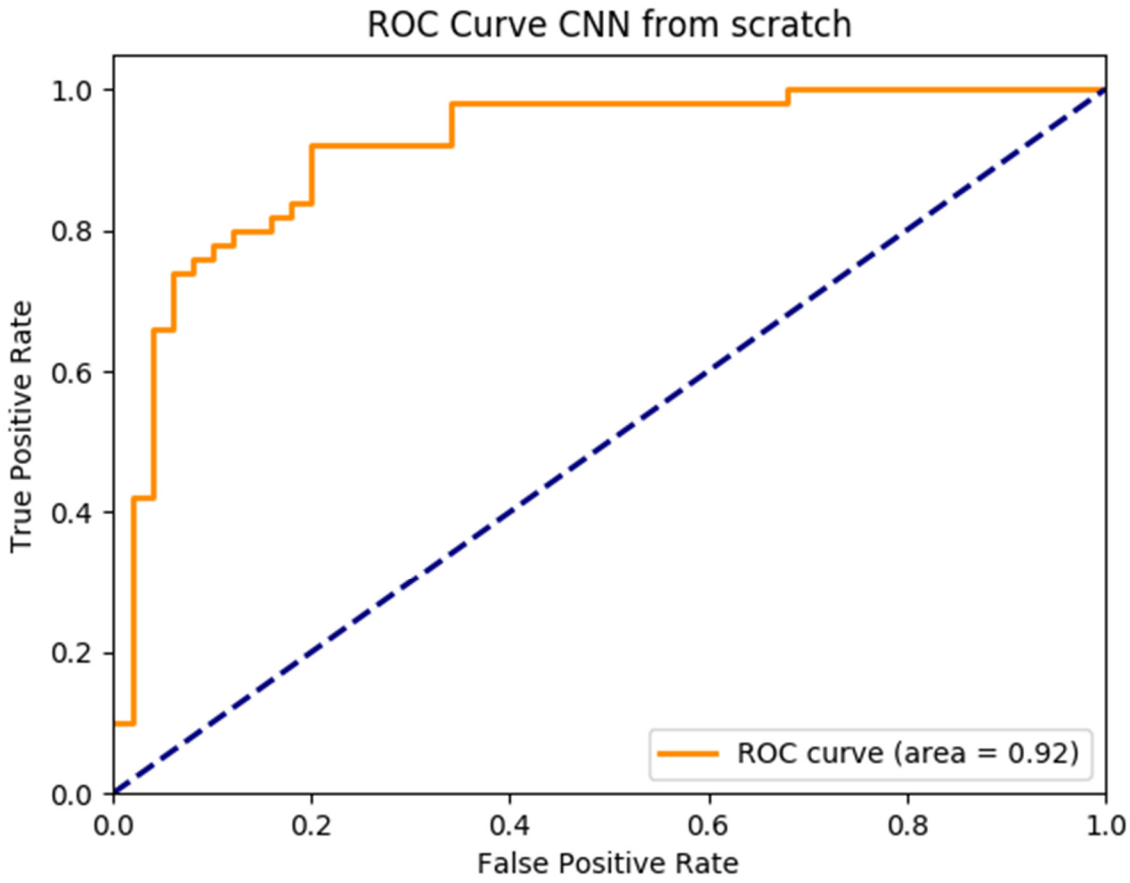
Below Data augmentation operations were performed, using the codes listed in the table below:

Code	Description	Used Values
Fliph	Horizontal Flip	Fliph
Flipv	Vertical Flip	Flipv
Noise	Adds random noise to the image	noise_0.01 noise_0.5
Rot	Rotates the image by the specified amount	rot_90 rot_-45
Trans	Shifts the pixels of the image by the specified amounts in the x and y directions	trans_20_10 trans_-10_0
Zoom	Zooms into the specified region of the image, performing stretching/shrinking as necessary	zoom_0_0_20_20 zoom_-10_-20_10_10
Blur	Blurs the image by the specified amount	blur_1.5

Results

First Model: CNN from scratch, no data augmentation

Simple Convolutional Neural Network with 3 layers. The results obtained until now can be shown on the ROC curve presented below:



Classification Report CNN from scratch, CV Folder.

- 110 epochs. No early stop.
- AUC: 0.9164

Class	Precision	Recall	f1-score	support
0.0	0.86	0.88	0.87	50
1.0	0.88	0.86	0.87	50
avg / total	0.87	0.87	0.87	100

Second Model: VGG16 + Dense Layer

Classification Report VGG16 + Dense Layer.

- 100 epochs. No early stop.

Class	Precision	recall	f1-score	support
0.0	0.87	0.92	0.89	50
1.0	0.91	0.86	0.89	50
avg / total	0.89	0.89	0.89	100

- AUC: 0.9496

Classification Report VGG16 + Dense Layer.

- 100 epochs.ModelCheckpoint. Best Val Accuracy
- AUC: 0.93

Class	Precision	recall	f1-score	support
0.0	0.82	0.94	0.88	50
1.0	0.93	0.80	0.86	50
avg / total	0.88	0.87	0.87	100

Third Model: CNN + Data Augmentation

Classification Report CNN Scratch with Data Augmentation.

- 100 epochs.ModelCheckpoint. Best Val Accuracy

- AUC: 0.9444

Class	Precision	Recall	f1-score	support
0.0	0.81	0.96	0.88	50
1.0	0.95	0.78	0.86	50
avg / total	0.88	0.87	0.87	100

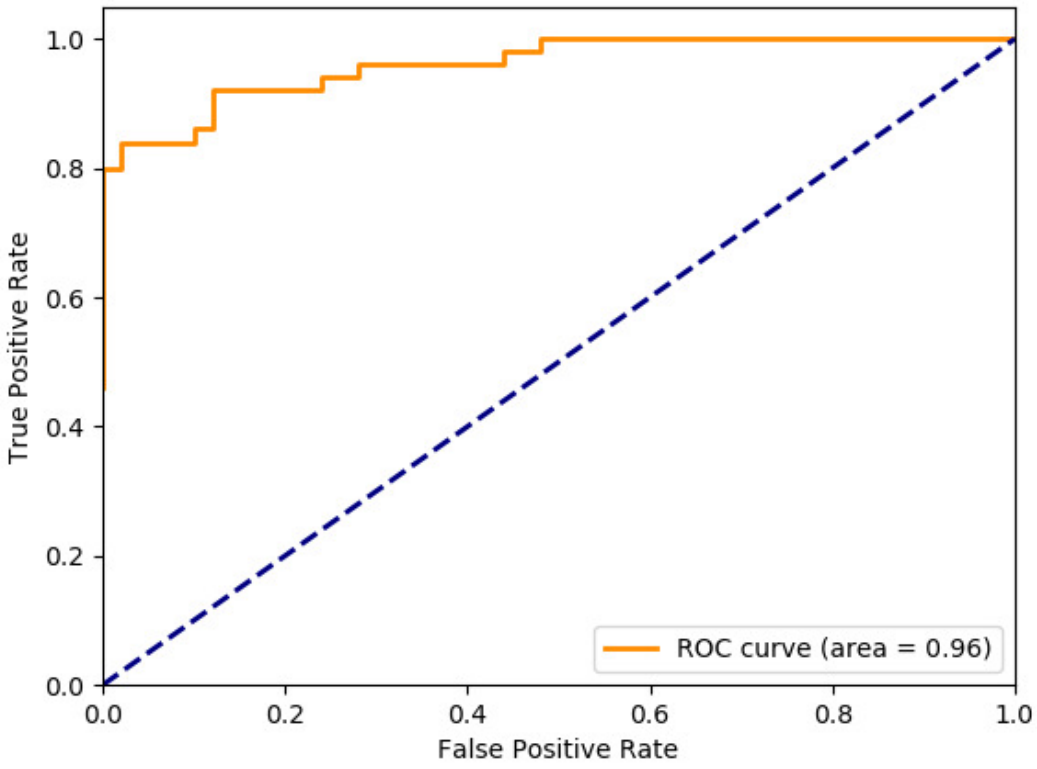
Fourth Model: ResNet50 + Dense Layer + Data Augmentation

We added a global spatial average pooling layer to the last convolutional output of ResNet50 Model trained on ImageNet followed by fully-connected layer with 'relu' activation and a fully connected layer, where the latter contains one node for each skin cancer category(melanomas, nevus, or SBK) and is equipped with a softmax.

Classification Report ResNet50 with Data Augmentation.

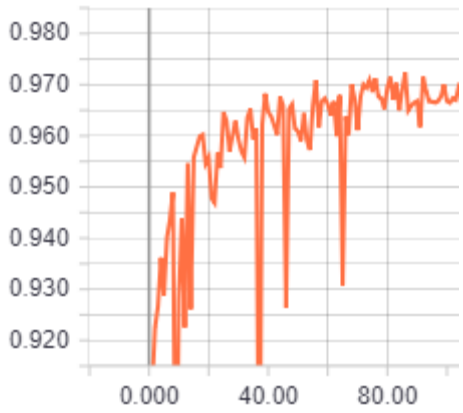
- 100 epochs. ModelCheckpoint. Best Val Accuracy
- AUC: 0.9612

Class	Precision	Recall	f1-score	support
0.0	0.88	0.88	0.88	50
1.0	0.88	0.88	0.88	50
avg / total	0.88	0.88	0.88	100

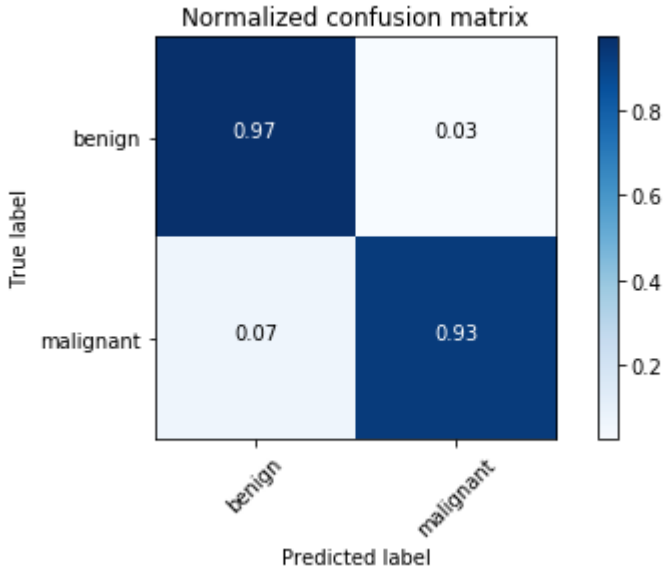


Robustness of the Best Model: Validation

Our data was split in training and validation subsets with an 80%/20% ratio.The training subset is used to train the model and the validation subset to check the result on data independent from the training set. It allows to know if the model overfits or not.We train our model during 100 epochs which lasted around 10 hours using Amazon EC2 instance. After 100 epochs, we reached a training accuracy of 99% and a validation accuracy of 96%.
Validation accuracy curve for 100 epochs is shown in below figure:-



On validation data, the model had an AUROC value of 0.9612 (which is good) and the confusion matrix displayed the following values :

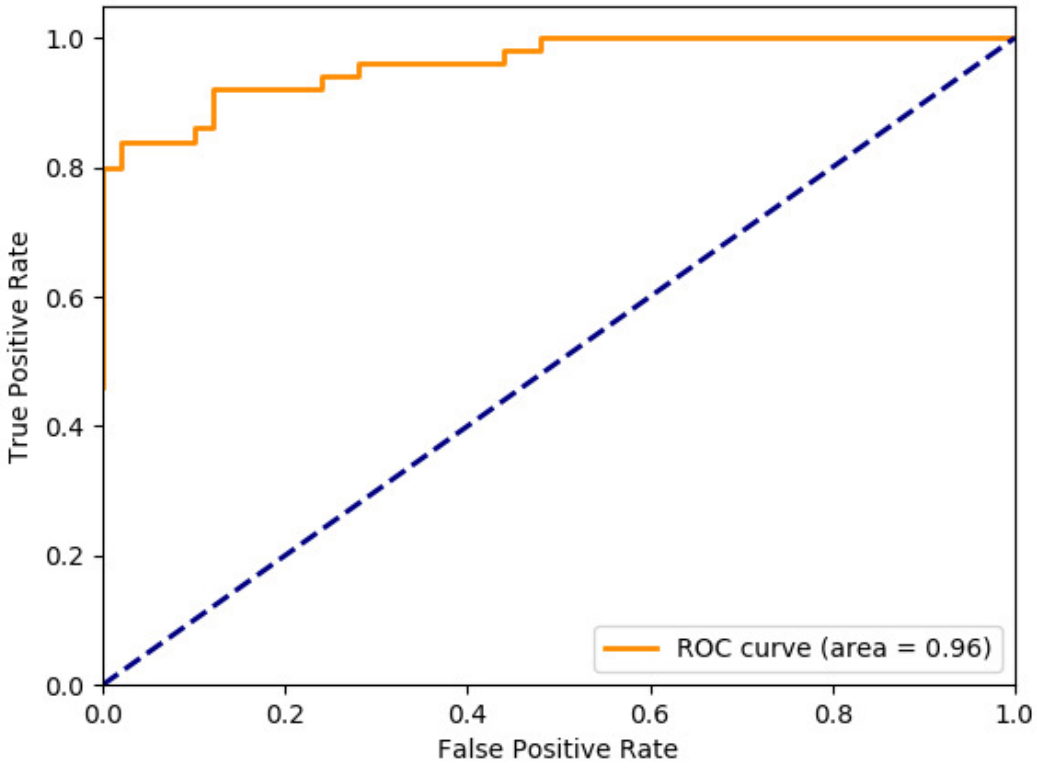


Confusion matrix after 100 epochs in percentage

As we can see, benign mole are better detected with only 3 % error, while malignant moles has 7 % error. This is logical. Indeed, the model learns better with more data and there are more pictures of benign mole.

Conclusion

The model had an ROC-AUC value of 0.9612 (which is good) .



Our model can be validated from the top scores from the ISIC competition and can be considered as second best model in the competition.

In this category ROC AUC scores can be found in the image below.

Rank	User	Title	Organization	Documentation	Date	Score
1	monty python	gpm-LSSSD	Multimedia Processing Group - Universidad Carlos III de Madrid		Wed, 1 Mar 2017, 12:57:35 pm	0.965
2	Kazuhisa Matsunaga	ResNet ensemble with normalized image	Casio and Shinshu University joint team		Wed, 1 Mar 2017, 11:18:03 pm	0.953
3	RECOD Titans	release (rc36xtrm) "alea jacta est"	RECOD Titans / UNICAMP		Wed, 1 Mar 2017, 11:42:07 pm	0.943
4	Xulei Yang	multi-task deep learning model for skin lesion segmentation and classification-3	Institute of High Performance Computing + National Skin Center, Singapore		Tue, 28 Feb 2017, 6:34:10 pm	0.942
5	T D	Last Minute Submission!!!!	University of Guelph - MLRG		Wed, 1 Mar 2017, 11:55:50 pm	0.935
6	Lei BI	EResNet (single scale w/o attributes)	USYD-BMIT		Wed, 1 Mar 2017, 8:04:42 pm	0.921
7	C V	all	icuff		Tue, 28 Feb 2017, 1:06:44 am	0.911
8	Cristina Vasconcelos	comb	icuff		Tue, 28 Feb 2017, 1:11:21 am	0.911
9	Masih Mahbod	Skin Lesion Classification Using Hybrid Deep Neural Networks	IPA		Wed, 1 Mar 2017, 12:51:43 pm	0.908
10	Dylan Shen	task3_final_RQ	Computer Vision Institute, Shenzhen University		Wed, 1 Mar 2017, 9:20:22 pm	0.886
11	Euijoon Ahn	DeepAhn	USYD-BMIT		Wed, 1 Mar 2017, 10:30:13 am	0.885
12	INESC TECNALIA	Final	INESC TEC Porto / TECNALIA		Wed, 1 Mar 2017, 7:05:40 pm	0.881
13	Vic Lee	task3_final_Alice	Computer Vision Institute, Shenzhen University		Wed, 1 Mar 2017, 9:11:31 pm	0.875
14	Balázs Harangi	Ensemble of deep convolutional neural networks	University of Debrecen		Wed, 1 Mar 2017, 8:25:16 pm	0.867
15	x j	finalv_L2C1_trir	CVI		Wed, 1 Mar 2017, 11:17:56 am	0.855
16	Rafael Sousa	Araguaia Medical Vision Lab - GooglAlexNet	Universidade Federal de Mato Grosso		Wed, 1 Mar 2017, 3:26:22 pm	0.840
17	Matt Berseth	Final Classification Submission	NLPLOGIX / WISEEYE.AI		Tue, 28 Feb 2017, 6:32:47 am	0.827
18	Dennis Murphree	Transfer Learning from Inception	Dennis Murphree		Wed, 1 Mar 2017, 11:06:33 pm	0.817
19	Wenhao Zhang	testPhase	CSMedical		Wed, 1 Mar 2017, 7:08:07 pm	0.817
20	Hao Chang	MYBrainAI	Yale		Wed, 1 Mar 2017, 11:53:55 pm	0.774
21	Jaisakthi S.M.	Lesion Classification	SSNMLRG		Wed, 1 Mar 2017, 9:25:02 pm	0.687
22	Wiselin JiJi	Dr JIJI P2 Test	Dr Sivanthi Aditanar College of Engineering		Thu, 2 Mar 2017, 12:46:52 am	0.498
23	Yanzhi Song	submit of yanzhi	song		Wed, 1 Mar 2017, 8:05:13 am	0.456

Reflection

First, I build a benchmark architecture from scratch and then improved my results using transfer learning technique on Resnet-50 model. It is computationally very expensive to train big models, I created an amazon EC2 instance with GPUs which can run several processes parallelly. The final model performs better than my benchmark model. As you can see from the classification report, the submission currently predicts that most of the images in the test dataset correspond to benign lesions.

My reason for choosing ResNet50 architecture is that ResNet50 model deals with vanishing gradients problem better than other models and also reduces overfitting problem along with number of parameters needed for computation. But in this experiment, it resulted in good AUC score but I believe the score can further be improved by adding batch normalization and shuffling the train set after data augmentation and also adding more images to reduce overall imbalance and overfitting.

The most difficult part for me was to get the experiments running on AWS EC2 Instance with GPU mode enabled. Although computational time was reduced with GPU option, there is cost involved which resulted in lower number of experiments being done.

Improvement

We can further improve the accuracy by tuning certain hyperparameters and choosing different optimizations techniques.

Due to time and computational cost it was not possible for me to run more experiments using different known architectures other than Resnet50 for this dataset. It's definitely possible that a different architecture would be more effective. Given enough time and computational power, I'd definitely like to explore the different approaches.

As the classes were heavily imbalanced, one of my hypotheses is if I generate further augmented dataset for the classes that have less data than the others, save them in the training set would help to improve model further.

References :

- https://codebox.net/pages/image-augmentation-with-python
- https://challenge.kitware.com/#challenge/583f126bcad3a51cc66c8d9a
- http://challenge2017.isic-archive.com/
- https://www.udacity.com (Dermatologist AI)
- Dermatologist-level classification of skin cancer with deep neural networks by Andre Esteva, Brett Kuprel, Roberto A. Novoa, Justin Ko, Susan M. Swetter, Helen M. Blau & Sebastian Thrun
- SKIN LESION ANALYSIS TOWARD MELANOMA DETECTION: A CHALLENGE AT THE 2017 INTERNATIONAL SYMPOSIUM ON BIOMEDICAL IMAGING (ISBI), HOSTED BY THE INTERNATIONAL SKIN IMAGING COLLABORATION (ISIC)
- Coursera DeepLearning.ai course