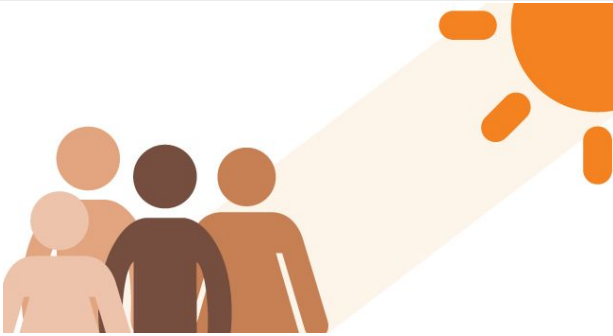
THINKFUL

Detecting Skin Cancer with Deep Learning

Thinkful Data Science Flex Program
Final Capstone

Author: Shela Usadi
(shelausadi@gmail.com)
Code: https://bit.ly/2Gxa93e

Background



The Problem

- Skin cancer is the most prevalent type of cancer worldwide
- Early detection improves chance of survival
- It is difficult to detect skin cancer by the naked eye
- Diagnosis by medical experts currently only have a 77.0344% accuracy

Proposed Solution

- Utilize current images of diagnosed skin lesions and use machine learning to detect skin cancer more accurately than unaided visual inspection by physicians
- Goal: build something that has an accuracy of close to 77.0344% or better

ISIC 2019

Data Source

- Our data source comes from the International Skin Imaging Collaboration (ISIC) website in the form of 2 CSV files and 25,331 images.
- CSV files:
  - Metadata
  - Groundtruth
- ISIC's mission: to reduce melanoma related deaths and unnecessary biopsies
- Planning to enter my solution into the ISIC 2019 competition (Deadline: August 9th)

```
[ ]  metadata.head(5)
```

	image	age_approx	anatom_site_general	lesion_id	sex
0	ISIC_0000000	55.0	anterior torso	NaN	female
1	ISIC_0000001	30.0	anterior torso	NaN	female
2	ISIC_0000002	60.0	upper extremity	NaN	female
3	ISIC_0000003	30.0	upper extremity	NaN	male
4	ISIC_0000004	80.0	posterior torso	NaN	male

```
[157] groundtruth.head(5)
```

	image	MEL	NV	BCC	AK	BKL	DF	VASC	SCC	UNK
0	ISIC_0000000	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1	ISIC_0000001	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	ISIC_0000002	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	ISIC_0000003	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	ISIC_0000004	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Cleaning the data

```
[ ]  metadata.isnull().sum()
```

```
image                    0
age_approx             437
anatom_site_general   2631
lesion_id             2084
sex                    384
dtype: int64
```

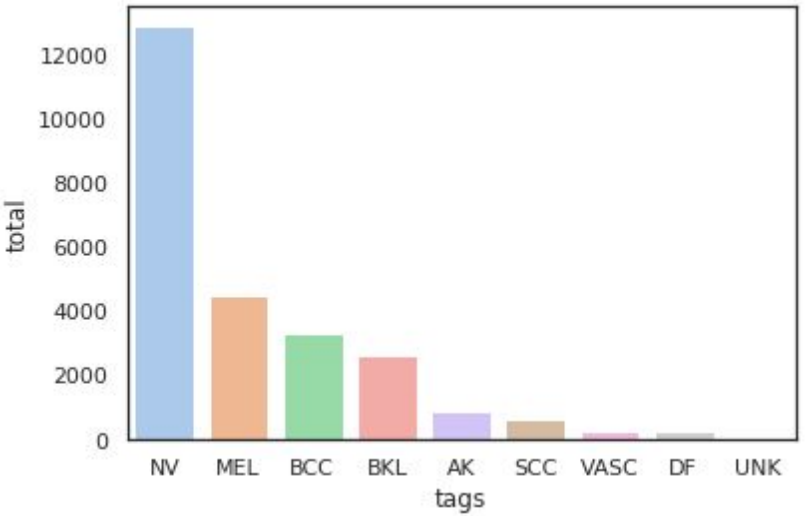
Remove rows with missing data from the dataset

```
data = pd.merge(metadata, groundtruth, on='image')
```

Merge the two CSV files so that we can explore the data more easily

Exploratory Data Analysis

What is the distribution of images by tags in the dataset?

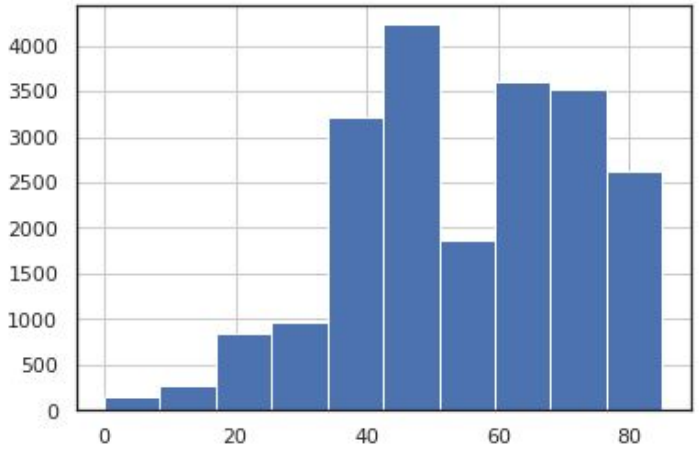


Legend
MEL - melanoma
NV - melanocytic nevus
BCC - basal cell carcinoma
AK - actinic keratosis
BKL - benign keratosis
DF - dermatofibroma
VASC - vascular lesion)
UNK - unknown

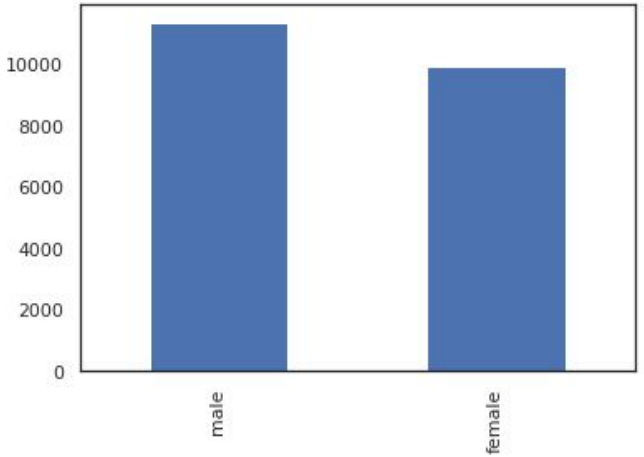
There's a disproportionate number of melanocytic nevus (NV) images

Exploratory Data Analysis

What is the distribution of the age of patients in this dataset?

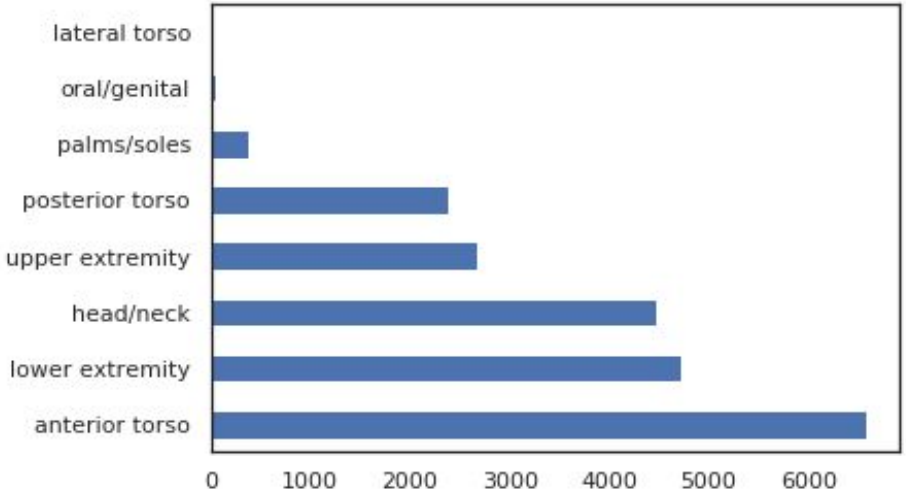


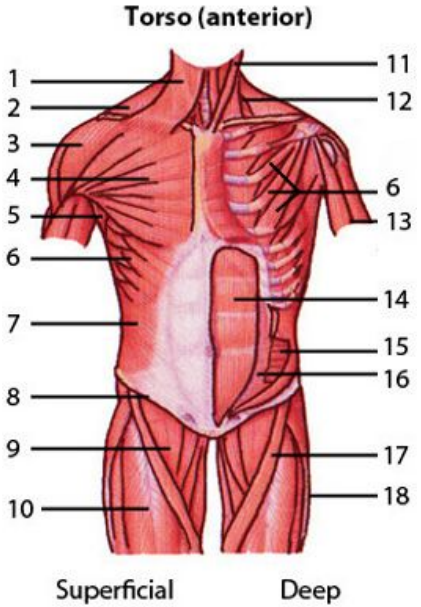
What is the distribution for sexes in this dataset?



Exploratory Data Analysis

Which anatomic areas are the most compromised to skin cancer in this dataset?





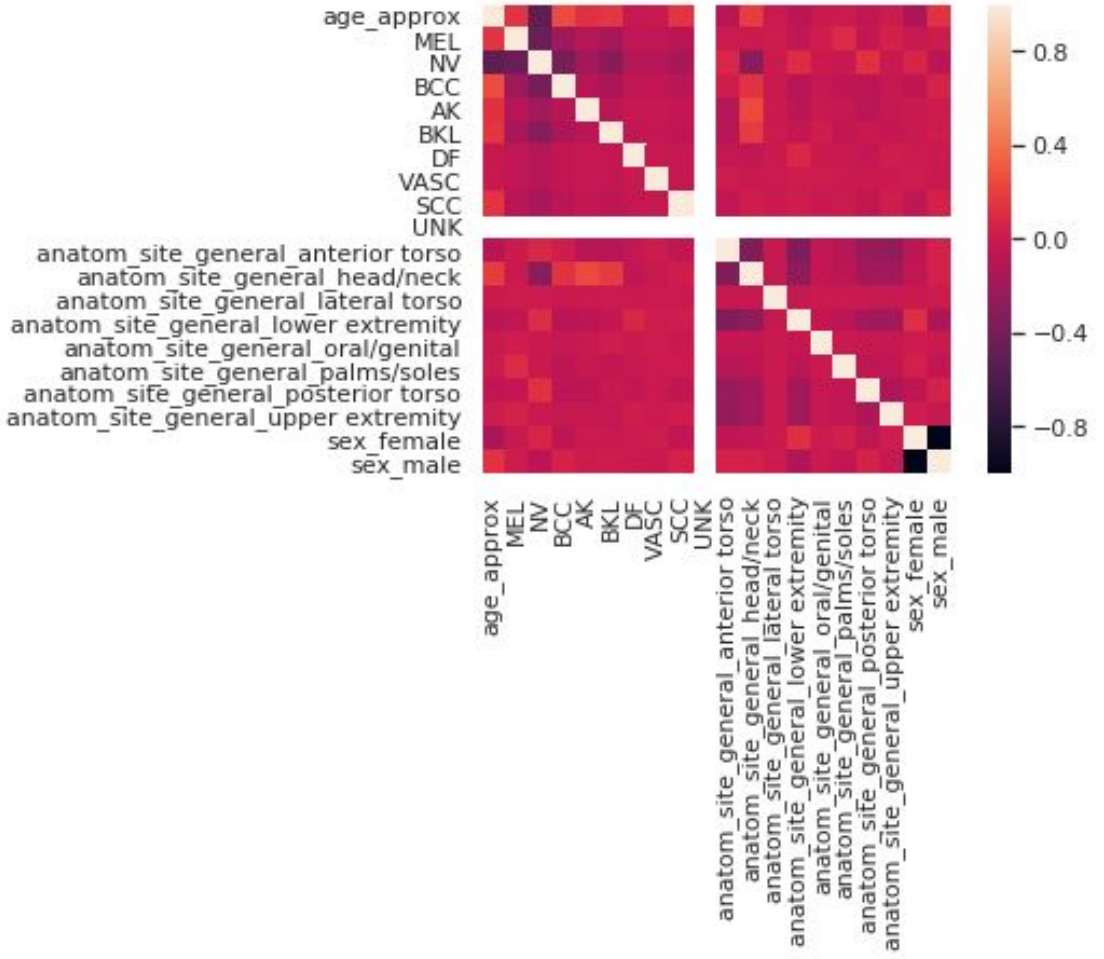
Correlation

```
one_hot_data.corr()['MEL'].sort_values(ascending=False).head(10)
```

```
MEL                                  1.000000
age_approx                           0.155374
anatom_site_general_palms/soles      0.105355
anatom_site_general_upper extremity  0.047432
anatom_site_general_oral/genital     0.017322
anatom_site_general_lateral torso    0.015550
sex_male                             0.012582
anatom_site_general_anterior torso   0.005012
anatom_site_general_head/neck       -0.004677
sex_female                          -0.012582
Name: MEL, dtype: float64
```

```
one_hot_data.corr()['sex_female'].sort_values(ascending=False).head(10)
```

```
sex_female                           1.000000
anatom_site_general_lower extremity  0.130057
NV                                   0.076396
anatom_site_general_palms/soles      0.039990
anatom_site_general_upper extremity  0.018617
VASC                                 0.010514
DF                                   0.005656
anatom_site_general_oral/genital     0.002601
AK                                  -0.003146
anatom_site_general_lateral torso   -0.006185
Name: sex_female, dtype: float64
```



Fastai deep learning library

```
[ ]  %reload_ext autoreload
     %autoreload 2
     from fastai.vision import *
```

- Released Oct 2nd, 2018
- Sits on top of PyTorch v1
- “Provides a single consistent API to the most important deep learning applications and data types”
- Going to use it to try resnet18, resnet35, and resnet50
    - Pre-trained by Imagenet
    - Faster and can use less data than if we start from scratch

Train/ Test Split

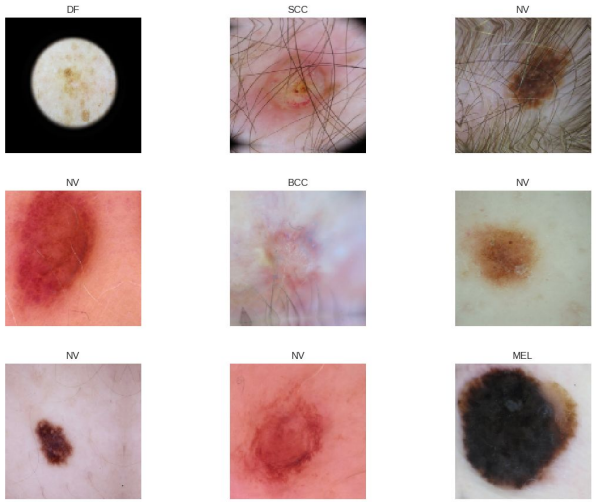
```
#create list of random indices
from numpy.random import RandomState
random_indices = np.random.randint(low=0, high=25331, size=5000)
```

```
#this are to be used in the split by index
train_indices = list(random_indices[0:math.floor(len(random_indices)*0.7)])
test_indices  = list(random_indices[math.floor(len(random_indices)*0.7):])
```

We are going to undersample the training set due to limitations by Colaboratory.

Get ImageList

```
np.random.seed(42)
src = (ImageList.from_csv(path,
                          '/gdrive/My Drive/ISIC_2019_data/truelabels.csv',
                          folder='ISIC_2019_Training_Input', suffix='.jpg')
       .split_by_idxs(train_idx=train_indices, valid_idx=test_indices)
       .label_from_df())
```



Finding Optimal Learning Rate

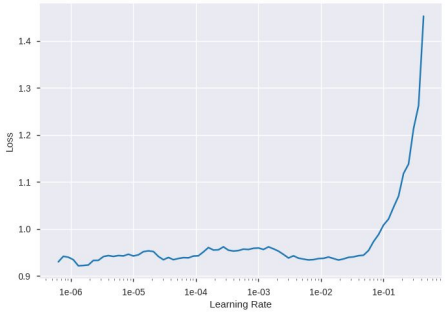
Each time we build a model, we find and plot the learning rate

Learning rate - the size of the step that we use to update the weights via gradient descent. During each iteration, we multiply learning rate by the gradient. By finding an optimal learning rate, we make the model train faster and generalize better.

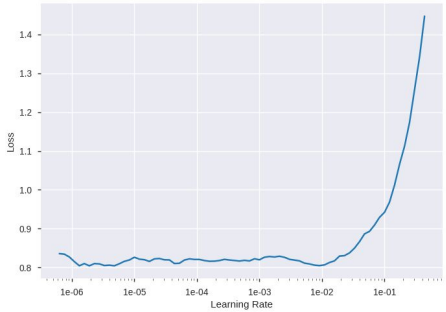
The learning rate plot allows us to find the learning rate that makes the loss go down the fastest.

Re-train the model with the optimal learning rate

```
[ ]  learn_resnet18.unfreeze()
     learn_resnet18.fit_one_cycle(10, max_lr=slice(1e-6,1e-2))
```



Learning rate plot for Resnet-34

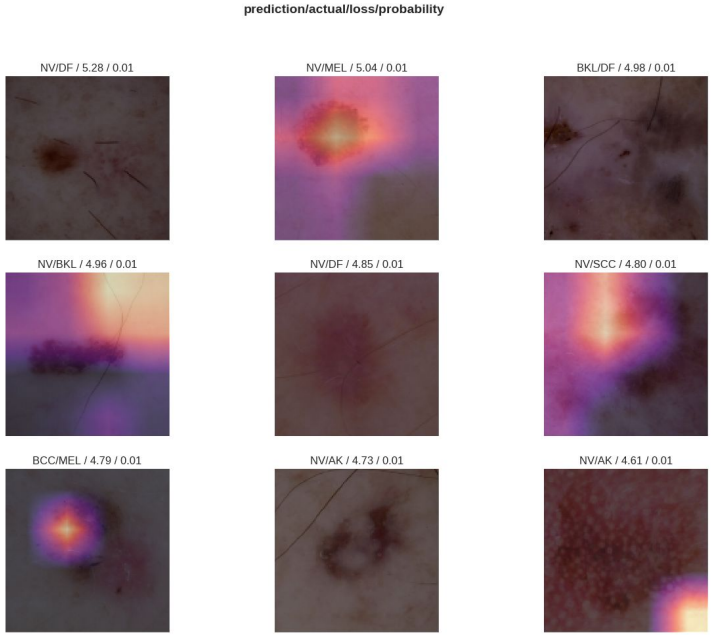


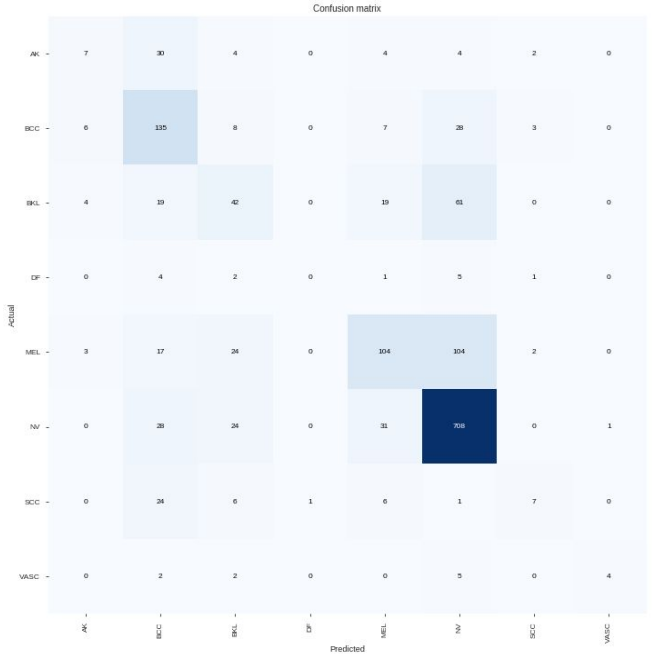
Learning rate plot for Resnet-18

Results

Model	Resnet - 18	Resnet - 34	Resnet - 50
Accuracy	73.00%	75.87%	75.47%
Top 10 most confused	('MEL', 'NV', 113), ('BKL', 'NV', 51), ('NV', 'MEL', 48), ('BCC', 'NV', 37), ('BKL', 'MEL', 33), ('AK', 'BCC', 28), ('NV', 'BCC', 26), ('MEL', 'BCC', 24), ('BKL', 'BCC', 23), ('BCC', 'MEL', 20)	('MEL', 'NV', 104), ('BKL', 'NV', 61), ('NV', 'MEL', 31), ('AK', 'BCC', 30), ('BCC', 'NV', 28), ('NV', 'BCC', 28), ('MEL', 'BKL', 24), ('NV', 'BKL', 24), ('SCC', 'BCC', 24), ('BKL', 'BCC', 19)	('MEL', 'NV', 91), ('BKL', 'NV', 51), ('NV', 'MEL', 32), ('BCC', 'NV', 28), ('BKL', 'MEL', 22), ('AK', 'BCC', 21), ('NV', 'BCC', 21), ('MEL', 'BKL', 20), ('SCC', 'BCC', 19), ('NV', 'BKL', 16)

Resnet-34 Most Confused





9 cases where the network performed the worst

Conclusion

Model	Resnet - 18	Resnet - 34	Resnet - 50
Accuracy	73.00%	75.87%	75.47%

- We've come very close to beating medical experts' diagnosis accuracy with only 5000 out of 25,331 data points
- If we use a larger sample, I think we can get an accuracy that's higher than 77.03%
- Weak Points:
  - Misdiagnoses can still happen
  - Still 1% lower than human medical expert's diagnosis accuracy

Next Steps

Short-term steps:

- Use Google Cloud Platform (Storage, Datalab) to run this with all 25k+ pictures in the dataset
- Experiment with Locality Sensitive Hashing
- Write the paper and submit it for the ISIC 2019 competition!

Long term steps:

- Build an application & web app that identify skin lesions by using phone camera and an affordable microscope attachment
- Be able to diagnose skin cancer easier, faster, cheaper and with more accuracy.

Next Project

AIRMED FOUNDATION

Airmed Foundation

The Airmed Foundation is an **open source initiative** working under the AGPL-3.0 License. We provide a **secure channel to store and transfer medical records**. We do this through the **Interplanetary File System (IPFS)** and **Hyperledger Fabric**. The conjunction of these technologies guarantees the immutability and availability of the data. We replicate all records through the IPFS network, using the **Bittorrent** protocol. We achieve secure access to files through **asymmetric cryptography**. We protect and store access keys in the Hyperledger Fabric **blockchain**.