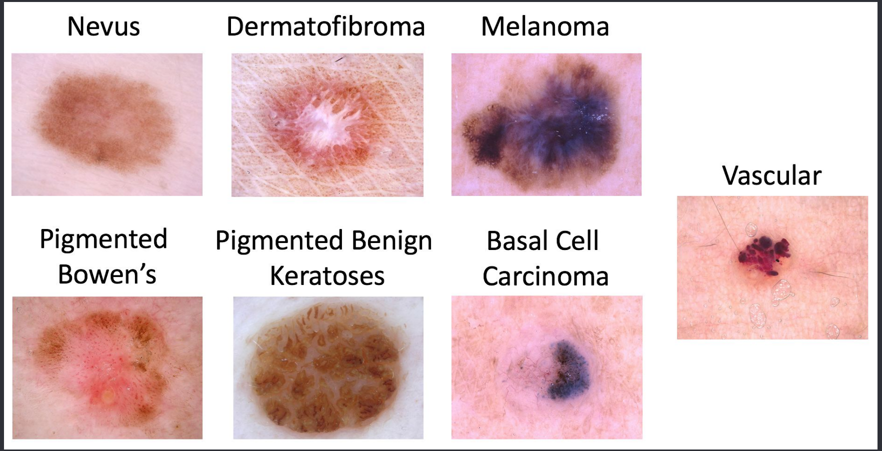
Skin Cancer Detection Project

By: Nikhil, Arjun, Mana, Hope, and Ethan

Problem Overview

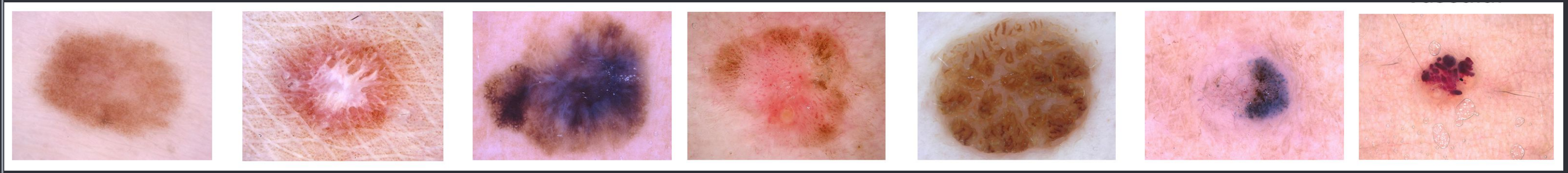
Introduction

- Skin cancer is the most common type of cancer in the world.
  - 9,500 cases are reported daily in the United States
  - 2-3 million cases are reported every year globally.
- Many cases go undetected, due to not having quality healthcare and equipment, or due to human error.



- Skin cancer is usually diagnosed visually, with screenings.
- Using machine learning would help use more accurately identify skin cancer, which would save millions of lives.

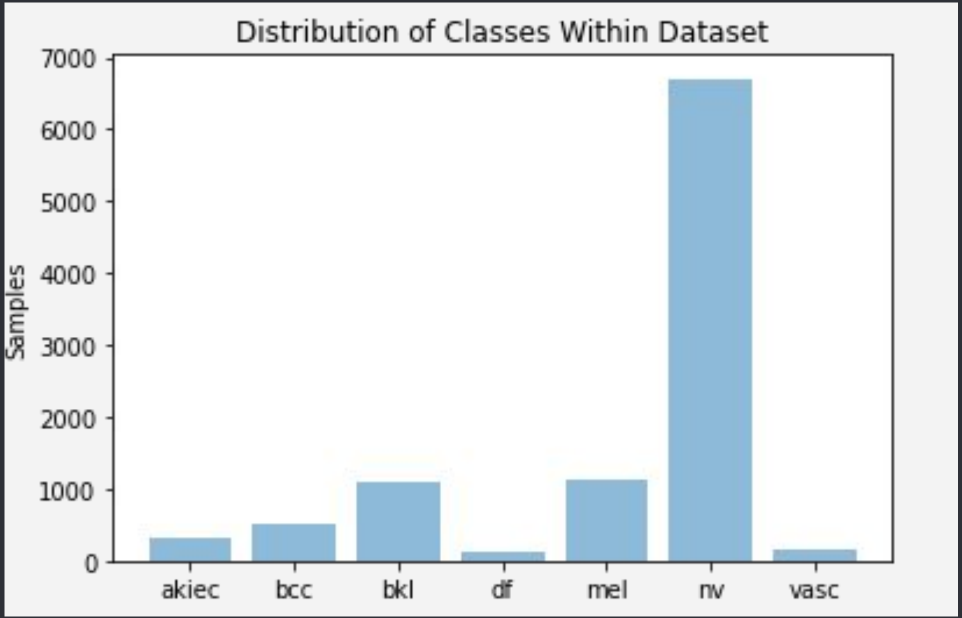
What are some features you notice in each skin cancer classes that could help identify them?



Processing Data

Exploring the Data

- 10,015 images
- 7 Classes



Does anybody see any problems with this dataset?

Preprocessing Data

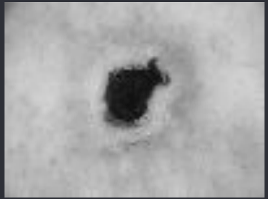
- Reducing data
  - Makes training data more equal across classes (142)
  - But also reduces amount of data
- Augmentations
  - technique used to increase the size of our data by using techniques like blurring, resizing, flipping, rotating, and color changing.
  - Prevents overfitting
  - Increases amount of DIVERSE data

Resize

Zoom

Blur

Flip

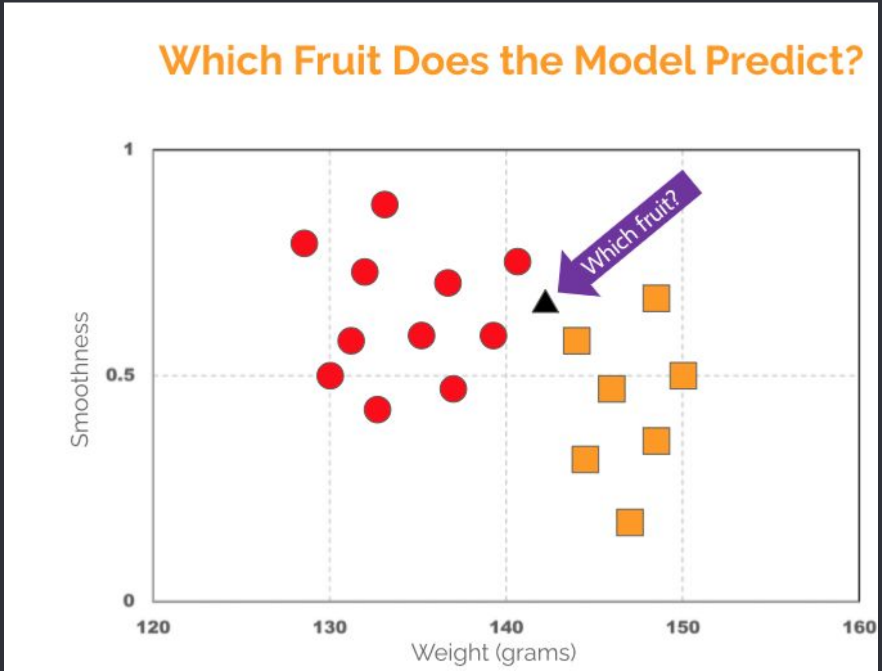


Greyscale

Model Design- KNN, CNN, Transfer Learning

K-Nearest-Neighbors:

knn = KNeighborsClassifier(n_neighbors = 3, “weight”, “algorithm”)



KNN Performance

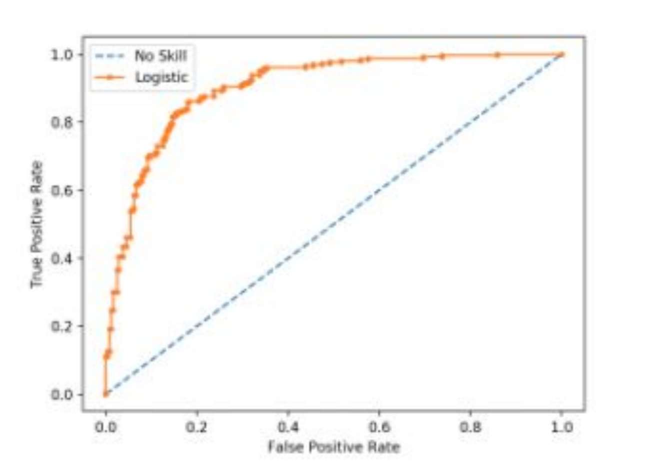
Optimizing Our Model:

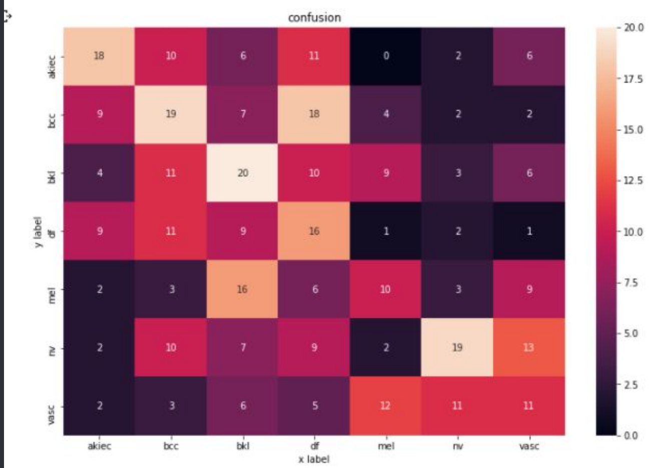
Receiver Operator Curve: relationship between true positive and false positive.

Area Under Curve: Metric demonstrates how close ROC is to random guessing. Higher AUC → better model

----

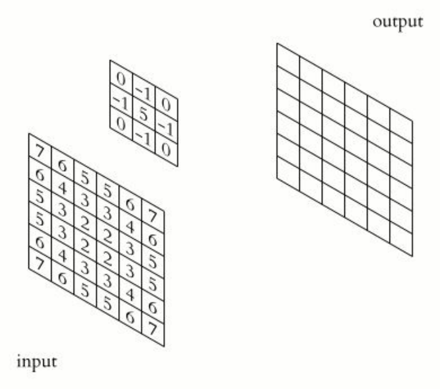
- Used for loop and AUC score to optimize parameters
- Final Accuracy: 34%





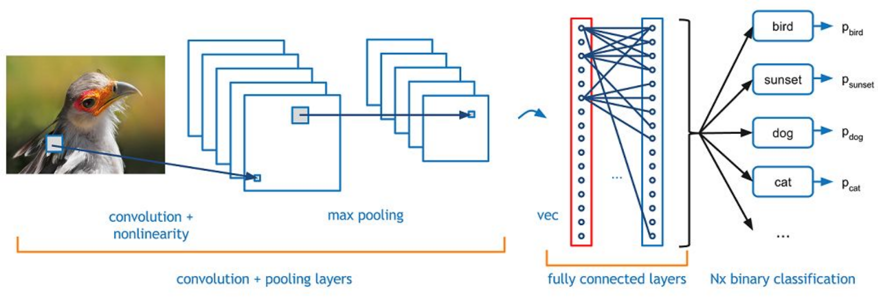
Convoluted Neural Networks (CNN)/Transfer Learning

- Convolutions - a mathematical operation that lets us find a pattern in a portion of an image.
- Apply kernel to image to achieve feature extraction
  - Kernel - pattern we’re looking more represented by numbers
- Transfer Learning - CNN w/ a pre-trained network



0	-1	0
-1	5	-1
0	-1	0

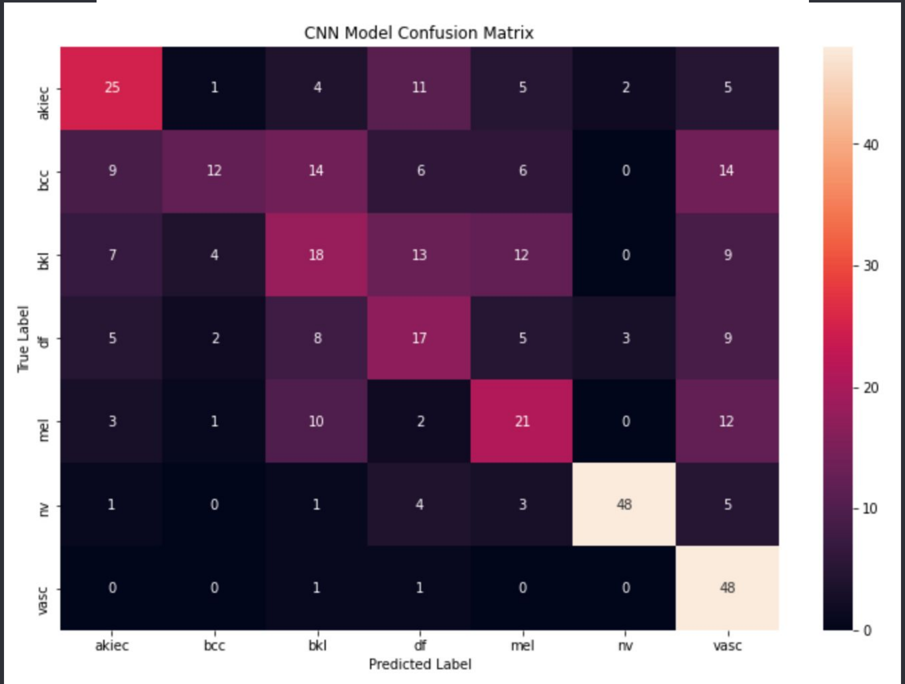
7	6	5	5	6	7
6	4	3	3	4	6
5	3	2	2	3	5
5	3	2	2	3	5
6	4	3	3	4	6
7	6	5	5	6	7



CNN Performance

- ◦ Dropout, layer size, # neurons, epochs optimized

```
The ROC AUC Score of the model is 0.83497
```

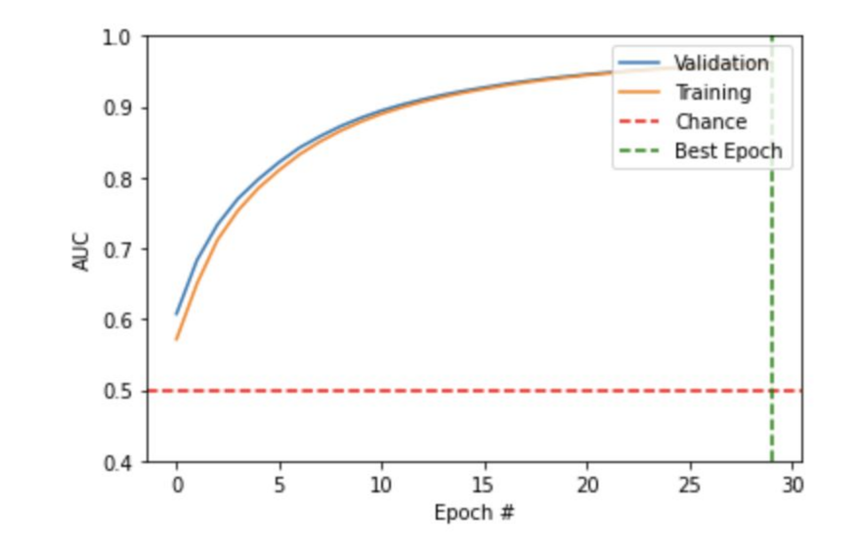


CNN Model Confusion Matrix

True Label \ Predicted Label	akiec	bcc	bkl	df	mel	nv	vasc
akiec	25	1	4	11	5	2	5
bcc	9	12	14	6	6	0	14
bkl	7	4	18	13	12	0	9
df	5	2	8	17	5	3	9
mel	3	1	10	2	21	0	12
nv	1	0	1	4	3	48	5
vasc	0	0	1	1	0	0	48

Transfer Learning CNN Performance

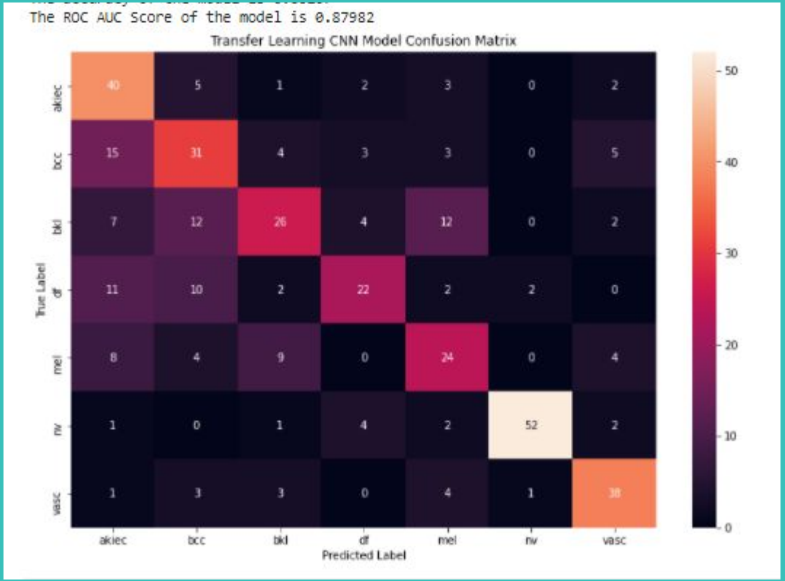
Transfer Learning- Pre Trained CNN



Optimizing Parameters

- Transfer Learning
- Hyper Parameters:
    - Image Size
    - Color
    - Data Augmentation
- ROC Score Increase

Image Size Trial CM



```
The ROC AUC Score of the model is 0.87982
```

Transfer Learning CNN Model Confusion Matrix

True Label \ Predicted Label	akiec	bcc	bkl	df	mel	nv	vasc
akiec	40	5	1	2	3	0	2
bcc	15	31	4	3	3	0	5
bkl	7	12	26	4	12	0	2
df	11	10	2	22	2	2	0
mel	8	4	9	0	24	0	4
nv	1	0	1	4	2	52	2
vasc	1	3	3	0	4	1	38

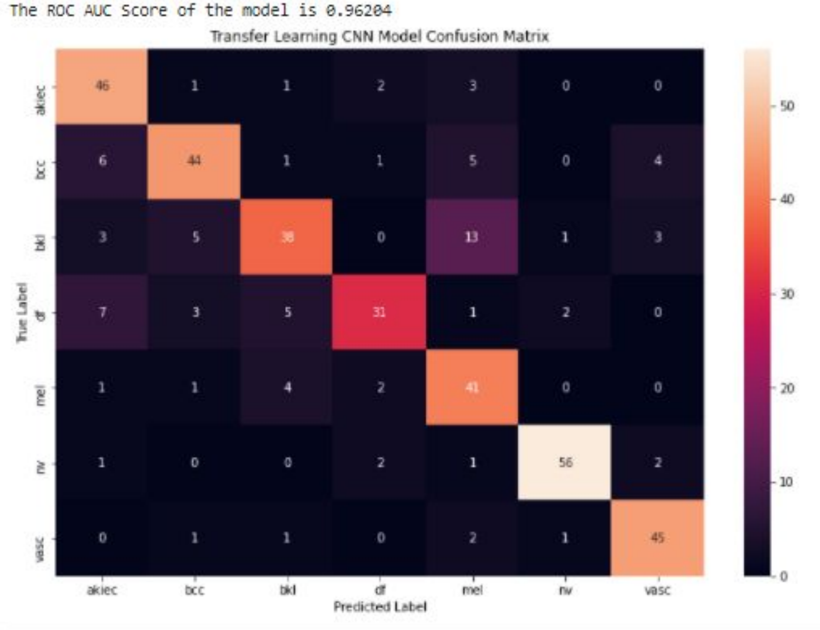
IMG Size = 50, 50 AUC Score: 0.87982

Key Takeaways

CNN + Larger Image Size

Balanced Data

Applicable Model



```
The ROC AUC Score of the model is 0.96204
```

Transfer Learning CNN Model Confusion Matrix

True Label \ Predicted Label	akiec	bcc	bkl	df	mel	nv	vasc
akiec	46	1	1	2	3	0	0
bcc	6	44	1	1	5	0	4
bkl	3	5	38	0	13	1	3
df	7	3	5	31	1	2	0
mel	1	1	4	2	41	0	0
nv	1	0	0	2	1	56	2
vasc	0	1	1	0	2	1	45

Final Model CM:
IMG Size = 224, 224, AUC Score: 0.87982

Thank You For Listening!