

Andrew Neuman

# Skin Lesion Identification



# What is the project about?

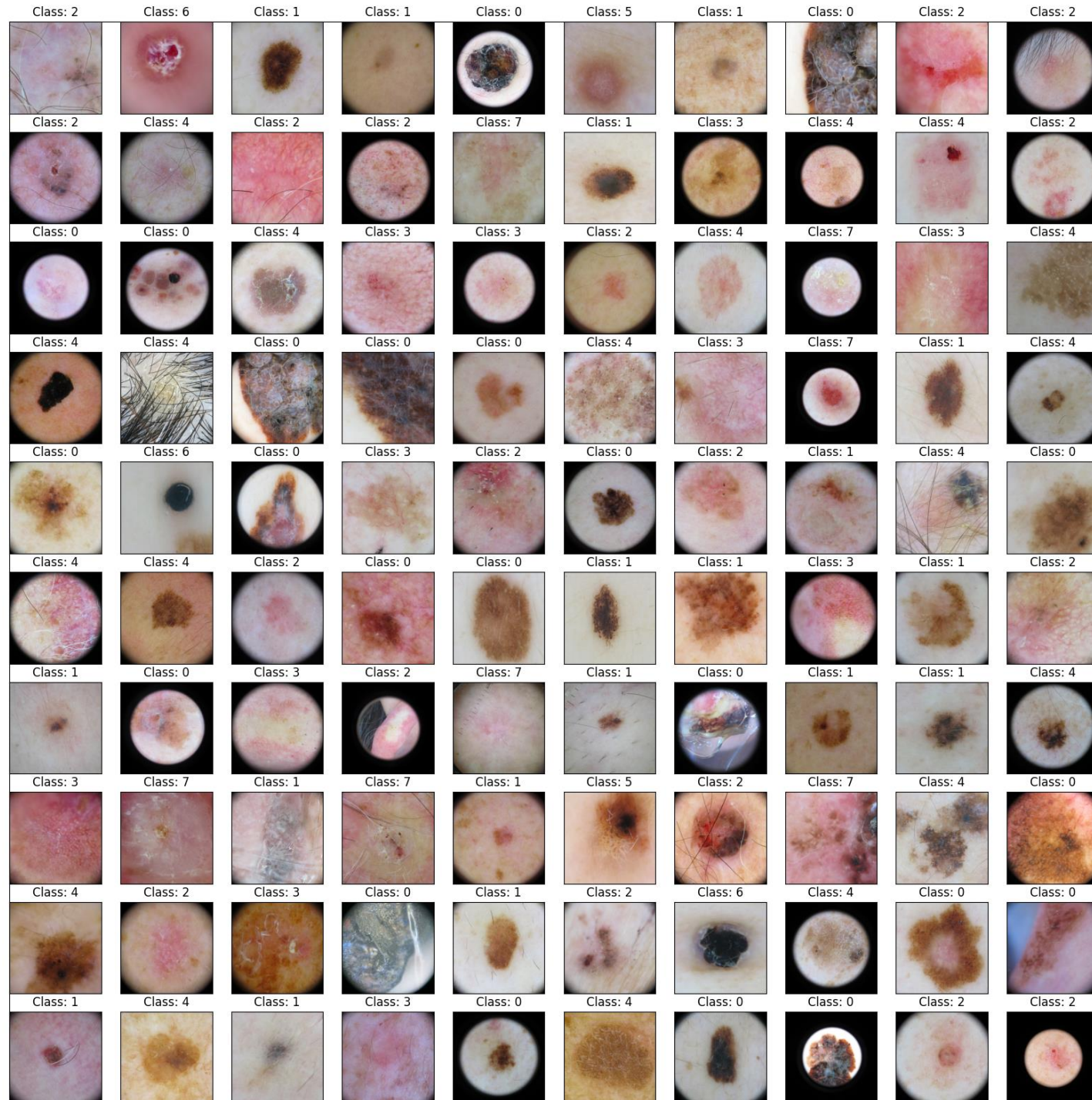
- The goal of the project is to accurately classify images of different skin lesions.
- Some of the lesions are cancerous while others are not.

# Why is it interesting?

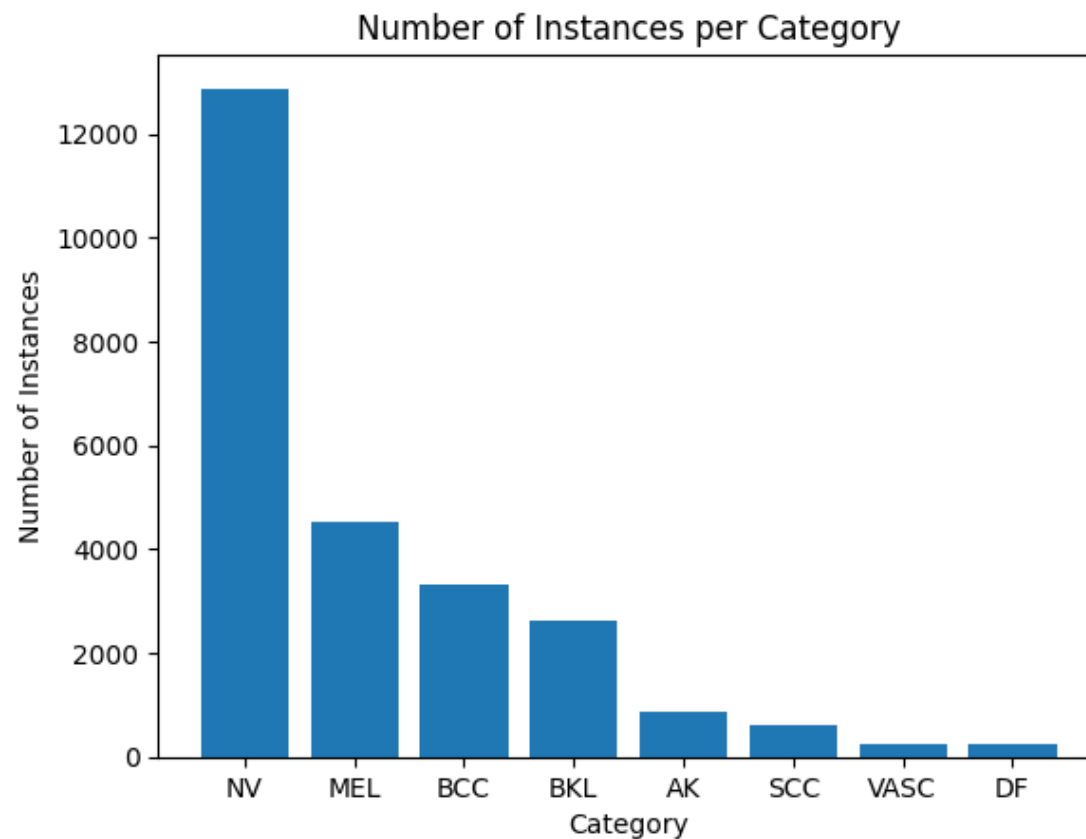
- Wait times to see a dermatologists can be over a year.
- Hope to develop something that might be able to be used as a prescreening tool.
- Ease nerves of people while they wait to see a specialist.



# Dataset



# Class Imbalance



Abbreviation	Full
NV	Melanocytic nevus
MEL	Melanoma
BCC	Basal cell carcinoma
BKL	Benign keratosis
AK	Actinic keratosis
SCC	Squamous cell carcinoma
VASC	Vascular lesion
DF	Dermatofibroma

# Dealing with Imbalanced Data

- Image augmentation is used to balance the dataset.
- Augmentations were selected that a Doctor could expect to see in practice.
- For instance, rotated images, flipped images, or slightly blurred images.

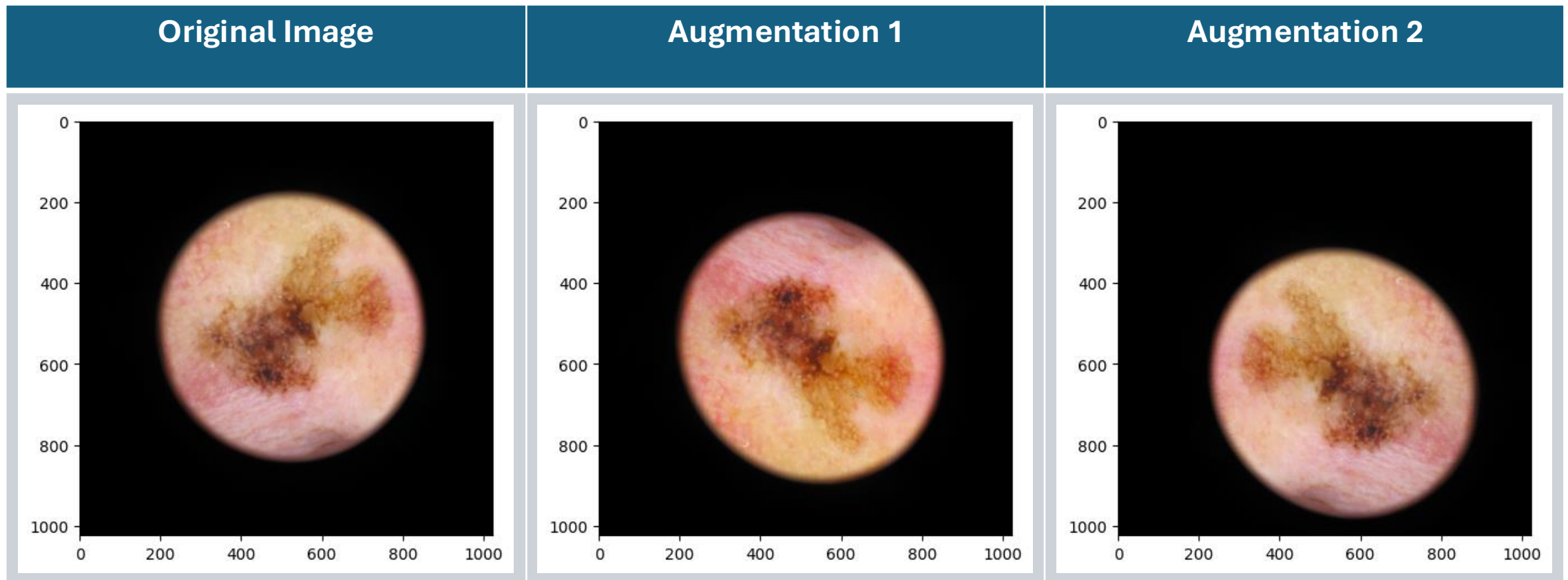


# Sample of Augmentation Code

```
import torch
import torchvision
from torchvision import datasets, transforms
from torchvision.transforms import v2

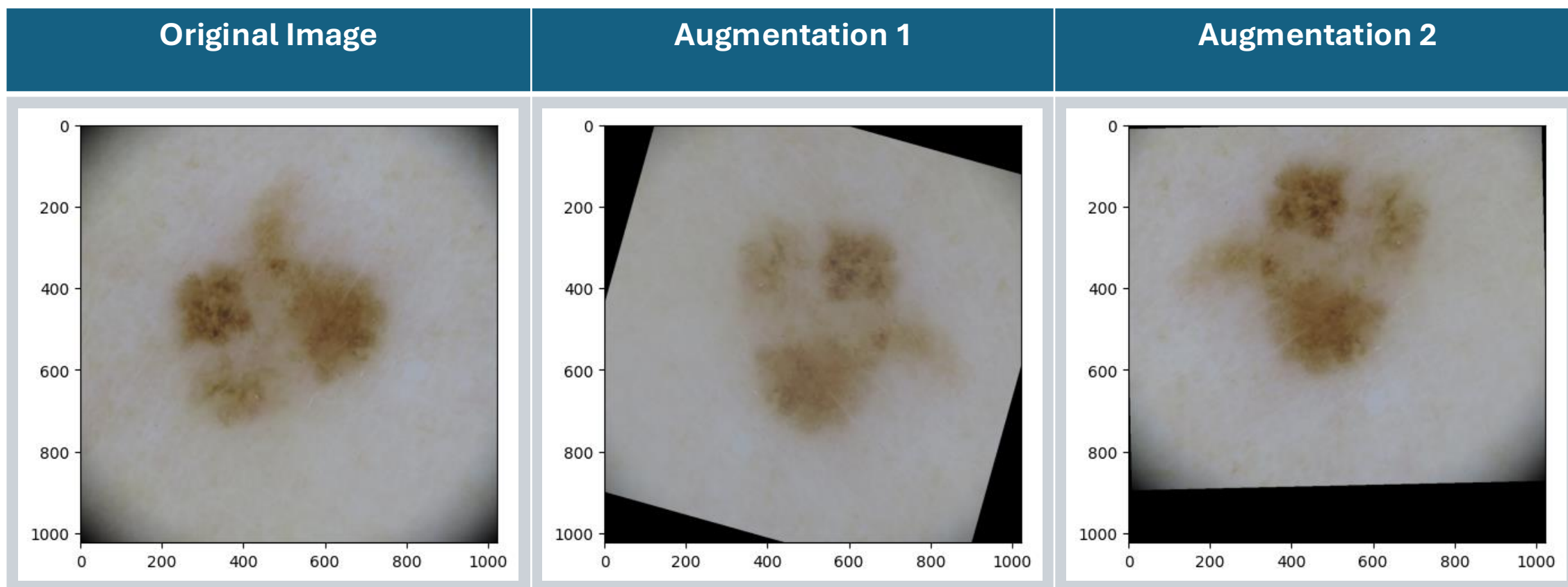
# Info about augmentation https://pytorch.org/vision/main/transforms.html
transforms = v2.Compose([
    v2.RandomHorizontalFlip(0.5),
    v2.RandomVerticalFlip(0.5),
    v2.RandomApply(transforms=[v2.RandomAdjustSharpness(sharpness_factor=2)], p=0.5),
    v2.RandomApply(transforms=[v2.RandAugment()], p=0.5),
    v2.RandomApply(transforms=[v2.RandomRotation(degrees=(0, 360))], p=0.6),
    v2.RandomApply(transforms=[v2.GaussianBlur(kernel_size=(5, 9), sigma=(0.1, 5.))], p=0.5),
])
```

# Original vs Augmented Image

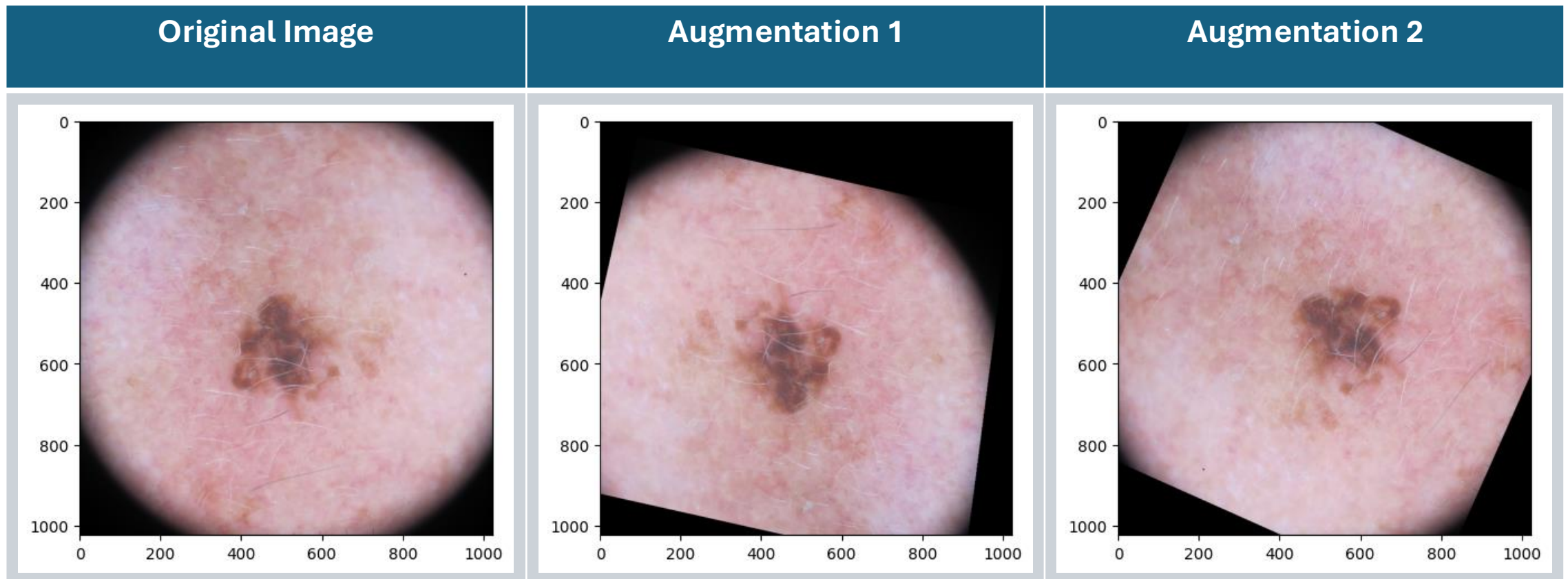




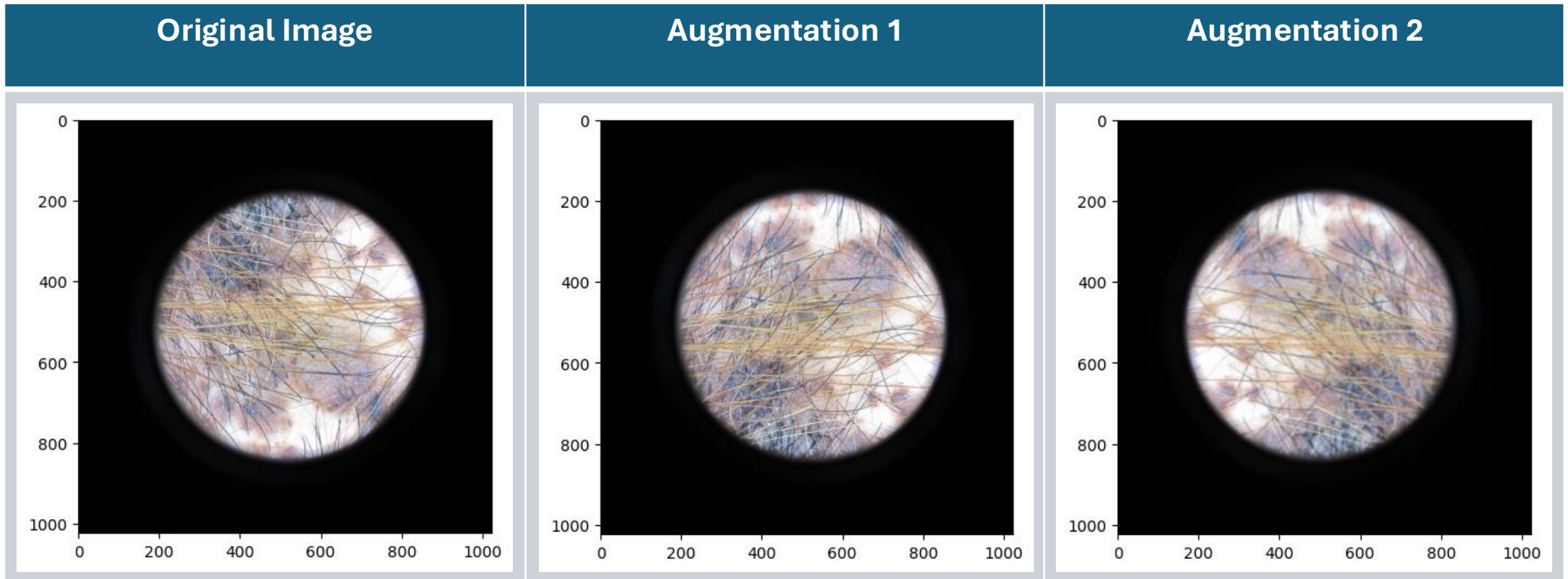
# Original vs Augmented Image



# Original vs Augmented Image



# Original vs Augmented Image

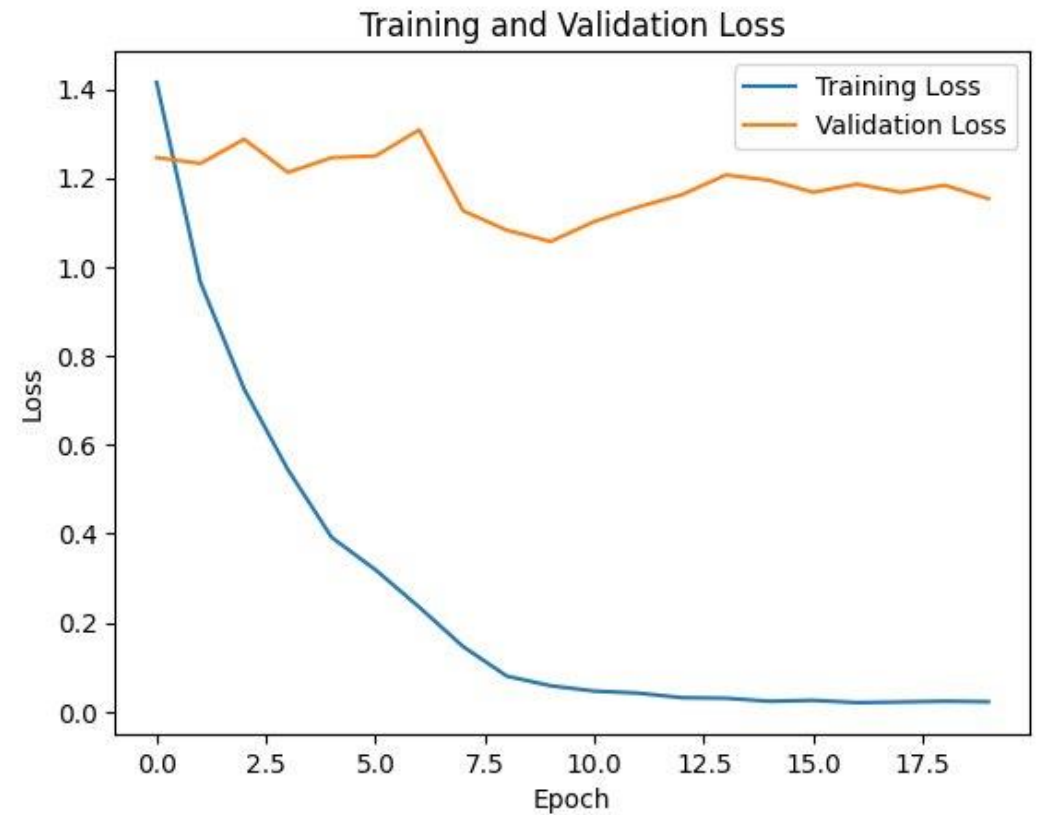
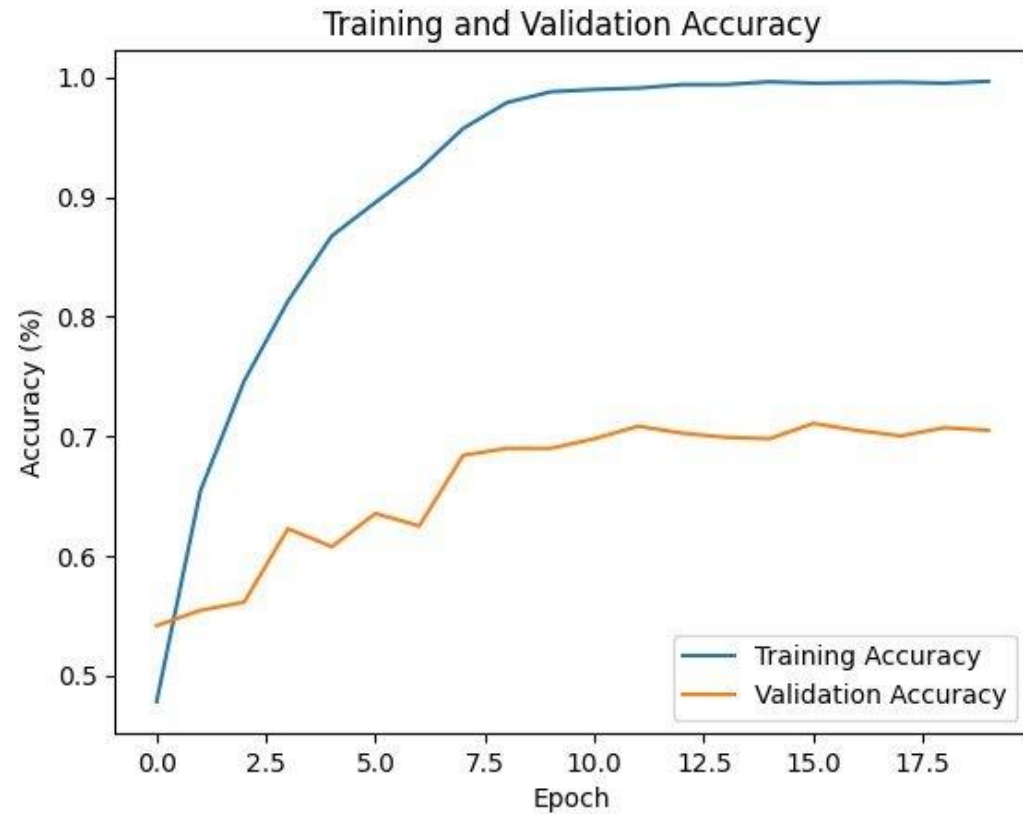


# Problem Formulation

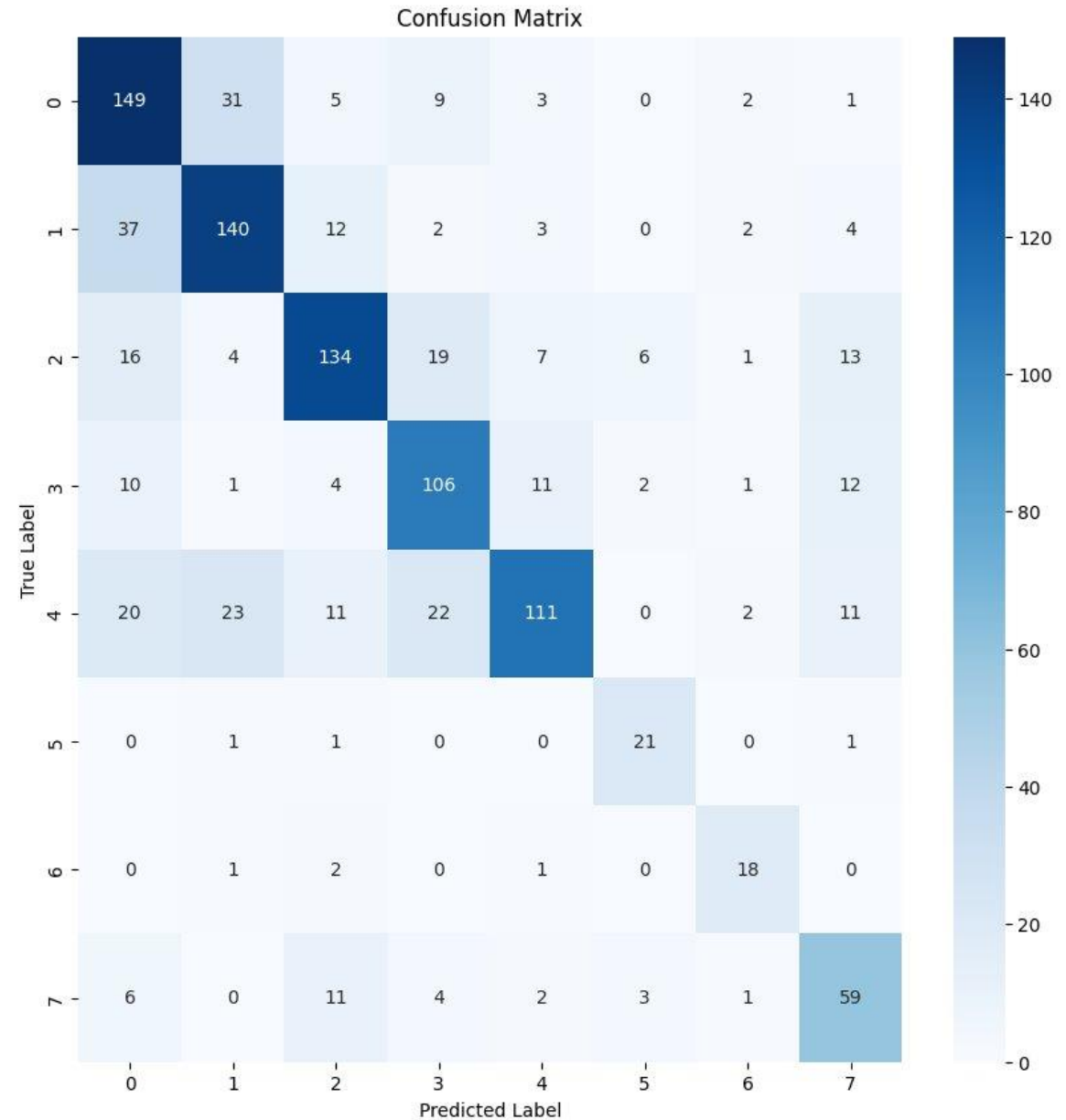
- Training set: Perfectly balanced using image augmentation.
- Validation/Testing set: Still imbalanced. Want these to represent the true sample.
- Various image sizes cut down to images of only size 1024x1024.
- Model: InceptionV3, fully trained
- Fine-tuned hyperparameters:
  - Epochs: 20
  - Learning Rate: 0.0001
  - Batch Size: 16
  - Weight Decay: 0.001
  - Learning rate decay scheduler: 0.1 every 7 epochs
  - Optimizer: Adam



# Results Training and Validation



# Results Testing – Confusion Matrix



# Results Testing – Classification Report

	precision	recall	f1-score	support
0	0.63	0.74	0.68	200
1	0.70	0.70	0.70	200
2	0.74	0.67	0.71	200
3	0.65	0.72	0.69	147
4	0.80	0.56	0.66	200
5	0.66	0.88	0.75	24
6	0.67	0.82	0.73	22
7	0.58	0.69	0.63	86
accuracy			0.68	1079
macro avg	0.68	0.72	0.69	1079
weighted avg	0.70	0.68	0.68	1079

# Conclusion

- These results are not as good as I had hoped for.
- Great learning experience.
- Very humbling.
- Still lots to learn.
- Not done working on it yet.
- Will continue to work on it until it is in a deployable state.