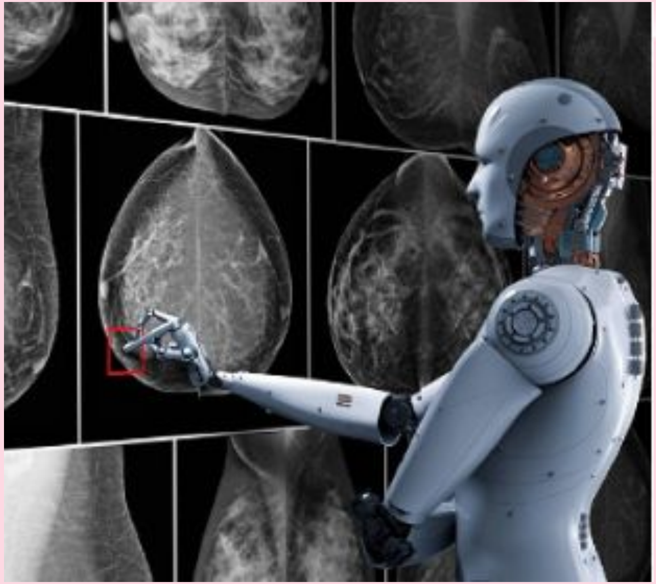
Detecting Breast Cancer with Machine Learning

By: Bryan Lange

robot



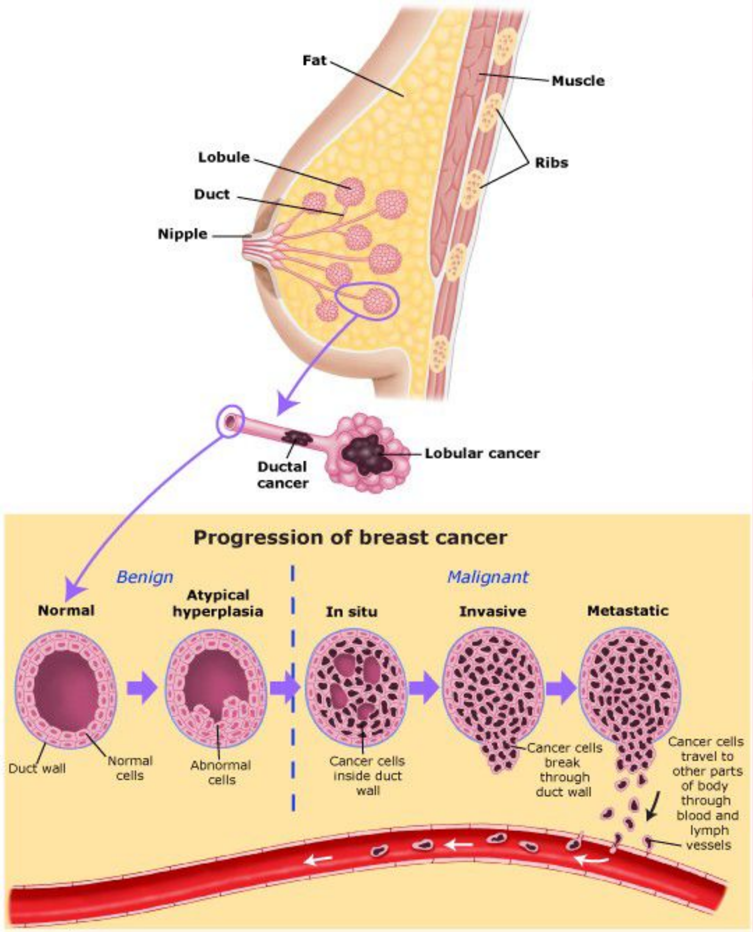
Presentation Overview

- Background information
- Data exploration
- Image augmentation
- Modeling process
- Transfer learning
- Model results
- Challenges & limitations
- Final recommendations

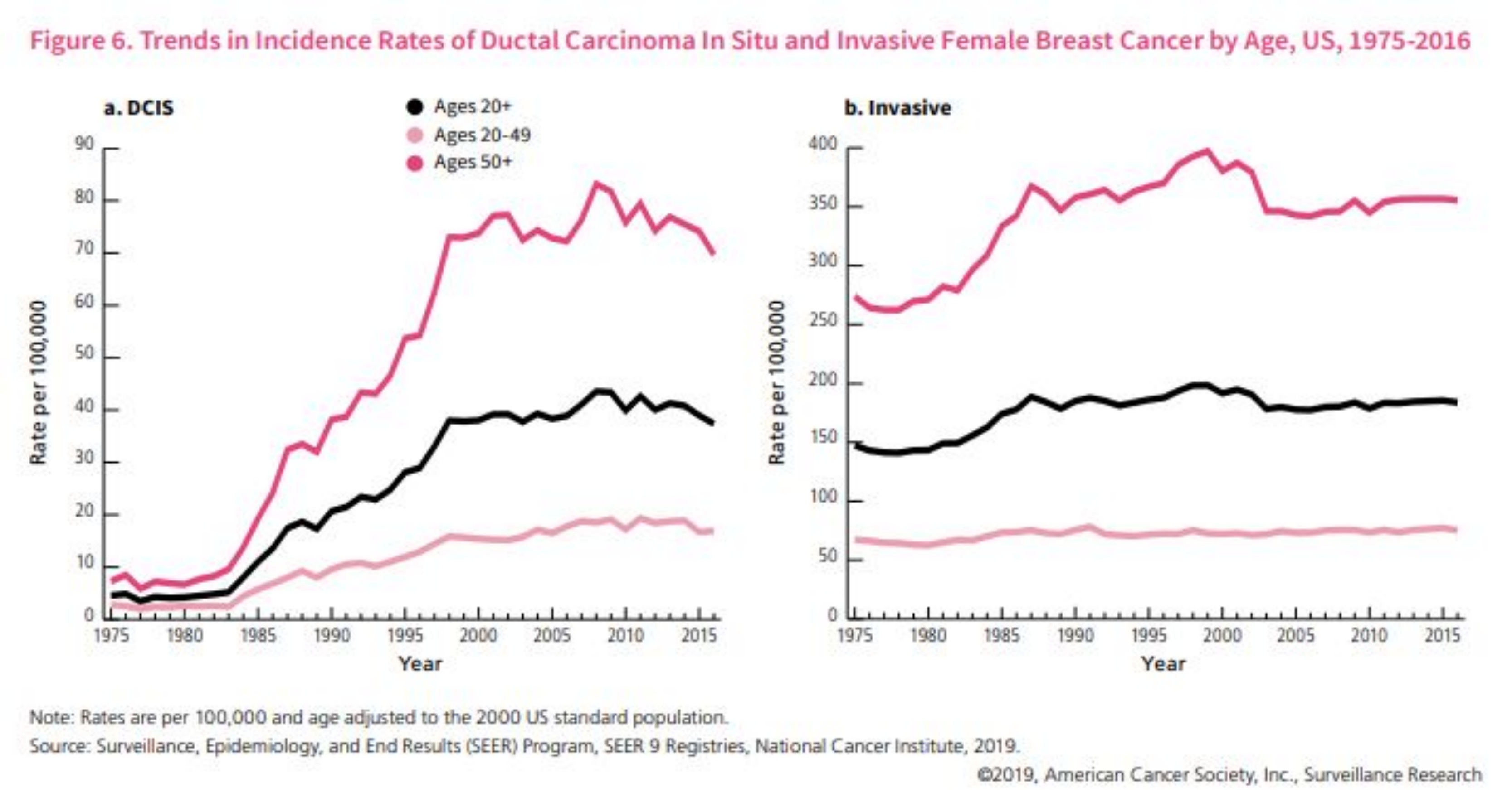
United States Breast Cancer Stats

- About 1 in 8 U.S. women will develop breast cancer over the course of their lifetime
- The American Cancer Society estimates over 275,000 new cases in 2020 with over 40,000 deaths attributed to breast cancer
- Second highest cancer death rate in American females (behind lung cancer)
- Over 3.5 million women with a history of breast cancer in the U.S.
- Breast cancer can be non-invasive, which means it does not spread beyond the lobule or duct, or it can be invasive where it spreads beyond the duct into the normal tissue
- 80% of malignant breast cancers are **Invasive Ductal Carcinoma** (IDC) which can invade surrounding tissue, lymph nodes and possibly other areas of the body

progression



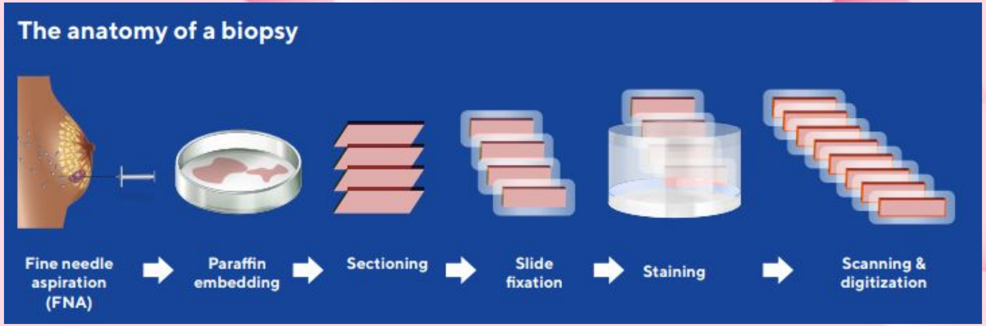
Invasive Ductal Carcinoma Over the Years



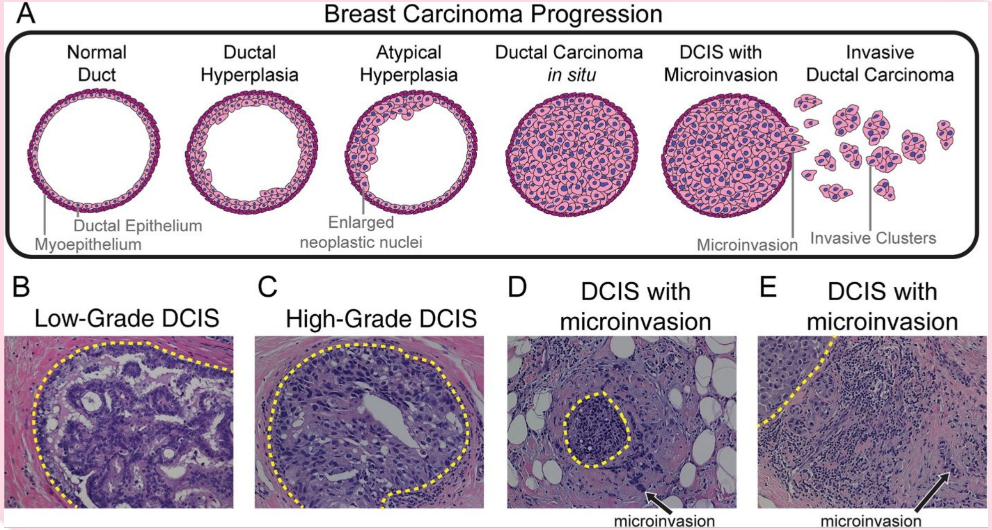
Biopsy and Histopathology

- ❖ A doctor removes a small amount of tissue (specimen) to be examined by a pathologist
- ❖ In the laboratory the specimen is cut into thin slices called histologic sections
- ❖ Next, the sections are stained with various dyes which help to distinguish parts of the cell
- ❖ The sections are then placed on a glass slide and are ready to be viewed under a microscope / scanner
- ❖ Here is where the pathologist can make a diagnosis and determine the grade, or how abnormal the tumor cells appear

biopsy



link



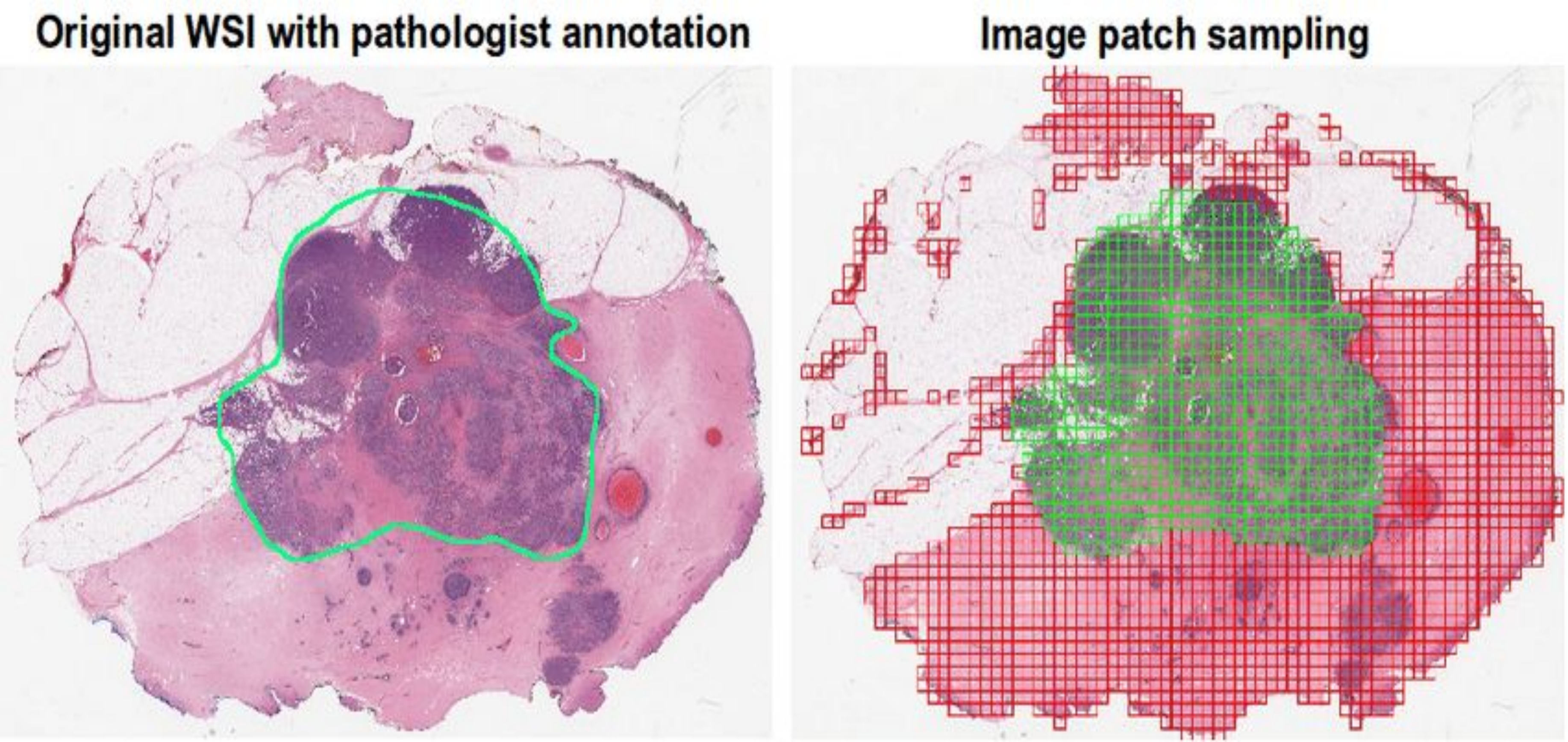
Problem Statement

- Invasive breast cancer detection can be a time consuming task
- A pathologist needs to scan large, mostly benign areas to ultimately hone in on a malignant region, which can be very tedious
- Errors can be made in the process and smaller malignant areas can be overlooked by the human eye
- Advancements in technology should be able to assist by automating part of the process and double checking the diagnosis given by the pathologist

Using histopathological images, can we train a neural network to accurately detect the presence of invasive ductal carcinoma?

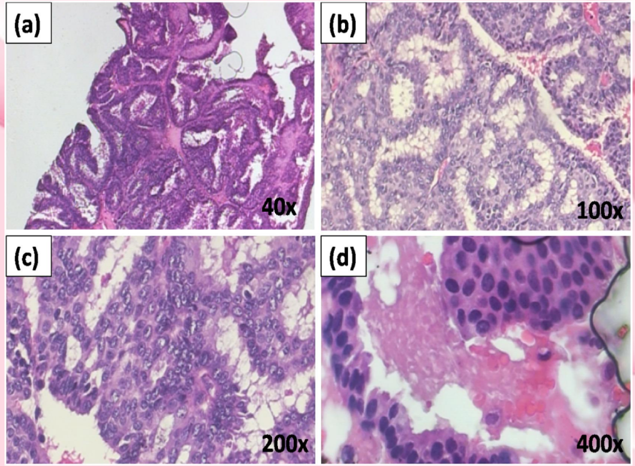
IDC Image Data

link



IDC Image Data

- The original dataset link consisted of 162 whole mount slides of breast cancer specimens
- The images were scanned at 40x
- Those images were then broken into 277,526 patches patches link
- Each patch is then reshaped to be 50 x 50 pixels
- From the patches:
  - 198,740 negative samples (0)
  - 78,786 positive samples (1)
- Nearly 72% of samples are negative, meaning we have an imbalance of classes



link

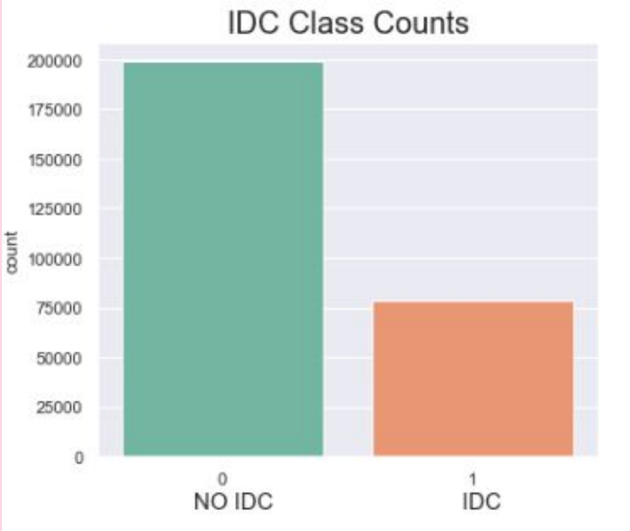
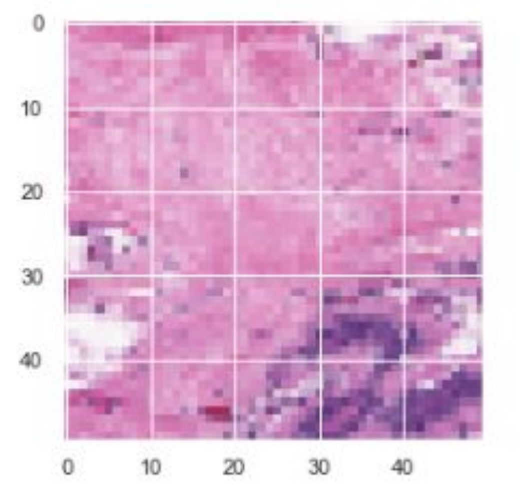
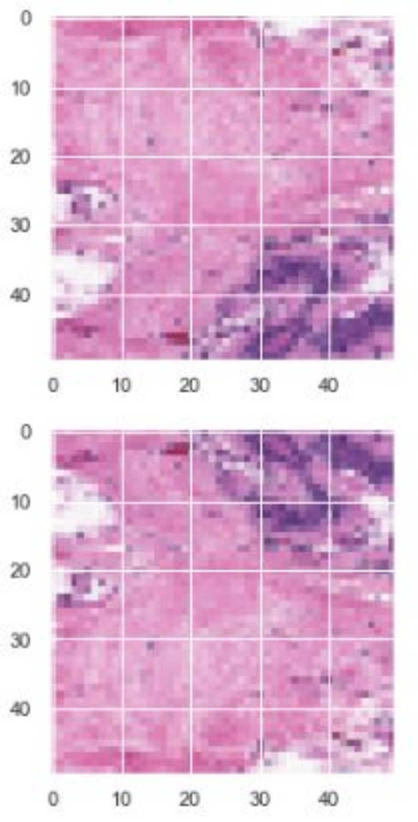


Image Augmentation

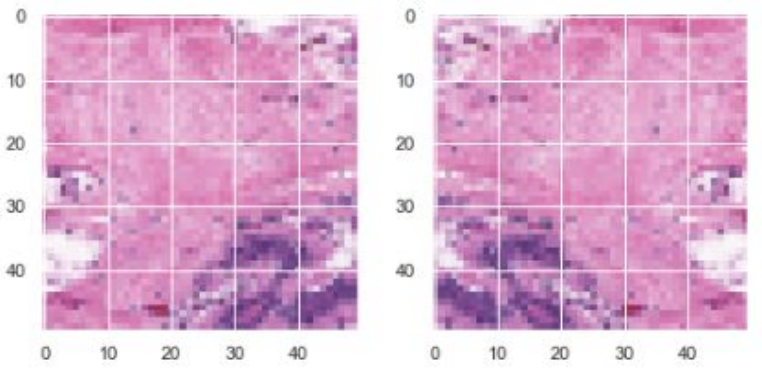


Original image

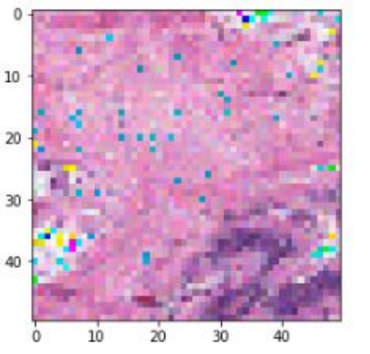
- The more data we train our model on, the better. Images can be augmented in a variety of ways to help increase diversity of data without actually collecting new data.



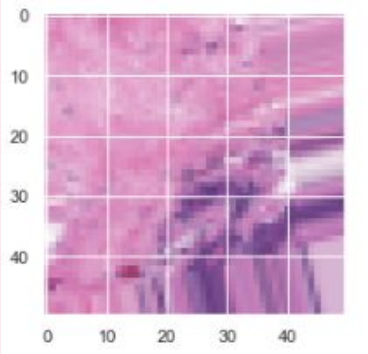
Vertical Flip



Horizontal Flip

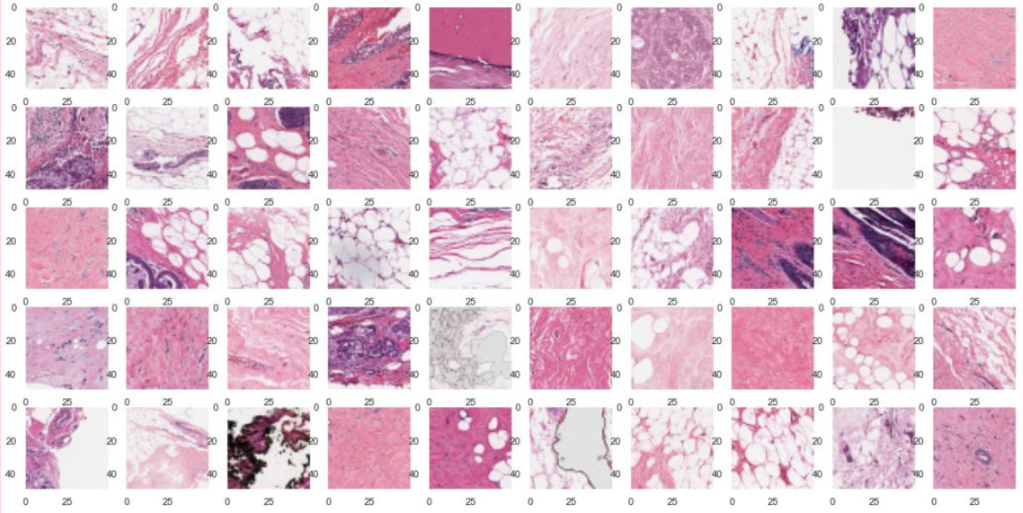


Noise

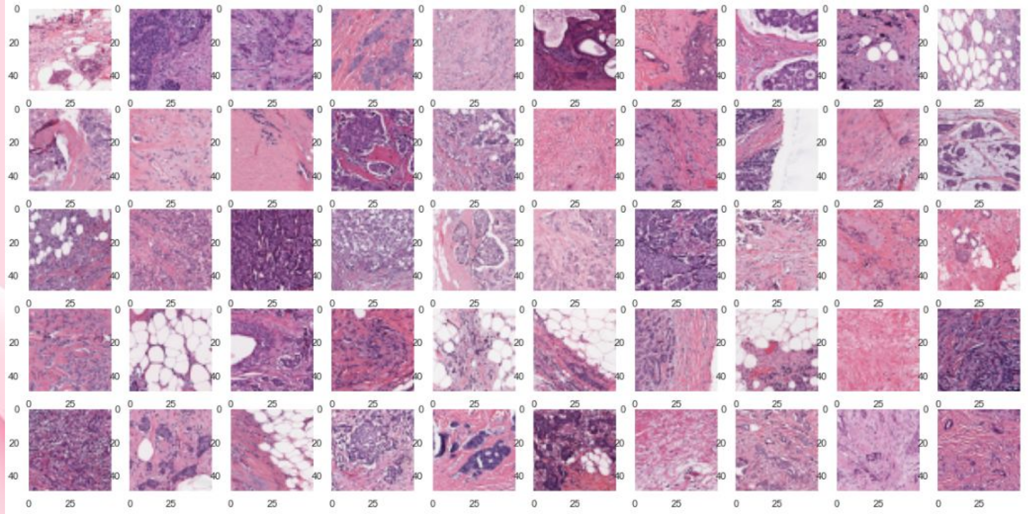


Blur

Negative Patches



Positive Patches



Neural Networks

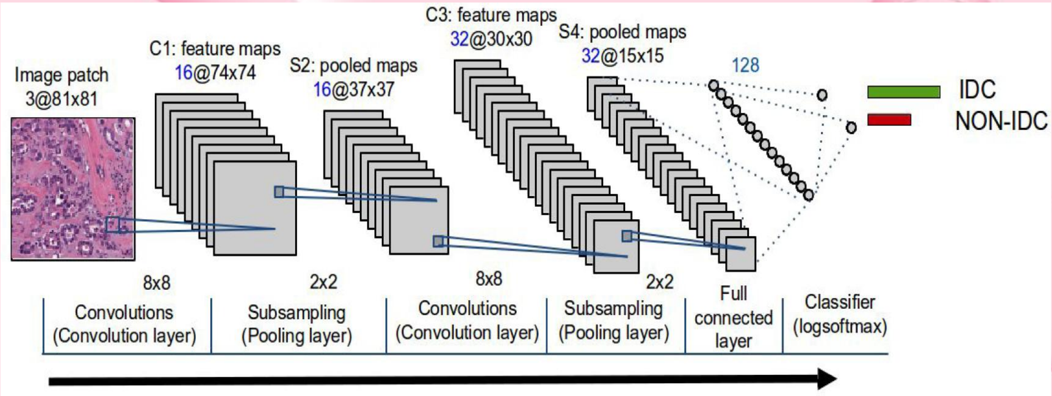
link

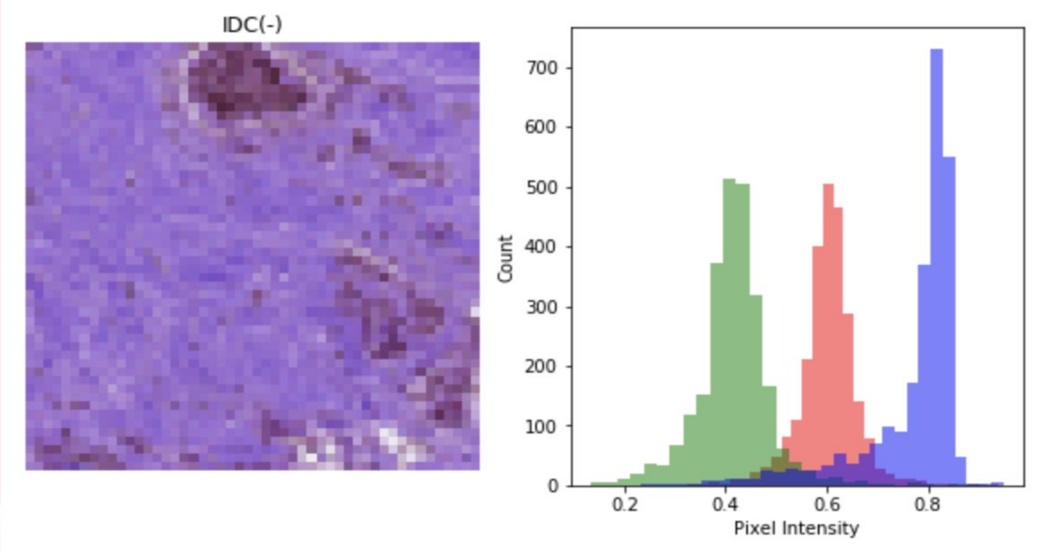
- A **neural network** is a series of algorithms that endeavors to recognize underlying relationships in a set of data

- A convolutional neural network (CNN, or ConvNet) is a class of deep neural networks, most commonly applied to analyzing image input

- My models will utilize **Keras**

- **Keras** is an open-source neural-network library written in Python. It is capable of running on top of TensorFlow and is designed to enable fast experimentation with deep neural networks.





Modeling Process

- Tech issues from the onset - needed to reduce the total images used
- Using random sampling, the classes are balanced before the modeling begins
- Data is divided into training, testing, and validation sets
- My first attempt utilized Conv2D layers in a Sequential Model to help identify essential features that classify the images and dropouts were used to help avoid overfitting the model
- This is a binary classification problem and we want to produce a ***probability*** so the sigmoid activation layer is generally used for the output
- Given the nature of the subject matter, our goal is to limit the number of false-negatives (**recall**) to ensure those with IDC do not walk away believing they are healthy

Predictions from model

```
array([[0.9746208 , 0.02506059],
       [0.97701275, 0.02278483],
       [0.38747153, 0.6134969 ],
       ...,
       [0.56684154, 0.4368201 ],
       [0.26279867, 0.7358743 ],
       [0.03667714, 0.96294326]]
```

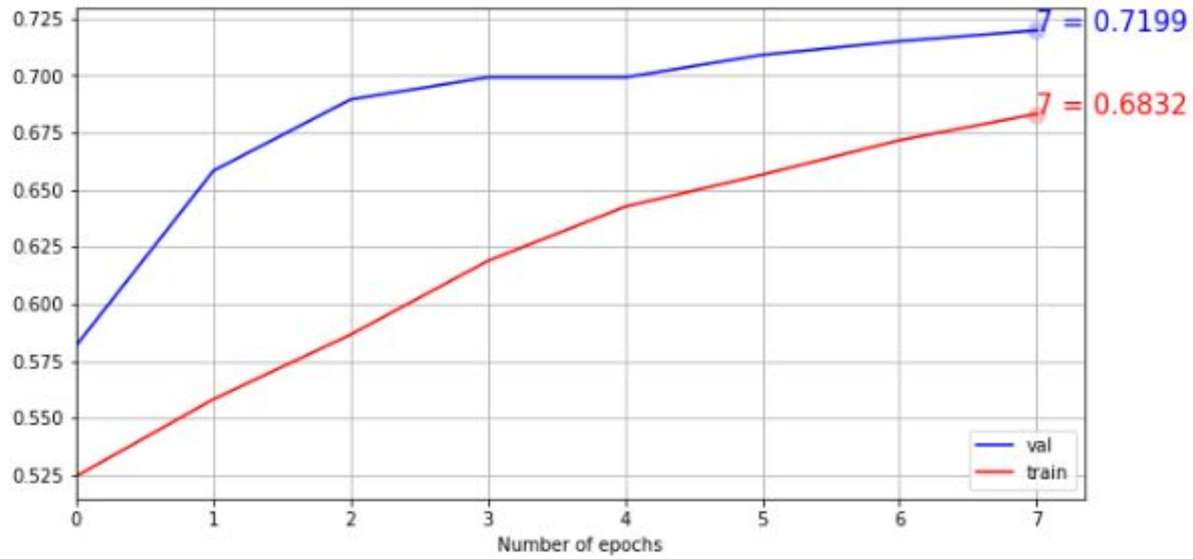
```
y_pred_labels[:3]
```

```
array([0, 0, 1])
```

Baseline Results

Keras CNN #1C - accuracy: 0.7198553681373596

	precision	recall	f1-score	support
IDC(-)	0.73	0.69	0.71	5117
IDC(+)	0.71	0.75	0.73	5117
accuracy			0.72	10234
macro avg	0.72	0.72	0.72	10234
weighted avg	0.72	0.72	0.72	10234

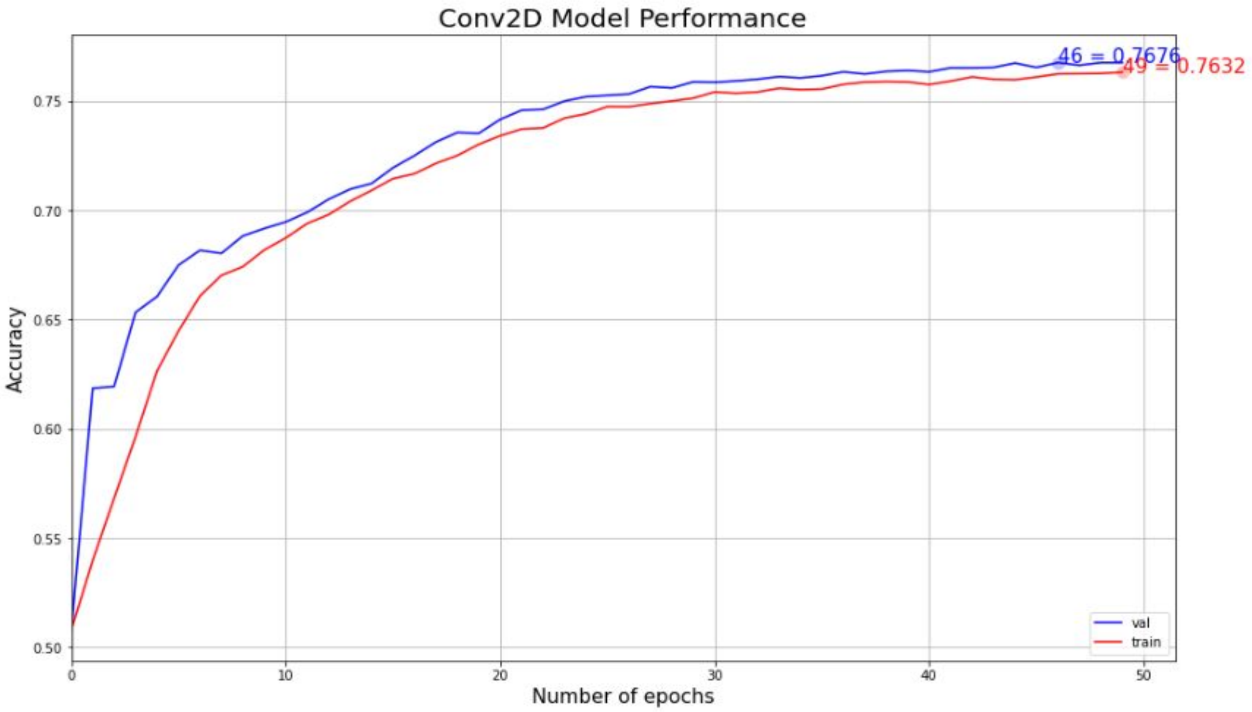


- ❖ Not overfit as the validation data actually outperformed the training set
- ❖ 72% accuracy
- ❖ Not a great recall score
- ❖ Plenty of room for improvement

Second Model

Keras CNN #1C - accuracy: 0.7675859332084656

	precision	recall	f1-score	support
IDC(-)	0.76	0.78	0.77	2502
IDC(+)	0.77	0.76	0.77	2502
accuracy			0.77	5004
macro avg	0.77	0.77	0.77	5004
weighted avg	0.77	0.77	0.77	5004



- ❖ Used 90000 images
- ❖ Trained for 50 epochs
- ❖ Improved recall and accuracy

- **Adjusted hyperparameters on the model but got stuck around 76% accuracy**

- **Needed more input features**

- **Switched to using Macbook and was able to process all 278,000 images**

- **I then tested out the difference by running a very simple feed forward model**

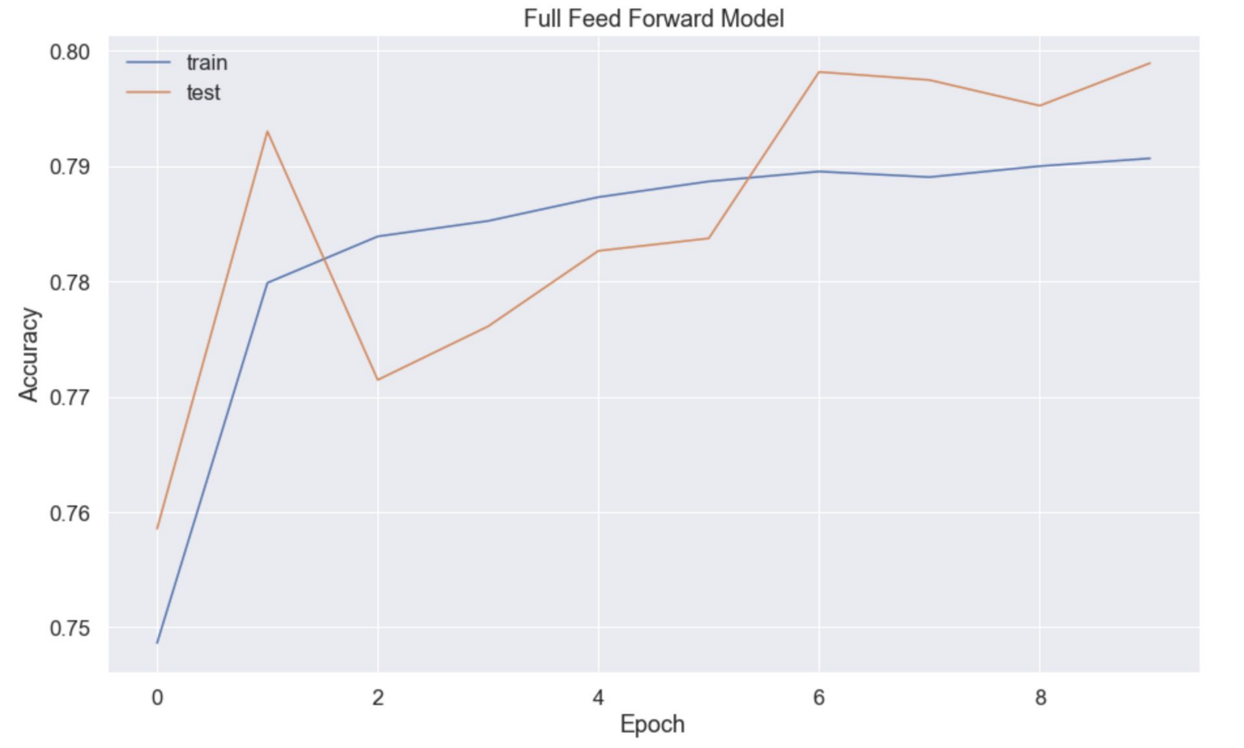
Feed Forward Model

```
Feed Forward Accuracy Score 0.7989068031311035

              precision    recall  f1-score   support

      IDC(-)       0.79      0.82      0.80      7867
      IDC(+)       0.81      0.78      0.79      7867

    accuracy                           0.80     15734
   macro avg       0.80      0.80      0.80     15734
weighted avg       0.80      0.80      0.80     15734
```

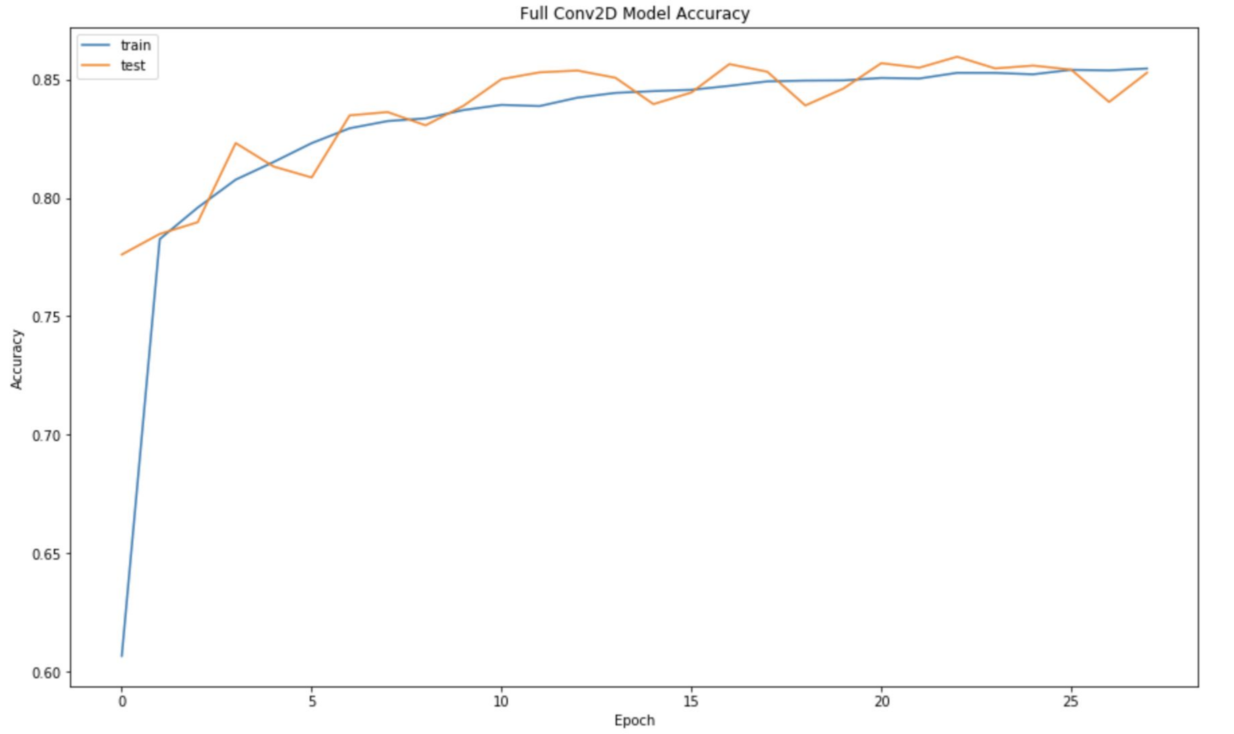


- ❖ Improved scores
- ❖ Trained for 10 epochs
- ❖ Best Recall score yet at 78%

CNN model on full dataset

CNN Accuracy Score 0.852929949760437

	precision	recall	f1-score	support
IDC(-)	0.91	0.78	0.84	7867
IDC(+)	0.81	0.92	0.86	7867
accuracy			0.85	15734
macro avg	0.86	0.85	0.85	15734
weighted avg	0.86	0.85	0.85	15734



- ❖ Increasing the amount of samples dramatically boosted results
- ❖ Accuracy jumps up to 85% and a 92% recall score is achieved

- **Again, no matter how much fine tuning was done, the model seemed to top out at around 85%**
- **Taking hours for models to run**
- **Took advantage of free GPU to accelerate the computations**
- **Allowed me to use transfer learning in newer models**

- **Transfer learning allows us to train deep networks using less data then we would need if we had to train from scratch**

- **In transfer learning, we are basically transferring the “knowledge” that a model has learned from a previous task to our current one.**

- **Transfer learning helps to save time and improve accuracy**

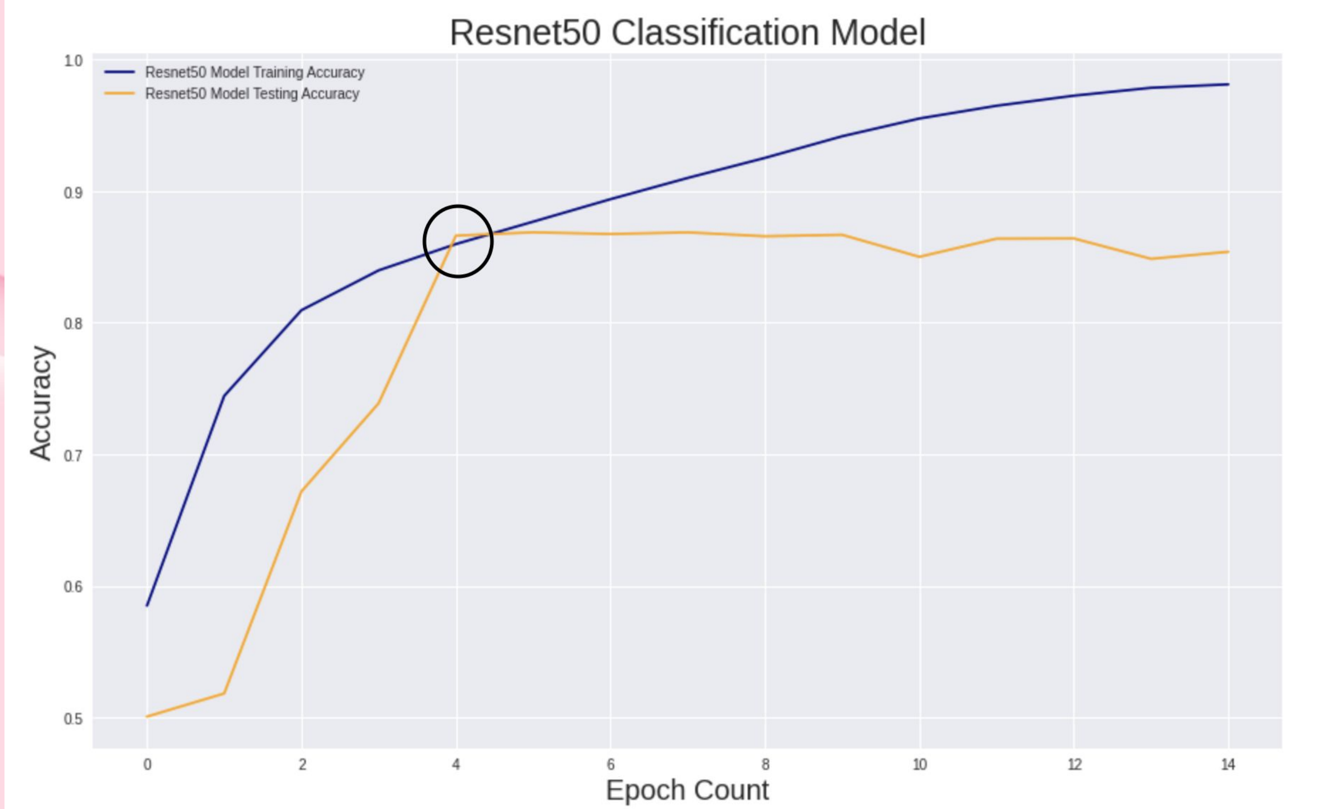
- **Many of the most popular transfer learning models are built on Google’s Imagenet**

link

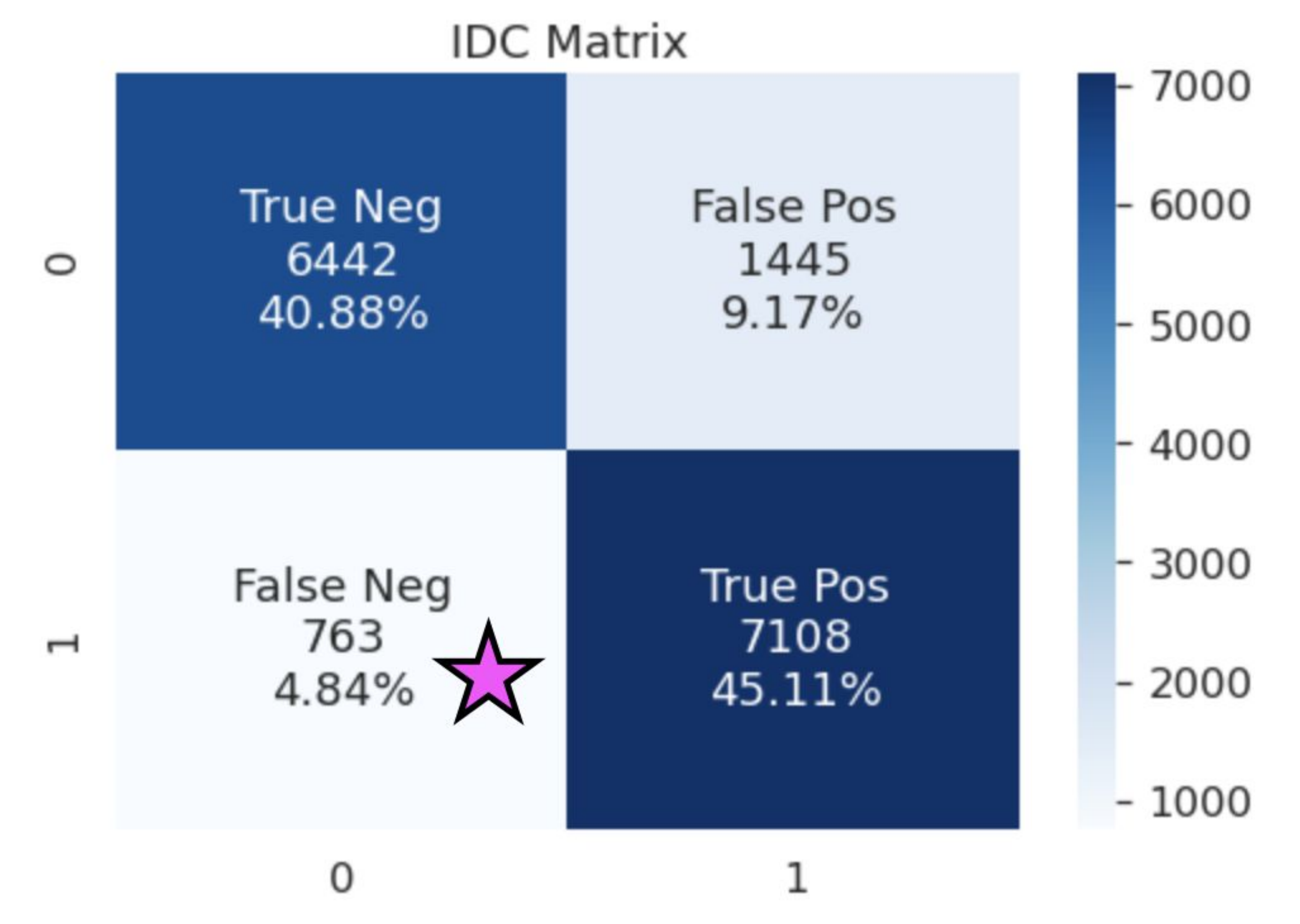
From Wikipedia: The ImageNet project is a large visual database designed for use in visual object recognition software research. More than 14 million images have been hand-annotated by the project to indicate what objects are pictured and in at least one million of the images, bounding boxes are also provided. ImageNet contains more than 20,000 categories with a typical category, such as "balloon" or "strawberry", consisting of several hundred images.

ResNet50 Model

	precision	recall	f1-score	support
a_no_idc	0.89	0.82	0.85	7887
b_has_idc	0.83	0.90	0.87	7871
accuracy			0.86	15758
macro avg	0.86	0.86	0.86	15758
weighted avg	0.86	0.86	0.86	15758

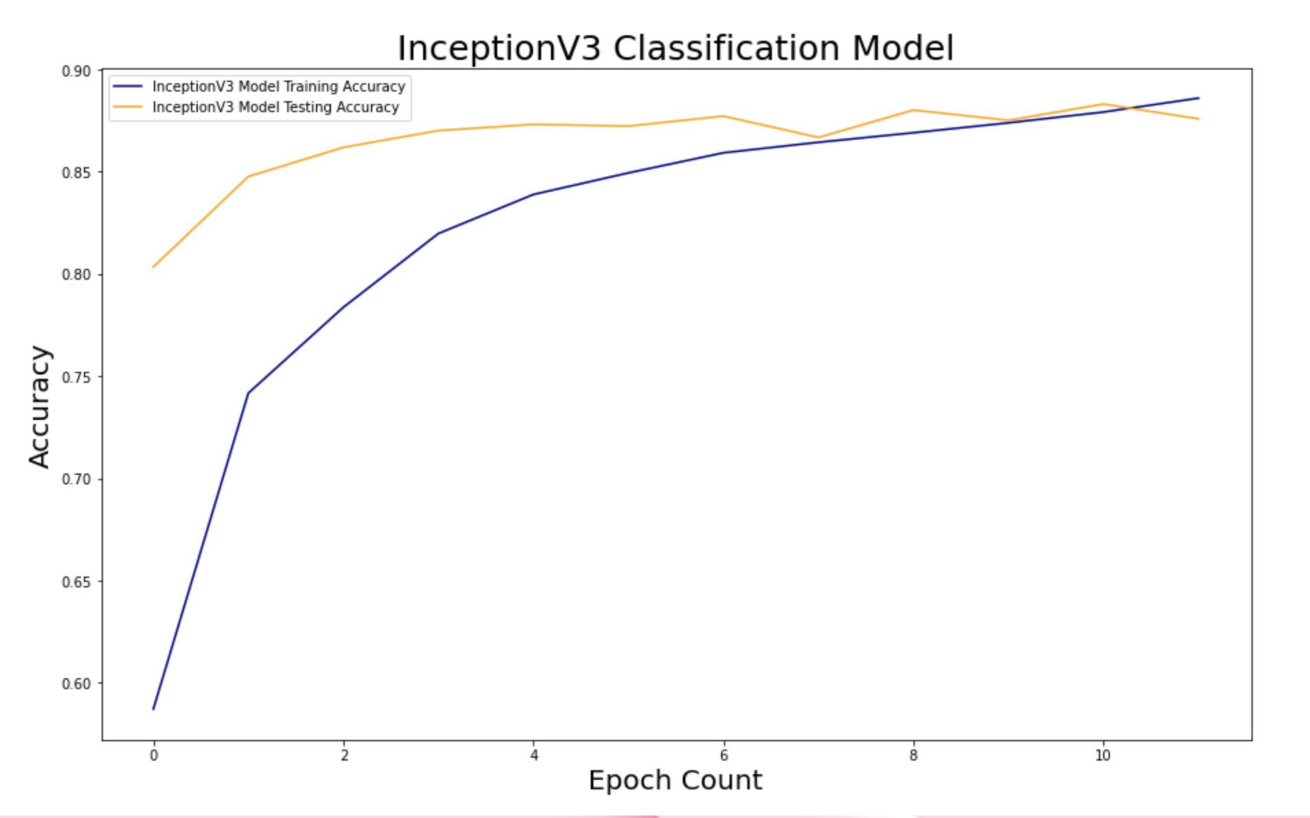


- ❖ The accuracy gets a little better then peaks at the 4th epoch around 86%
- ❖ Using the ResNet50 model as a base then adding my own layers, I ended up with over 28,000,000 total parameters

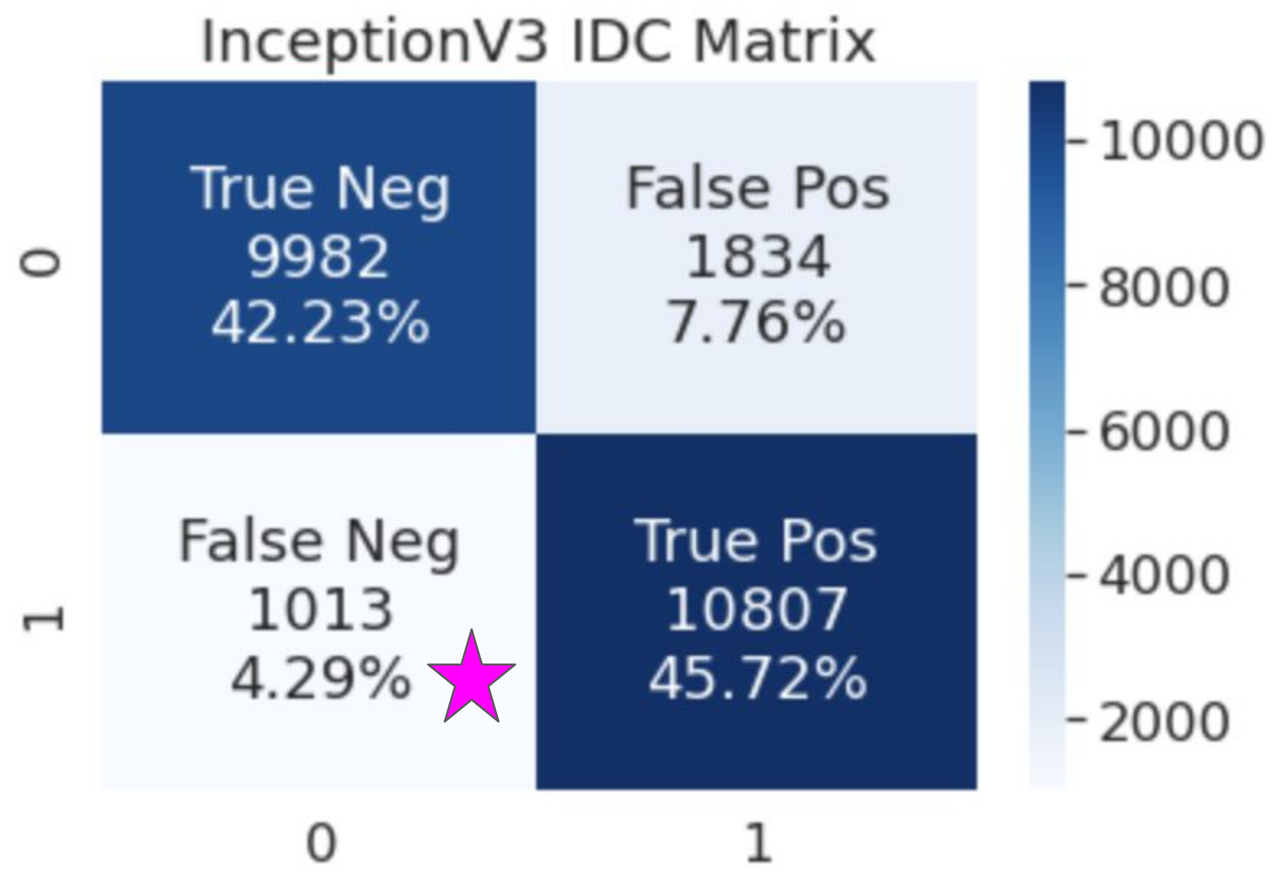

IDC Matrix
True Neg
6442
40.88%
False Pos
1445
9.17%
False Neg
763
4.84%
True Pos
7108
45.11%
0
1
0
1
7000
6000
5000
4000
3000
2000
1000

InceptionV3 Model

	precision	recall	f1-score	support
a_no_idc	0.91	0.84	0.88	11816
b_has_idc	0.85	0.91	0.88	11820
accuracy			0.88	23636
macro avg	0.88	0.88	0.88	23636
weighted avg	0.88	0.88	0.88	23636



- ❖ The InceptionV3 Model had the best overall accuracy and a recall score of 91%
- ❖ Not as overfit as the ResNet50 model
- ❖ Used to help classify leukemia



InceptionV3 IDC Matrix
True Neg
9982
42.23%
False Pos
1834
7.76%
False Neg
1013
4.29%
True Pos
10807
45.72%
0
1
0
1
10000
8000
6000
4000
2000

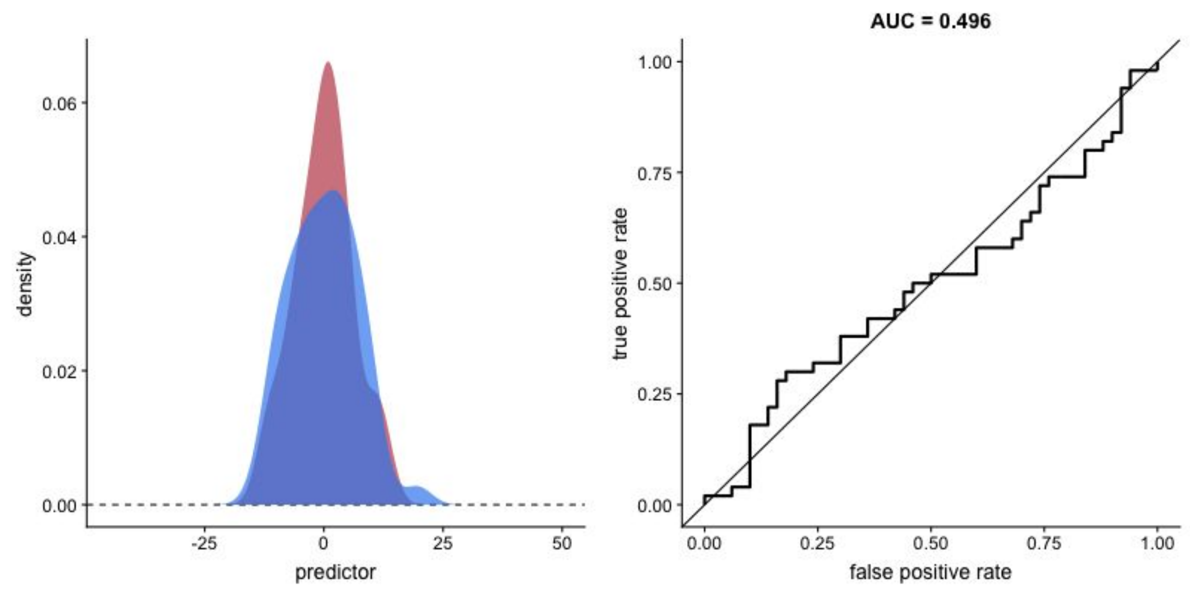
ROC, AUC, precision, and recall visually explained

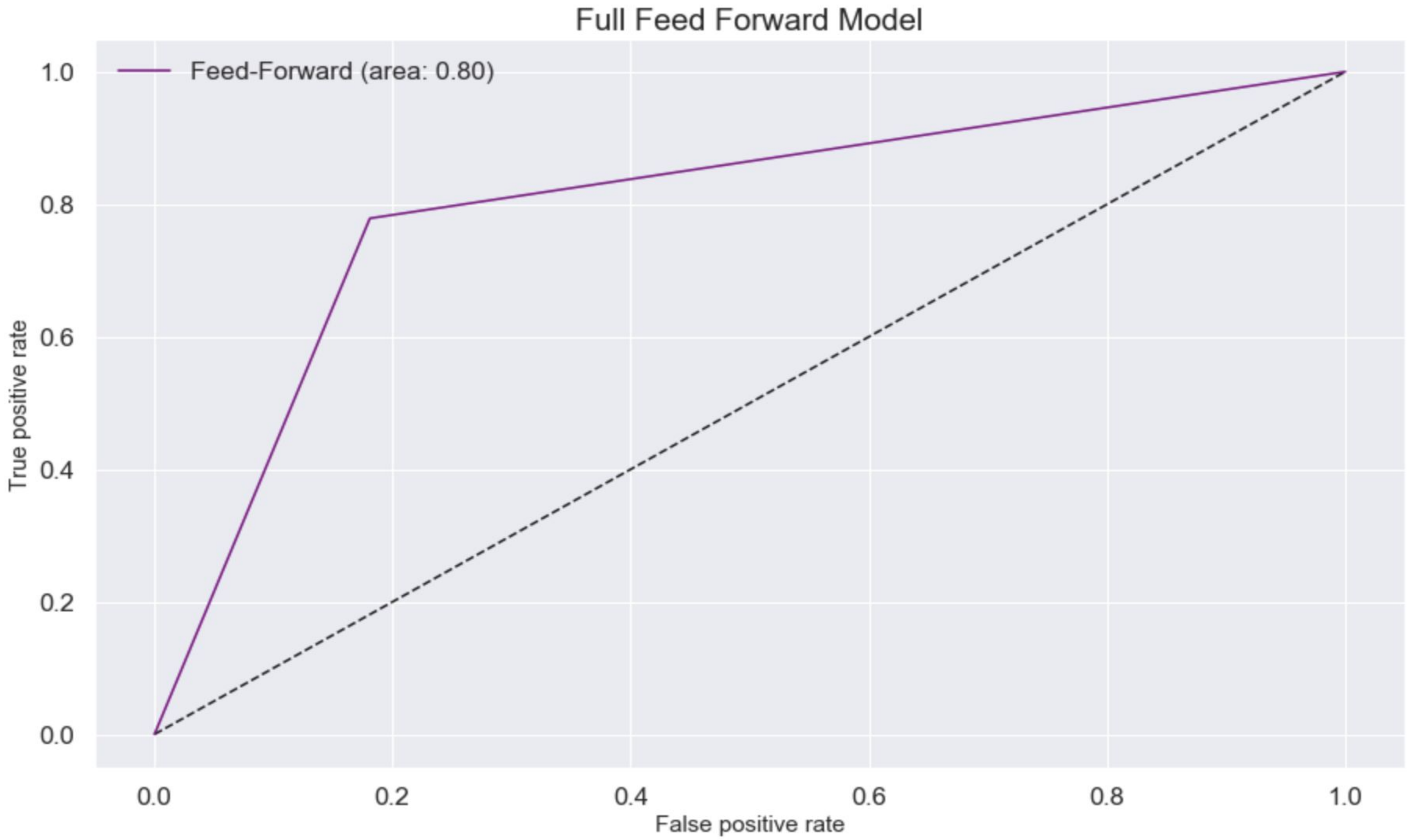
- A receiver operating characteristic (ROC) curve displays how well a model can classify binary outcomes
- An ROC curve is generated by plotting the false positive rate of a model against its true positive rate, for each possible cutoff value
- Often, the area under the curve (AUC) is calculated and used as a metric showing how well a model can classify data points

https://paulvanderlaken.com/2019/08/16/roc-auc-precision-and-recall-visually-explained/

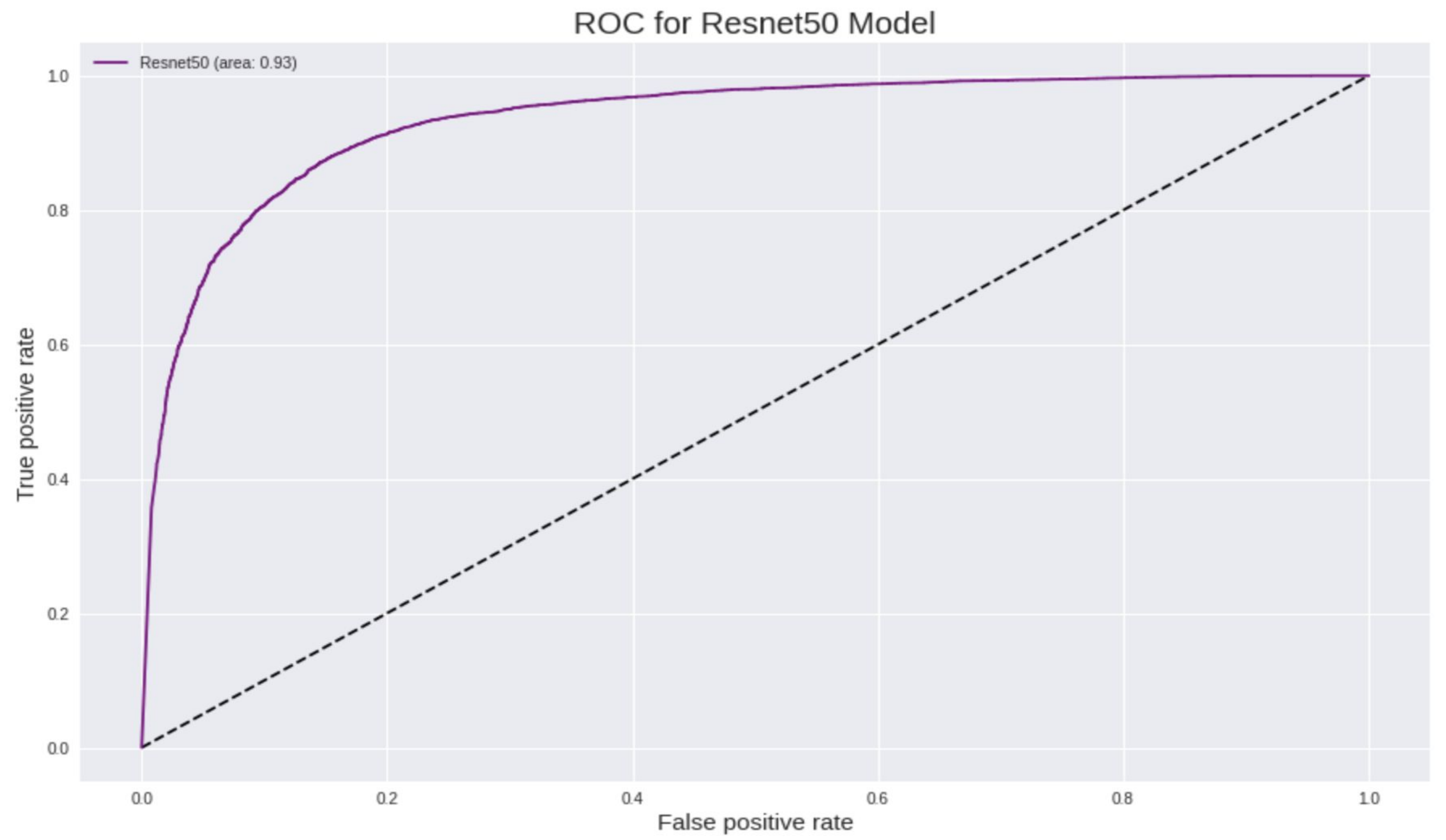
link

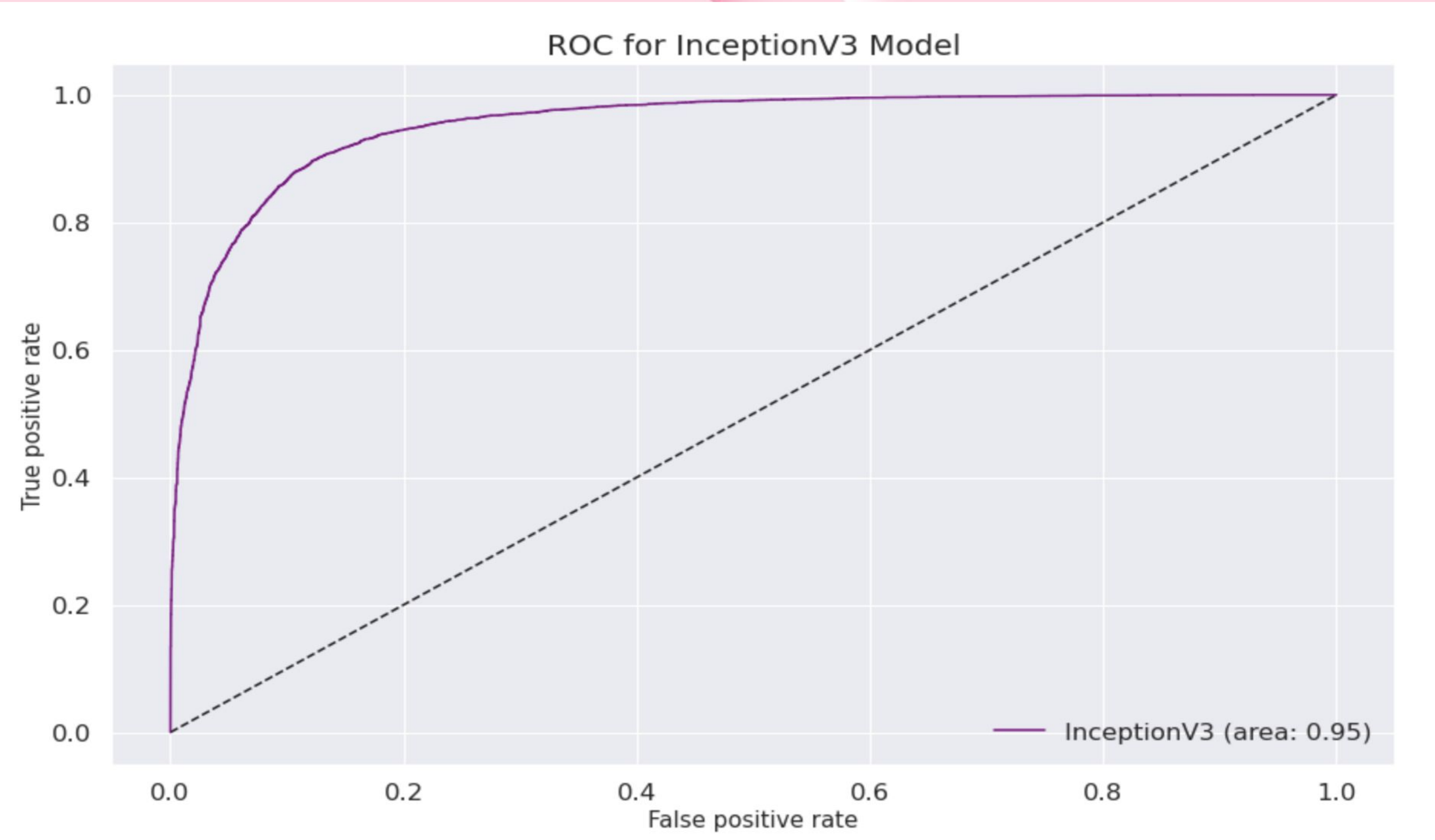
Predicted malignant and actually malignant *(True Positive)*	Predicted malignant but actually benign *(False Positive)*
Predicted benign but actually malignant *(False Negative)*	Predicted benign and actually benign *(True Negative)*





Full Feed Forward Model
Feed-Forward (area: 0.80)
True positive rate
False positive rate
0.0
0.2
0.4
0.6
0.8
1.0


ROC for Resnet50 Model
Resnet50 (area: 0.93)
True positive rate
False positive rate
0.0
0.2
0.4
0.6
0.8
1.0
0.0
0.2
0.4
0.6
0.8
1.0


ROC for InceptionV3 Model
1.0
0.8
0.6
0.4
0.2
0.0
True positive rate
0.0
0.2
0.4
0.6
0.8
1.0
False positive rate
InceptionV3 (area: 0.95)

	Accuracy	Recall	ROC
Feed Forward Model	80%	78%	80%
CNN Model	85%	92%	86%
ResNet50 Model	86%	90%	93%
InceptionV3 Model	88%	91%	95%

Challenges and Limitations

1) Computational power and image data
2) Better image quality would help
3) Lack of domain expertise
4) Large neural nets can have high variance and overfit models

Conclusions and recommendations

1) The learning rate is the most influential hyperparameter
2) Trained models alleviate pathologist work and provide a second set of eyes
3) Transfer learning and ensemble methods will continue to assist in accurate and reliable classification

ARTIFICIAL INTELLIGENCE, DIAGNOSTICS, HEALTH TECH

Google's AI beats humans at detecting breast cancer — sometimes

A retrospective study published in Nature shows Google's DeepMind AI outperformed radiologists in detecting breast cancer. But it won't be replacing them anytime soon.

By ELISE REUTER

Post a comment / Jan 3, 2020 at 5:08 PM

Google's DeepMind AI outperformed radiologists in detecting breast cancer, according to a retrospective study published in Nature on Wednesday. After being trained on thousands of mammograms, the system was able to better identify breast cancer cases than the radiologists that had made their initial assessments.

link

Thank you!