Breast Cancer Classification

Predicting IDC in Breast Cancer Histopathology Images

Final Presentation

W207 Applied ML
Fall 2022

Heesuk Jang
Tatianna Martinez
Kesha Julien
Rachael Phillips

Research Question

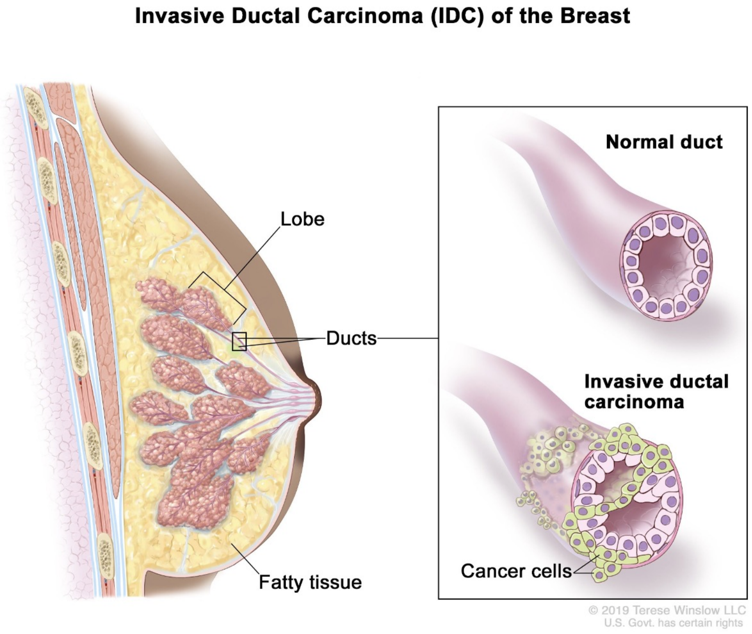
Can we accurately diagnose Invasive Ductal Carcinoma (IDC) using machine learning technologies given breast cancer histology images ?

Invasive Ductal Carcinoma (IDC) is the most common subtype of all breast cancers, accounting for 80% of all breast cancer diagnoses.

Our Focus is being able to automate the identification and classification of IDC would help to save time and resources for pathologists.

The application of machine learning and AI service network system can promote quality of medical services in rural areas where medical providers are not available.

What is Invasive Ductal Carcinoma (IDC)?



What is IDC?

Invasive ductal carcinoma (IDC), also known as **infiltrating ductal carcinoma**, is a type of breast cancer that starts in the milk ducts of the breast and moves into nearby tissue. In time, IDC may spread (metastasize) through the lymph nodes or bloodstream to other areas of the body.

How is IDC Found?

Pathologists typically focus on the regions which contain the IDC, rather than the whole mount slide. As a result, one of the common pre-processing steps for identification is to delineate the exact regions of IDC inside of a whole mount slide.

Dataset

Dataset from Kaggle

Size of Data Set:

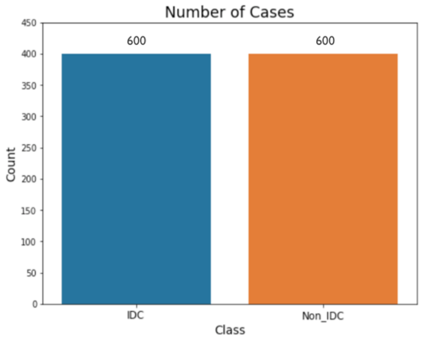
277,524 Images and divided into train, validate, and test
Focused on 1,200 randomly sampled images (Train/Test/Val = 800/200/200)

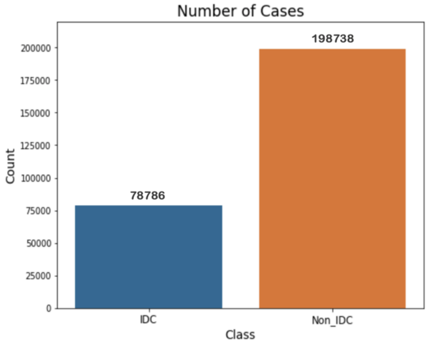
Main Features:

Is IDC Present? 0 or 1
Mount slide cross section

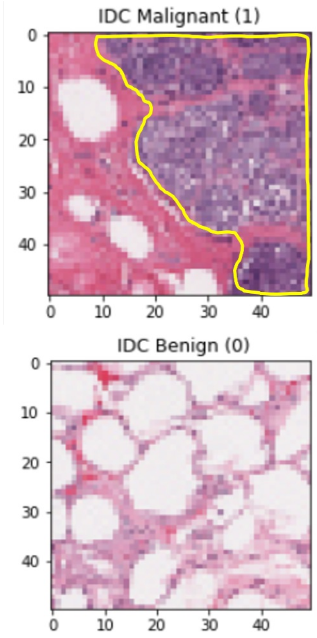
Class Imbalance

72 % (198,738) non IDC vs 28% (78,786) IDC





Breast Histopathology Images



How is the Data Organized?

Dataset Overview

162 whole mount slides, scanned at 40x
C indicates the class, 0 is non-IDC and 1 is IDC

Initial EDA and PreProcessing

There are no missing values or values other than 0 or 1 in our Target
Pulled out "**bad**" images (Train/Test/Val = 3/1/3)
Used a total of 1,193 images for our Train/Test/Validation sets (797/199/197)
The same data is used for both the Baseline and Advanced Model

Transformations Applied to Images

Grayscale
Augmentation
Normalization

Bad Images

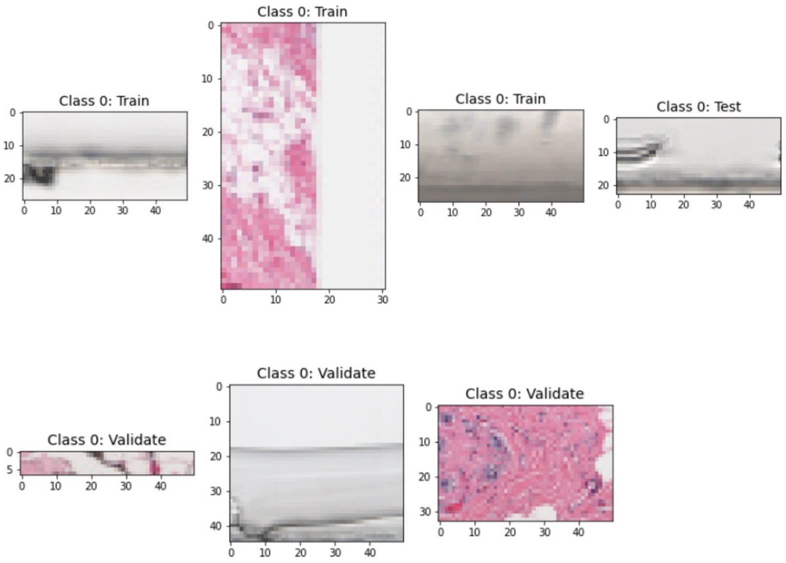
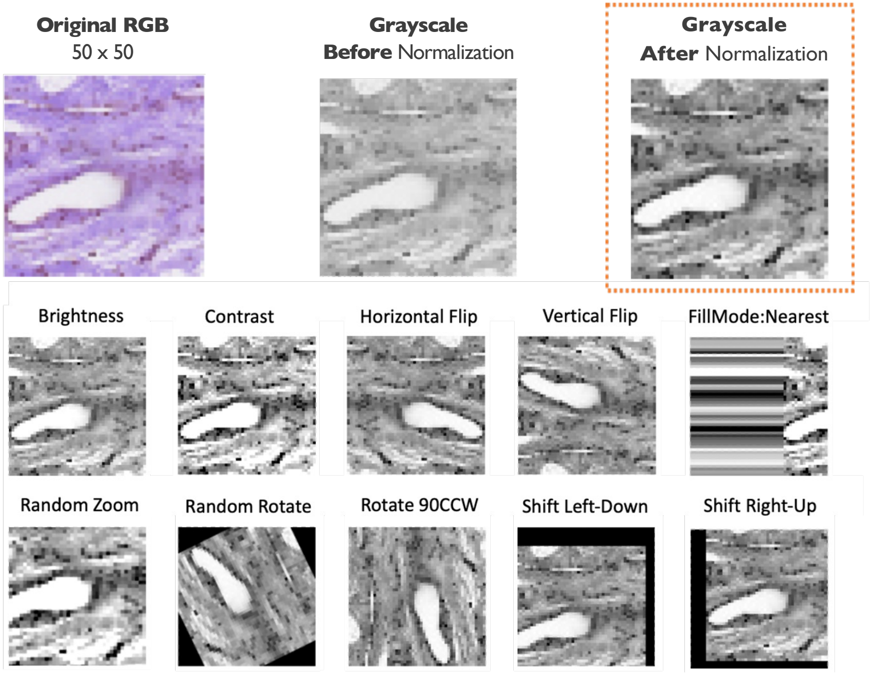


Image Augmentation



Augmented the grayscale images using OpenCV

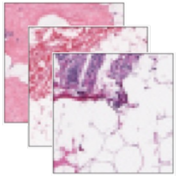
Normalized the augmented images using Keras ImageDataGenerator

Used 9,564 training images (12X more data) to train the machine learning models

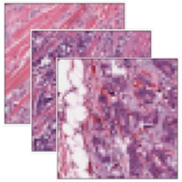
Our Approach

Input Images

IDC Benign (0)



IDC Malignant (1)



Models Used

01 Baseline Model
Support Vector Machine (SVM)

02 Model 2
Random Forest

03 Model 3
Convolutional Neural Networks (CNN)

04 Model 4
CNN Transfer Learning

Evaluation Metrics

01 **Test Accuracy**

02 **Validation Accuracy**

03 Precision

04 **Recall**

05 **F1 Score**

06 Zero One Loss

07 **ROC Curve and AUC**

Support Vector Machines (SVM) as Our Baseline

Key Parameters for SVM

- **Gamma:** Defines how far the influence of a single training example reaches values lead to biased results
- **C:** Controls the cost of miscalculations
- **Kernel:** Mathematical functions (Ex: Linear, RBF, Polynomial)

We used **GridSearch** in our model to test multiple values for each parameter. We found that:

- The RBF kernel performed the best
- Accuracy score of Train: **0.77**
- Accuracy score for Test: **0.72**
- Accuracy score for Validation: **0.69**

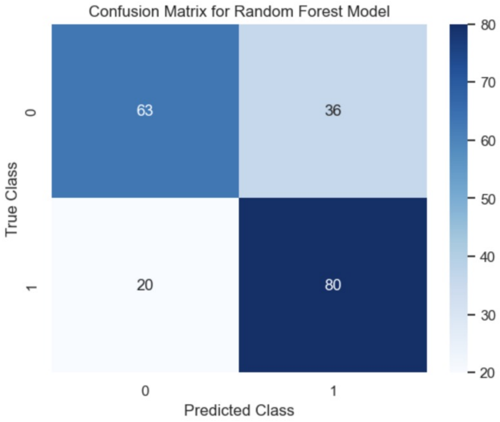
Classification Report

	precision	recall	f1-score	support
0	0.75	0.65	0.70	99
1	0.69	0.79	0.74	100
accuracy			0.72	199
macro avg	0.72	0.72	0.72	199
weighted avg	0.72	0.72	0.72	199

Random Forest

Key Parameters for Random Forest

- N_estimators
- Max_features
- Max_Depth
- Min_samples_split
- Min_leaf_samples
- criteria



Hyperparameter Tuning

We used **RandomSearchCV** in our model
We found that:

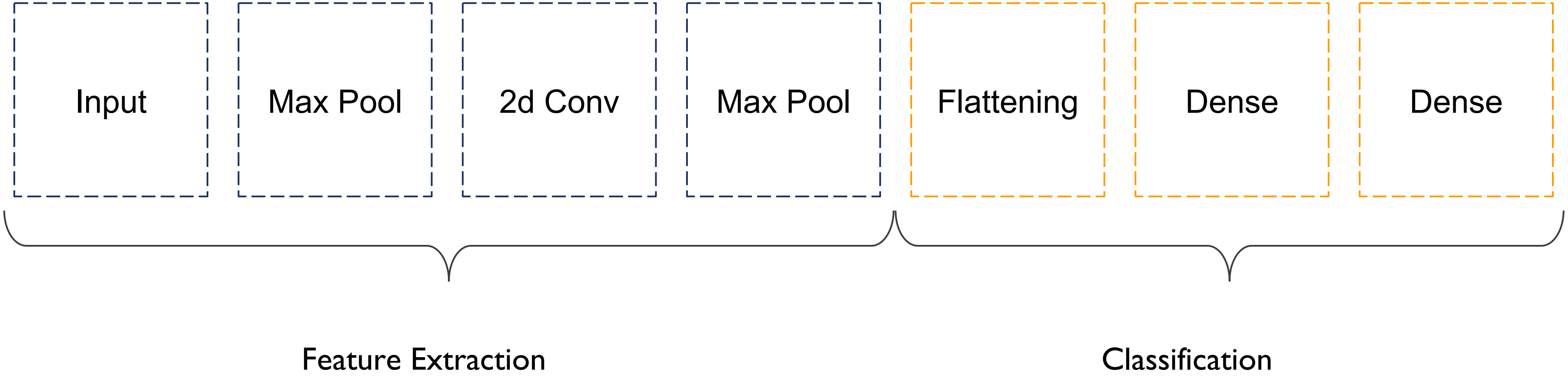
- Accuracy score of Train: 0.76
- Accuracy score for Test: 0.72
- Accuracy score for Validation: 0.70

Classification Report

	precision	recall	f1-score	support
0	0.76	0.64	0.69	99
1	0.69	0.80	0.74	100
accuracy			0.72	199
macro avg	0.72	0.72	0.72	199
weighted avg	0.72	0.72	0.72	199

CNN Model

Model Architecture:



CNN Model

Hyperparameter Tuning Options

Hyperparameter	**Values**
Number of Layers	1, 2, 3, 4
Filter Sizes	8, 16, 32, 64, 128
Kernel Sizes	1, 2, 3, 4, 5
Number of Dense Layer Units	8,16,32,64,128, 256, 521
Learning Rates	0.1, 0.001, 0.0001, 0.00001

CNN Model

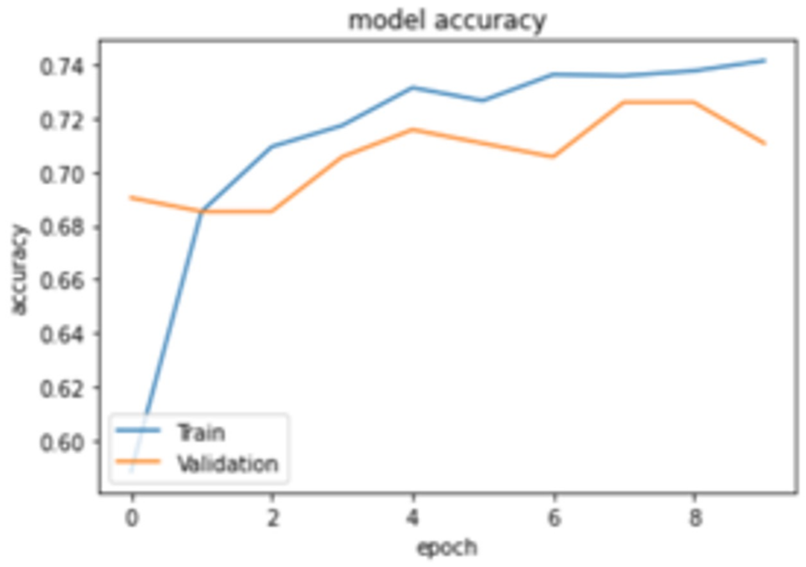
Hyperparameter Tuning

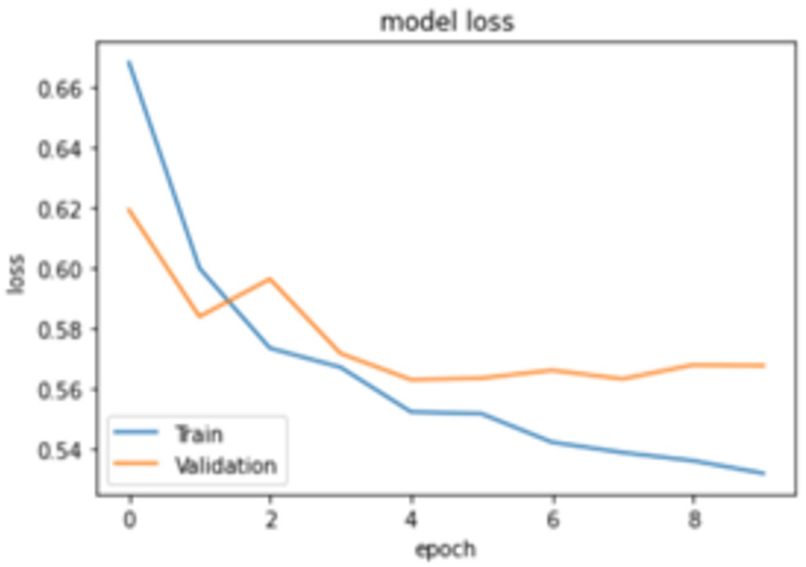
```
Model: "sequential_50"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d_159 (Conv2D)         (None, 50, 50, 16)        32

 max_pooling2d_159 (MaxPooli  (None, 25, 25, 16)       0
 ng2D)

 conv2d_160 (Conv2D)         (None, 25, 25, 8)         3208

 max_pooling2d_160 (MaxPooli  (None, 12, 12, 8)        0
 ng2D)

 conv2d_161 (Conv2D)         (None, 12, 12, 8)         1608

 max_pooling2d_161 (MaxPooli  (None, 6, 6, 8)          0
 ng2D)

 flatten_50 (Flatten)        (None, 288)               0

 dense_100 (Dense)           (None, 521)               150569

 dense_101 (Dense)           (None, 2)                 1044

=================================================================
Total params: 156,461
Trainable params: 156,461
Non-trainable params: 0
_________________________________________________________________
```

	Number of Filters	**Kernel Size**
Input Layer	16	1
Conv2D 1	8	5
Conv2D 2	8	5
Dense	521	

CNN Model

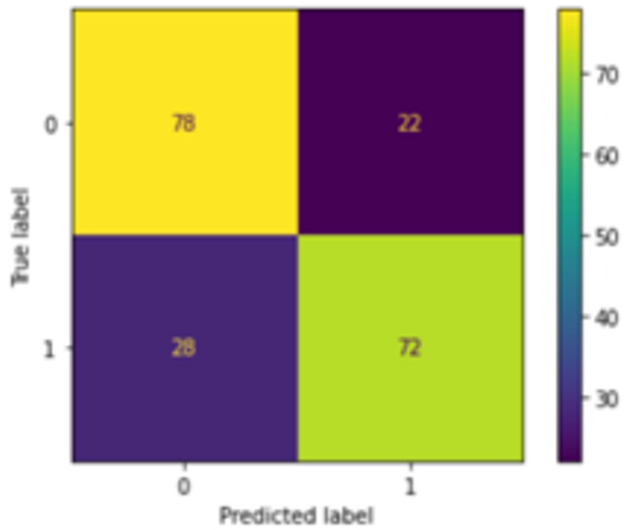
Model Results





CNN Model

Model Results and Metrics



Classes	Accuracy	Precision	Recall	F1-Score
0	0.78	0.74	0.78	0.76
1	0.72	0.77	0.72	0.74

Transfer Learning with CNN

Applied Transfer Learning

01 VGG16

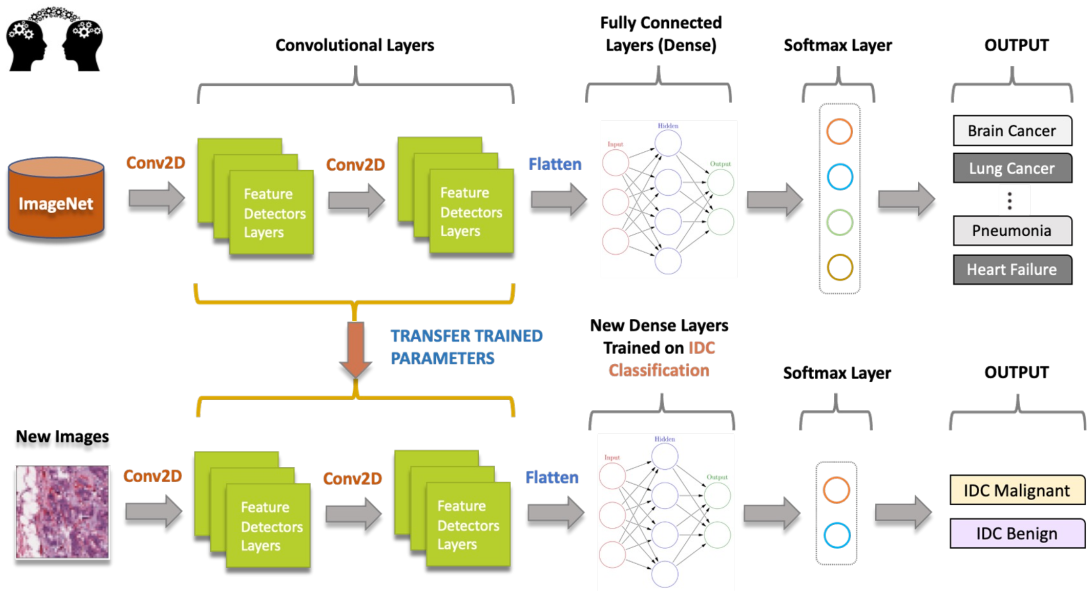
02 VGG19

03 ResNet50

04 DenseNet201

05 ResNet152V2

Transfer Learning Process



Tuning Hyper Parameters with Keras Random Search Tuner

Hyper Parameters

Tuning Trials	Execution Per Trial	Batch Size	Epochs	Objective	Image Size
60	2	64	20	Validation Loss	50 x 50 x 3

01 Optimizer
SGD, Adam, Rmsprop

02 Activation
ReLU, Leaky ReLU, Tanh, Gelu

03 Dense Units
[min=**128**, max=**1024**, step=**128**]

04 Learning Rate
[le-2, le-3, le-4]

05 Dropout Rate
[0.2, 0.3, 0.4, 0.5, 0.6]

06 Add Dense Layer
True, False

07 Brightness Delta
[0.2, 0.3, 0.4]

08 Contrast Factor
[2, 3, 4]

09 Rotate Range
[-90, 90]

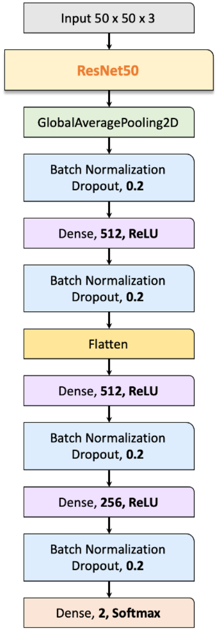
10 Shift Width and Height
[-12, 12]

TL Model 1 - CNN Transfer Learning with All Frozen Layers

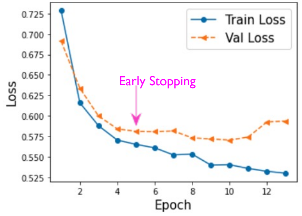
Top Models by Transfer Learning

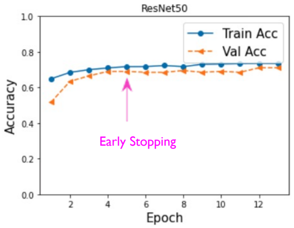
TL Model	Test Acc	Train Acc	Val Acc
DenseNet201	0.74	0.79	0.72
VGG16	0.74	0.71	0.68
ResNet152V2	0.74	0.79	0.71
VGG19	0.73	0.72	0.7
ResNet50	0.72	0.73	0.71

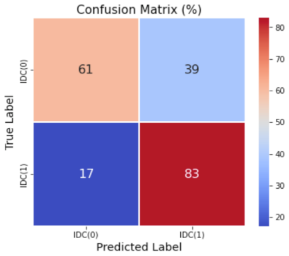
CNN with ResNet50

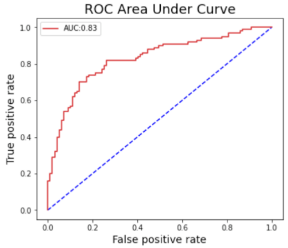


Test Accuracy	Train Accuracy	Val Accuracy	Learning Rate	Activation	Optimizer
0.72	0.73	0.71	0.01	ReLU	SGD







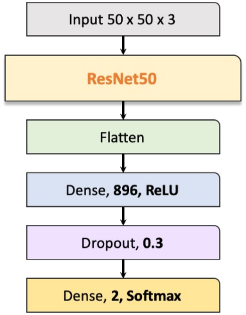


TL Model 2 - CNN Transfer Learning with Unfrozen Last Layer

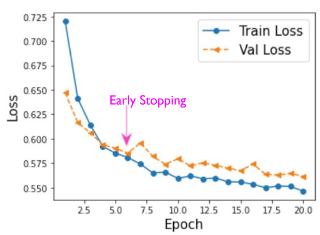
Top Models by Transfer Learning

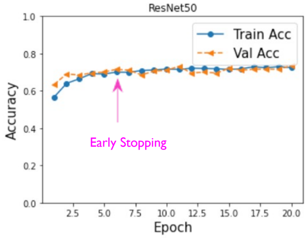
TL Model	Test Acc	Train Acc	Val Acc
DenseNet201	0.77	0.8	0.73
ResNet50	0.76	0.72	0.74
VGG16	0.74	0.78	0.71
ResNet152V2	0.71	0.76	0.66
VGG19	0.7	0.77	0.71

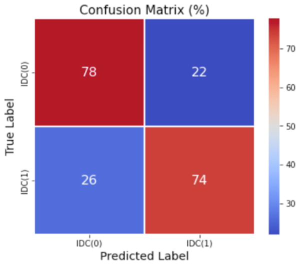
CNN with ResNet50

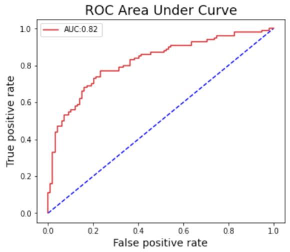


Test Accuracy	Train Accuracy	Val Accuracy	Learning Rate	Activation	Optimizer
0.76	0.72	0.74	0.0001	ReLU	Adam









Results Before and After Transfer Learning

Possible Sources of the Insignificant Improvement

with Transfer Learning

01 Mismatch in domain

02 Stronger augmentation

03 Increased size of the fine-tune dataset

Conclusions - Model Summary

Model	Test_Acc	Train_Acc	Validation_Acc	Precisions	Recall	F1Score	ROC_AUC	ZeroOneLoss
Random Forest	0.72	0.76	0.70	0.68	0.80	0.74	0.71	0.28
CNN without Transfer Learning	0.75	0.74	0.71	0.75	0.75	0.75	0.72	0.25
M1: CNN with ResNet50	0.72	0.73	0.71	0.73	0.72	0.71	0.83	0.28
M1: CNN with DenseNet201	0.74	0.79	0.72	0.74	0.74	0.74	0.80	0.26
M1: CNN with ResNet152V2	0.74	0.79	0.71	0.74	0.74	0.74	0.80	0.26
M1: CNN with VGG16	0.74	0.71	0.68	0.74	0.74	0.74	0.82	0.26
M1: CNN with VGG19	0.74	0.72	0.69	0.74	0.74	0.74	0.80	0.26
M2: CNN with ResNet50	0.76	0.72	0.74	0.77	0.74	0.75	0.82	0.24
M2: CNN with DenseNet201	0.77	0.80	0.73	0.81	0.72	0.76	0.81	0.23
M2: CNN with ResNet152V2	0.71	0.76	0.66	0.71	0.70	0.70	0.81	0.29
M2: CNN with VGG16	0.74	0.78	0.71	0.75	0.74	0.74	0.83	0.26
M2: CNN with VGG19	0.70	0.77	0.71	0.68	0.79	0.73	0.81	0.30

Conclusions - Top 5 Models

Model	Test_Acc	Validation_Acc	Recall	F1Score	ROC_AUC
M2: CNN with ResNet50	0.76	0.74	0.74	0.75	0.82
CNN without Transfer Learning	0.75	0.74	0.75	0.75	0.72
M1: CNN with ResNet50	0.72	0.71	0.72	0.71	0.83
M2: CNN with DenseNet201	0.77	0.80	0.73	0.81	0.72
Random Forest	0.72	0.70	0.80	0.74	0.71

Conclusions - Lessons Learned & Limitations

Lessons Learned

Removing “bad” images from our dataset **improved accuracy** by 2-4%

Learned that certain fields have “**best practices**” for image augmentation

Important to work off the same dataset and augmented images for **accurate comparisons**

Limitations

Need significantly more processing power to run the original full set of data

Computationally expensive to maintain GPU capability to run pre-trained models

Limited knowledge on the differences among transfer learning methods and their impacts

Complex classification task on the histopathology images by nature

Looking Forward

Being able to handle a larger dataset

Utilize patient contextual data along with histopathology images to improve the classification results

Selection of better-suited transfer learning techniques for medical dataset

Hyperparameter tuning for a number of epochs and batch size for optimal performance

Employ various filter techniques to enhance the histopathology images to handle unstructured pattern

Project Contributions

Team Member	Research	EDA	SVM	Random Forest	CNN	CNN Transfer Learning	Presentation Slides
Rachael Phillips	X	X	X	X			X
Tatianna Martinez	X	X		X			X
Kesha Julien	X	X			X		X
Heesuk Jang	X	X			X	X	X

Reference

- **Kaggle** - https://www.kaggle.com/datasets/paultimothymooney/breast-histopathology-images
- **NIH National Cancer Institute** - https://www.cancer.gov/publications/dictionaries/cancer-terms/def/invasive-ductal-carcinoma
- **Penn Medicine Abramson Cancer Center** - https://www.pennmedicine.org/cancer/types-of-cancer/breast-cancer/types-of-breast-cancer/invasive-ductal-carcinoma#:~:text=Invasive%20ductal%20carcinoma%20(IDC)%2C,other%20areas%20of%20the%20body.