Novel method for improving deep-learning accuracy of diagnosing breast cancer using different image augmentations

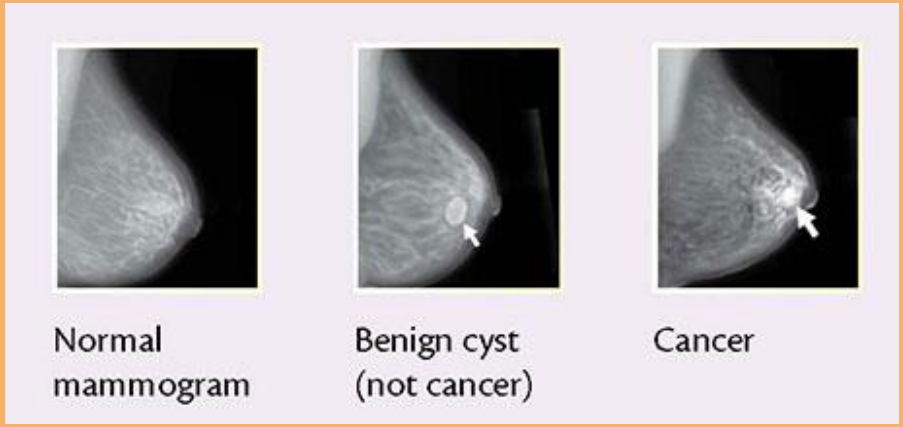
Pratik Bharadwaj

Abstract

According to the CDC, 1 in 8 U.S. women (about 12%) will develop invasive breast cancer over the course of their lifetime. As it grows, it can metastasize throughout the body causing serious health issues and death in many cases. Mammography remains the most effective means available to detect cancer in its earliest stages. However, overdiagnosis and overtreatment are two prevalent risks of mammography screening. These risks cause unnecessary mental and physical pain in addition to exorbitant cost as almost 4 billion dollars are spent on correcting misdiagnoses (CNBC). These adverse consequences can be mitigated by more accurate diagnosis of breast cancer using machine learning, specifically deep learning. This paper utilized deep learning to classify the nature of breast tumors gathered from the Curated Breast Imaging Subset of the Digital Database for Screening Mammography (CBIS-DDSM). The dataset consisted of multiple views of 3567 patient mammograms with full pathologies and annotations. Factors such as resolution, affinity, and contrast were artificially augmented for each image, to account for the various acquisition methods of mammograms as well as to better assess the neural network's predictions. The neural network achieved a validation accuracy of 82% in predicting the pathology of the tumor using a resolution of 200x200, as well as a contrast ratio of 2.0 and different affine transformation. Using this algorithm, issues regarding the overdiagnosis and overtreatment of breast tumors can be greatly mitigated and hence allowing for accurate diagnosis. Future work includes proving the generalizability of this model by testing it on other publicly available datasets as well as utilizing transferred learning via deeper neural network structures to improve validation accuracy.

Breast Cancer - Background

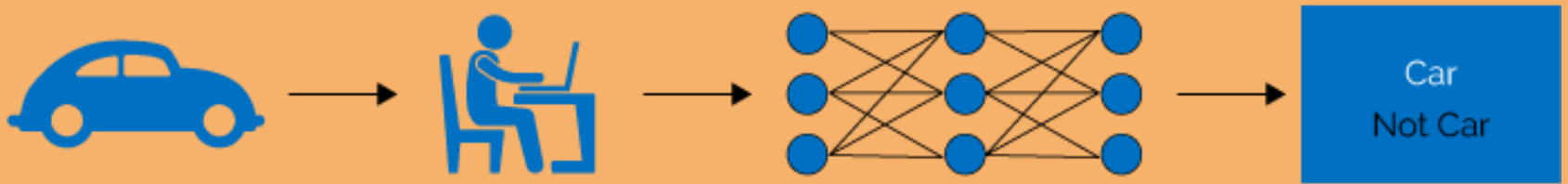
- 1 out of every 8 women are diagnosed with breast cancer
- About 6.8% percent of women die from the disease
- Mammography most effective way of early breast cancer diagnosis
  - Problems: misdiagnosis and overtreatment; 30% FP rate
  - 4 billion dollars spent annually for corrections
- Deep Learning can be used to eliminate misdiagnosis



Deep Learning-Background

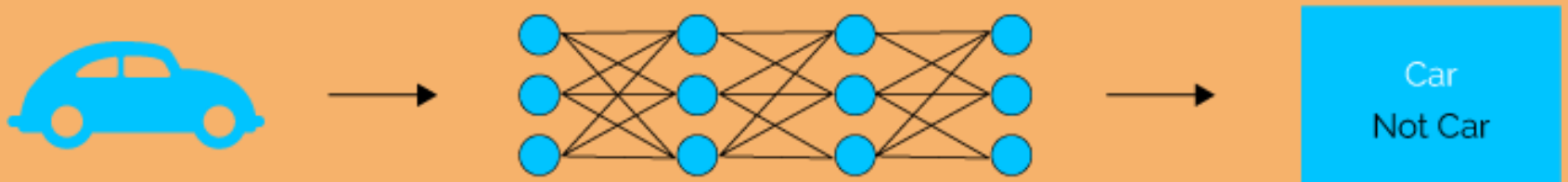
- Deep-learning software attempts to mimic the activity in layers of neurons in the neocortex
- Learns to recognize patterns in digital representations of sounds, images, and other data.
- Used in medical, audiovisual, and technological fields
- Many types of deep learning models used to diagnose breast cancer in today's research
- However, current implementations fail to generalize to other datasets as they have memorized and outputted the labels of one dataset

Machine Learning



Input → Feature extraction → Classification → Output

Deep Learning



Input → Feature extraction + Classification → Output

Literature Review

- effective way of diagnosing fine needle aspirates by comparing different machine learning algorithms' performance on classifying data [1]
- achieved peak accuracy of 96% diagnosing FNA's, benign sample majority; more equal spread necessary for accurate stats [1]
- CNN's used to directly classify pre-segmented breast masses in mammograms using transferred learning [2]
- Many pre-built CNN structures used, highest recall achieved without augmentation [2]
- These researchers focused on peak accuracy in 1-2 datasets, without accounting for differences in many mammogram datasets.
- This work focuses on analyzing the effect of different augmentations on the dataset to improve generalizability

Purpose

After analyzing multiple research papers focused on breast cancer diagnosis using machine learning, it was discovered that current research was only concerned with achieving peak accuracy in 1-2 datasets, without accounting for variation found between different sets of mammograms, reducing transferability of the model, thus failing to accurately diagnose unseen data.

How does this information loss (variability) across different mammogram images affect diagnostic accuracy?

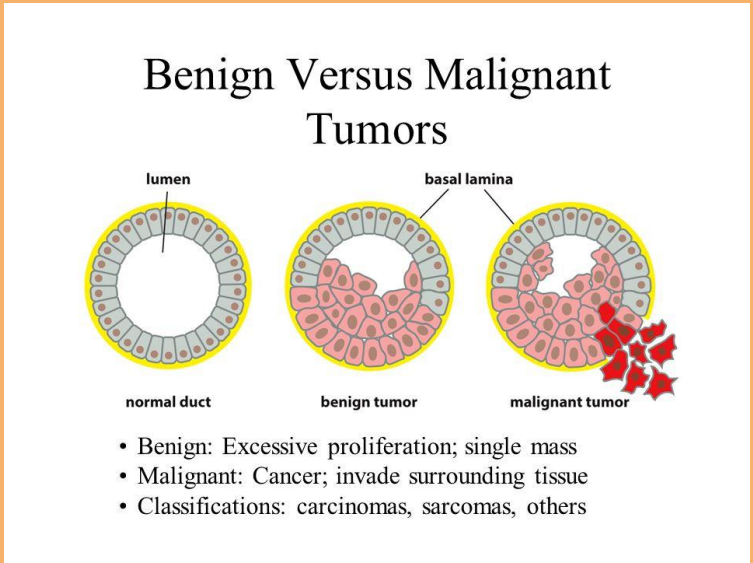
Can current radiology accuracy be improved through deep learning by simulating variation between acquisition methods of mammograms?

Hypothesis

1) Using mammogram data, it is hypothesized that simulation of different image variants through artificial augmentation will produce good predictive accuracies
2) It is also hypothesized that producing large samples of data using these augmentations will allow for better model generalizability, improving diagnostic accuracy

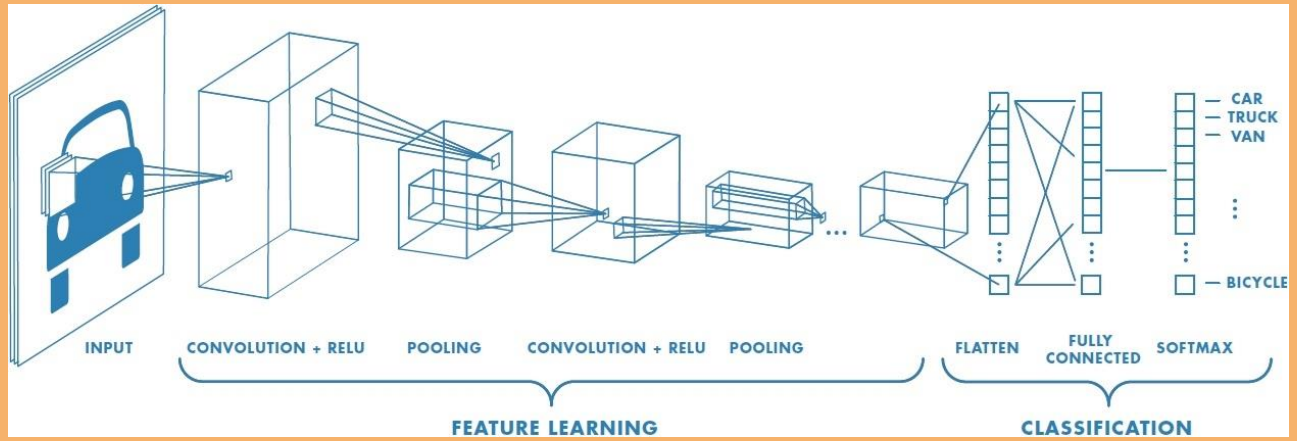
Data

- CBIS-DDSM dataset, benign and malignant tags
- 3567 total images, equal split
- BIRAD tags also present
  - 1 - lowest level of malignancy
  - 6 - highest level of malignancy
- This work focused on binary classification
  - Sufficient to do binary classification as current prognostic methods only rely on nature of tumour treatment



Convolutional Neural Network (CNN)

- Convolutional neural network diagnosed benign and malignant mammograms
- Backpropagation
- SGD
- Layers
  - Convolution
  - Pooling
  - Dropout
  - Dense
  - Fully Connected



[3]

https://becominghuman.ai/extract-a-feature-vector-for-any-image-with-pytorch-9717561d1d4c

Materials

- Computer
- Linux Server running Ubuntu 16.04
- TitanX with 12GB RAM - GPU
- CBIS-DDSM dataset
- Keras API
  - open source neural network API
  - enables fast experimentation with deep neural networks
- Python
  - Pre-process all data
  - One-Hot-Encode Metadata
  - Build Neural Network
  - Generate Accuracy Statistics

Keras

Methods

- Resize images to 25 x 25, 50x50, 100x100, 200x200, 400x400, 800x800 png
- One Hot Encoding of Metadata
- Image Augmentations
  - Resolution changes
  - Affine transformations (rotations, zooms, width shifts)
  - Contrast changes (ratios of 1,2,3,4,5)

Benign	0	1
Malignant	1	0

Example of one-hot encoding

Methods cont.

- Resolution
  - Bash script used to resize images into different resolutions
- Contrast
  - rounding algorithm used to change contrast
  - floor function for lower value pixels, ceiling for higher value pixels
  - level of contrast indicates level of grayscale (e.g. higher means only black and white)
- Affine Transformations
  - Keras API (DataGen) used to augment images further
  - Rotation: 90º
  - Zoom: 30%
  - Width Shift: 10%

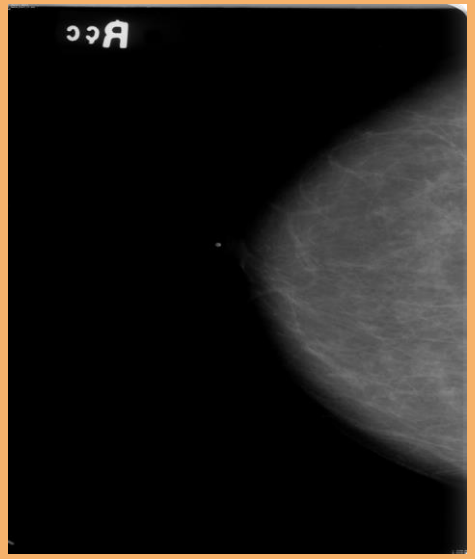
Cross-validation

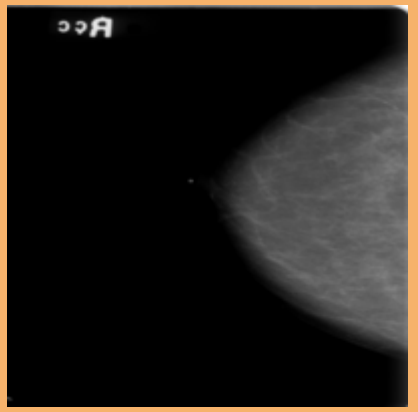
- Two datasets with labels were used to test the model
- Training: correct matched labels, used to discover relationships
- Testing: no labels, model outputs probabilities of labels
- Train: 80%, Test: 20%
- Trained for 72 epochs
  - Accuracy
  - Validation Accuracy
  - Loss
  - Validation Loss
- Highest val accuracy selected for model
- Validation accuracy = (TP + TN) / n

Methods

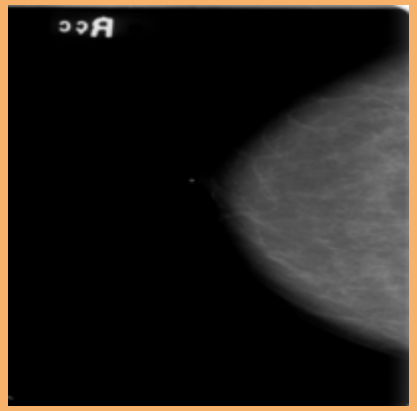
1) **Gathering of Breast Cancer Data:** All data for this project was taken from the CBIS-DDSM dataset. Using the curated images from the dataset containing both Mediolateral Oblique (MLO) and Cranial Caudal (CC) views, a dataset of 3,567 mammograms was garnered. Images were roughly split even between benign and malignant samples, and images and metadata were converted into comma-separated values (CSVs) and stored onto a virtual machine running Linux for further parsing
2) **Image Pre-Processing:** After analyzing upwards of 30 different mammogram datasets, the most common variables were selected for augmentation. A number of different transformations was applied to the data. Bash script was used to change the resolution of the images, and a rounding algorithm was used to vary the contrasts of the images. Keras' ImageDataGenerator library was used to generate different tensor batches for affine transformations
3) **Building Neural Network:** This neural network structure is loosely based on a combination of AlexNet and GoogleNet, two very popular CNN structures in deep learning works. Activation functions and amount of layers were achieved through extensive trial-and-error and research to increase non-linear properties of the model to garner best possible accuracy
4) **Running the Data:** Images were converted into matrices with grayscale pixels ranging from 0-255. The algorithm applied one of the pre-processing functions to these matrices, before metadata was one-hot-encoded. This model used 10-fold cross-validation to split data into 80-20 train/test split. Random batches of 128 images were run through the CNN for 72 epochs (iteration of all samples), outputting accuracy statistics
5) **Tuning the Model:** The highest validation accuracy was selected as final accuracy as this scores the transferability of the model. Based on this accuracy, the model would be fine tuned using feature selection to achieve best accuracy possible. Epoch graphs and confusion matrices were produced to determine extent of overfitting.

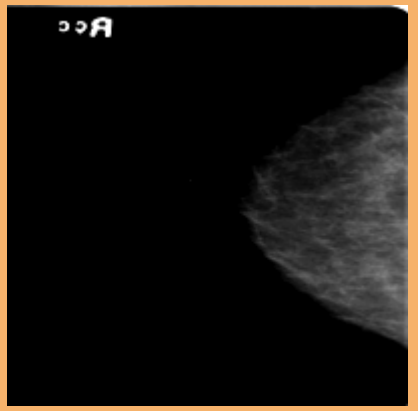
Resolution:



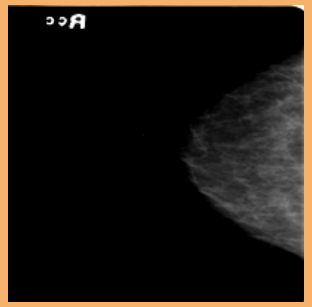


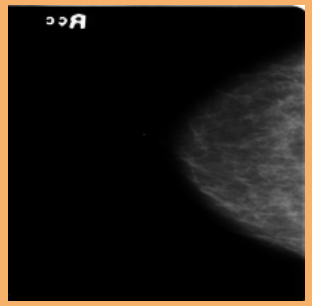
Contrast:

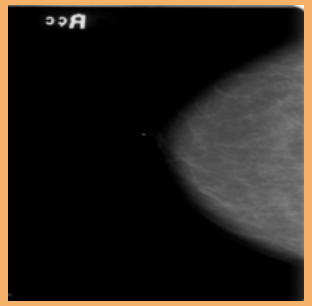


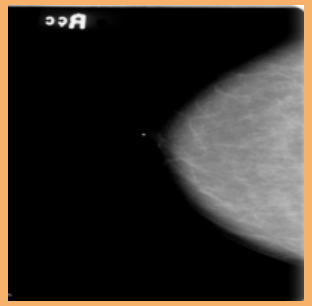


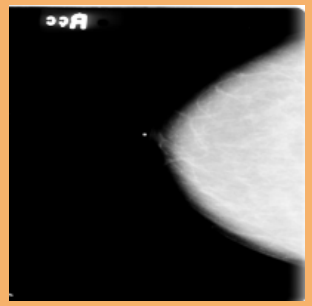
Contrast Transformations







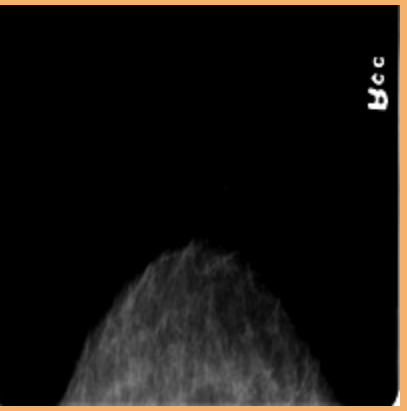




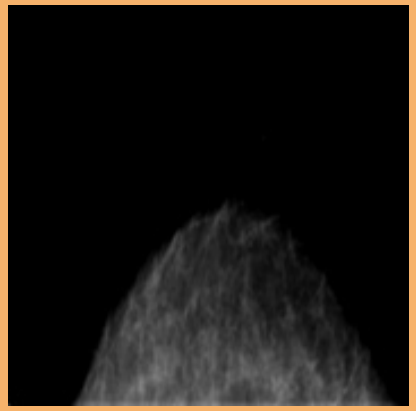
1.0 2.0 3.0 4.0 5.0

ImageDataGenerator Augmentations

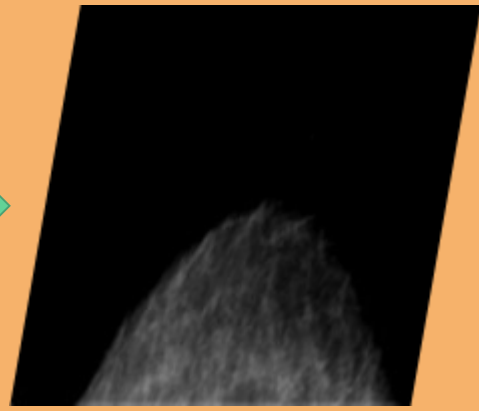
Rotation



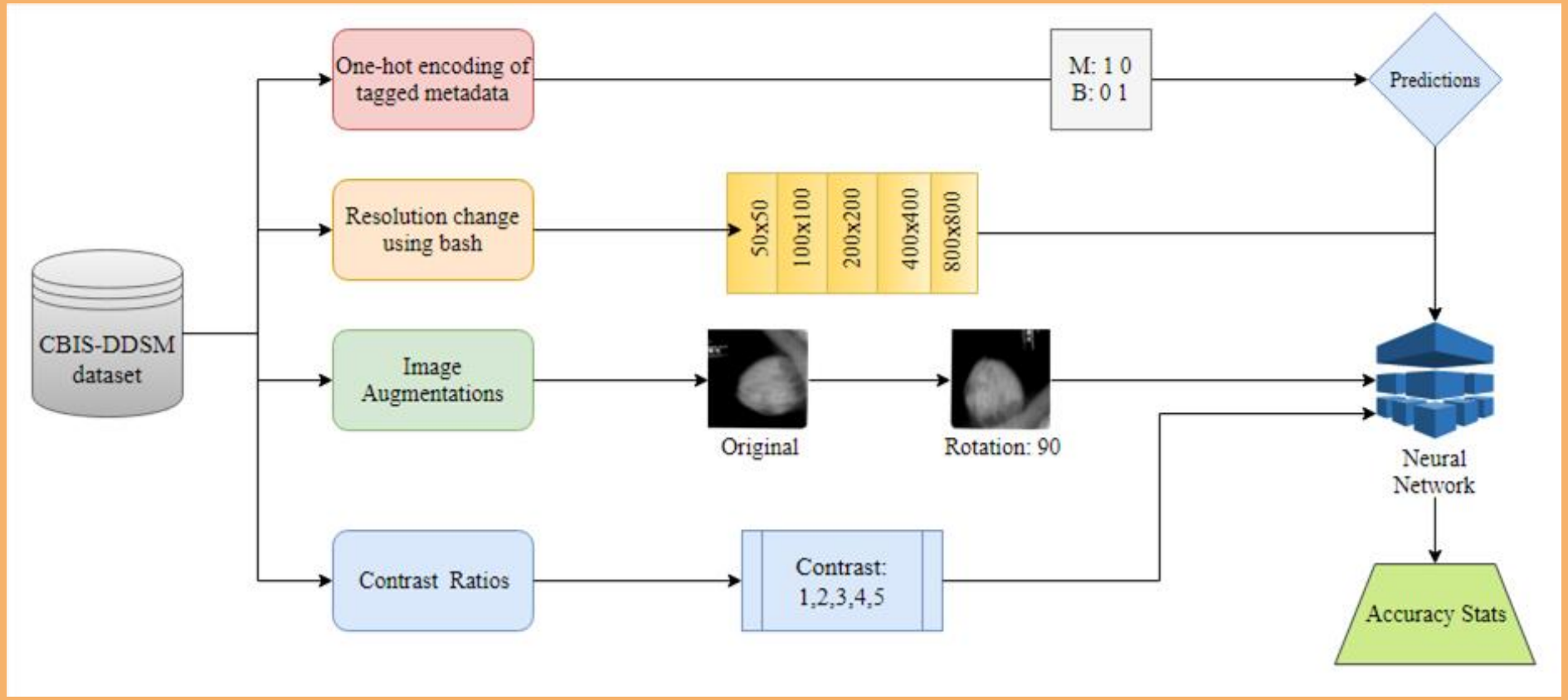
Zoom:



Width Shift:



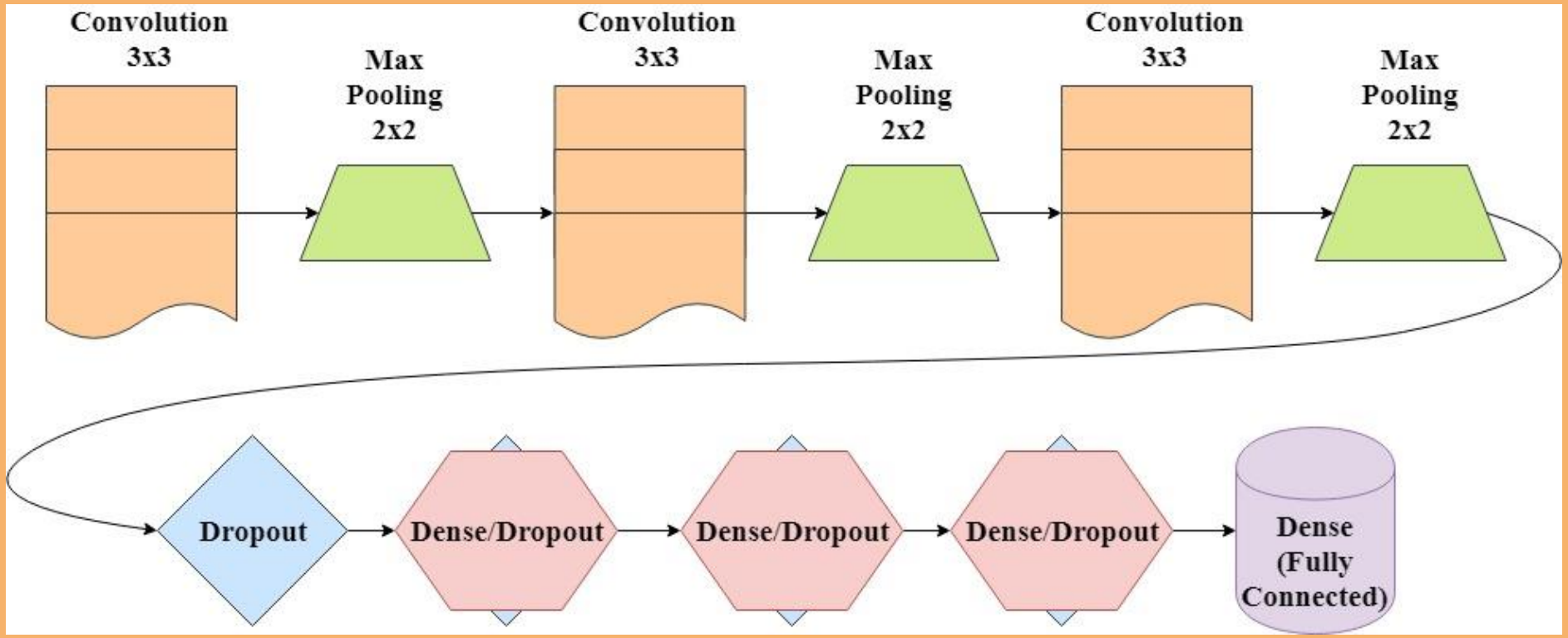
Methodology Flowchart



CNN Structure used in this project

Layer	Kernel Size	Repetitions	Activation
convolution	3x3	x 3	relu
max pooling	2x2	x 1	
convolution	3x3	x 3	relu
max pooling	2x2	x 1	
convolution	3x3	x 3	relu
max pooling	2x2	x 1	
Layer	**Units**	**Dropout Value**	**Activation**
Dropout		0.25	
Dense	128	0.5	relu
Dense	512	0.5	relu
Dense	1024	0.5	relu
Dense (FC)	2		sigmoid

CNN Structure Flowchart



Code

```
import tensorflow as tf
from keras.backend.tensorflow_backend import set_session
config = tf.ConfigProto()
config.gpu_options.per_process_gpu_memory_fraction = .95
config.gpu_options.visible_device_list = "0"
set_session(tf.Session(config=config))

from keras.preprocessing.image import ImageDataGenerator

from keras import backend as K
K.set_image_dim_ordering('th')  # a lot of old examples of CNNs
# have the format (colors, rows, columns)  (1,28,28)
#  tensorflow expects (28,28,1)

img_rows = 50
img_cols = 50

import glob
allImageFileNames = glob.glob("/media/hdd0/unraiddisk2/cbis-ddsm/DOI/*/*/*/*50*50.png")
import numpy as np
import pandas as pd
calcification = pd.read_csv("/media/hdd0/unraiddisk2/cbis-ddsm/calcification_50.csv").values
masses = pd.read_csv("/media/hdd0/unraiddisk2/cbis-ddsm/masses_50.csv").values

def makeNumericalFromString(y):
 y_num = np.zeros(y.shape[0])
 for i in range(y.shape[0]):
   if y[i,11] == "MALIGNANT":
     y_num [i] = 0
   elif y[i,11] == "BENIGN":
     y_num [i] = 1
   else:
     y_num [i] = 1
 return y_num

y_calc_num = makeNumericalFromString(calcification)
y_mass_num = makeNumericalFromString(masses)

y_num = np.append(y_calc_num,y_mass_num)
from keras.utils import np_utils
y = np_utils.to_categorical(y_num )

from scipy.misc import imread, imsave, imresize
```

Code-2

```
      y_num [i] = 1
    else:
      y_num [i] = 1
  return y_num

y_calc_num = makeNumericalFromString(calcification)
y_mass_num = makeNumericalFromString(masses)

y_num = np.append(y_calc_num,y_mass_num)
from keras.utils import np_utils
y = np_utils.to_categorical(y_num )

from scipy.misc import imread, imsave, imresize

norm = 0;
roi = 1;
crop = 2;

a = np.zeros( (calcification.shape[0] + masses.shape[0] , img_rows, img_cols))
a_index = 0
for i in range(calcification.shape[0]):
 filename = calcification[i,norm]
 print str(a_index)+" Loading "+filename
 a[a_index] = imresize( imread(filename, True) , (img_rows, img_cols) ).astype('float32') / 255
 a_index += 1

for i in range(masses.shape[0]):
 filename = masses[i,norm]
 print str(a_index)+" Loading "+filename
 a[a_index] = imresize( imread(filename, True) , (img_rows, img_cols) ).astype('float32') / 255
 a_index += 1

#comment out if doing contrast
a = a[:,np.newaxis,:,:]
#contrast


nb_classes = 2
NB_CLASS = 2

from keras.models import Sequential,model_from_json
from keras.layers import Dense, Activation, Flatten, Dropout, Convolution2D, MaxPooling2D

```

Code-3

```
from keras.layers import Dense, Activation, Flatten, Dropout, Convolution2D, MaxPooling2D


model = Sequential()
nb_filters = 8  # number of convolutional filters to use
nb_pool = 2     # size of pooling area for max pooling
nb_conv = 3     # convolution kernel size

model.add(Convolution2D(nb_filters, nb_conv, nb_conv, border_mode='valid', input_shape=(1, img_rows, img_cols), activation='relu'))
model.add(Convolution2D(nb_filters, nb_conv, nb_conv, activation='relu'))
model.add(MaxPooling2D(pool_size=(nb_pool, nb_pool)))

model.add(Convolution2D(nb_filters, nb_conv, nb_conv, activation='relu'))
model.add(MaxPooling2D(pool_size=(nb_pool, nb_pool)))
model.add(Convolution2D(nb_filters, nb_conv, nb_conv, activation='relu'))
model.add(MaxPooling2D(pool_size=(nb_pool, nb_pool)))

model.add(Dropout(0.25))
model.add(Flatten())
model.add(Dense(128, activation='relu'))
model.add(Dropout(0.25))
model.add(Dense(512, activation='relu'))
model.add(Dropout(0.25))
model.add(Dense(1024, activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(nb_classes, activation='softmax'))
model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])

from sklearn.cross_validation import train_test_split
X_train,X_val,Y_train,Y_val = train_test_split(a,y,test_size=0.2)
#model.fit(a,y,batch_size = 128, nb_epoch = 32, verbose = 1)
#model.fit(X_train, Y_train, validation_data=(X_val, Y_val), batch_size=32, nb_epoch=30, verbose=1)

import matplotlib.pyplot as plt

history50 = model.fit(X_train, Y_train, validation_data=(X_val, Y_val), batch_size=8, nb_epoch=72, verbose=1)

```

Results-Resolution

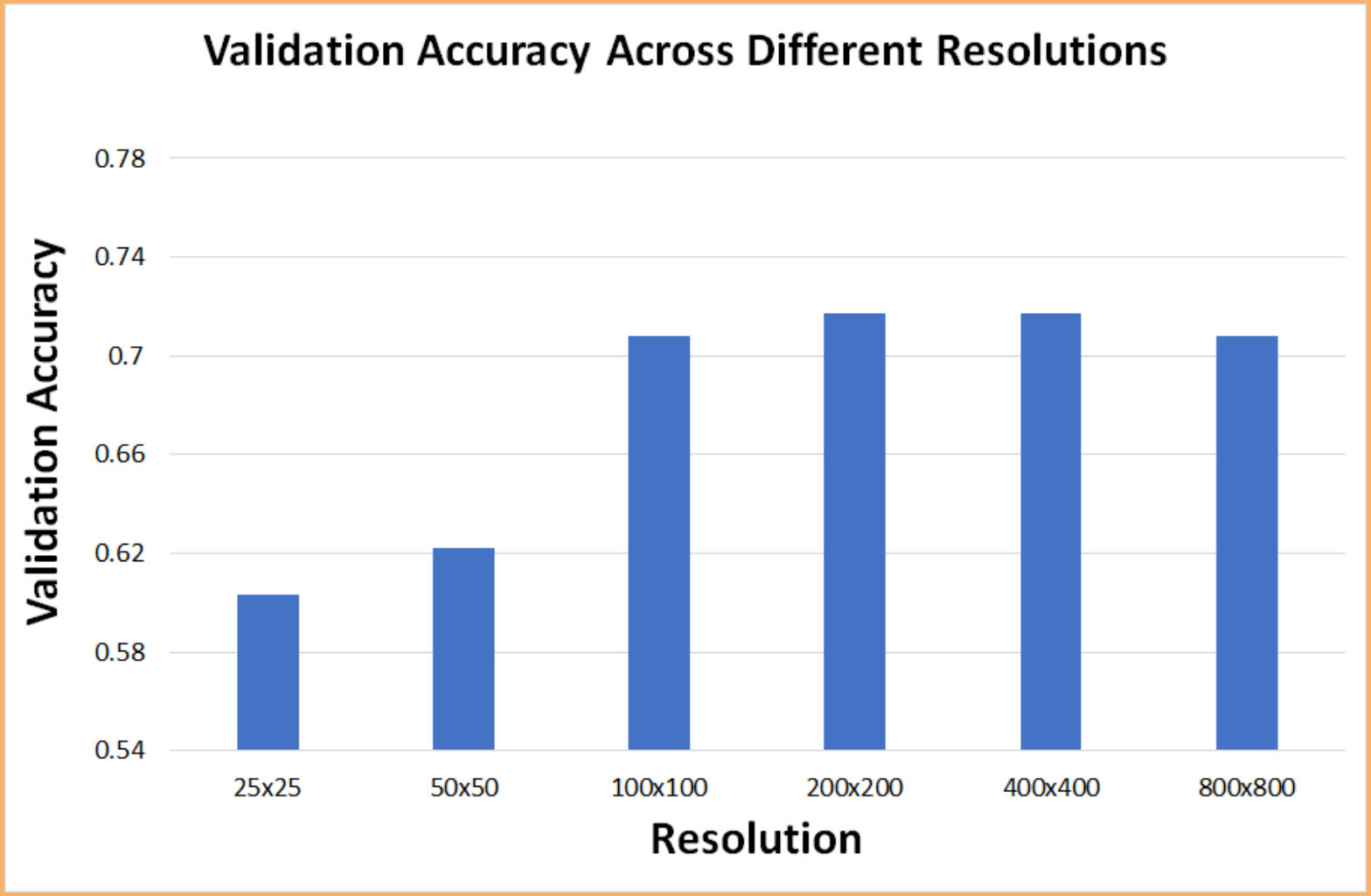
Resolution	Validation Accuracy
25x25	0.603
50x50	0.622
100x100	0.708
200x200	**0.717**
400x400	0.717
800x800	0.708

Validation Accuracy Table-Contrast and Datagen constant

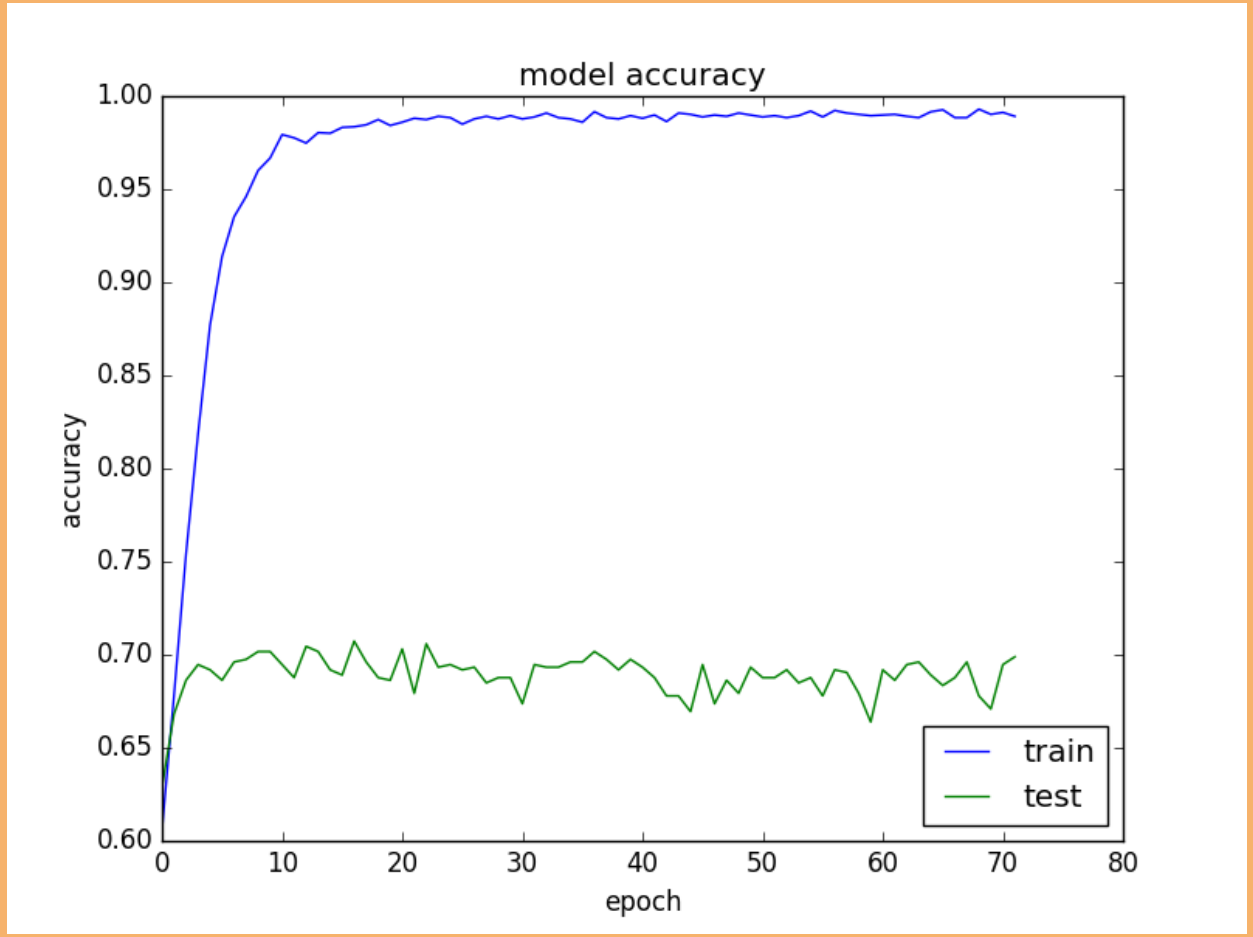
Confusion Matrix - 200x200 resolution

n = 714	Malignant	Benign
Predicted Malignant	164 (64.8%)	**126 (27.3%)**
Predicted Benign	89 (33.2%)	335 (72.7%)

0.27 false positive rate


Validation Accuracy Across Different Resolutions
Validation Accuracy
0.78
0.74
0.7
0.66
0.62
0.58
0.54
25x25
50x50
100x100
200x200
400x400
800x800
Resolution

Resolution Accuracy Chart



200x200 resolution: Drops in validation accuracy indicate overfitting

Results-Contrast

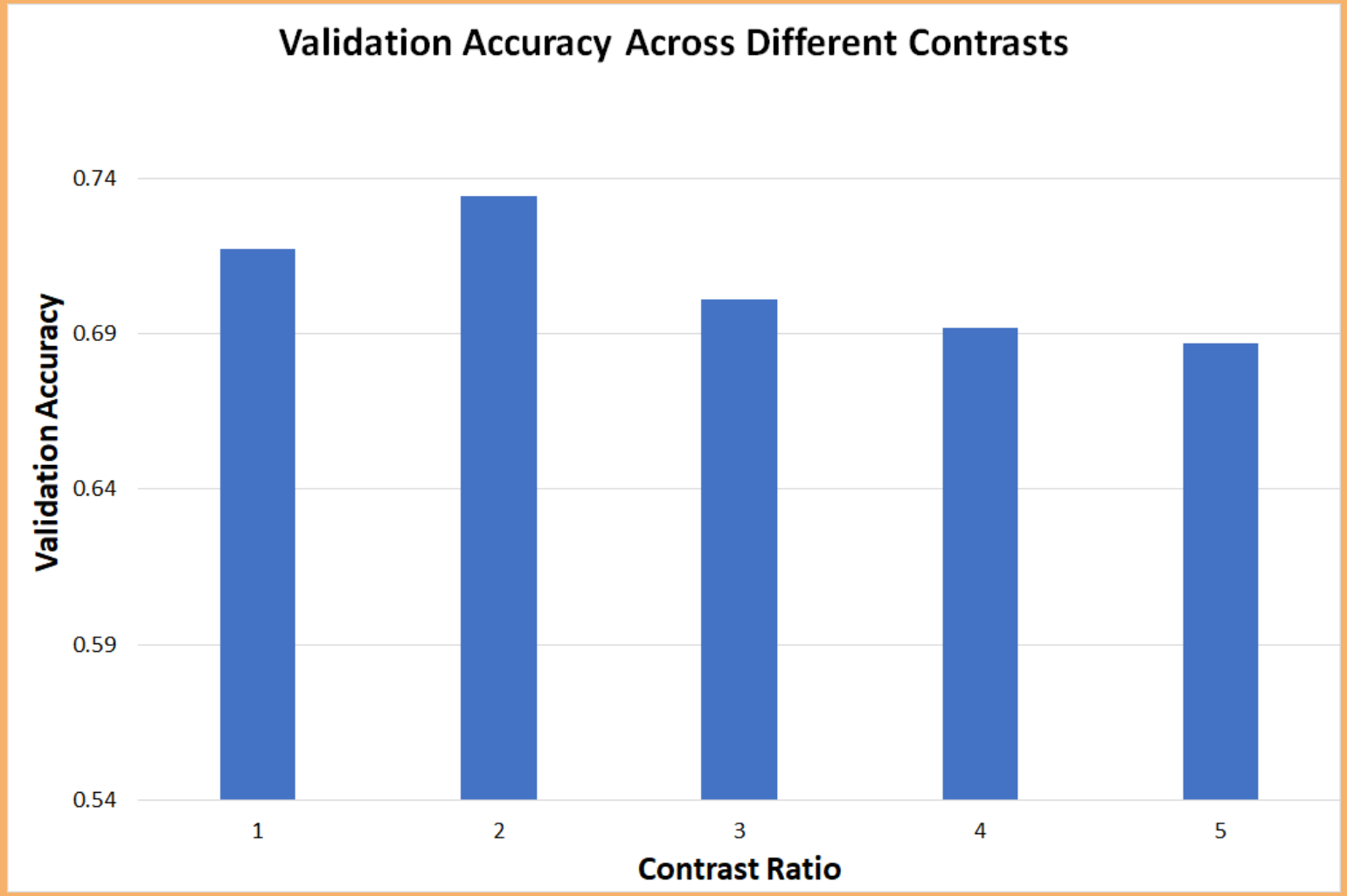
Contrast Ratio	Validation Accuracy
1.0	0.717
2.0	**0.734**
3.0	0.701
4.0	0.692
5.0	0.687

Validation Accuracy Table

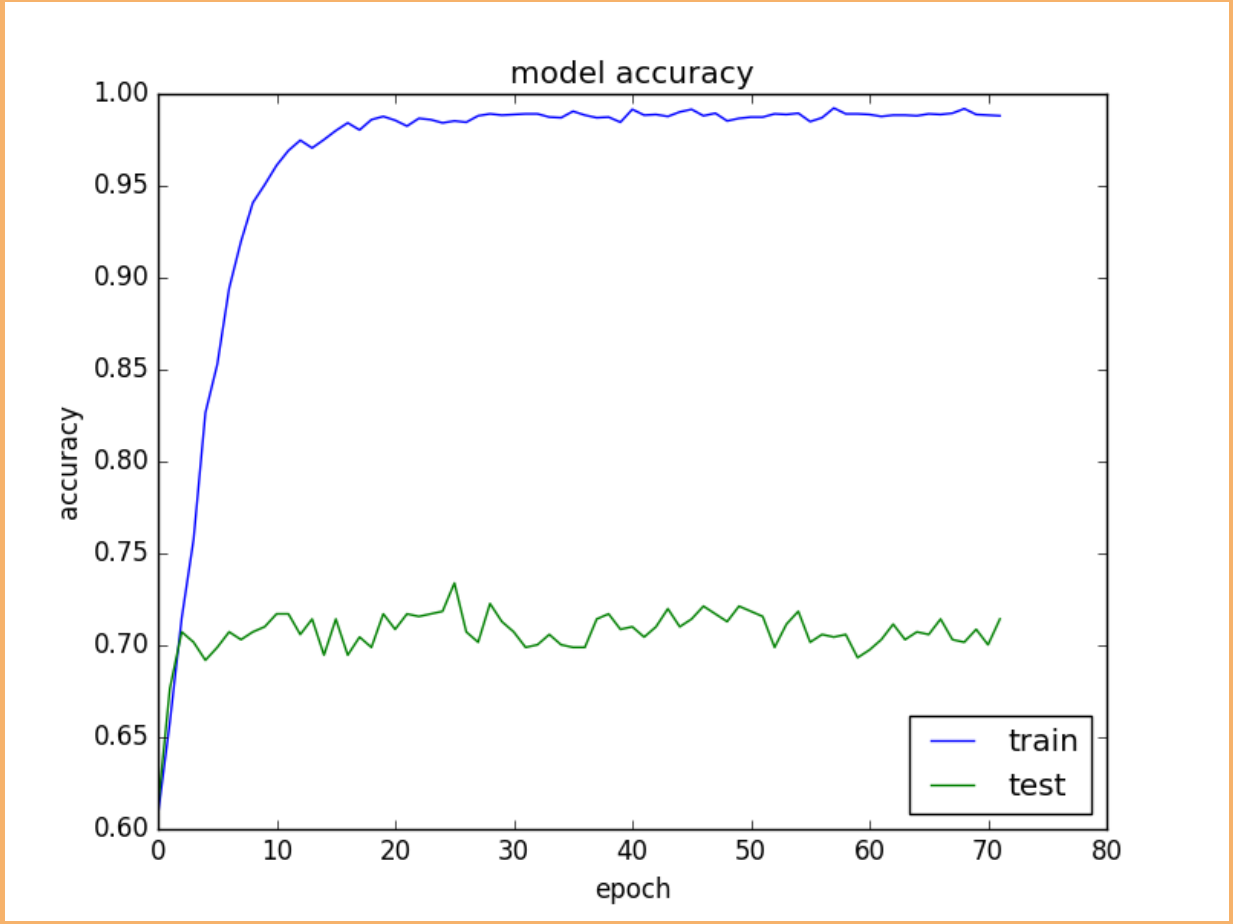
Confusion Matrix - 2.0 Contrast Ratio

n = 714	Malignant	Benign
Predicted Malignant	165 (67.9%)	**117 (24.8%)**
Predicted Benign	78 (32.1%)	354 (75.2%)

0.25 false positive rate


Validation Accuracy Across Different Contrasts
0.74
0.69
0.64
0.59
0.54
Validation Accuracy
1
2
3
4
5
Contrast Ratio

Contrast Accuracy Chart



2.0 contrast ratio: Better validation accuracy but overfitting present due to lack of samples.

Results: DataGenerator

Transformations	Composite Validation Accuracy
Rotation: 90º Zoom: 30% Width Shift: 10%	**0.824**

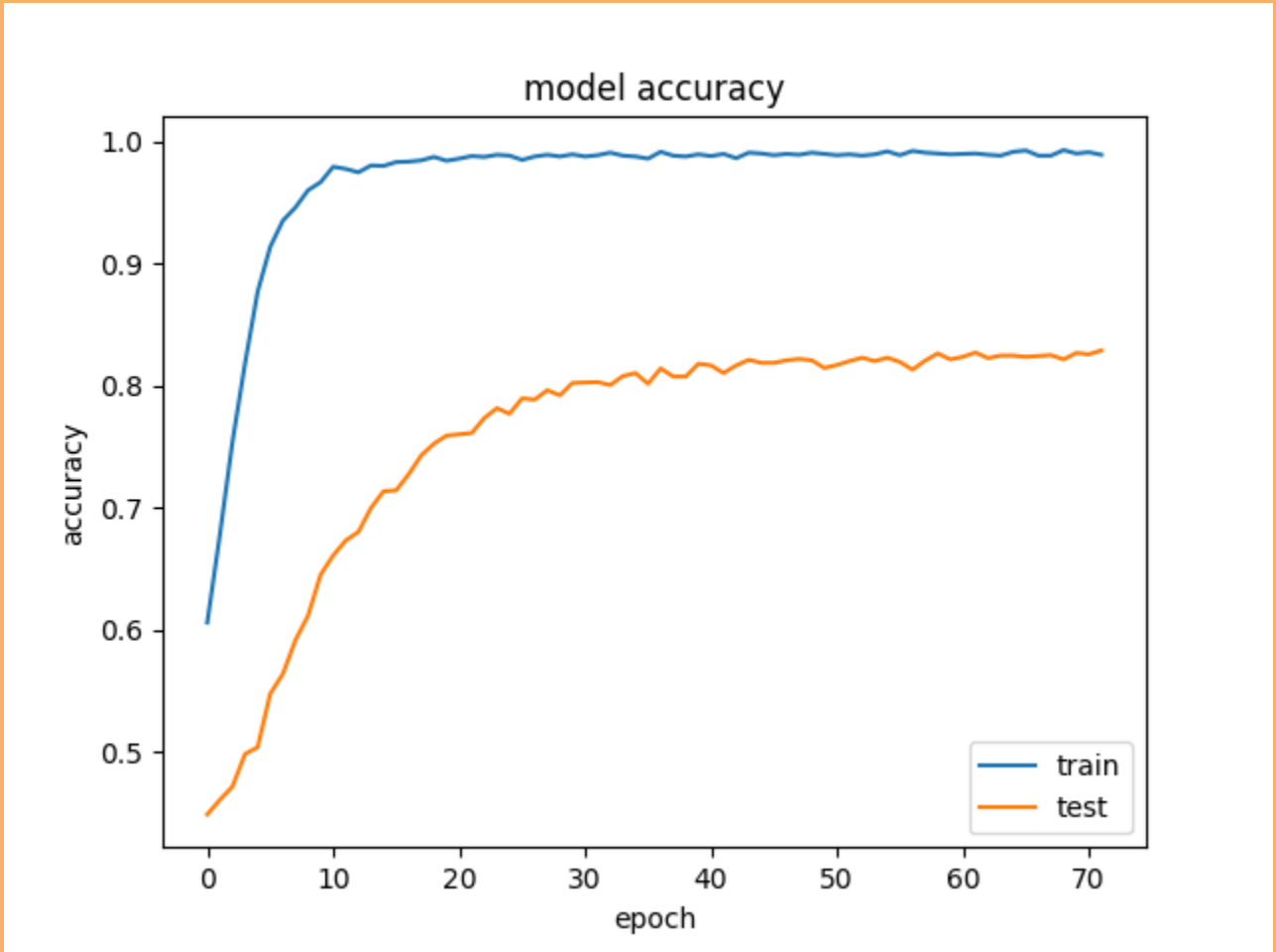
Validation Accuracy Table: Composite Accuracy with optimal contrast and resolution

Confusion Matrix

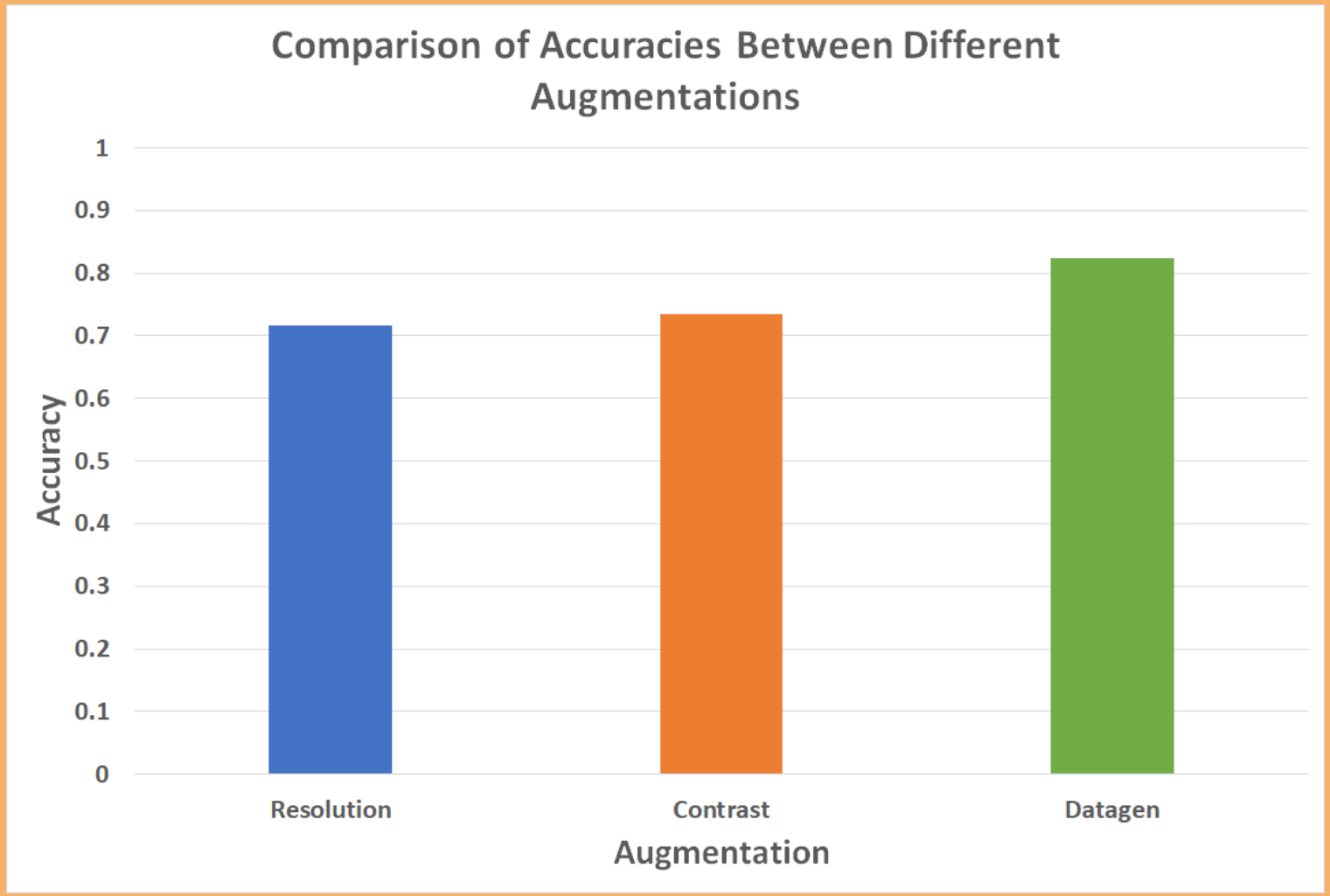
n = 2856	Malignant	Benign
Predicted Malignant	863 (81.1%)	**311 (17.4%)**
Predicted Benign	201 (18.9%)	1481 (82.6%)

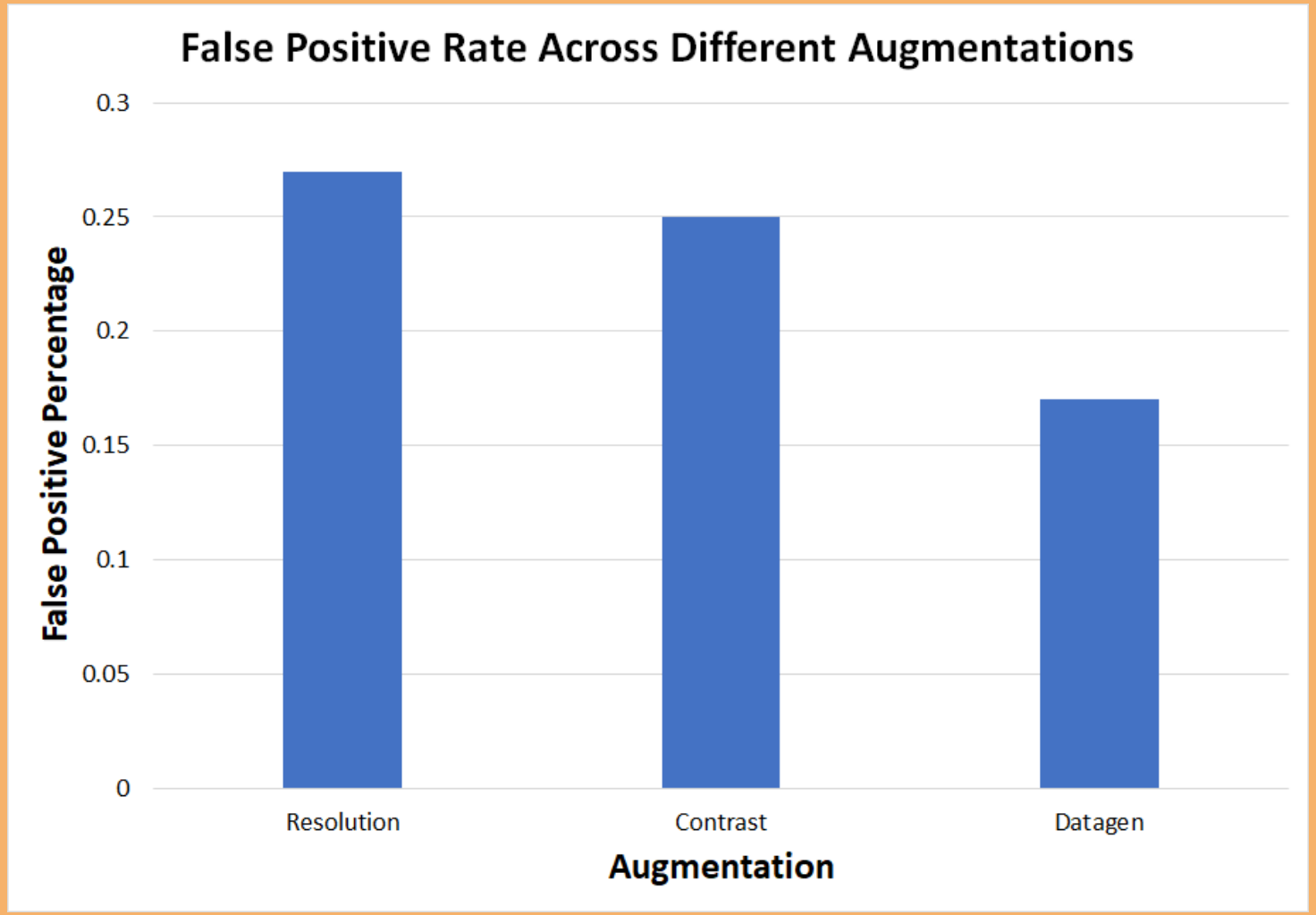
0.17 false positive rate

DataGen Accuracy Chart



Composite accuracy: increase in sample size drastically reduces overfitting


Comparison of Accuracies Between Different Augmentations
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
Accuracy
Resolution
Contrast
Datagen
Augmentation


False Positive Rate Across Different Augmentations
0.3
0.25
0.2
0.15
0.1
0.05
0
False Positive Percentage
Resolution
Contrast
Datagen
Augmentation

Analysis

- This model performed most effectively on 200x200 resolution
- Since 714 images were validated upon, overfitting was present in first two augmentation batches
- Fitting on datagen performed the best (no overfitting due to lack of sharp drops) because sufficient samples produced to cover variation in images
- Contrast potentially highlighted cancerous areas in breast, reducing noise in images, allowing CNN to fit better
- First two augmentation procedures took ~1 hr to run through 72 epochs, Datagen fitting took ~3 hrs to run, faster than reported in literature
- Augmentations accounted for most types of variation found in mammogram images, contributing to success of neural network

Conclusions

- **Demonstration of Increased Accuracy after Introducing more samples of augmented data**
  - The dataGenerator proved most effective as the neural network was able to train on multiple image variants found across mammography datasets. A large increase in accuracy (~10%) was observed after adding additional samples of data. This classifier performed very well despite limited amount of data being available on public datasets
- **Improved Generalizability of Model while Reducing Model Runtime**
  - The most important contribution of this work - being able to generalize to unseen data is a prevalent issue in deep learning diagnosis of breast cancer. This robust CNN structure did not overfit on validation data while maintaining a consistent accuracy of 82%. The reduction in runtime contributes to the efficiency of this neural network in diagnosing new data
- **Reduction of False Positives from Current Prognostic Methods**
  - The final model produced a false positive rate of 17%, 30 percent lower than average radiologists performance. This is the first project in the field that utilizes actual image variants in order to reduce false positives, eliminating anxiety for patients who receive this diagnosis

Applications

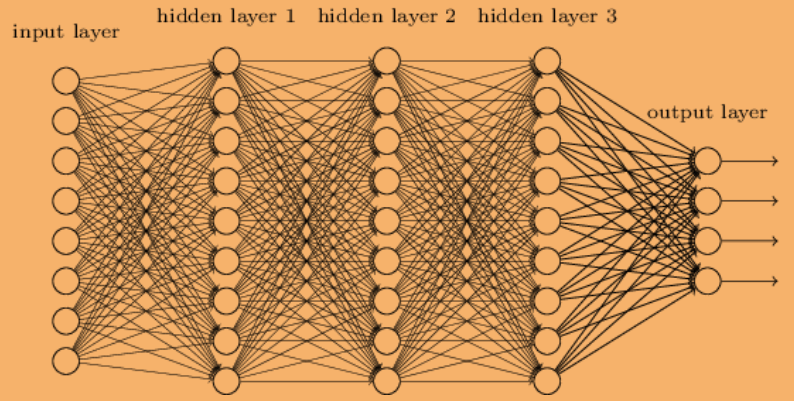
With this research scientists can:

- Better design mammogram acquisition methods to produce better results from deep learning algorithms using augmentations provided
- Use CNN's to better diagnose breast cancer, reducing mental stress and anxiety for patients as well as save money on performing unnecessary callback tests for false positives

This research has the potential to surpass human-caused error in breast cancer diagnosis, assisting radiologists to better assess their prediction

Future Work

- Producing saliency maps and class activation maps
  - Allows for better assessment of CNN's predictions
  - Highlight important regions responsible for predictions
  - Can also detect region of interest and severity of tumor
- Incorporate Deeper CNN structures
  - GoogleNet, AlexNet, etc.
  - Deeper CNN → transfer learning advantage
  - Acquisition of more data required to run deeper neural networks



Bibliography

[1]: “U.S. Breast Cancer Statistics.” *Breastcancer.org*, www.breastcancer.org/symptoms/understand_bc/statistics.

[2]: Ap. “The High Cost of Breast Cancer ‘False Positives.’” *CBS News*, CBS Interactive, 7 Apr. 2015, www.cbsnews.com/news/the-cost-of-breast-cancer-false-positives/.

[3]: Litjens, Geert, et al. “A Survey on Deep Learning in Medical Image Analysis.” *[1702.05747] A Survey on Deep Learning in Medical Image Analysis*, 4 June 2017,

[4]: “ImageNet Classification with Deep Convolutional Neural Networks.” *ACM Digital Library*, Curran Associates Inc., dl.acm.org/citation.cfm?id=2999257.

[5]: “Using Convolutional Neural Networks for Image Recognition.” *Embedded Vision Alliance*, www.embedded-vision.com/platinum-members/cadence/embedded-vision-training/documents/pages/neuralnetworksimagerecognition.

Bibliography

[6]: Wolberg, MD William H. “Computerized Breast Cancer Diagnosis and Prognosis From Fine-Needle Aspirates.” *Archives of Surgery*, American Medical Association, 1 May 1995, jamanetwork.com/journals/jamasurgery/article-abstract/596210?redirect=true.

[7]: Lévy, Daniel, and Arzav Jain. “Breast Mass Classification from Mammograms Using Deep Convolutional Neural Networks.” *[1612.00542] Breast Mass Classification from Mammograms Using Deep Convolutional Neural Networks*, 2 Dec. 2016, arxiv.org/abs/1612.00542.

[8]: “Mammography and Breast Imaging Resources.” *American College of Radiology | American College of Radiology*, www.acr.org/Clinical-Resources/Breast-Imaging-Resources.

[9]: Elmore, Joann G., et al. *Journal of the National Cancer Institute*, U.S. National Library of Medicine, 18 Sept. 2002, www.ncbi.nlm.nih.gov/pmc/articles/PMC3142994/.

[10]: Nover, Adam B., et al. “Modern Breast Cancer Detection: A Technological Review.” *International Journal of Biomedical Imaging*, Hindawi, 28 Dec. 2009, www.hindawi.com/journals/ijbi/2009/902326/.