Gautham Meenakshisundaram (44468423)
Macquarie University - Session 1, 2019

The Effect of Resolution Transformations on the Classification of Chest X-Rays

1 Introduction

It is of common knowledge that information will be lost when images are rescaled from higher resolution sizes to lower resolution sizes. In order to test this hypothesis, classification models using both conventional and deep learning models are to be adopted on a chest X-ray dataset for various image sizes. Amongst the multiple performance metrics such as Accuracy, Precision, Recall and F1 score available to evaluate the classification models, Accuracy providing the ratio of correctly predicted observations to total number of observations seems to be the obvious performance measure to be considered.

2 Related Work

A paper [1] on the labelling, classification and localization of diseases in the chest X-ray dataset was published along with the chest X-ray images.

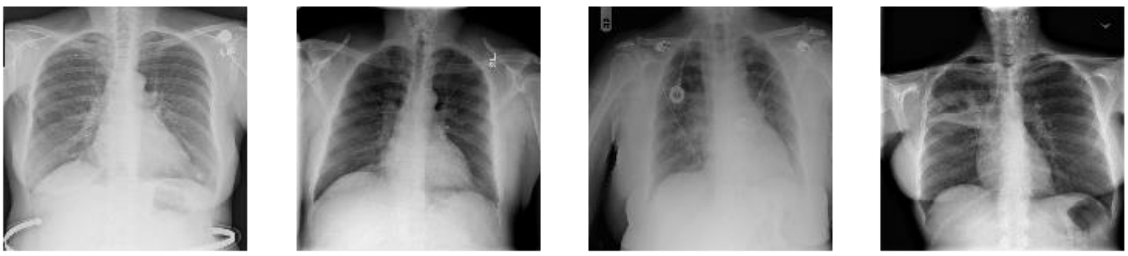
3 Dataset

On 27th September of 2017, the National Institutes of Health (NIH) in United States of America released a new database 'ChestX-ray14' containing 112,120 frontal view chest X-ray images in the public domain to promote scientific research. These X-ray images of 32,717 unique patients were collected from 1992 to 2015 and stored in the hospital's Picture Archiving and Communication System (PACS). A 5 percent sample of the original dataset – 5,606 images – available on Kaggle[1] is considered for our analysis, taking into account the memory required for handling image classification.

3.1 Images

The X-ray images stored in DICOM (Digital Imaging and Communications in Medicine) header files are extracted and rescaled to 1024 x 1024 bitmap images while ensuring no significant loss of intensity ranges and detail contents.

Figure 1: Random X-Ray Images



----

[1]https://www.kaggle.com/nih-chest-xrays/sample

3.2 Labels

As per the inputs from radiologists, a list comprising of 14 frequently observed and diagnosed pathologies - Atelectasis, Cardiomegaly, Effusion, Infiltration, Mass, Nodule, Pneumonia, Pneumathorax, Consolidation, Edema, Emphysema, Fibrosis, Pleural Thickening and Hernia was shortlisted first and then searched across associated radiological reports. A range of Natural Language Processing (NLP) techniques were employed to detect the pathology keywords in the radiological reports while removing negation and uncertainty. DNorm used for disease recognition and normalization, and MetaMap used for the detection of biological concepts were the two machine learning tools employed in the mining of radiological reports. The reports were split and tokenized using NLTK (Natural Language ToolKit) and parsed through the Bllip parser using David McCloskys biomedical model. As a result, 51,708 images were labelled with single or multiple pathologies from the 14 pathologies and the remaining 60,412 images were labelled with a ‘No Finding’ label.

3.3 Evaluation

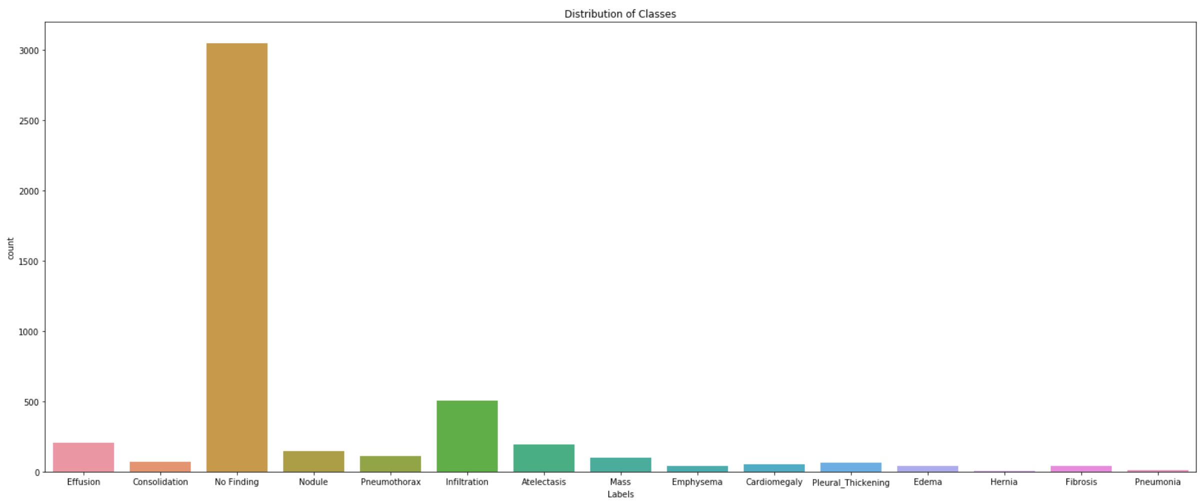
The reports from OpenI which is also a chest X-ray image dataset with 4,143 images, was considered as a gold standard for evaluating the methods. Furthermore, two annotators were asked to mark the 14 pathologies individually across a randomly selected sample of 900 reports.

4 Methods

4.1 Class Distribution

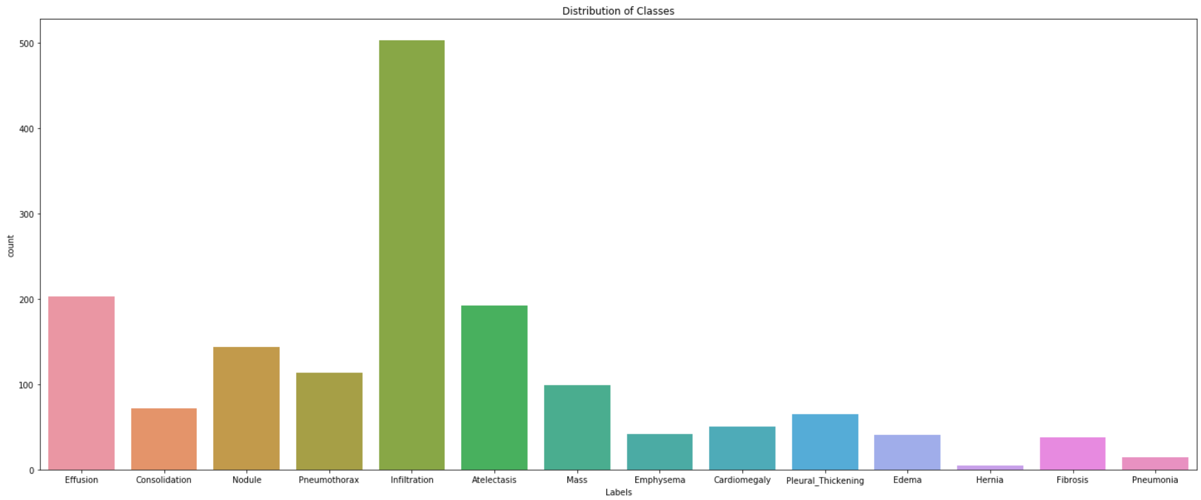
The dataset consists of both images with single pathology and multiple pathologies. A single label approach was undertaken as the multiple label approach will be complicated with the number of unique classes being as high as 244. After excluding images with multiple pathologies, the total number of images was 4,626. Upon examining the distribution of classes, we find that the dataset is skewed towards the ‘No Finding’ label with differences in thousands, when compared with other labels.

Figure 2: Distribution of Classes



When the 'No Finding' label is also excluded, the classes are distributed evenly to some extent except for the 'Infiltration' label which at least has differences in hundreds, when compared to other labels. Finally, a reasonable size of 1,582 images is considered for our analysis.

Figure 3: Distribution of Classes



4.2 Image Sizes

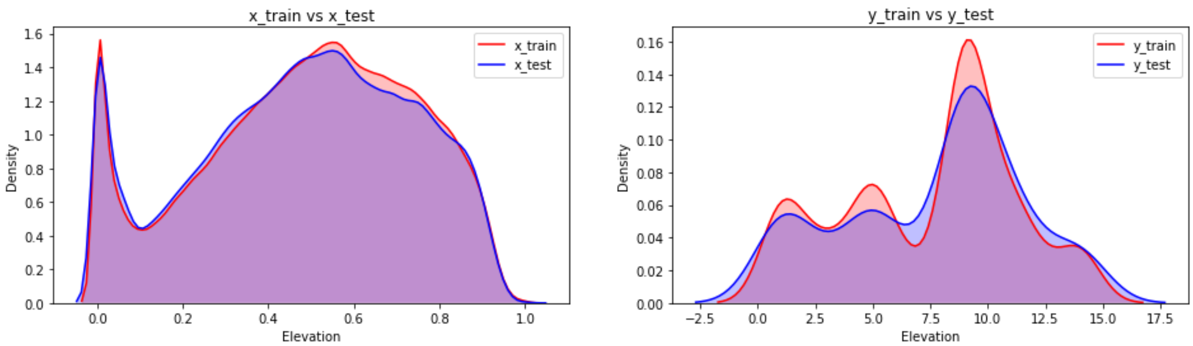
Initially, the common image sizes such as 16x16, 32x32, 64x64, 128x128, 256x256 were considered but then the minimum input size for our deep learning model was from 46x46. In order to perform a fair comparison of classification models from both conventional and deep learning methods, the following image sizes were finalised - 46x46, 64x64, 128x128, 224x224, and 256x256. The image size 224x224 was also considered as it is the default input size for most of the pre-trained models in deep learning. However, image sizes beyond 256x256 were not considered due to the memory required for handling large resolution images.

4.3 Pre-processing

The images are resized and saved as float values in numpy arrays for further pre-processing. The labels are saved as float values in numpy arrays for conventional methods and are converted to one hot vectors for Convolutional Neural Networks (CNN). This one hot encoder performs binarization of the labels which are category values and include them as a feature to train the CNN model. The image array is normalized to values ranging between 0 to 1 to nullify the effect of any scaled values. Furthermore, the shape of the image array is converted from 4 dimensions to 2 dimensions for the input in conventional methods.

The dataset is then split into standard training (80%) and test (20%) dataset sizes for the analysis which results in 1,265 images for training and 317 images for testing. The distribution of training and test datasets in the density plots seems to be largely similar.

Figure 4: Density plots of Training and Test datasets



4.4 Analysis

In addition to a deep learning model in Convolutional Neural Network, conventional methods such as K-Nearest Neighbors, Support Vector Machine, Naïve Bayes, Decision Tree and Random Forest are employed to determine the effect of resolution transformations in the classification of X-ray images.

4.4.1 K-Nearest Neighbors

K-Nearest Neighbors (KNN) is a supervised learning algorithm which is very simple to implement yet produces efficient results. The algorithm assumes that similar data points are in proximity using distance functions which is the logic it applies during classification tasks. KNN is suitable for classification tasks and even multi-class problems. Moreover, it only has one hyper parameter to modify and provides multiple distance functions in Euclidean Distance, Hamming Distance, Manhattan Distance, Minkowski Distance. It is also time consuming to find the optimal number of neighbors, n=1 for this analysis.

4.4.2 Support Vector Machine

Support Vector Machine (SVM) is a supervised learning algorithm which separates classes by drawing a boundary between them using the best possible hyperplane. SVM provides good accuracy but consumes a lot of time while training large datasets such as in our case.

4.4.3 Naïve Bayes

Naïve Bayes classifier is a probabilistic model based on Bayes theorem, used for classification tasks. According to Bayes theorem, the probability of A occurring depends on whether B has occurred, denoting they are independent. Likewise, the algorithm assumes that all classes are independent of each other and have an equal effect on the outcome. Multinomial, Bernoulli and Gaussian are the three types of Naïve Bayes classifiers and out of which, the latter has been adopted for our analysis.

4.4.4 Decision Tree

Decision Tree has a flowchart structure in which each node represents a test, each branch represents the outcome of the test and each leaf node represents the class label. The algorithm is widely considered to be to be highly effective in producing high accuracy outcomes for classification tasks.

4.4.5 Random Forest

Random Forest is nothing but an ensemble model made up of multiple Decision Trees. The model trains each tree on a different sample to reduce variance and splits the nodes by considering only a subset of the classes. The final outcome is arrived by averaging the outcome of each individual tree. This ensemble model uses the outcomes of random uncorrelated decision trees to produce the best possible outcome.

4.4.6 Convolutional Neural Network

A Convolutional Neural Network (CNN) is a deep learning model is based on the connectivity pattern of the multiple layers containing neurons in the human brain. The CNN model used for this analysis consists of 4 sets of convolutional and max pooling layers with ReLu activation layers between them. A couple of dropout layers are inserted after the 2^{nd} and 4^{th} set of layers. The last layers of the model comprises of flatten, dense, ReLu activation, dropout, dense and Softamax activation layers. The layers in the CNN model shares the parameters and input amongst themselves equally to reduce the individual load and speed up the training. The algorithm is sophisticated enough to not require much pre-processing of the data, when compared with conventional models.

5 Results

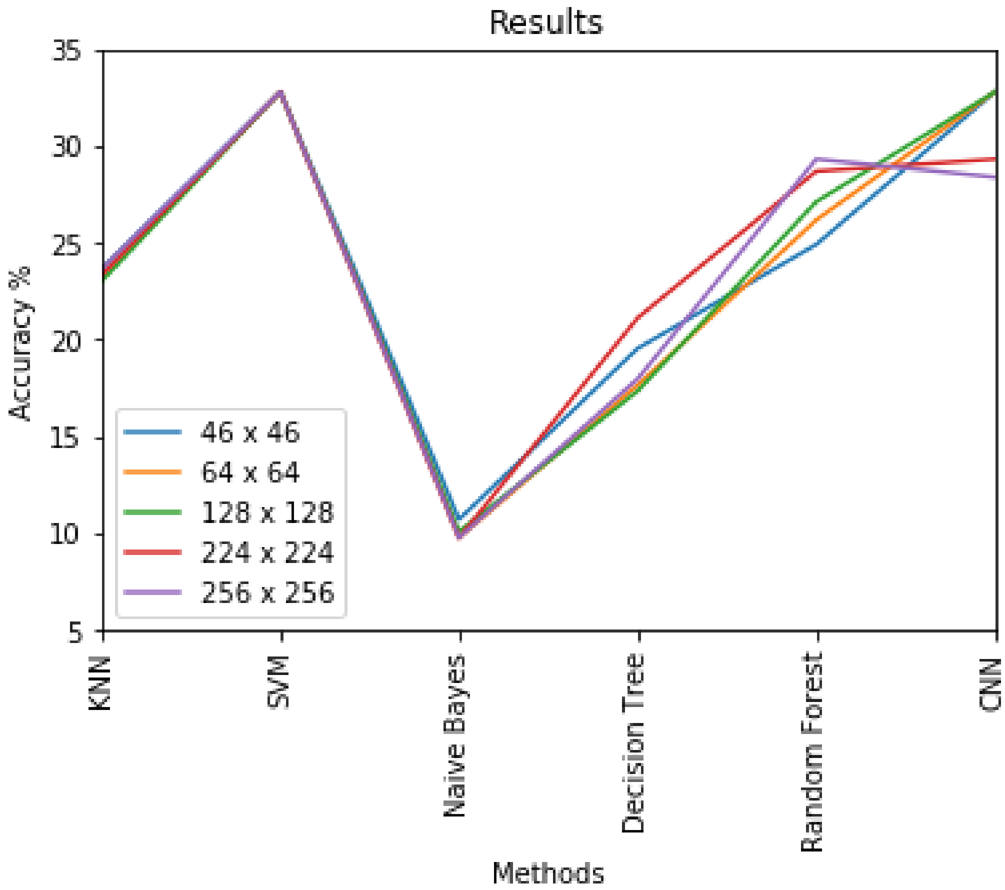
The accuracy of the KNN model hovers around 23% across all the pixel size which might be due to its sensitivity to dimensionality, when the input image had to be converted from 4 dimensions to 2 dimensions. The accuracy of the SVM model stay constantly at 32.80% across all the pixel sizes confirming that it handles high dimension data well. The Naïve Bayes model has produced low accuracy ranging around 9% to 10% which can be attributed to its over-reliance on the assumption of independence between the classes.

Table 1: Accuracies of Classification Models

Methods	46 x 46	64 x 64	128 x 128	224 x 224	256 x 256	Trend
KNN	23.65%	23.34%	23.02%	23.34%	23.65%	
SVM	32.80%	32.80%	32.80%	32.80%	32.80%	
Naive Bayes	10.72%	9.77%	10.09%	9.77%	9.77%	
Decision Tree	19.55%	17.66%	17.35%	21.13%	17.98%	
Random Forest	24.92%	26.18%	27.12%	28.70%	29.33%	
CNN	32.80%	32.80%	32.80%	29.33%	28.39%	

The accuracy of the Decision Tree model varies from 17.35% to 21.13% with no discernible trend which might be due to its susceptibility to over-fitting. The accuracy of Random Forests shows an increasing trend starting from 24.92% for 46x46 size to 29.33% for 256x256 size, for which the decorrelated trees and its reduction of variance in the dataset might be credited. The accuracy of the CNN model stays constant at 46x46 size till 128x128 size and then drops down with a decreasing trend to 28.39% for 256x256 size.

Figure 5: Accuracies of Classification Models



6. Conclusion

As per common knowledge, it was assumed that the accuracy of a classification model will decrease from higher resolution sizes to lower resolution sizes due to information loss but the outcomes that have been produced does not support that hypothesis and has rather left us with more questions. How does the accuracy of the SVM model stay constantly at 32.80% across all the pixel sizes? Likewise, does the accuracy of KNN model hovering around 23% with no increase or decrease indicate imbalanced data? Does the varying accuracy of the Decision Tree model with no discernible trend indicate over-fitting of data? Would Precision and Recall have been a better performance metric compared to Accuracy, considering that the dataset might be class imbalanced?

These questions require further examination and can be taken up at a later stage as future work. The class distribution can be balanced by over-sampling the dataset and if required, hyper-parameter tuning can be performed and weights can also be added to the CNN model.

References

[1] Wang, X., Peng, Y., Lu, L., Lu, Z., Bagheri, M. and Summers, R. (n.d.). (2017). *ChestX-ray8: Hospital-scale Chest X-ray Database and Benchmarks on Weakly-Supervised Classification and Localization of Common Thorax Diseases*.

Towards Data Science. (2019). *Machine Learning Basics with the K-Nearest Neighbors Algorithm*.

Kdnuggets.com. (2019). *Support Vector Machines: A Simple Explanation*.

Towards Data Science. (2019). *Naive Bayes Classifier*.

Medium. (2019). *Decision Trees — A simple way to visualize a decision*.

Displayr. (2019). *What is a Random Forest? | Displayr*.

Towards Data Science. (2019). *A Comprehensive Guide to Convolutional Neural Networks — the ELI5 way*.