Machine Learning for Signal Processing Course Project
Project Final Report
Date: 05/09/2023

Project Title: Identifying and classifying types of pneumonia within chest X-ray images with machine learning techniques.

Team Members: Noah Hamilton and Samuel Washburn

Table of Contents

Identifying and classifying types of pneumonia within chest X-ray images with machine learning techniques

1 INTRODUCTION

This project classifies images of chest X-Rays between normal healthy lungs and pneumonia infected lungs. The motivation behind this project comes from the leading cause of child mortality being pneumonia [1]. Nearly all cases of childhood pneumonia occur in developing countries. These countries lack the expertise and resources to diagnose and properly treat cases of pneumonia. There are two types of pathogens that cause pneumonia: bacterial and viral; these two types require distinct types of treatment.

A previous study was done first using Optical Coherence Tomography (OCT) images and then the a similar model was taken and applied to X-Rays images [2]. This study by Kermany used a Convolutional Neural Network that was pretrained using the ImageNet dataset and it achieved roughly 92% accuracy [2]. This project compares other Machine Learning techniques and image classifiers such as Support Vector Machine (SVM), K-Nearest Neighbors (KNN), and a custom trained Convolutional Neural Network (CNN). These image classifiers will detect whether a chest X-ray is normal healthy or pneumonia infected. Successful execution of this project provides insights into the use of machine learning techniques for the medical field.

This project has been successfully completed with the creation of three image classifiers around 70% overall accuracy using various MLSP techniques including but not limited to data pre-processing, data conditioning, feature extraction, and finally modelling and prediction. Out of the three models tested (SVM, KNN, and CNN) the SVM performed the best overall while the CNN performed exceptionally well at detecting pneumonia, but very poorly at detecting whether a patient was healthy. Selecting a different test set significantly improved the performance of CNN, which caused suspicion regarding the quality of the original test set.

2 PROJECT OBJECTIVE

This project aims to demonstrate that machine learning can be successfully applied to classification of medical images by identifying pneumonia in chest X-ray images. This project would use the same data set that Kermany used and perform different Machine Learning techniques such as Support Vector Machines, K nearest neighbor clustering to compare against the Convolutional Neural Network approach taken by Kermany [1]. This project would also attempt to improve Kermany's accuracy by not using a transfer learning approach and fine tuning the CNN design and parameters. This project aims to reduce the number of misdiagnosed X-rays specifically X-Rays that are infected with pneumonia but get classified as healthy.

3 PROJECT TASKS

The tasks in the project plan closely follow the machine learning for signal processing paradigm steps of Sensor, Signal Capture, Data Conditioning, Feature Extraction, and Modelling and Prediction.

For Sensor and Signal Capture, this project uses a pre-compiled dataset of chest X-rays images that was downloaded from Mendeley Datasets. For data conditioning, the dataset was balanced to even out the number of normal vs infected chest X-rays along with the application of histogram equalization and trimming. For feature extraction, principal component analysis was performed to reduce the dimensionality of the data for the KNN and SVM while convolution and pooling layers handle the feature extraction within the CNN. Finally, for modelling and prediction, three image classifiers were modelled (SVM, KNN, and CNN) and the test set was predicted upon by each of the three models.

3.1 Sensor

The sensor is an X-Ray machine. X-Rays are part of the electromagnetic spectrum such as visible light and radio waves. The machine makes a small burst of X-Ray radiation at the area of interest and the radiation passes through the body and is recorded by a detector on the other side of the body. The detector records the radiation received into an image, which is electronically stored in a digital file. In a chest X-ray, "the ribs and spine absorb much of the radiation and appear white or light gray, while lung tissue absorbs little radiation and appears dark." These X-ray images will be used as data for the image classification [4].

3.2 Signal Capture

The data were acquired via an internet download from the Mendeley Data website. The images were taken from pediatric patients from the Guangzhou Women and Children's Medical Center in Guangzhou, China. The data was compiled and labelled by the University of California San Diego. The dataset is 2GB, so the team's computers were capable of storing and processing this quantity of data. The data consists of 5,856 images of chest X-rays. Images are in jpeg format. The images are labeled with their diagnoses - either "normal" or "pneumonia". Within the pneumonia category, there is an additional label as to whether the pneumonia is bacterial or viral. The images are divided into three sets – training, validation, and test. The dataset has about 3x as many pneumonia images as normal images. The dataset has already been cleaned to remove poor quality images. The image labels (normal or pneumonia) were given by three experts, so the data are clean and reliable [2] [3] [5].

3.3 Data Conditioning

Several techniques were used to pre-process the data before feature extraction. The first technique was dataset balancing. In the original dataset from Kaggle, the "normal" class and the "pneumonia" class were imbalanced in both the training set and test set. There were about 1,300 normal images in the training set, but about 3,900 pneumonia images. The team balanced the training set for improved training, as well as the test set. When balancing the dataset, the team selected images to remove based on which file sizes were smallest. Images with an abnormally small file size may have poor quality or small resolution, so removing small images reduces the risk of poor data.

The team also implemented contrast enhancement through histogram equalization. Histogram equalization involves the redistribution of grey-scale intensities such that the quantities of pixels in each intensity "bin" are level. In other words, the pixel intensity values are spaced out in the grey-scale spectrum to maximize contrast in the image [6]. Histogram equalization was used on the chest X-rays to maximize the contrast between bones, pneumonia in the chest (lighter regions that aren't bones), and black regions. MATLAB has a built-in function 'histeq' that makes histogram equalization (HEQ) simple. The result of HEQ on a chest X-ray with viral pneumonia is shown below in Figure 1. The left image is an image before HEQ. The image looks cloudy within the chest region, and the viral pneumonia (the wispy, "cob-web" looking features) is tricky to distinguish. After HEQ (right image), the viral pneumonia stands out more because the darker areas in the chest (no pneumonia) are driven toward black. The team discusses the impact of HEQ on performance in the 'SVM' section of the results.

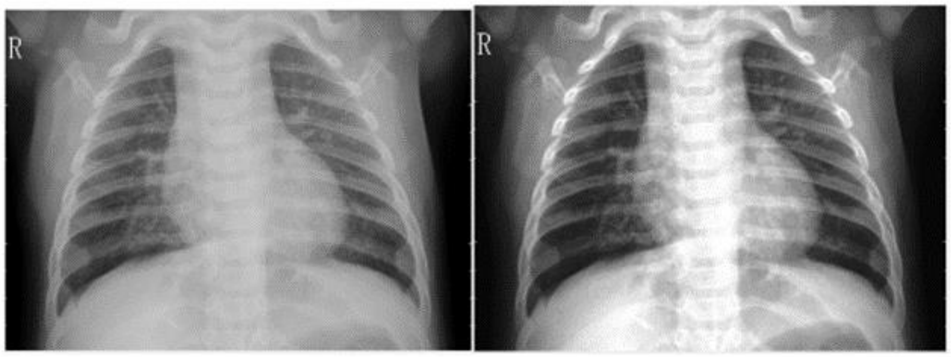


Figure 1: A chest X-ray with viral pneumonia – before and after histogram equalization (left and right images, respectively).

Lastly, the team implemented 'trimming' of the images. In a chest X-ray, the primary area of interest for classifying pneumonia is the lungs, since pneumonia exists in the lungs. Consequently, the borders of the images are considered "noise" since they aren't relevant to the presence of pneumonia. To improve the signal-to-noise ratio of the images, the team trimmed off the borders of the images. An example of trimming is shown below in Figure 2. In the MATLAB code, the user has the option to set four parameters: top trim, bottom trim, right trim, and left trim. These parameters are between 0 and 1, and they set the percentage of the image dimension to trim away. The team discusses the impact of trimming on performance in the 'SVM' section of the results. Note that HEQ and trimming were only executed in the SVM model – the KNN and CNN models did not utilize these techniques.

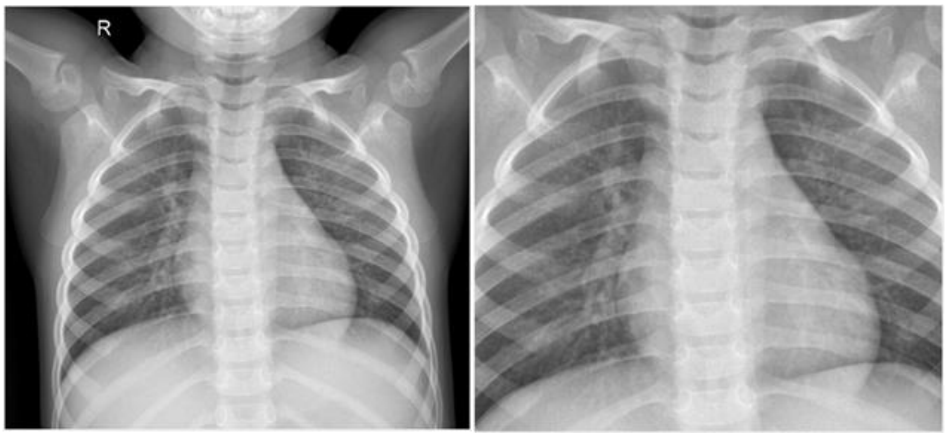


Figure 2: A chest X-ray before and after trimming (left and right images, respectively).

3.4 Feature Extraction

The feature extraction process was completed in two ways: principal component analysis (PCA) and the use of Convolution and Pooling layers.

PCA was used in order to reduce the dimensionality of the data for input into the KNN and the SVM. PCA transforms the data linearly to best represent the dimensions of the data that have the highest variation. Built in functions for PCA within both MATLAB (SVM) and Python (KNN) were used to compute the PCA. It was then decided to compare the results of the classifiers using different percentages of the variation of levels. So for example, the first couple of PCA components represent about 30% of the variation in the data. The variation was varied between 30% and 70% to find the greatest accuracy. In the plots below, it can be seen that around 45-55% variance achieves the highest accuracy levels. This makes sense as reducing the data dimensions allows for the models to key in on the most important pieces of the data. PCA was used in the KNN and SVM and not needed with implementation within the CNN as the convolution and pooling layers take care of the feature extraction.

Convolution and pooling layers reduce the dimensionality of the data and easily implemented within the TensorFlow Python Library. The Conv2D layer establishes kernels that are convolved with the input images. As the kernel parameters are trained with the training data, the features that are important to a normal image vs a pneumonia image are implicitly learned in the CNN. When adding a Conv2D layer within the TensorFlow Python library, there are multiple parameters that can be set. The first parameter sets how many kernels the images will be convolved with in this layer [7]. During training, 160 parameters were learned in the first Conv2D layer in the model. The next parameter sets the size of the kernel window to 3 pixels by 3 pixels. To extract larger features from the chest X-rays, the team may consider using a larger kernel size, perhaps 5x5 or 7x7, in the first Conv2D layer. When the output size of a layer starts to get smaller, a 3x3 or 1x1 kernel size is more appropriate to extract features. Also, source [7] mentions that more kernels, for example 32 or 64, should be used in Conv2D layers that are deeper into a CNN. The activation parameters set the activation function that the data is passed through after the data is convolved with the kernels [7]. A MaxPooling layer after the Conv2D layer reduces the dimensionality of the data by taking the maximum of some subset of pixels. The combination of the Conv2D and MaxPooling layers create the feature extraction within the CNN.

3.5 Modeling/Prediction

Three different image classification techniques were used for modelling and prediction of the chest X-rays: K Nearest Neighbors (KNN), Support Vector Machine (SVM), and Convolutional Neural Network (CNN).

The KNN is an unsupervised machine learning classifier that uses the distance between feature vectors to predict the most common class. The KNN has one main tunable parameter and that is the number of neighbors used for comparison. It is also possible to use different distance metrics for computing the distance; however, in this case Euclidean distance between feature vectors was used. The KNN was implemented using the sklearn python library.

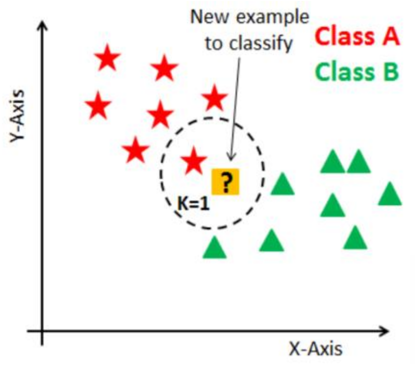


Figure 3: Example of the KNN image classifier [8]

The SVM is a supervised machine learning classifier that makes use of a linear boundary to make predictions. The SVM can be configured with different kernels and different box constraints. Linear, Gaussian, and Polynomial kernels were compared as well as box constraints ranging from 0.9, 0.09, and 0.009. The SVM was implemented using the

MATLAB built in `svm()` function.

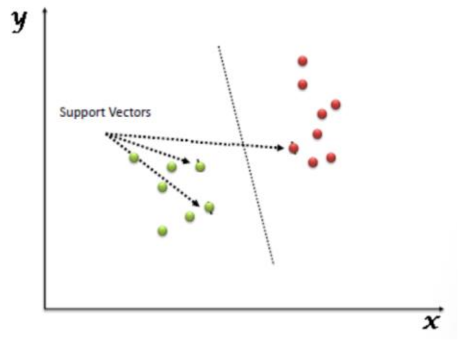


Figure 4: Example of the SVM classifier using a linear boundary to predict classes [8]

The CNN is an unsupervised machine learning classifier that uses the concept of neurons, hidden layers, convolution and pooling to make predictions. In the CNN, the number and types of layers as well as the activation functions can be varied to improve performance. Input data is fed through the network to compute an output and for training back propagation techniques are used to calculate the weightings connecting the hidden layers within the CNN.

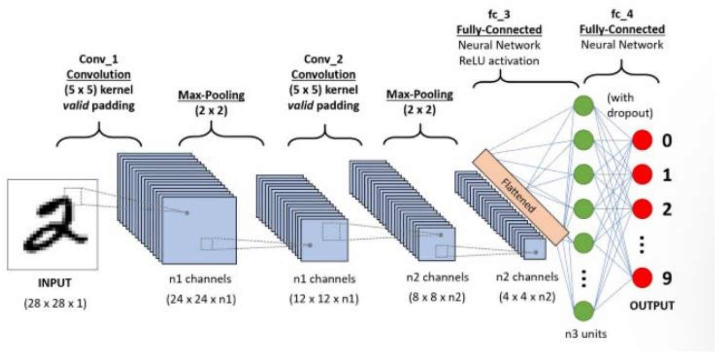


Figure 5: Example of the CNN [8]

Each of the models were trained using the training data set and tested on the test dataset. The results are shown below in the Results and Conclusions section.

4 RESULTS

4.1 K-Nearest Neighbor

The KNN was first run using all PCA components across the validation dataset in a process called cross-validation. This was used to find the "k" value or number of neighbors that would be used in the computation of the closest neighbor. It was found that running from 1 to 14 neighbors that 13 was found to be highest scoring k value as seen below in Figure 6. Increasing the number past 14 began to degrade the accuracy and take substantial amount of computational time to compute.

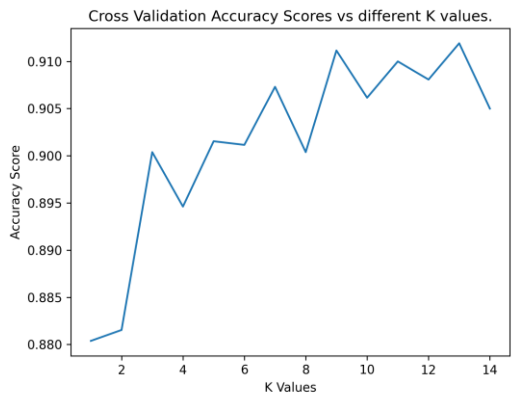


Figure 6: Generated Plot of Cross Validation scores over varying neighbor counts.

Using the highest scoring K-value, the KNN classifier was compared using the percent of variance in the components used from PCA: 30-70% at 5% increments. The classification accuracy is shown below in Figure 7. The highest overall accuracy was at 55% of the total variance at around 77% accuracy.

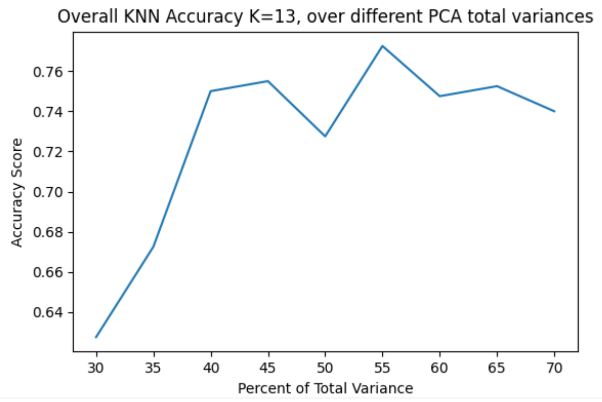


Figure 7: Overall KNN accuracy using the best k-value over different PCA total variances.

4.2 Support Vector Machine

The performance of the SVM was compared across several different parameters, including:

- Percent of variance in the components used from PCA: 30%-70% at 5% increments.
- Linear kernels, Gaussian kernels, and polynomial kernels.
- Box constraint values: 0.009, 0.09, 0.9
- Data conditioning though HEQ and trimming vs. no HEQ and no trimming.

The classification accuracy of the SVMs is shown in the plots below. Figure 8 shows the accuracy using a linear kernel across different box constraints and total variances. The best performance was seen between 45% and 55% total variance. The performance did not vary significantly across different box constraint values. The highest performance was 78.8%, achieved with a box constraint of 0.9 and a total variance of 45%.

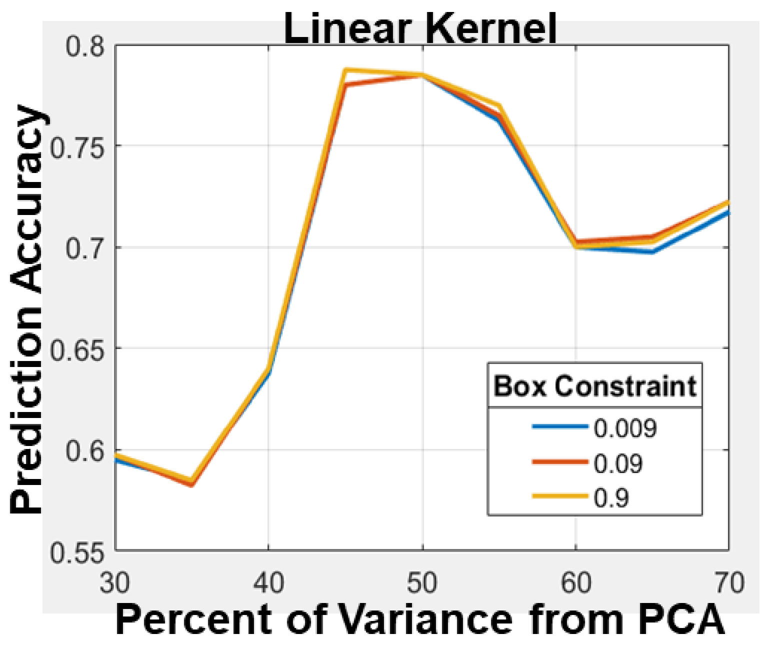


Figure 8: SVM performance using a linear kernel.

Figure 9 shows the SVM accuracy using a Gaussian kernel across different box constraints and total variances. Best performance was seen using 45-50% of total variance, but performance dropped off sharply when total variance was about 50%. Unlike the linear kernel, the box constraint value had a large impact for the Gaussian kernel. A box constraint of 0.9 resulted in an accuracy above 75%, while a box constraint of 0.009 resulted in accuracy below 65%. A higher box constraint value produced better results. The highest performance was about 76%, achieved with a box constraint of 0.9 and a total variance of 45%.

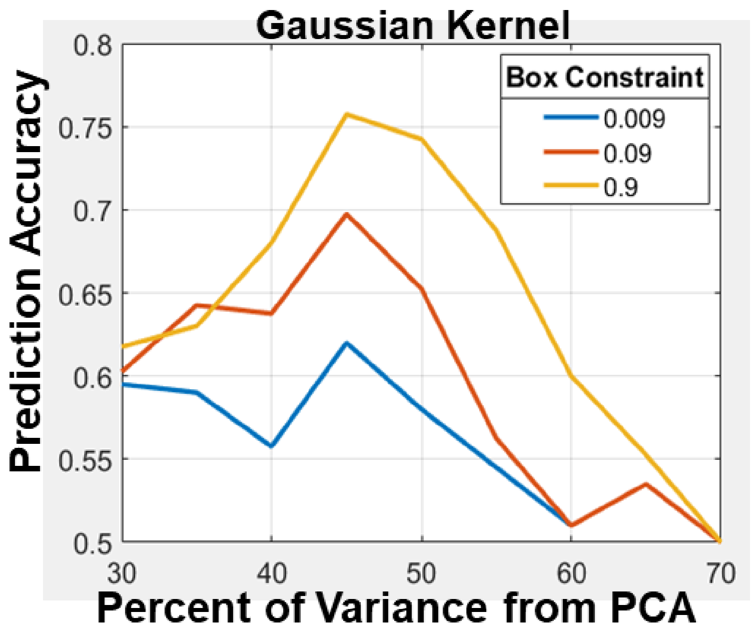


Figure 9: SVM performance using a Gaussian kernel.

Figure 10 shows the SVM accuracy using a polynomial kernel across different box constraints and total variances. Again, best performance was seen using 45-55% of total variance. The accuracy profile vs. total variance looks similar to the linear kernel but with slightly less accuracy. Box constraint had minimal impact on performance, although the lowest box constraint had slightly lower accuracy between 45-50% variance. The highest performance was about 78%, achieved with a box constraint of 0.9 and a total variance of 50%.

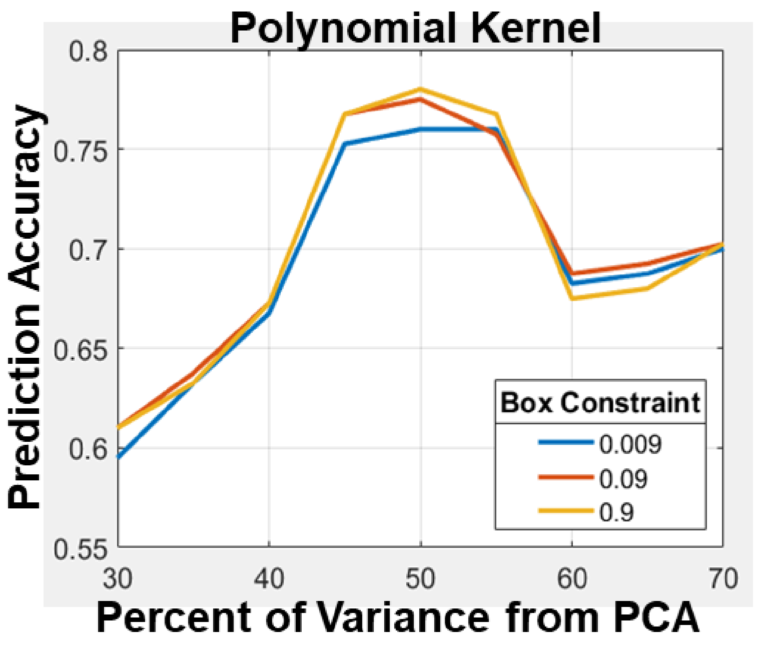


Figure 10: SVM performance using a polynomial kernel.

As mentioned in the ‘Data Conditioning’ section, HEQ and trimming were tried for the SVM model in attempt to improve performance. Interestingly, neither HEQ nor trimming improved the prediction accuracy of the SVM model. After projecting onto the principal components, perhaps the effects of the equalization and the trimming were lost. For this particular project, HEQ and trimming were not effective for improving the performance of an SVM. HEQ and trimming were not attempted for KNN or CNN.

4.3 Convolutional Neural Network

A CNN was built using the Tensorflow keras library in Python . The CNN consisted of three convolutional layers, three max pooling layers, a flattening layer, and two dense layers. An illustration of the layers in the CNN is shown in Figure 11. To develop this model, the team learned from two Youtube tutorials that described how to develop image classifiers for color images using Tensorflow keras [9] [10]. The code in these tutorials was adapted to work for the classification of the grayscale chest X-rays.

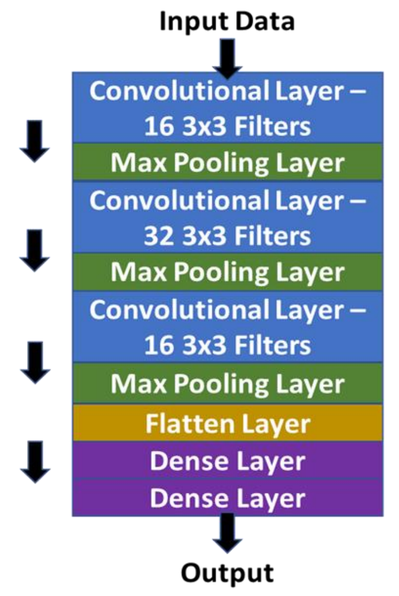

Input Data
Convolutional Layer – 16 3x3 Filters
Max Pooling Layer
Convolutional Layer – 32 3x3 Filters
Max Pooling Layer
Convolutional Layer – 16 3x3 Filters
Max Pooling Layer
Flatten Layer
Dense Layer
Dense Layer
Output

Figure 11: Illustrated layers in the team's CNN.

A summary of the CNN was outputted in Python. The shape of each layer, as well as the number of parameters that can be trained for each layer, are shown in Table 1 below.

Table 1: Shape and number of parameters for each layer in the CNN.

Layer Type	**Output Shape**	**Number of Parameters**
Convolution 2D	254 x 254 x 16	160
Max Pooling 2D	127 x 127 x 16	0
Convolution 2D	125 x 125 x 32	4,640
Max Pooling 2D	62 x 62 x 32	0
Convolution 2D	60 x 60 x 16	4,624
Max Pooling 2D	30 x 30 x 16	0
Flatten	14,400	0
Dense	256	3,686,656
Dense	1	257
Total Parameters:	3,696,337	
Trainable Parameters:	3,696,337	

For the CNN model, 10% of the training set images were separated to be the validation set. The parameters in the CNN were trained by passing the training data through the model 15 times. Each pass is called an epoch. A plot of the loss on the training data and the validation data as the epochs progressed is shown below in Figure 12. As we expect, the loss on the training set and validation set steadily decreases. If we performed too many epochs, the model may be trained too tightly to the training data, which would hurt the model's generalization to the validation set and test set [11].

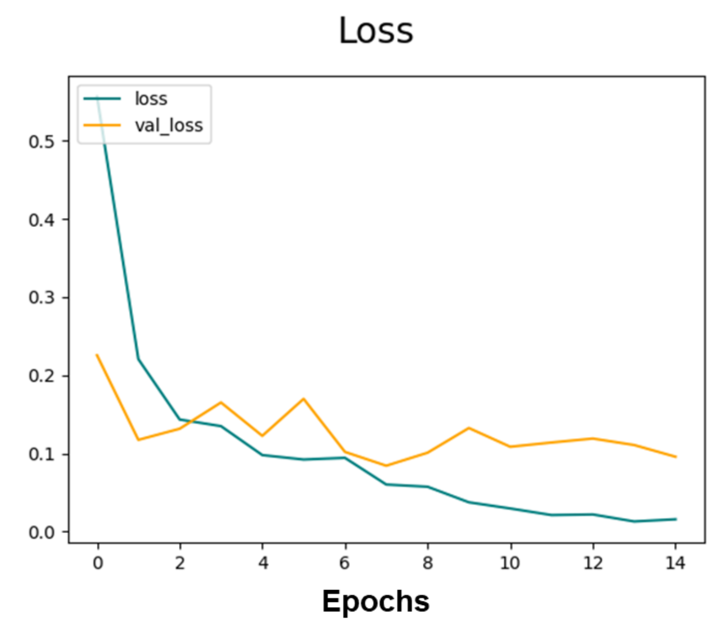


Figure 12: Plot of training loss and validation loss vs. number of epochs.

A plot of the training accuracy and validation accuracy during training is shown below in Figure 13. As we expect, the training and validation accuracy improves with each pass through the data [11].

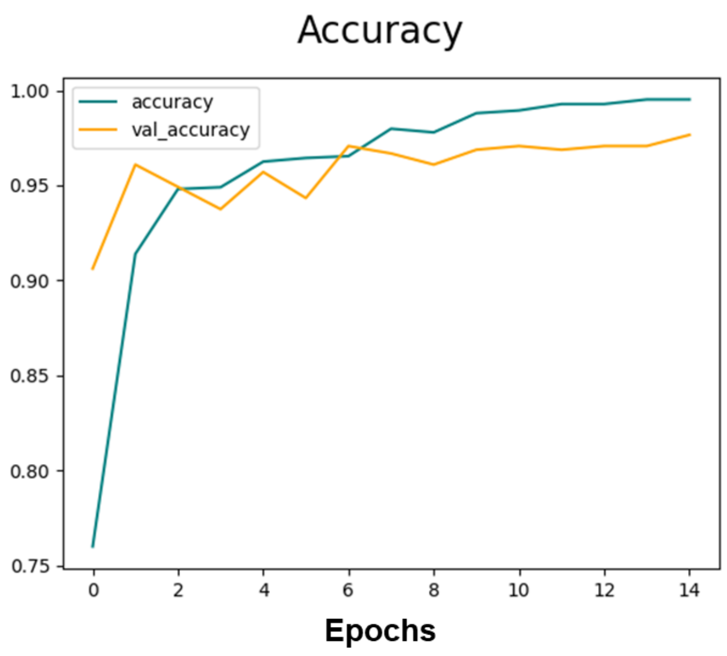


Figure 13: Plot of training accuracy and validation accuracy vs. number of epochs.

After the CNN was trained, the test set was passed through the model for classification [11]. Sadly, the CNN only classified the test set with an accuracy of 66%. The CNN correctly classified pneumonia images 99% of the time, but had a major problem with false positives.

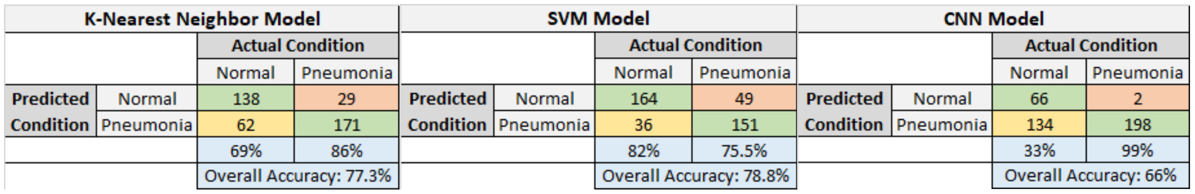
The CNN classified normal images correctly only 33% of the time.

Given that CNN is supposed to be highly successful with image classification, the team was suspicious at the low accuracy of 66%. Through more investigation, the team was concerned that the test set on Kaggle had significant differences from the training set, which was causing poor generalization from the training phase to test phase. To pursue this issue further, the team formed a new test set by pulling 200 images from each class from the training set. So instead of training on 1,300 images from each class, and then testing with the test set, the CNN was trained on 1,100 images from each class, and then tested on the unseen 200 images from each class.

*This new attempt at training and testing performed much better – **93% accuracy**! This prediction accuracy was more in family to what the team expected from a CNN. This major improvement increased the team's suspicion that the test set on Kaggle may not be representative of the two classes that were trained on. For consistency in comparison, the team will include the performance of the original attempt in section 4.4 (66% accuracy), but please note that a significantly higher accuracy was achieved with CNN in the latter attempt (with a new test set).*

4.4 Comparison of the three models via confusion matrices.

In order to compare the performance of each of the classifiers, confusion matrices are generated to do a side-by-side comparison. A confusion matrix is a more in-depth description of the results. The confusion matrix shows the number of false positives, false negatives, true positives, and true negatives. In this case a false positive is classifying an X-ray that is truly normal as an infected patient while a false negative is classifying an X-ray that is truly infected as a healthy patient. The latter (false negative) is much worse in this cases. The true negative and true positive are the cases when the image classifier correctly classifies healthy patients as healthy and pneumonia infected as patients as infected.



K-Nearest Neighbor Model			
		Actual Condition	
		Normal	Pneumonia
Predicted Condition	Normal	138	29
	Pneumonia	62	171
		69%	86%
		Overall Accuracy: 77.3%	

SVM Model			
		Actual Condition	
		Normal	Pneumonia
Predicted Condition	Normal	164	49
	Pneumonia	36	151
		82%	75.5%
		Overall Accuracy: 78.8%	

CNN Model			
		Actual Condition	
		Normal	Pneumonia
Predicted Condition	Normal	66	2
	Pneumonia	134	198
		33%	99%
		Overall Accuracy: 66%	

Figure 14: Generated Confusion matrices for the KNN, SVM, and CNN showing a more detailed insight into the accuracy of each of the models.

In Figure 14, the red highlighted cells are the false negatives which are the worst cases while the yellow highlighted cells represent the false positive cases, which are inaccuracies, but have minor consequences relative to the false negatives. The green highlighted cells represent the cases where the model correctly predicted the actual condition. The above confusion matrices are taken from the highest performing variations of each of the models. The SVM has the highest overall performance; however has the lowest accuracy when it comes to predicting a pneumonia infected patient while the CNN model has the worst overall performance (on the original test set), but has extremely high pneumonia prediction. The CNN appears to be biased towards classifying images as infected. With the newly formed test set, the CNN performed better than 90%, which is a significant improvement.

5 CONCLUSIONS

This project demonstrated that machine learning is a viable approach for classifying pneumonia in chest X-rays. Both the KNN and SVM model achieved classification accuracies of above 75%. When using the Kaggle test set, the CNN was less than 70% accurate, but when using a different test set, the CNN was more than 90%, which better meets the expectations of a CNN. With the new test set, the CNN performed about the same as Kermany's CNN [2], which is exciting. This project showed that multiple different machine learning models can be used with success to attack the same problem. Through experience and extensive testing, it is up to the designer to select the model that performs best for a given problem. In a broader sense, this project has shown that machine learning can be effective for classifying medical conditions in images. An opportunity for future work could be to apply the lessons learned from this project to classify other medical conditions in images. Examples of other medical images that could be classified include MRIs and CT scans. Another area for future work would be to decrease the rate of false positives in the models. In general, it was found that that the models struggled more with classifying normal images – the models seemed to have a bias toward picking "pneumonia".

6 SUMMARY

This project demonstrated the ability to apply learned MLSP techniques to real world applications through use of multiple image classifiers (KNN, SVM, and CNN) and the MLSP paradigm (sensor, data conditioning, feature extraction, and prediction). It was found that the SVM performed the best (highest overall accuracy), which was expected as SVM is particularly suited to make binary decisions. Furthermore, the CNN performed the best on images of Pneumonia to a high accuracy while mis-classifying Normal images with a much worse accuracy. This project highlights the ability for Machine Learning models to be used to aid medical professionals, but also necessitates the fact that it Machine Learning is only so accurate just like humans. The combination of Machine Learning tools like the ones created in this project and humans can lead to increases in efficiency in patient care.

7 APPENDIX:

Link to Source Code: https://github.com/theRealNoah/mlsp-image-classification

8 REFERENCES

- [1] Rudan, Igor & O'Brien, Katherine & Nair, Harish & Liu, Li & Theodoratou, Evropi & Qazi, Shamim & Luksic, Ivana & Walker, Christa & Black, Robert & Campbell, Harry & Reference, Group. (2013). Epidemiology and etiology of childhood pneumonia in 2010: Estimates of incidence, severe morbidity, mortality, underlying risk factors and causative pathogens for 192 countries. J Glob Health. 3. 10401. 10.7189/jogh.03.010401.

- [2] D. S. Kermany, M. Goldbaum, W. Cai, C. C. S. Valentim, H. Liang, S. L. Baxter, A. McKeown, G. Yang, X. Wu, F. Yan, J. Dong, M. K. Prasadha, J. Pei, M. Y. L. Ting, J. Zhu, C. Li, S. Hewett, J. Dong, I. Ziyar, A. Shi, R. Zhang, L. Zheng, R. Hou, W. Shi, X. Fu, Y. Duan, V. A. N. Huu, C. Wen, E. D. Zhang, C. L. Zhang, O. Li, X. Wang, M. A. Singer, X. Sun, J. Xu, A. Tafreshi, M. A. Lewis, H. Xia, and K. Zhang, "Identifying medical diagnoses and treatable diseases by image-based Deep Learning," Cell, vol. 172, no. 5, 2018.

- [3] Kermany, Daniel; Zhang, Kang; Goldbaum, Michael (2018), "Labeled Optical Coherence Tomography (OCT) and Chest X-Ray Images for Classification", Mendeley Data, V2, doi: 10.17632/rscbjbr9sj.2

- [4] X-Ray Basics: https://www.radiologyinfo.org/en/info/chestrad#7ce6e42dff284a8fa1ddf997ceb3c708

- [5] Kaggle Inspiration: https://www.kaggle.com/datasets/paultimothymooney/chest-xraypneumonia

- [6] Intro to Contrast Enhancement: https://iq.opengenus.org/contrast-enhancementalgorithms/#:~:text=Machine%20Learning%20(ML),-More%20Less%20Up&text=Contrast%20enhancement%20is%20an%20essential,is%20importa nt%20for%20human%20interpretation

- [7] Explaining the Conv2D Layer in a Tensorflow Keras CNN Model https://pyimagesearch.com/2018/12/31/keras-conv2d-and-convolutional-layers/

- [8] https://iq.opengenus.org/basics-of-machine-learning-image-classification-techniques/

- [9] Simple Demonstration of CNN in Python from ‘NeuralNine’. Classification of Color Images from Multiple Classes https://www.youtube.com/watch?v=t0EzVCvQjGE

- [10] In-Depth Demonstration of CNN in Python from Nicholas Renotte. Classification of Color Images from Two Classes https://www.youtube.com/watch?v=jztwpsIzEGc

- [11] Intro to CNN: https://towardsdatascience.com/convolution-neural-network-for-imageprocessing-using-keras-dc3429056306

- [12] Intro to CNN: https://www.analyticsvidhya.com/blog/2021/06/image-processing-using-cnna-beginners-guide/

- [13] Intro to Feature Extraction: https://www.analyticsvidhya.com/blog/2019/08/3-techniquesextract-features-from-image-data-machine-learning-python/

- [14] Feature Extraction - https://www.mathworks.com/discovery/feature-extraction.html