Classification of X-ray images with common machine learning algorithms

Choi, Moon-ki choi0652@umn.edu

Choi, Won Joon choix471@umn.edu

Kim, Hyunwoo kim00186@umn.edu

Lee, Garim lee02169@umn.edu

CSCI 5523 Final Project - Due May 4th, 2021

A link for codes on GitHub: https://github.com/dulusian/Pneumonia_detection

1 Motivation

Diagnosis of lung diseases such as pneumonia through chest X-ray images requires considerable reliability because small mistakes in diagnostic tools have a great impact on humans. In this project, our team "Life Saver" will propose four reliable diagnostic tools based on machine learning frameworks learned from the course CSCI5523 - Introduction to Data Mining by Dr. Vipin Kumar. The project will have a strong impact on pneumonia diagnosis for three reasons:

- The use of diagnostic tools can determine how far lung diseases such as pneumonia have occurred and diagnose diseases caused by viruses or germs.
- The use of diagnostic tools provides a reliable model for diagnosing patients' chest conditions, preventing professional errors in accurately diagnosing lung diseases and conditions.
- The diagnostic tools are based on x-ray images with pneumonia, so they can be extended indefinitely to another model with x-ray photos with other diseases.

From this project, we expect the four machine learning models used in diagnostic tools to be able to standardize and accurately diagnose diseases such as pneumonia and lung disease and compare the various other machine learning models we used to see which of the models is most suitable for screening pneumonia.

2 Related work

Convolutional neural network (CNN):

Convolutional neural network (CNN) has had groundbreaking results in the field of Deep Learning. CNN has its strength in pattern recognition such as image classification and voice recognition [1]. Artificial neural network (ANN) uses a naive approach of training parameters whereas CNN reduces this complex task in large datasets of the modern world by taking a different route. For example, if the image that we are trying to train on is 32×32 pixels with 3 RGB channels, there would be $32 \times 32 \times 3$ with only one neuron. By adding more neurons and layers, parameters that will be updated become very large which takes a considerable amount of time. Convolution reduces complexity by taking a portion of the image per neuron. We will discuss more about the process of the convolutional neural network further in this project.

Using deep learning algorithm for detecting pneumonia from X-rays:

Our project focuses on the classification of pneumonia X-rays images. This article experiments using the convolutional neural network with the rectified linear unit (ReLU) as the activation function and Adam optimization function to test the result [2].

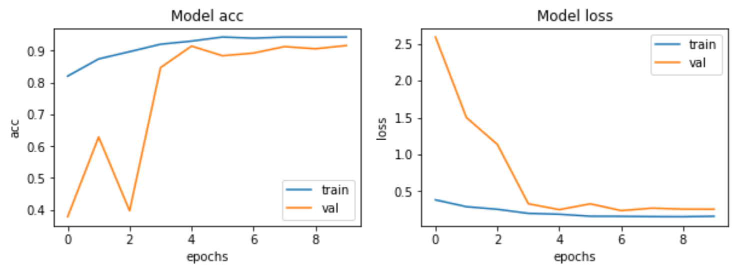

Figure 1: Credit: Abhinav Sagar

As we can see from the figure, model accuracy and loss improves within 10 epochs. The model achieves 90% accuracy on the validation set which can be improved as the model is further trained. In this project, we will use various algorithms to compare the results and discuss the tradeoffs among the models created.

3 Data Logistics

The dataset is organized into 3 folders which have train, test, and validation datasets. All dataset contains subfolders for each image category (Pneumonia/Normal). There are 5,863 X-Ray images (JPEG) and 2 categories (Pneumonia/Normal) in total.
Chest x-ray images (frontal larynx) were selected as a retrospective cohort for children aged 1 to 5 at Guangzhou Women's and Children's Hospital. All chest X-ray imaging was performed as part of the patient's routine clinical treatment.
For the analysis of chest X-ray images, all chest radiographs were initially screened for quality control by eliminating all low-quality or non-readable scans. Then, diagnostic scores on images were scored by two specialists before they were deleted for AI system training. To account for all rating errors, the evaluation set was also confirmed by a third expert [3].

The normal chest x-ray (left panel) shows clear lungs with no abnormal surgical site in the image. Bacterial pneumonia (middle) typically exhibits focal lobar bonds in the right upper lobe (white arrow), while viral pneumonia (right) exhibits a more diffuse "interval" pattern in both lungs.

4 Understanding models

Based on the purpose of the project, classification was conducted to detect pneumonia or normal as a binary classifier. Considering that the widely used machine learning algorithms for image classification are Support Vector Machine (SVM), Artificial Neural Network (ANN), Convolutional Neural Network (CNN), and transfer learning, the purpose of this part is to briefly describe each algorithm.

First, SVM is a widely used supervised learning model for classification that uses linear or nonlinear decision boundaries to separate classes [4]. It is also known that SVM performs notably in image classification tasks such as image recognition due to its outstanding generalization capability and reputation in the training data set to achieve high accuracy [5]. Second, neural networks can be used for challenging areas such as image classification [4]. Specifically, ANN and CNN were chosen because they were proven to show successful performance in tasks using images, such as image classification, object identification, and face recognition. Lastly, transfer learning allows the classification of a new dataset using related pre-trained data. It is known that, if a model is trained on enough dataset for image classification, the model using transfer learning will show adequate accuracy in classifying another model. A detailed explanation of each model is followed.

4.1 Support Vector Machine (SVM)

The strategy of SVM is to find an optimal maximum margin that separates the classes by focusing on the training samples located at the edge of the class distribution [4]. Thus, the main idea of SVM is to derive a hyperplane by maximizing the margin between two classes, which in this case, Pneumonia and Normal. In addition, there are also two conditions for using SVM: first, all data vectors belonging to the same class are placed on the same side of the hyperplane. Second, the distance or margin between the nearest data vectors in the two classes is maximized [6]. In general, linearly separable hyperplanes cannot classify input data without some errors. Therefore, we decided to find the separated hyperplane using nonlinear SVM. In the process, the data are transformed into higher dimensional spaces using nonlinear transformations that distribute the data so that linear separation hyperplanes can be found. However, the large dimensionality of feature space leads to inefficient mapping problems, this can be achieved by kernel tricks instead of explicitly computing transformations in feature space. Kernel tricks are played by replacing kernel functions instead of internal products of two transformed data vectors, and using kernel tricks significantly reduces computational tasks [7].

Applying the process above, we consider SVM-based classification for X-ray images with different kernels such as linear, polynomial, radial basis, and sigmoid. The comparative performance of SVM with respect to different kernels is performed. In our SVM, principle component analysis (PCA) is performed here for feature reduction. Our results show that SVMs with RBF kernels achieved the highest classification accuracy, so we decide to use it.

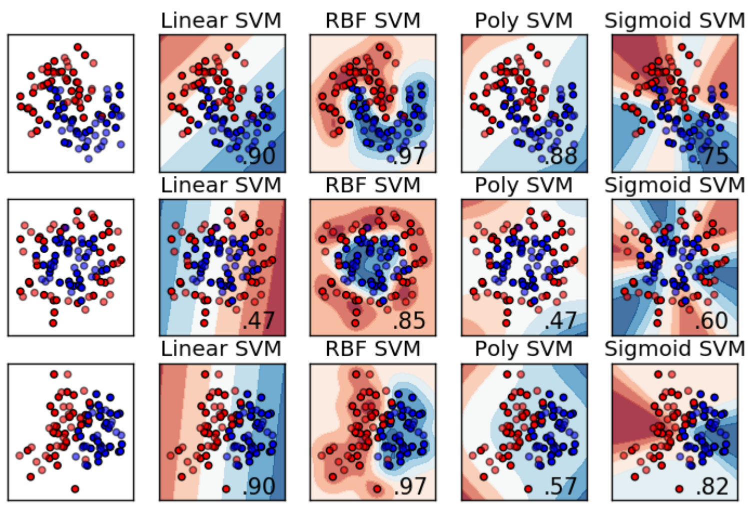

Figure 2: Visualization of different SVM Kernels Use

The figure above shows monochromatic training points and translucent inspection points. The lower right score indicates the classification accuracy of the test set.

4.2 Artificial Neural Network (ANN)

Artificial neural network (ANN) is a machine learning algorithm inspired by biological neural networks. Although ANN can have a high time and computational complexity, it is a powerful classification model because complex non-linear functions can be learned as a composition of simple processing units [4, 8].

ANN consists of the processing units, which are also known as nodes or artificial neurons. Nodes are connected by edges from which nodes send and receive signals. The output of each node is computed by complex non-linear functions of the input signals and transmitted to other nodes of the next layer. Each edge has a weight that adjusts and determines the strength of the signal at a connection. Nodes are aggregated into the given layer. If more than one hidden layer is set to create a model, it is called a multi-layer neural network. The nodes in each layer operate on the given activation function, such as linear function, sigmoid function, and sign function. Because multi-layer neural networks perform well with any type of classification, we selected a multi-layer neural network approach for the ANN model. The graphical representation of ANN is presented in the below figure.

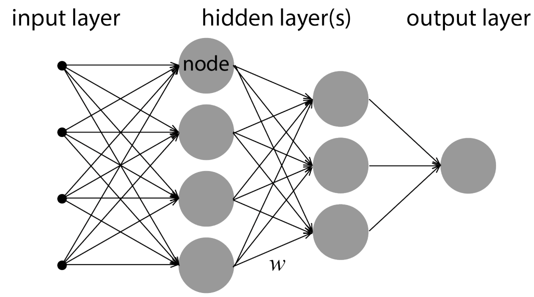

Figure 3: Graphical representation of ANN

4.3 Convolutional Neural Network (CNN)

A convolutional neural network (CNN) is one of the applications of neural networks which is widely used to analyze images. One of the key features of CNN is known as space invariant artificial neural networks because of the convolution kernels or filters to capture the important features of the image while reducing the entire data size [4, 9]. The convolutional layer is the most important element of the CNN. The layer is composed of a set of kernels or filters that can learn with a small region but it can be extended to entire data with forwarding process. The graphical representation of CNN is presented in the below figure.

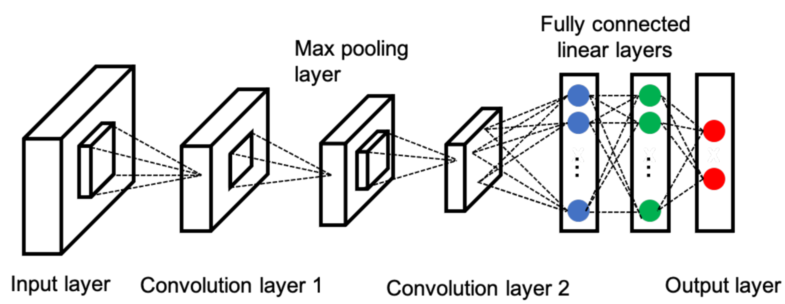

Figure 4: Graphical representation of CNN

4.4 Transfer Learning

4.4.1 Concept of transfer learning

Transfer learning is a technique that uses the knowledge gained from pre-trained models to apply it to different but related problems. Since the state-of-the-art net for images is a convolutional neural network (CNN), we used the pre-trained models from ImageNet in PyTorch. With the pre-trained models, we only fine-tune the latter layers which contain the output layers. By taking this route, researchers often get better results than the ones they created from scratch. For our particular case, since the model that we created with ANN and CNN resulted in high accuracy, we decided to proceed with transfer learning to compare the result created from scratch to the pre-trained models. To get the best results, we compared our models to GoogleNet, ResNet18, ResNet34, and ResNet 50.

4.4.2 GoogleNet

The network architecture of GoogleNet is quite different than the other state-of-the-art networks, such as VGGNet, ZFNet, and AlexNet. The key difference is that it uses the inception module. The inception module uses 1×1 convolution for the purpose of dimensionality reductions. The most straightforward way of improving the performance of networks is to increase the depths and widths of networks. However, this can lead to an overfitting problem since there will be a large number of parameters with limited labels. Also, the training can be very expensive which can take a vast amount of computational resources. GoogleNet contains a 1×1 convolution with 128 filters and rectified linear activation, a fully connected layer with 1024 units and rectified linear activation, a dropout layer with 70% ratio of dropped outputs, and a linear layer with softmax loss as the classifier [10]. The figure 3 visualizes the idea of the inception module.

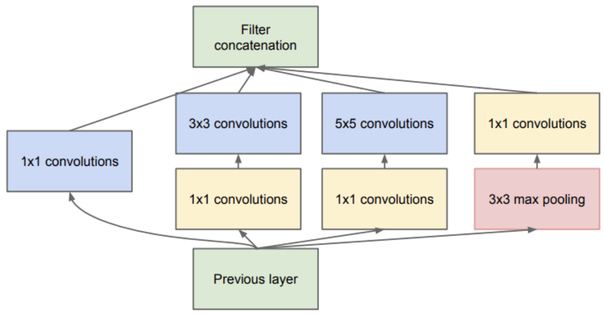

Figure 5: Inception module with dimension reduction.

4.4.3 ResNet

ResNet is an abbreviated term from Residual Neural Network. The motivation for creating such a model was that when deeper networks start to converge, vanishing/exploding of gradients can occur. This has been largely addressed by the use of Stochastic Gradient Descent(SGD). However, there exists another problem which is degradation. The degradation problem occurs when we add more layers to the net. We expect the accuracy of training and test set to be better when we add more layers but accuracy saturates and then degrades rapidly [11]. ResNet introduces deep residual learning framework. The basic understanding of deep residual learning is that we add shortcut connections to other layers by skipping one or more layers. The reason behind adding these shortcut connections is that if we add more layers that operate identity mapping, the network with added layers should not have a higher training error than the original network. With this idea, if the network created is very similar to the added layers, we can safely skip the connections that have a similar outcome.
The following figure shows the simple residual learning block:

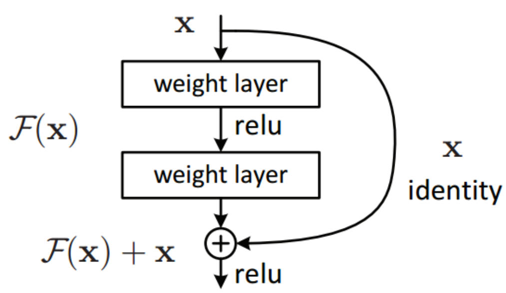

Figure 6: Single residual learning block

5 Data pre-processing

The chest X-ray images contained three parts: train, val, and test. The labels for the X-ray images were either Normal or Pneumonia. Since the dataset was not balanced we wanted to explore the difference between models we obtain from ANN, CNN, and transfer learning by training on both balanced and unbalanced datasets. Originally, the dataset contained 1341 normal and 3875 pneumonia images for the train set, 8 normal and 8 pneumonia images for the validation set, and 234 normal and 390 pneumonia images for the test set. As we can see, the dataset was not balanced and only contained 8 images for each label in the validation set. To experiment models in different settings, we created both balanced and unbalanced dataset. A key thing to note here is that we wanted to train on balanced and unbalanced dataset and compare the accuracy using same validation and test set. This way, we can verify which circumstance will give better result in test set. The training and test set were divided by 8:2 ratio and training and validation set also were divided by 8:2 ratio in balanced dataset. The following is the number of images in each phase in balanced and unbalanced dataset, respectively:

Number of images (balanced data)		
	Normal	Pneumonia
Train	1012	1012
Val	254	254
Test	317	317

Number of images (unbalanced data)		
	Normal	Pneumonia
Train	1012	3702
Val	254	254
Test	317	317

The following is different data-preprocessing steps taken for different models:

- For SVM, the images were converted to a grayscale value and integrated into an image size of 150x150. The images were then normalized by 255 and principle component analysis (PCA) is performed for feature reduction from 3-dimension to 2-dimensions for RBF parameters.
- For ANN, CNN, and transfer learning, we prepared our data using PyTorch's data transformer.
- For ANN, the data was normalized using a mean of [0.5, 0.5] to use grayscale images instead of RGB images, considering a high time and computational complexity of ANN. The batch size was set to 8.
- For CNN and transfer learning, we normalized our data using the mean of [0.485, 0.456, 0.406] and standard deviation of [0.229, 0.224, 0.225] since there are three channels for RGB images. The batch size was set to 8 both CNN and transfer learning.
- A key note on the validation set and test set is that for ANN, CNN, and transfer learning, we wanted to compare the accuracy of the models trained on balanced and unbalanced set. Therefore, just for the comparison part, the number of epochs and batch size were set equal across the models.

The following images are the examples of original image, transformed RGB image, and transformed grayscale image:

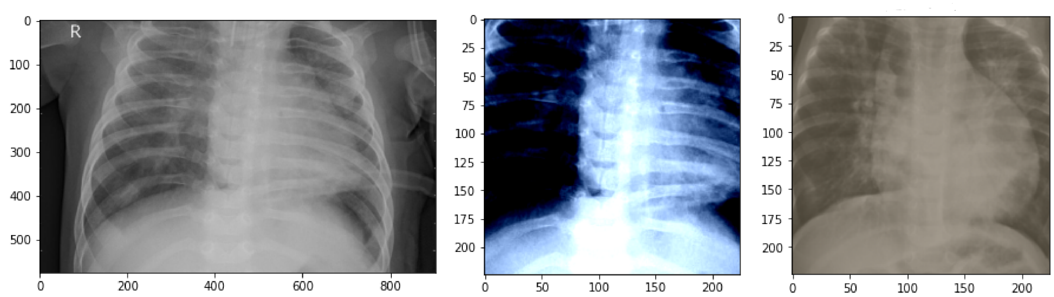

Figure 7: Original x-ray image vs. transformed x-ray images

6 Training and testing with different models

6.1 SVM

One of the challenges that we face with SVM classification is its time complexity for training. In fact, SVM took approximately five to six hours to classify all image data through resizing and reshaping. Instead, we were able to obtain great result values by continuous comparisons for faster image processing and efficient use of algorithms. For example, each kernel has its own essential parameters, especially RBF kernels we choose to use typically have cost and gamma values as hyper-parameters, and we obtained the optimal values.

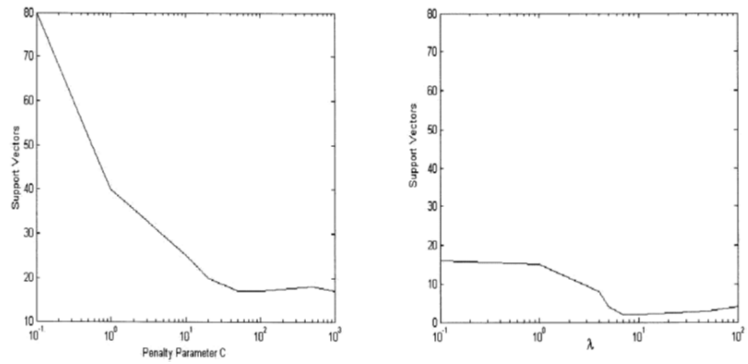

Figure 8: Number of support vectors against penalty parameter cost and gamma

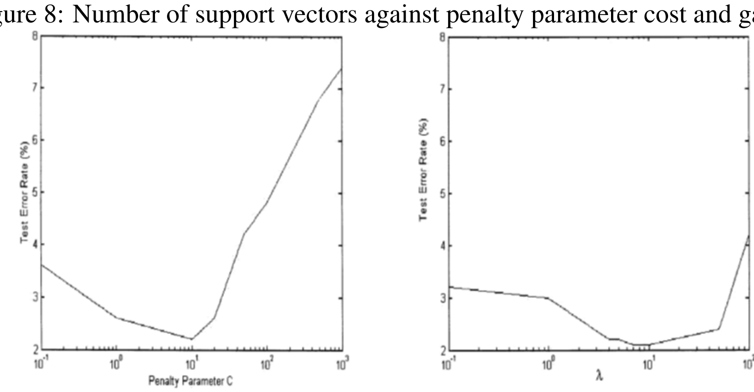

Figure 9: Test error against penalty parameter cost and gamma

In RBF kernel, cost determines how many data samples are allowed to be placed in different classes [12]. We confirmed that setting the C value low leads to the presence of outliers in most cases. On the other hand, higher C values led to finding general decision boundaries by observing presence of outliers more carefully. In addition, gamma determines the distance at which one data sample exerts influence [12]. Gamma is associated with the standard deviation of the Gaussian function which leads to smaller standard deviation with higher gamma values. We also confirm that the larger gamma, the shorter the distance at which one data pointer exerts influence, while the lower gamma, the larger it is. Eventually, the final model of the SVM diagnostic tool was selected with optimal cost and gamma values.

Test Accuracy and the best Parameters			
	Accuracy	Cost(C)	Gamma(γ)
Balanced	0.953	10	0.0005
Unbalanced	0.942	5	0.005

6.2 ANN

Artificial neural network (ANN) models with different hidden layer sizes and node sizes were created. Across the cases, responsive activation functions (RELU) were used for activation and 0.001 learning rate was employed. The number of hidden nodes were set to reasonable values based on the previous literature, which suggested that a large number of hidden nodes may degrade the generalization ability of a network because large hidden nodes may have spurious connections [13]. For case 1, one input layer, four hidden layers, and one output layer were used. The number of nodes for each hidden layer was set to 256, 128, 128, and 64, respectively. For case 2, we used four hidden layers with each node size 512, 256, 128, and 64, respectively. Lastly, for case 3, we used three hidden layers with each node size 256, 128, and 64, respectively. After each ANN model was created, the model was trained using training data and validated using validation data with 20 epochs. All these experiments were conducted separately for balanced and unbalanced training data. The below figures show the training loss of models using balanced and unbalanced training data, separately for the three cases.

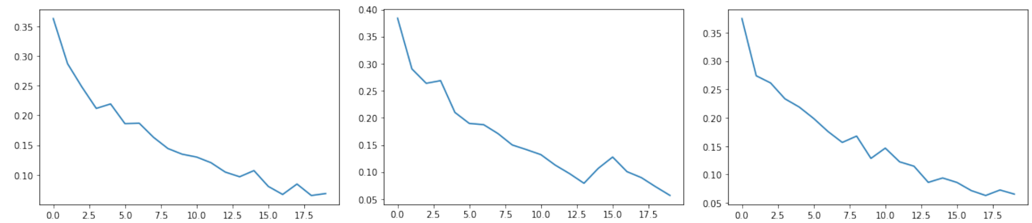

Figure 10: Training loss of balanced data. left: case 1; middle: case 2; right: case 3

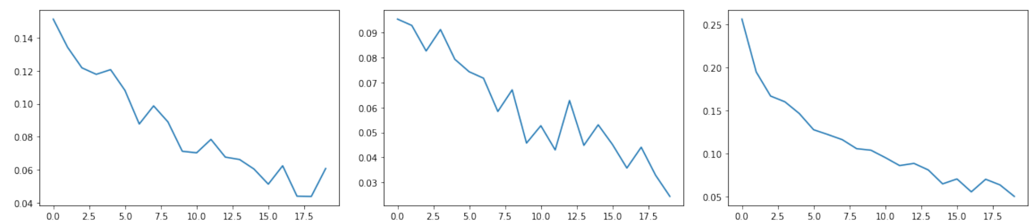

Figure 11: Training loss of unbalanced data. left: case 1; middle: case 2; right: case 3

For each scenario, the model with the highest validation accuracy was used to predict the output for test data. The below table shows the test accuracy of the three cases for both balanced and unbalanced data. For our data, ANN model performed well when trained using a balanced set than an unbalanced set in case 2 and case 3, while it performed well with the unbalanced training data in case 1. The test accuracy of the model trained with the balanced data was highest in case 3. Although the test accuracy of the model trained with the unbalanced data set was highest in case 1, the difference in the test accuracy between the cases was not large. Thus, the case 3 model was selected as the final model.

Test Accuracy of ANN		
	Balanced	Unbalanced
Case1	0.909	0.937
Case2	0.937	0.934
Case3	0.948	0.934

6.3 CNN

Convolutional neural network (CNN) model in our project is composed of combination of two 2D convolution layers (input dimension = 3, output dimension = 3, kernel size = 4 and 3) and 2D max-pooling (kernel size = 4). Then, a linear transformation with the RELU as an activation function result in 2 final outputs. The number of epoch is 20 and the model is evaluated and saved after computing the accuracy through the validation data at the end of each epoch. To find the best model with higher accuracy, the number of linear layer, the output dimension of the convolutional layer, and the number of node in linear transformation are changed and models are tested. For the CNN, we tested three cases with different parameter setting. Case 1 has original model as a reference. Case 2 model has output dimension as 2 for the second convolutional layer. This approach was adopted to prevent the overfitting of the model, and such change leads better accuracy when dealing with test data. Lastly, case 3 model has another linear layer. We expect that such additional layer can increase the accuracy. To prevent overfitting as well as case 2, we reduce the number of node in linear layers. The below table shows the test accuracy of the three CNN models for both balanced and unbalanced data.

Test Accuracy of CNN		
	Balanced	Unbalanced
Case1	0.905	0.947
Case2	0.893	0.936
Case3	0.901	0.938

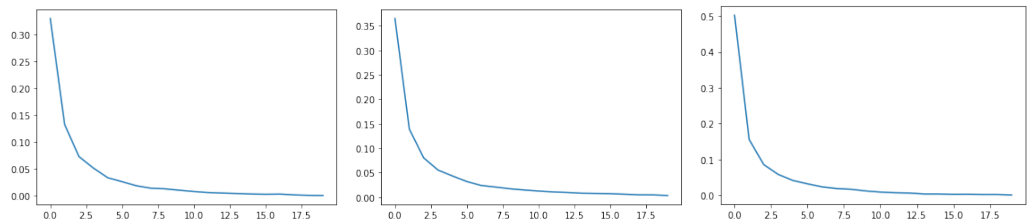

Figure 12: Training loss of balanced data. left: case 1; right: case 2

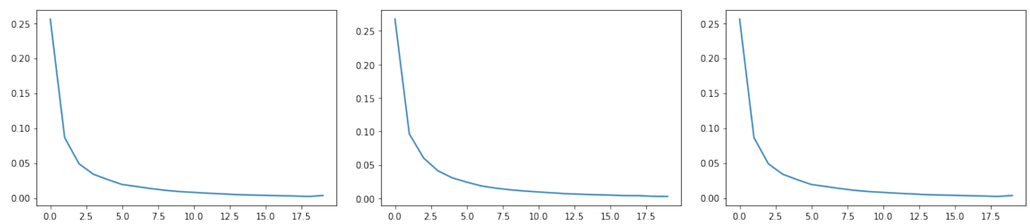

Figure 13: Training loss of unbalanced data. left: case 1; middle: case 2; right: case 3

6.4 Transfer learning (GoogleNet, ResNet)

Both GoogleNet and different versions of ResNet were trained using both balanced and unbalanced dataset. The main reason for training these models in both balanced and unbalanced conditions is to experiment how it will affect both training phase and test accuracy from those models. For convenience, the following figures only contain training loss of GoogleNet and ResNet18, respectively, trained on balanced and unbalanced dataset:

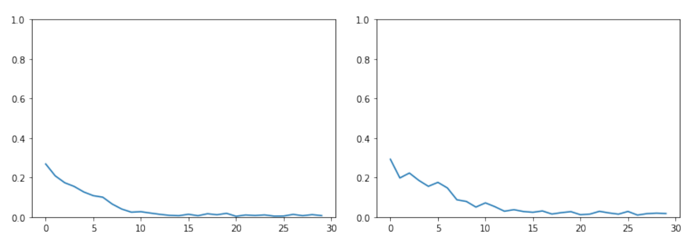

Figure 14: Training loss of balanced dataset from GoogleNet and ResNet18

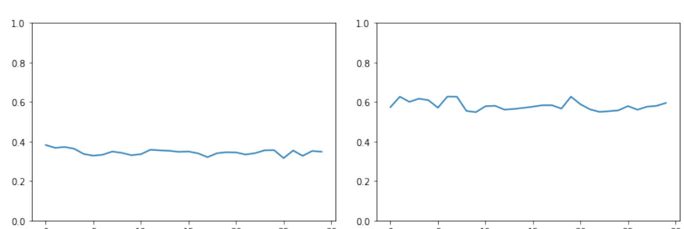

Figure 15: Training loss of unbalanced dataset from GoogleNet and ResNet18

As we can infer from the figures, models trained on balanced dataset shows downtrend of training loss in each epoch. On the other hand, models trained on unbalanced dataset maintains relatively high training loss than the ones trained on balanced dataset and do not show downtrend of the losses.

7 Comparing accuracy from different models

To control possible effects of different hyperparameters between the models and keep the output comparable and consistent, we set the same batch size 8 and the same number of epoch 20 for ANN, CNN, and pre-trained models. The accuracy of the models is shown in the below table.

Test Accuracy of the models		
	Balanced	Unbalanced
SVM	0.953	0.942
ANN	0.948	0.934
CNN	0.905	0.947
GoogleNet	0.937	0.937
ResNet18	0.945	0.932
ResNet34	0.943	0.931
ResNet50	0.935	0.931

8 Conclusion and Future works

In this project, we tested and compared various models (SVM, ANN, CNN, and transfer learning) to diagnose pneumonia based on the chest x-ray images. The main results of the current project are twofold. First, as shown in section 7 "comparing accuracy from different models", SVM showed the highest accuracy (0.953), followed by the ANN (0.948) and ResNet (0.945), all trained by the balanced data. Although we did not formally hypothesize any superiority or inferiority of specific models, we expected that CNN and transfer learning commonly used for image data would overwhelm SVM and ANN. The finding can be attributed to that ANN and SVM may perform well enough for our data. Because features in the chest x-ray image are relatively simple compared to the image of complex object detection task, ANN and SVM may be able to diagnose pneumonia well, even though they are not generally suitable for complex image data.

Second, SVM, ANN, GoogleNet, and ResNet showed higher accuracy when trained by balanced data than the unbalanced data. This finding may be because when a model is trained on unbalanced data, the model likely predicts the class in the test data as the higher portion in the training data. Also, in contrast to other models, CNN showed high accuracy when trained by the unbalanced data. Because the reason why CNN showed high accuracy with unbalanced training data is not clear, future works to understand such findings will be needed.

We can say that these machine learning models created in our project can be used as a diagnostic tool for a real application. This because the test accuracy of all models exceeds at least 90%, and these models are simple, inexpensive, and fast to diagnose pneumonia using chest x-ray images.

Although we adopted various widely used machine learning algorithms to detect pneumonia and compared the accuracy, this project offers room for further improvement for future steps. First, changing the parameters (i.e., the number of layers, the number of nodes, dimension of data) may improve accuracy. Thus, a future work using more various parameters would be beneficial to identify the characteristics of the models when it is applied to current x-ray image data. Furthermore, future work can explore which parameter set is proper to each model for better model performance. Only empirically changing the parameters does not necessarily guarantee the right direction in the approach. Therefore, a library or module that can help much effectively tuning the model would be useful to improve our models.

Second, future studies can explore the pre-processing procedure of the x-ray image data. In the current project, we transformed the dimension of image data for different models for data size reduction and convenience. Also, we changed the image's resolution to reduce the computational burden. It would be helpful to investigate further the impact of such data transformation on model training for future steps. Application of different dimensionality reduction algorithms for images such as t-Distributed Stochastic Neighbor Embedding (t-SNE) or autoencoders may enhance the time complexity with a similar performance of the models.

Each member's contribution to the project

All members in our group contributed equally to the project by splitting the tasks. The detailed contributions of each member are as follows:

- **Choi, Moon-ki:** Convolutional Neural Network, Motivation, Related work, Comparing accuracy from different models, and Conclusion and future work.
- **Choi, Won Joon:** Transfer learning, GoogleNet, ResNet, Data-preprocessing, Related Work, and Comparing accuracy from different models.
- **Kim, Hyunwoo:** Support Vector Machine, Motivation, Related work, Data logistics, and Comparing accuracy from different models.
- **Lee, Garim:** Artificial Neural Network, Understanding models, Comparing accuracy from different models, and Conclusion and future works.

References

1. Albawi, S., Mohammed, T. A. & Al-Zawi, S. *Understanding of a convolutional neural network* in *2017 International Conference on Engineering and Technology (ICET)* (2017), 1–6.
2. Sagar, A. *Deep Learning for Detecting Pneumonia from X-ray Images* Nov. 2019. `https://towardsdatascience.com/deep-learning-for-detecting-pneumonia-from-x-ray-images-fc9a3d9fdba8`.
3. Kermany, D. S., Zhang, K. & Goldbaum, M. *Large Dataset of Labeled Optical Coherence Tomography (OCT) and Chest X-Ray Images* in (2018).
4. Tan, P.-N., Steinbach, M., Karpatne, A. & Kumar, V. *Introduction to Data Mining (2nd Edition)* 2nd. ISBN: 0133128903 (Pearson, 2018).
5. Song, Q., Hu, W. & Xie, W. Robust support vector machine with bullet hole image classification. *IEEE transactions on systems, man, and cybernetics, part C (applications and reviews)* **32,** 440–448 (2002).
6. Vapnik, V. & Chervonenkis, A. A note on one class of perceptrons. *Automation and Remote Control* **25** (Jan. 1964).
7. Camps-Valls, G. & Bruzzone, L. Kernel-based methods for hyperspectral image classification. *Geoscience and Remote Sensing, IEEE Transactions on* **43,** 1351–1362 (July 2005).
8. Jain, A. K., Mao, J. & Mohiuddin, K. M. Artificial neural networks: A tutorial. *Computer* **29,** 31–44 (1996).
9. Valueva, M. V., Nagornov, N., Lyakhov, P. A., Valuev, G. V. & Chervyakov, N. I. Application of the residue number system to reduce hardware costs of the convolutional neural network implementation. *Mathematics and Computers in Simulation* **177,** 232–243 (2020).
10. Szegedy, C. *et al.* Going Deeper with Convolutions. *CoRR* **abs/1409.4842.** arXiv: `1409.4842`. `http://arxiv.org/abs/1409.4842` (2014).
11. He, K., Zhang, X., Ren, S. & Sun, J. Deep Residual Learning for Image Recognition. *CoRR* **abs/1512.03385.** arXiv: `1512.03385`. `http://arxiv.org/abs/1512.03385` (2015).
12. Mashao, D. *Comparing SVM and GMM on parametric feature-sets* in ().
13. Kwok, T.-Y. & Yeung, D.-Y. Constructive algorithms for structure learning in feedforward neural networks for regression problems. *IEEE transactions on neural networks* **8,** 630–645 (1997).