Evaluation of CNN and SVM Covid Chest X-Ray Classifiers

Rohit Hiraman Rajput
Department of Mechanical and Industrial Engineering (MS Robotics)
Northeastern University
Boston, MA,USA
rajput.ro@northeastern.edu

Troy Gilman
Department of Electrical and Computer Engineering (BS CE)
Northeastern University
Boston, MA, USA
gilman.tr@northeastern.edu

***Abstract*— The spread of the SARS-CoV-2 virus has made the COVID-19 disease a worldwide epidemic. X-ray imaging is a non-invasive technique to identify if individuals have symptoms of disease in their lungs. However, diagnosis by this method needs to be made by a medical specialist, which can limit mass diagnosis of the population. Image processing tools can support diagnosis by ruling out negative cases. Advanced artificial intelligence techniques such as Deep Learning have shown high effectiveness in identifying patterns such as those that can be found in diseased tissue. This study analyzes the effectiveness of a VGG16-based Deep Learning model and SVM (Support vector machine) in the identification of COVID-19 and pneumonia using torso radiographs. It also consists of evaluation between these 2 methods with variation in data size. Results show an increase in CNN accuracy as the training sample size increases, meanwhile the SVM classifier accuracy peaks when training with a sample size of 200 with insignificant improvements to accuracy when training with more samples.**

Keywords—Covid19; pandemic; deep learning; neural networks; X-ray; medical images formatting; classifier

I. INTRODUCTION

Coronavirus variants are enveloped, unsegmented, and positive-sense single-stranded RNA viruses. Six variants of coronavirus are known to cause disease in humans, most of them generally cause mild respiratory disease; however, fatal coronavirus variants have periodically emerged in recent decades, such as the 2002 Severe Acute Respiratory Syndrome Coronavirus (SARS-CoV) and the Middle East Respiratory Syndrome Coronavirus in 2012. In December 2019, the Office of the World Health Organization in China was informed of cases of pneumonia of unknown etiology detected in Wuhan, and a new coronavirus variant, called SARS-CoV-2, was extracted from samples of the lower respiratory tract of several patients [1].

In other viral diseases that affects breathing, such as influenza or SARS, the damage produced to the lungs can be observed using pulmonary X-ray images. So, it is logical to think that this relationship is maintained with COVID-19 patients, since this disease mainly attacks the lungs. However, classic pneumonia patients experience some symptoms similar to those COVID-19 patients in the early stages of the contagion, although with lower virulence. Even so, this fact must be taken into account to correctly diagnose this disease. However, the study of medical images has experienced great progress with the inclusion of Machine Learning systems capable of automatically extracting the necessary characteristics to make a correct diagnosis[2].

Moreover, while in Machine Learning the user gives the system a large number of rules to solve the problem, in Deep Learning the user gives the system a network model and only a few instructions to modify the model when errors occur. However, Deep Learning requires a longer training time compared to Machine Learning. SVM classifiers also work well with shorter fitting times and high accuracy [3,4].

Both Machine Learning models we use in this study require a dataset made up of several images corresponding to ill patients and healthy patients (all of them previously labeled by a professional). Using this knowledge, neural network-based systems are able to automatically analyze those images and extract the characteristics necessary to diagnose the illness. These systems require several design steps such as data pre-processing, correct choice of the network architecture, model training (that sometimes requires supervision), among others. This process is similar with all machine learning classifiers.

So, based on these premises, this study consists of using Machine Learning techniques applied to medical X-ray images of patient's lungs to obtain a classifier aid for COVID-19 diagnosis. It is important to emphasize that there are other imaging tools to detect COVID-19 like RM or CT; however, the objective of this work is not to obtain images from patients, but using an existing dataset that meets all the requirements. To achieve this purpose, a public dataset containing X-ray images from healthy, pneumonia infected, and COVID-19 infected patients from all over the world is used. This dataset is entirely made up of X-ray images and this is the justification of choosing X-ray images as our feature data type. With the data included in the dataset, CNN and SVM models are trained and the classification results and comparisons are presented in the work.

The rest of the paper is divided in the following way—first, the dataset and its preprocessing steps are described in detail. Next, both model architecture are explained including different stages implemented in each. Next, the results obtained from the model training processes are evaluated and comparisons are detailed in the Results and Discussion section. Finally, conclusions are presented.

II. DATASET AND ARCHITECTURE

A. Dataset

We are using a publicly available dataset with X-ray images from healthy, pneumonia and covid-19 patients Collected from various medical professional datasets with publicly available at https://www.kaggle.com/datasets/prashant268/chest-xray-covid19-pneumonia.

The dataset Consists of 1200 images in total with 400 for each class. This was then preprocessed, and 100 randomly selected samples were separated from the dataset for testing purpose. The training sets were randomly selected from the remaining samples with no overlap to best assess the model's performance.

Figure 1 show an example of X-ray image and show dataset visualizations.

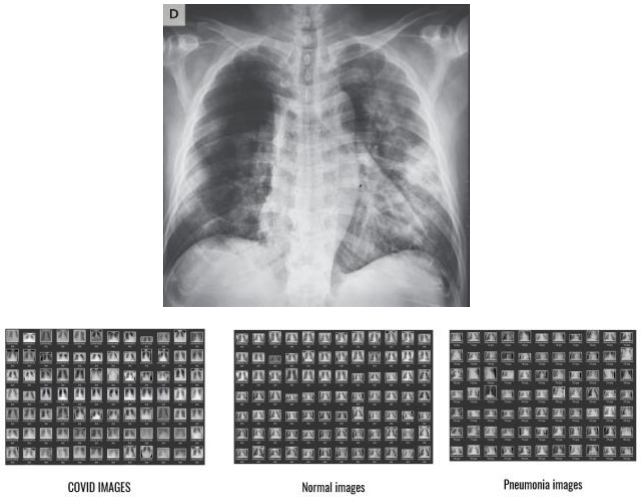

Fig. 1. Example of sample X ray and dataset visualizations

B. Pre-processing

The preprocessing is done step by step as following:

- We resize all the images in dataset to 224x224 as VGG16 requires that shape of input image. Next, we convert the dataset into an array as a 3-dimensional matrix.
- To avoid overfitting to our dataset, we balance the number of samples from each class.
- After that we normalize the pixel values to 0 to 1 to optimize it for our CNN as it increases the accuracy of CNN.
- We shuffle the dataset to avoid class lumpiness in our dataset
- We split off the testing dataset (20%) and store it as an array. The remaining dataset is considered as our train dataset and stored as an array.
- Due to this, we can implement training sample size variation for evaluating model performance.

III. SYSTEM MODELS

For this classification problem we are using two different models which are CNN and SVM. Our Convolutional Neural Network used the VGG16 (visual geometry model) model as its head. VGG16 is a pretrained CNN model that is trained to extract RGB image features. For our SVM, we are using the C-Support Vector Classifion model from Sci-kit and fitting the classifier to our dataset.

Detailed explanations of our models and implementations are explained below:

A. VGG16 NET MODEL[5]

VGG16 was proposed by Karen Simonyan and Andrew Zisserman of the Visual Geometry Group Lab. Figure 2 shows architecture of the system.

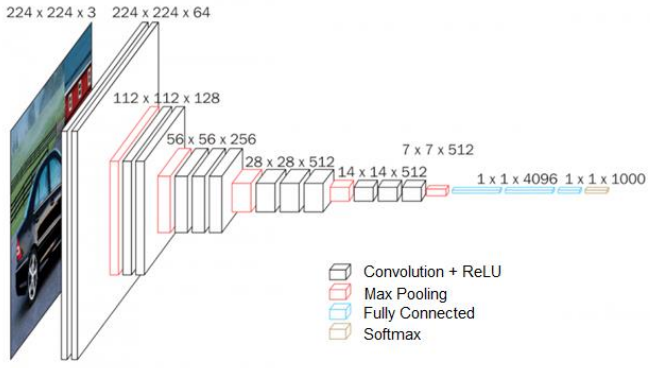

Fig. 2. VGG16 architecture

Architecture walkthrough for the system is as follows:

- The first two layers are convolutional layers with 3x3 filters. First two layers use 64 filters that results in a 224x224x64 volume as same convolutions are used. The filters are always 3x3 with stride of 1.
- After this, pooling layer are used with max-pool of 2x2 size and stride 2 which reduces height and width of a volume from 224x224x64 to 112x112x64.
- This is followed by 2 more convolution layers with 128 filters. This results in the new dimension of 112x112x128.
- After pooling layer are used, volume is reduced to 56x56x128.
- Two more convolution layers are added with 256 filters each followed by down a sampling layer that reduces the size to 28x28x256.
- Two more stacks, each with 3 convolution layers are separated by a max-pool layer.
- After the final pooling layer, 7x7x512 volume is flattened into a Fully Connected (FC) layer with 4096 channels and soft-max output of 1000 classes.
- This structure is used on our own 3 class dataset. Via transfer learning we get good feature extraction.

B. Mathmatical Model

- The ImageNet dataset contains images of fixed size of 224x224 and with RGB channels. So, we have a tensor of (224, 224, 3) as our input. This model processes the input images and outputs a vector of 1000 image feature likelihoods. This vector represents the classification probability for the corresponding classes.

$$C = \begin{bmatrix} y0 \\ y1 \\ y2 \\ y3 \\ . \\ . \\ y990 \end{bmatrix}$$

- To make sure these probabilities add to 1, we use soft-max function. This soft-max function is defined as :

$$P(y=j \mid \Theta^{(i)}) = \frac{e^{\Theta^{(j)}}}{\sum_{j=0}^{k} e^{\Theta_k^{(j)}}}$$

softmax function

where $\Theta = w_0 x_0 + w_1 x_1 + \ldots + w_k x_k = \sum_{i=0}^{k} w_i x_i = w^T x$

- After this we take the 5 most probable candidates into the vector. An example of random values and ground truth vector is defined as the following:

$$C = \begin{bmatrix} 780 \\ 0 \\ 1 \\ 2 \\ 990 \end{bmatrix} \quad G = \begin{bmatrix} G0 \\ G1 \\ G2 \end{bmatrix}$$

- Then the error function as follows:

$$E = \frac{1}{n} \sum_k min_i d(c_i, G_k)$$
$$where\, d = 0\, if\, c_i = G_k\, else\, d = 1$$

- If all the categories in ground truth are in the Predicted top-5 matrix, the loss becomes 0.

C. Model parameter input and functions used for CNN:

- For testing of CNN model, we use following parameters for our experiment:
  - Test Data n = 100
  - Train Data n = 20, 50, 200, 500
  - Epochs = 20
  - Batch Size = 10
  - Learning Rate = 0.001
- We use TensorFlow and Keres for defining and training this model. We use Scikit-learn's machine learning repo for performance analysis, matplotlib for plotting and OpenCV for image processing.
- We use "binary cross-entropy" loss for this problem and the Adam optimizer for optimization.

D. SVM model

Although Support Vector Machines (SVM) can be used as a classification approach, it is also employed in regression problems. It can easily handle multiple continuous and categorical variables. SVM constructs a hyperplane in multidimensional space to separate different classes. SVM generates optimal hyperplane in an iterative manner, which is used to minimize error. The core idea of SVM is to find a maximum marginal hyperplane (MMH) that best divides the dataset into classes. Figure 3 shows simple visualization of the model used in this evaluation:

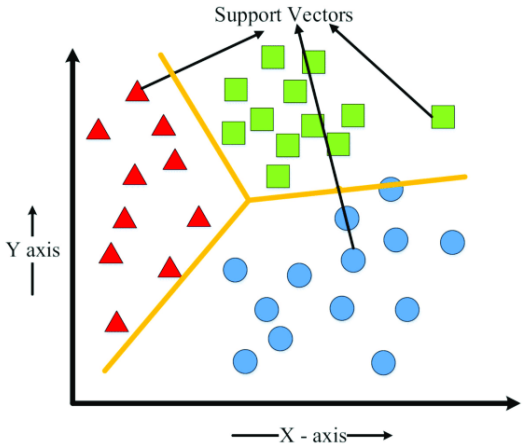

Fig. 3. Simple visualization of SVM model

Algorithms such as Support Vector Machines were designed for binary classification and do not natively support classification tasks with more than two classes. We use a one-to-one approach for using binary the classification algorithm for multi-class classification. This approach splits the multi-class classification dataset into multiple binary classification datasets and fits a binary classification model on each.

E. Model parameter input for SVM:

- For this problem we use Scikit-learn C-Support Vector Classification with Radial Basis Function (RBF) kernel.
- For testing of the SVM model, we used following parameters:
  - Test Data n = 100
  - Train Data n = 20, 50, 200, 500
  - One-to-One scheme for multiclass support

IV. Results and Comparison

A. CNN Results

This section includes results from training our CNN on datasets of 20, 50, 200, and 500 samples. Figures 4-7 show training loss, validation loss, training accuracy, and validation accuracy across 20 epochs for each respective training sample size.

Fig. 4. CNN Training Plot: N=20

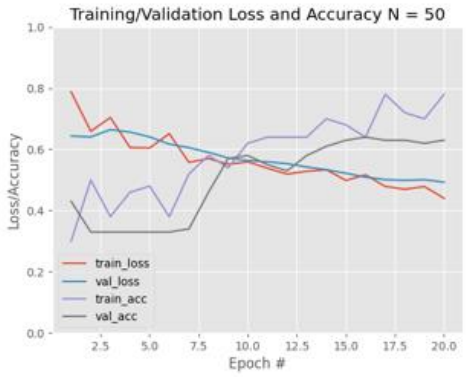

Fig. 5. CNN Training Plot: N=50

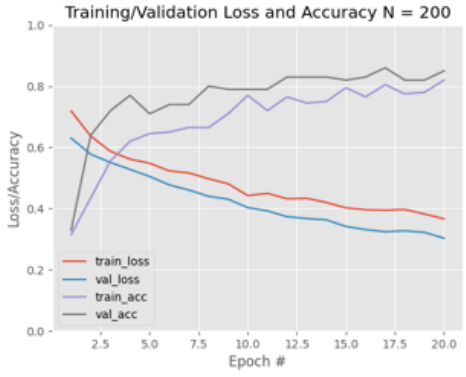

Fig. 6. CNN Training Plot: N=200

Fig 7. CNN Training Plot: N=500

As the number of training samples increases, there is a clear trend of improved performance metrics. The validation accuracy steadily improves from 40% to 90% as the training sample size increases from 20 samples to 500 samples.

Figure 8 shows the confusion matrices for each respective training sample size. The key for these confusion matrices is as follows: 0=normal, 1=covid, 2=pneumonia.

Fig. 8. CNN Confusion Matrices: N=20, 50, 200, 500

There is a clear trend in these CNN confusion matrices. As the number of training samples increases the classification predictions become more accurate. There is sequential improvement across each training sample size.

B. *SVM Results*

This section includes results from training our SVM on datasets of 20, 50, 200, and 500 samples. Figure 9 shows the confusion matrices for each respective training sample size. The key for these confusion matrices is as follows: 0=normal, 1=covid, 2=pneumonia.

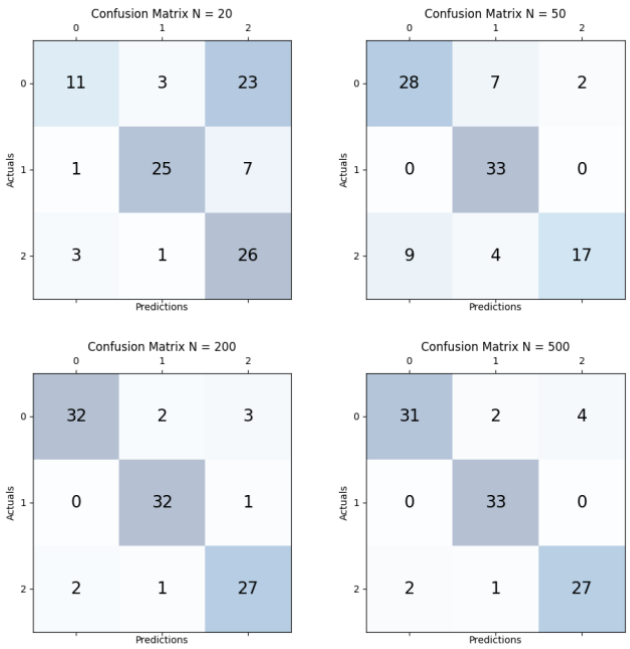

Fig. 9. SVM Confusion Matrices: N=20, 50, 200, 500

There is a clear trend in these SVM confusion matrices. As the number of training samples increases the classification predictions become more accurate. However, there is no significant improvement in the predictions when training on 200 samples vs. 500 samples.

C. *Model Results Comparison*

This section includes a comparison of the validation accuracies between the CNN and SVM models across all training sample sizes. Table 1 shows a comparison between the number of training samples vs. validation accuracy for the CNN and SVM models.

TABLE I. MODEL ACCURACIES COMPARISON

Train Size	**CNN vs SVM Accuracies Across Training Sample Sizes**	
	CNN	***SVM***
N=20	0.40	0.62
N=50	0.63	0.78
N=200	0.85	0.91
N=500	0.90	0.91

[a.] All accuracies are on a 0 to 1 scale

Figures 10 and 11 show plots of the number of training samples vs. validation accuracy for the CNN and SVM models respectively.

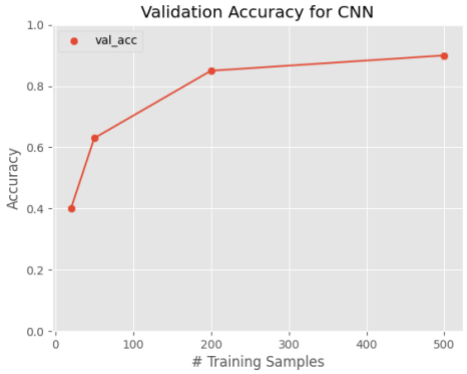

Fig. 10. CNN Validation Accuracy Plot

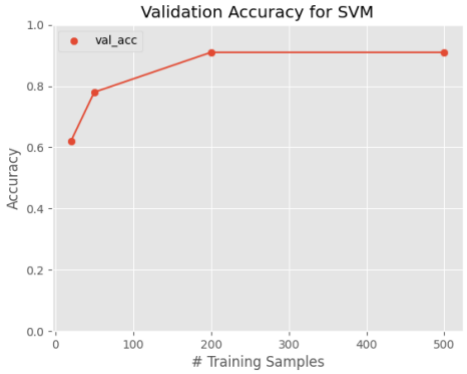

Fig. 11. SVM Validation Accuracy Plot

Although the SVM outperforms the CNN when trained on smaller sample sizes, model performance is equivalent when trained with ~500 samples.

V. CONCLUSION

- A pre-processing stage was done to all the X-ray images due to images being sourced from different machines with different calibrations, which caused a significant variation in the histogram of the images.
- At least ~200 samples are required to train a model to solve this problem (0.85+ accuracy) with 3 classes (normal, covid, pneumonia) for both CNN and SVM model.
- The training time for CNN is longer and its accuracy mostly depends on the dataset provided and its size. On the other hand, SVM achieves its accuracy with smaller datasets.
- SVM slightly outperforms CNN with smaller training sets but equivalent with ~500 samples.
- This approach for both classifiers models could be extended to classify more types of lung issues and decrease the need for medical professionals for this task.

References

[1] Repici, A.; Maselli, R.; Colombo, M.; Gabbiadini, R.; Spadaccini, M.; Anderloni, A.; Carrara, S.; Fugazza, A.; Di Leo, M.; Galtieri, P.A.; et al. Coronavirus (COVID-19) outbreak: What the department of endoscopy should know. Gastrointest. Endosc. 2020, 92, 192–197.

[2] Ker, J.; Wang, L.; Rao, J.; Lim, T. Deep learning applications in medical image analysis. IEEE Access 2017, 6, 9375–9389.

[3] Luna-Perejón, F.; Domínguez-Morales, M.J.; Civit-Balcells, A. Wearable fall detector using recurrent neural networks. Sensors 2019, 19, 4885. K. Elissa.

[4] Domínguez-Morales, M.J.; Luna-Perejón, F.; Miró-Amarante, L.; Hernández-Velázquez, M.; Sevillano-Ramos, J.L. Smart Footwear Insole for Recognition of Foot Pronation and Supination Using Neural Networks. Appl. Sci. 2019, 9, 3970.

[5] Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. arXiv 2014, arXiv:1409.1556.

[6] Civit-Masot J, Luna-Perejón F, Domínguez Morales M, Civit A. Deep Learning System for COVID-19 Diagnosis Aid Using X-ray Pulmonary Images. *Applied Sciences*. 2020; 10(13):4640.