Transfer Learning-Based CNNs vs Independently Designed CNN for the Classification of Chest X-Ray Images

1st Walid Rahman
Columbia University School of Engineering and Applied Science)
Dept. of Biomedical Engineering
New York, NY
wr2255@columbia.edu

***Abstract*—Convolutional neural networks (CNNs) allow for the classification of a wide variety of medical images captured by various imaging modalities, such as MRI, CT, and X-ray. CNNs offer an efficient classification method that can approach, match, or supersede the precision and recall in classification achieved by doctors. Kermany et al. used transfer learning techniques to develop a CNN that classified images of macular degeneration. They trained a CNN on an ImageNet dataset of 1000 categories of images. They then took the weights determined by the CNN and used them in a new CNN in which they inputted their medical images. In this binary classification task of determining if a chest X-ray image shows pneumonia or not, the CNN achieved an accuracy of 92.8 percent, sensitivity of 93.2 percent, and specificity of 90.1 percent. In this project, a CNN without transfer learning approaches and two transfer learning based CNNs were designed in order to determine if transfer learning based CNNs can offer the same precision, recall, and accuracy in the classification of medical images as an independently designed CNN. A CNN using Inception V3 performed just as well as the independently designed model in terms of recall and precision while a CNN using VGG16 performed significantly worse than both. Additionally, multiple iterations of the independent CNN were designed to achieve maximum optimization between precision and recall. Thus, transfer learning approaches for medical image classification can be just as good as independent models, but independent models may be able to attain higher precision in classification than transfer learning based models.**

I. Introduction

Convolutional neural networks (CNNs) allow for the classification of a wide variety of medical images captured by various imaging modalities, such as MRI, CT, and X-ray. CNNs offer an efficient classification method that can approach, match, or supersede the precision and recall in classification achieved by doctors. CNNs can be designed with any number of layers and types of layers, such as convolutional, pooling, normalization, fully connected layers, etc. With every layer, weights are determined by the neural network and eventually modified during back-propagation steps to minimize loss and improve accuracy of a model. The weights of one CNN are in part determined by the image dataset it was trained on. However, if a CNN is trained on a variety of image datasets, its weights and architecture may be useful for the training and testing of another dataset. The use of one pretrained architecture and/or set of weights on for a new dataset is called transfer learning. The pretrained architecture is used to train and test a new input of data. This is extremely useful as it allows for the reusability of neural networks. Additionally, pre-trained neural networks can serve as part of a larger architecture to further simplify the development of neural networks. These pre-trained CNNs should offer significant generalizability to be used for other image datasets. However, one big problem for pre-trained CNNs available for use, such as Inception, VGG, and ResNet, stems from the image dataset that was used to train them. They are typically trained on the imagenet dataset and are trained such that they hold weights based on training on the imagenet dataset. Thus, such networks are better suited for determining features of images similar to the ones in the imagenet dataset. The imagenet dataset doesn't contain medical images obtained in medical image modalities such as X-Ray, CT, and MRI. Thus, the imagenet weights may not allow for ideal classification and segmentation of medical images. Regardless of this potential issue, the networks can still detect image features such as edges, background-foreground differences, and shapes. Thus, they can still be very useful for medical image classification.

Kermany et al., in Identifying Medical Diagnoses and Treatable Diseases by Image-Based Deep Learning, used transfer learning techniques to develop a CNN that classified images of macular degeneration. They trained a CNN on an ImageNet dataset of 1000 categories of images. They then took the weights determined by the CNN and used them in a new CNN in which they inputted their medical images. The CNN then used those pre-trained weights as well learned weights through training to ultimately produce classification outputs for their macular degeneration with an accuracy of 96.6 percent, sensitivity of 97.8 percent, and specificity of 97.4 percent. They applied this model to a dataset of chest X-ray images. In this binary classification task of determining if a chest X-ray image shows pneumonia or not, the CNN achieved an accuracy of 92.8 percent, sensitivity of 93.2 percent, and specificity of 90.1 percent [1].

In this project, the same chest X-ray image dataset used by Kermany et al. was used to create three different CNNs: two using transfer learning and another not using transfer learning (independently designed CNN). Each CNN was tasked with classifying the image dataset for showing either pneumonia or the absence of pneumonia. The purpose of designing the transfer learning-based CNNs and an independently designed CNN was to determine which CNN architecture could deliver higher performance overall. Additionally, the performance of the independently designed CNN was be iteratively improved to bring its performance close to or higher than the performances of the transfer-learning based CNNs. It is expected that the transfer learning-based CNNs will still deliver higher performance as they will utilize architectures that have already been trained to be efficient classifiers. This may highlight the advantages that transfer learning-based CNNs offer over independently designed CNNs and may also speak to the generalizability of transfer-learning based architectures for the classification of a wide variety of medical image datasets. If the independently designed CNN significantly outperforms the transfer learning-based CNNs, it may offer insight as to how different CNN architectures can be used in place of predesigned and pretrained architectures used in transfer learning-based approaches.

II. Methods

A. Dataset Acquisition and Processing

The image dataset is provided by Kaggle contributor Paul Mooney. It is the same dataset used by Kermany et al. and contains a total of 5856 chest X-ray JPEG images. Of this total, 1583 are labeled as normal to specify the absence of pneumonia, and 4273 are labeled as pneumonia to specify the presence of pneumonia. The downloaded data was stored into different training set and testing set folders by Paul Mooney. In the training set, there are 1349 normal images and 3883 pneumonia images. In the testing set, there are 234 normal images and 390 testing images [2]. All data was stored locally and accessed via code. A function named "image processor". This function takes the image directory as an input, determines the label of each image based on file name, and stores the labels in in an array. Then, it converts each image corresponding to the labels into an array and stores it in array holding all image arrays. Using this function, all training and testing data was stored into arrays that allowed for use in convolutional neural networks. It was noted using plotting that image dataset wasn't truly even between images classified as showing pneumonia versus images classified as not showing pneumonia. Thus, it was determined that precision, recall, and an F1 score were the best metrics to determine the effectiveness of any classifiers designed. A confusion matrix was, therefore, designed for each set of results.

B. Model Designs and Deployment

Independent Model: Six different architectures were designed without using any previously trained architecture. Each of these architectures differed in the amount of 2D convolutional layers, pooling layers they had, the use of batch normalization, and the number of layers overall. Each network also contained a dropout layer and a dense layer for final classification. Each of these models were initialized using a Sequential model from the Keras API. A summary of each model can be found in the associated python notebooks. The RMSprop optimizer was used because it has shown excellent results with minibatches. RMSprop normalizes gradients per iteration using the magnitude of recent gradients. It speeds up stochastic gradient descent by increasing horizontal learning speed. The learning rate was changed multiple times until it was determined that an initialized learning rate of .00005 obtained the best results. Callbacks were utilized to adjust the learning rate per minibatch. The batch size of 32 gave the best results on an initial architecture (called "arch 1") and so was used for all subsequent models. Binary cross-entropy loss was used as the loss function. Models named "arch 1", "arch 2", and "arch 3" used 6 epochs for training. Models named "arch 4", "arch 5", and "arch 6" used 10 epochs. The number of epochs were optimized after multiple training sessions.

Inception V3 and VGG16 Models: The Inception models are designed by Google and have been trained using imagenet. In this project, Inception V3 and VGG16 were used in two ways: (1) the CNN architecture was employed and additional layers were added in a manner similar to Kermany et al in order to produce a binary classification at the final layer, and (2) the CNN archiecture and imagenet weights from training on the imagenet dataset was employed with the same additional layers for binary classification. The purpose of using these two methods was to see if the use of the imagenet weights would make a difference in classification performance. When only the CNN architecture was used, the weights were determined by training on the chest X-ray dataset. RMSprop was again used as the optimizer and a learning rate of .00005 was determined as the best learning to start at. Callbacks were utilized to adjust the learning rate per minibatch. A batch size of 32 gave the best results. Binary cross-entropy loss was used as the loss function. Six epochs were used for each model since model performance didn't improve after 6 epochs. All models were trained and tested using Jupyter Notebooks.

C. Model Performance Metrics

Given the skewed dataset with more images showing the presence of pneumonia than not, it was determined the best metrics for model performance evaluation were recall, precision, F1 score and AUC. First, a confusion matrix based on model test results was made for each model after training. A function called "statmaker" was designed to take the confusion matrix array as an input and output recall, precision, F1 score, and ROC curve with AUC.

III. Results

The results of each model is shown in Figure 1. Accuracy is not the best indicator of model performance in this case binary classification was performed. In this case, precision and recall

are the ideal metrics. The F1 score provides a good estimate of overall model performance. The best performing independent architecture and transfer learning based architecture are highlighted in green while the worst performing architectures are highlighted in orange. Please note that "NW" refers to no imagenet weights being used while "W" means that imagenet weights were used.

Model	Recall	Precision	F1 Score	AUC	Accuracy
Arch 1	0.8153	0.8641	0.8391	0.8008	0.8045
Arch 2	0.8974	0.8537	0.875	0.8205	0.8397
Arch 3	0.8615	0.8773	0.8693	0.8303	0.8381
Arch 4	0.8487	0.8531	0.851	0.8026	0.8141
Arch 5	0.8282	0.8411	0.8346	0.7838	0.7949
Arch 6	0.9872	0.7026	0.821	0.7581	0.731
IncV3-NW	0.9179	0.792	0.8503	0.7581	0.7981
IncV3-W	0.9795	0.71	0.8233	0.6564	0.7372
VGG16-NW	1	0.625	0.7692	0.5	0.625
VGG16-W	1	0.625	0.7692	0.5	0.625

Fig. 1. Model results for each model

Loss and accuracy curves as well confusion matrices for each model are located in each model's respective Python notebook.

IV. Discussion

It is important to establish what defines a good model. Ideally, a good classifier should have a balance between precision and recall. In the case of pneumonia classification using chest X-ray images, recall tells us: of all the images that show pneumonia, how many of those does the model correctly classify as showing pneumonia? Precision then tells us: of those that were classified as showing pneumonia, how many actually show pneumonia? The F1 score is a combination of precision and recall and offers an idea as to the overall performance of the model. The AUC is another good performance metric that tells us how well a model distinguishes between classifications. The higher the AUC, the better the classifier. The closer the AUC is to 0.5, the worse of a classifier the model is. Accuracy is a measure of how well a model correctly classifies images.

In clinical settings where these models can be deployed, a model achieving high recall is very "safe." A model achieving very high recall classifies more images as true positives. In this case, a model achieving very high recall would classify most images as showing pneumonia. However, this may also mean that there will be a lot of false positive classifications. For diagnostic purposes, it may be best to achieve high recall simply because it leaves fewer patients at risk of developing pneumonia. However, many patients will be misdiagnosed as having pneumonia. In that case, other symptoms should be checked for to possibly rule out pneumonia. However, there often exists a trade-off between recall and precision. In this case, a more precise model offers more accurate results. A more precise model will generally be a better overall classifier since it will be able to better classify images that don't actually show pneumonia as not showing pneumonia. In a clinical setting, this can save doctors time, medication, and money. However, some patients actually have pneumonia will be misdiagnosed as not having pneumonia since the recall may be affected. Due to the existence of this trade-off, the best measure of model classification performance is the F1 score. The higher the F1 score, the better a model is at being both precise and sensitive.

A. Independent Models

Of all the independent models, "Arch 2" gave the best performance overall. It had fairly good recall at .8974 and decent precision at .8573. The F1 score was .8750. Thus, this model is both sensitive and precise and can be used in clinical settings with confidence. Although the recall is less than 90 percent, it prevents the misdiagnosis of patients who don't actually have pneumonia as having pneumonia. "Arch 6" achieved phenomenal recall at .9872. Thus, this model was able to classify images as showing pneumonia fairly well. However, it performed terribly at properly classifying the images as indicated by the precision. As good as "Arch 6" is at classifying images as showing pneumonia, it should not be deployed in clinical settings simply due to its lack of precision. The lack of precision would ultimately cost doctors time and money, and the improper use of medication by patients who don't actually have pneumonia.

B. Transfer Learning Based Models

It should first be noted that VGG16 performed terribly in both cases of using Imagenet weights and not using Imagenet weights. The exact same results were achieved for both models. Both models classified every image as showing the presence of pneumonia. Although high recall is good, these models are effectively useless since they offer no actual advantage to doctors. The odd case that both models achieved the same results may have been due improper initialization of weights. This was tested, but the results stayed the same anyway.

The Inception V3 models fared much better than the VGG16 models. The model that used the Imagenet weights achieved recall similar to "Arch 6", but although did similarly terribly when it came to precision. Therefore, for the same reasons as specified for "Arch 6", this model would not be useful to deploy. However, when the Inception V3 architecture alone was used and weights were determined through training on the images, a good recall of .9179 was achieved and a decent precision of .7920 was achieved. The F1 score was .8503, which was almost as higher as the F1 score of model "Arch 2." Thus, both the Inception V3 model without Imagenet weights and model "Arch 2" were good models for chest X-ray image classification. However, it should be noted that the precision of the Inception V3 model without Imagenet weights achieved a precision that was much lower than that of most of the independent models. It's recall was marginally higher than that of "Arch 2" but it was not as precise as "Arch 2". Thus, "Arch 2" is ultimately the better model. The close success of

the Inception V3 model without Imagenet weights indicates that transfer learning based approaches can be a good starting point for medical image classification.

It was highlighted earlier that Imagenet contains many images, but no medical images in the X-ray modality. This is probably the reason why the transfer learning based models using the Imagenet weights weren't very precise. However, when their arhcitectures were trained on the actual medical images, they did fairly well, as in the case of Inception V3. Thus, the approach used by Kermany et al. was partially proven: transfer learning can be a good method for medical image classification, but the Imagenet weights may not be the ideal weights for actual model deployment.

"Arch 2" had a much simpler architecture than the Inception V3 architecture: it had fewer layers and fewer processes to work through. However, it performed better than all the models. This suggests that although transfer learning based approaches are good methods to use to build medical image classifiers, they are not the optimal methods. Independently designed architectures can be better at classifying medical images. Regardless of these findings, it is important to note that currently developed architectures such as Inception V3 are good starting points for any medical professional wanting to develop medical image classifiers for use in practice. With adjustment to architectures or the use of similar architectures, good classifiers can be designed.

REFERENCES

[1] Kermany, D. S., Goldbaum, M., Cai, W., Lewis, M. A., Xia, H., Zhang, K. (2018). Identifying Medical Diagnoses and Treatable Diseases by Image-Based Deep Learning. Cell,172, 1122-1131. doi:https://doi.org/10.1016/j.cell.2018.02.010

[2] Mooney, Paul. (2018, March). Chest X-Ray Images (Pneumonia), Version 2. Retrieved 4/11/2019