Pneumonia Classification using Convolutional Neural Network and EfficientNet

Quynh Le*
Hang Liu*
Jiyeon Song*
Golokesh Patra*
q3le@ucsd.edu
hal064@ucsd.edu
jis036@ucsd.edu
gpatra@ucsd.edu
Halıcıoğlu Data Science Institute
La Jolla, California, USA

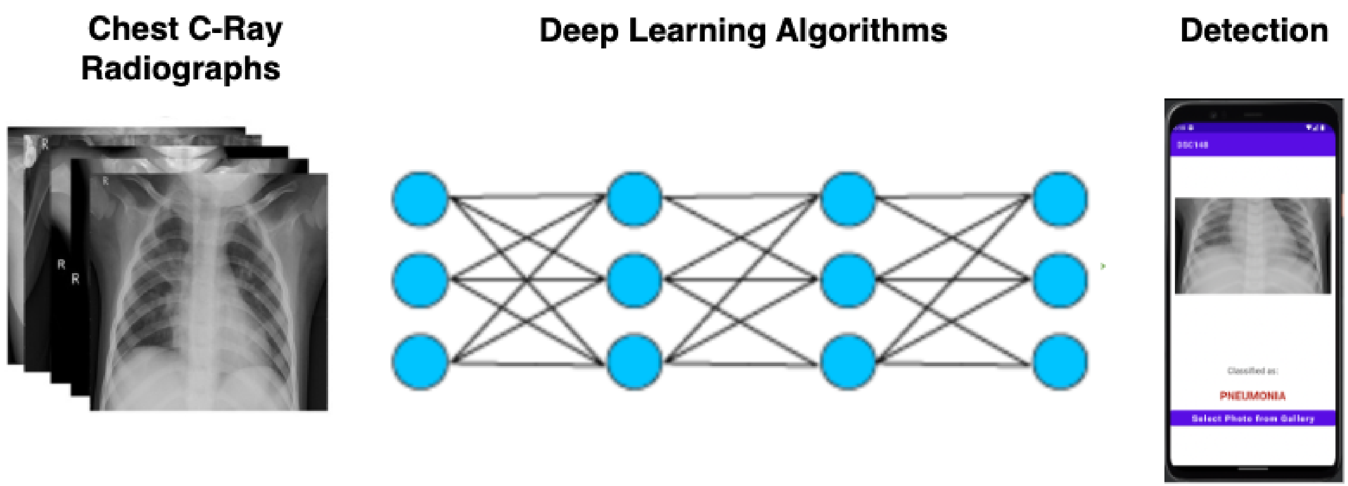


Figure 1: Pneumonia Image classification pipeline

ABSTRACT

Pneumonia is a respiratory disease that affects millions of people worldwide. Early and accurate diagnosis is critical for effective treatment and management of the disease. In recent years, machine learning techniques have been increasingly used for medical image analysis, including the diagnosis of pneumonia. In this paper, we propose a pneumonia classification method using a Convolutional Neural Network (CNN) and the EfficientNet architecture. The proposed method uses chest X-ray images to distinguish between normal and pneumonia cases. We evaluated the proposed methods on a publicly available dataset (on kaggle - Chest X-Ray Images), achieving an accuracy of 76% for the CNN and 88% for the EfficientNet. Our results demonstrate the potential of using deep learning techniques for the accurate diagnosis of pneumonia, which could have significant clinical implications for the early detection and treatment of the disease.

Codes and demo are available at Github repository: https://github.com/radiangle/pneumonia-classification

*All authors contributed equally to this research.

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/XXXXXXX.XXXXXXX

CCS CONCEPTS

• **Classification**;

KEYWORDS

Pneumonia, Data Mining, Convolutional Neural Network, EfficientNet, TensorFlow, Pytorch, Image Classification, Deep Learning

ACM Reference Format:
Quynh Le, Hang Liu, Jiyeon Song, and Golokesh Patra. 2023. Pneumonia Classification using Convolutional Neural Network and EfficientNet . In *DSC 148 '23: UCSD Introduction to Data Mining , March 19, 2023, San Diego, CA.* ACM, New York, NY, USA, 6 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

With the rapid advancement of technology, there has been a significant increase in the amount of visual data that is being generated and consumed. Images are one of the most common forms of visual data, especially in medicine. In order to make sense of the amount of data, it has become essential to develop efficient methods of image classification. In this project, we used Convolutional Neural Networks, and EfficientNet to classify chest X-Rays of normal lungs and lungs with pneumonia.

2 LITERATURE

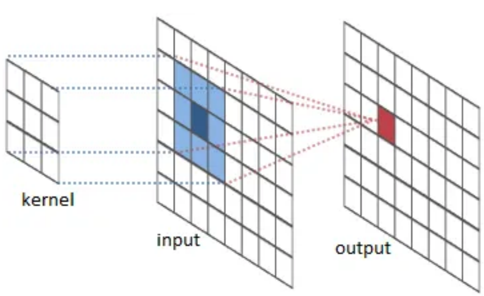
Computer vision and Image classification have this fundamental challenge of having various applications across a wide range of fields. Face identification, object recognition, medical diagnosis, and autonomous driving can all benefit from categorizing or labeling pictures based on their content. In order to discover underlying patterns and characteristics of different classes of images, image classification algorithms utilize large datasets. After exposure to a large number of labeled photos during training, they learn to generalize from this training data and successfully classify unseen images. Additionally, by modifying an image classification model's architecture, hyperparameters, and/or training methods, the algorithms can be made to work well on a particular dataset. Overall, image classification is a critical task in computer vision, and advances in this area have led to significant progress in many fields.

The technique is typically carried out using machine learning algorithms effective at processing visual data, such as convolutional neural networks (CNNs). CNNs are one of the many deep-learning algorithms using convolutional layers to learn and extract complicated image characteristics. Often used for image classification, their ability to accurately identify images and learn crucial properties from input data makes them popular. The EfficientNet algorithm is another well-liked deep learning technique for classifying images. It is made to be computationally effective while achieving cutting-edge accuracy in classifying images. As a result, image classification is used in many industries such as medical, self-driving cars, and security, where precise visual data classification is necessary to make critical decisions.

As a part of this Project, we would consider **CNN** and **EfficientNet** Algorithms.

2.1 Convolution Neural Networks (CNN)

(1) Convolution Layer :

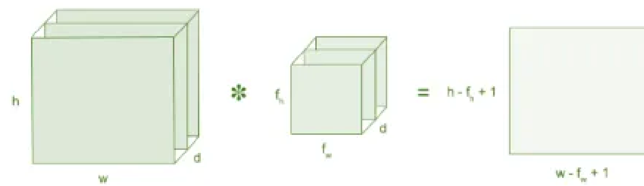


The initial stage of a Convolutional Neural Network (CNN) is referred to as the convolutional layer, which employs small patches of input information to extract characteristics from an input image. The convolution process maintains the connection between pixels and acquires knowledge about image features through a filter or kernel. It's mathematical operation takes two inputs such as image(matrix) and a filter or kernel.

$$An\ Image\ matric\ of\ Dimensions\ (\mathbf{h\ x\ w\ x\ d})$$

$$A\ filter\ (\mathbf{f_h\ x\ f_w\ x\ d})$$

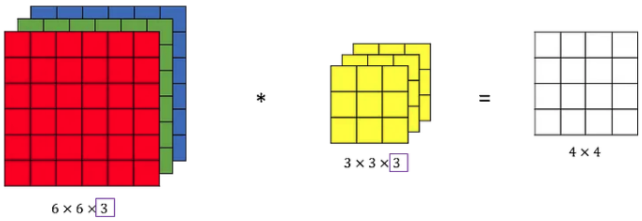
$$Outputs\ a\ volume\ dimension\ (\mathbf{h}\ -\ \mathbf{f_h}\ +\ \mathbf{1})\ \mathbf{x}\ (\mathbf{w}\ -\ \mathbf{f_w}\ +\mathbf{1}\ x\ 1$$



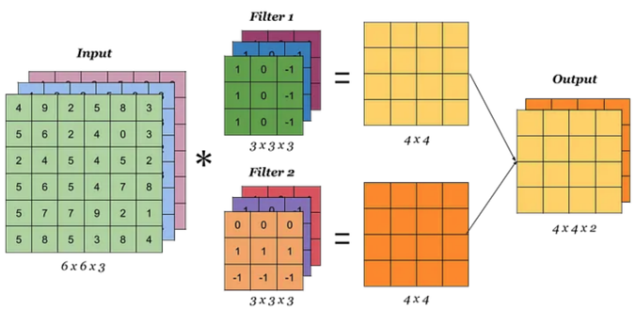
The convolution layer is responsible for the transformation of the input image to extract features from it. To obtain this, the input image is convolved with a kernel (or filters).
A kernel is a tiny matrix that has a smaller height and width than the image it will convolve. It is also called a convolution matrix or convolution mask. This kernel moves over the image's height and width and calculates the dot product of the kernel and the image at every location. The distance that the kernel moves is called the **stride length**. In the image above, we have the input image, the kernel where the stride length is 1. The output image is also known as the convolved feature.
Sometimes, the filter does not perfectly fit the input image, In such scenarios a **Padding** might be added.
A padding can be added in the following ways -

a. Pad the picture with Zeros (Also called Zero Padding) so that the filter fits the input image perfectly or
b. Drop the part of the image where the filter did not fit. This is called valid padding which keeps only the valid parts of the image

To achieve a more accurate depiction of a practical scenario, we will now examine a colored image, specifically an RGB image that consists of three channels. It's worth noting that since the input image comprises three channels, the number of filters or kernels for each channel must also be three. In other words, the kernel's depth or number of channels must match the input image's depth or number of channels. This can be expressed as follows:
When required to extract more features from an input image using convolution, multiple kernels/filters are required. Even in such cases the size of all the kernels must be the same. After the convolution, the convolved features of the input image are then stacked one after the other to create an output



so that the number of channels is equal to the number of filters used. A representation of the above is -



The **Activation Function** is the final element of the convolution layer, responsible for introducing non-linearity in the output. **ReLu** or **Tanh** are commonly used as activation functions in the convolution layer.

ReLU is an abbreviation for Rectified Linear Unit, which performs a non-linear operation that produces an output of $f(x) = \max(0,x)$. ReLU's purpose is to enable the ConvNet to learn non-linear patterns in real-world data, where negative linear values are insufficient. Although other non-linear functions, such as tanh or sigmoid, can be used as an alternative to ReLU, most data scientists prefer ReLU due to its superior performance compared to the other two.

The same can be mathematically represented as - **Forward Propagation of convolution layer** -

$$\mathbf{Y_i} = \mathbf{B_i} + \sum_{\mathbf{j=1}}^{\mathbf{n}} \mathbf{X_j} * \mathbf{K_{ij}} , i = 1, 2....d$$

Here-

Y is the Ouput

X is the Input Matric (Image)

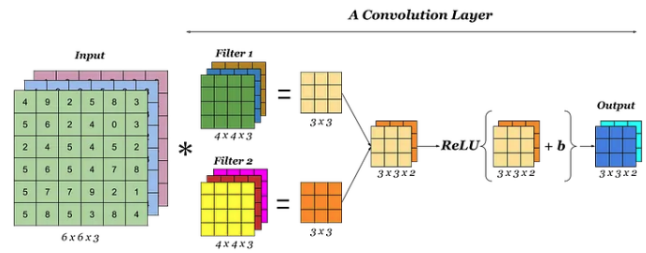
K is the Kernel/Filter

B is the Added Bias

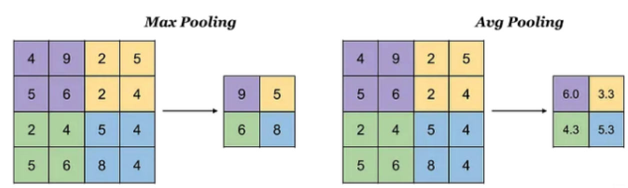
Matrix Visualisation for the above equation -

$$\begin{bmatrix} Y_1 \\ Y_2 \\ . \\ . \\ . \\ Y_n \end{bmatrix} = \begin{bmatrix} B_1 \\ B_2 \\ . \\ . \\ . \\ B_n \end{bmatrix} + \begin{bmatrix} K_{11}\ K_{12} \ldots K_{1n} \\ K_{21}\ K_{22} \ldots K_{2n} \\ . \\ . \\ . \\ K_{d1}\ K_{d2} \ldots K_{dn} \end{bmatrix} * \begin{bmatrix} X_1 \\ X_2 \\ . \\ . \\ . \\ X_n \end{bmatrix}$$

The final Convolution Layer can be represented as -



(2) Pooling Layer:
The pooling layer is employed to decrease the dimensions of the input image, which typically follows a convolutional layer in a convolutional neural network. The objective of adding a pooling layer is twofold: to enhance the computational efficiency and to enhance the durability of some of the identified features. The pooling procedure employs a kernel and a stride, as demonstrated in the example below, which utilizes a 2x2 filter to pool a 4x4 input image with a stride of 2. Max pooling and average pooling are the two most frequently used pooling methods in convolutional neural networks.

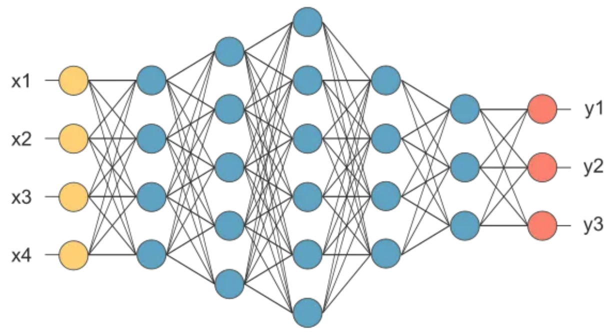


There are different types of Pooling are -

a. **Max Pooling** : Max pooling involves selecting the largest element from the rectified feature map. Alternatively, average pooling can also be used by selecting the average value. Another method is sum pooling, which involves calculating the sum of all elements in the feature map.
b. **Average Pooling** : Average pooling is a technique used in convolutional neural networks to downsample the feature maps. It involves dividing the input feature map into non-overlapping patches and then computing the average value of each patch. The result is a reduced map with a lower spatial resolution but the same number of channels as the input map.
c. **Sum Pooling** : Sum pooling is a type of pooling operation used in deep learning to reduce the spatial dimensions of feature maps. In sum pooling, the values of each feature map are summed within a fixed size window or kernel, resulting in a single value for each feature map.

(3) Fully Connected Layer :
At the conclusion of a convolutional neural network, a fully connected layer is placed to capture complex relationships among

high-level features. To accomplish this, the feature map generated by the preceding layer is flattened into a vector, which is then passed through the fully connected layer. The output of this layer is a one-dimensional feature vector.



The feature map matrix shown in the diagram will be transformed into a vector (x1, x2, x3, ...) and then combined using fully connected layers to create a model. To classify the resulting outputs, an activation function such as softmax or sigmoid is applied.

2.2 EfficientNet :

EfficientNet is a convolutional neural network architecture designed to be both computationally efficient and achieve state-of-the-art accuracy on various image recognition tasks.

EfficientNet uses a technique called compound coefficient to scale up models in a simple but effective manner. Instead of randomly scaling up width, depth or resolution, compound scaling uniformly scales each dimension with a certain fixed set of scaling coefficients. Using the scaling method and AutoML, the authors of efficient developed seven models of various dimensions, which surpassed the state-of-the-art accuracy of most convolutional neural networks, and with much better efficiency.

In otherway, EfficientNet achieves its efficiency through a novel compound scaling method that uniformly scales the network width, depth, and resolution in a systematic way. This method helps to balance the tradeoff between computational efficiency and model accuracy.

Compound Model Scaling :

In order to develop the method of compound scaling, the impacts of each scaling technique on the model's performance and efficiency were studied systematically. It was concluded that while scaling a single dimension can enhance the model's performance, balancing the scale across all three dimensions - width, depth, and image resolution - based on the variable available resources is the optimal approach to improve the overall model performance.

The reasoning behind the design of these networks is that if the input image is larger, more layers are required to increase the receptive field, and more channels are needed to capture finer details of the image. By implementing the compound scaling technique, previous CNN models, such as MobileNet and ResNet, were able to achieve improvements in both model efficiency and accuracy, with ImageNet accuracy increasing, compared to other arbitrary scaling methods.

Performance :

EfficientNet is a type of neural network architecture that surpasses traditional CNN models in terms of both accuracy and efficiency.

EfficientNet's superior performance is due to its use of a compound scaling technique that balances the scale across multiple dimensions, including width, depth, and resolution, while considering available resources. This approach enables EfficientNet to achieve a better trade-off between accuracy and computational efficiency compared to other CNN models.

EfficientNet also incorporates a novel convolutional block called the "MBConv block," which utilizes depthwise and pointwise convolutions to reduce computation while maintaining accuracy. Moreover, EfficientNet employs a technique known as "swish activation," which has proven to be more effective than conventional activation functions like ReLU.

The combination of the compound scaling technique, MBConv block, and swish activation function, along with other features, has led to EfficientNet achieving state-of-the-art performance on various image classification tasks. Furthermore, it requires fewer computational resources than traditional CNN models.

3 DATASET

We obtain the Chest X-Ray Images (Pneumonia) dataset from the Kaggle page: Chest X-Ray Images (Pneumonia), which the Kaggle page obtains the data originally from the Mendeley Data Website: Labeled Optical Coherence Tomography (OCT) and Chest X-Ray Images for Classification. The dataset was collected from retrospective cohorts of pediatric patients of one to five years old from Guangzhou Women and Children's Medical Center, Guangzhou.

The Chest X-Ray Images (Pneumonia) dataset contains 5,856 chest X-ray images in JPEG format and each Image is labeled as either 'Normal' or 'Pneumonia' based on the diagnosis of the patient.

The dataset is divided into three subsets: train, test, and validation, with 5,216, 624, and 16 images, respectively. Inside training dataset, we have 1341 normal case and 387 Pneumonia cases. For test dataset, we have 234 normal and 390 pneumonia cases. Lastly, we have 8 for each category inside Validation dataset.

	Normal	Pneumonia	Total
Type			
test	234	390	624
train	1341	3875	5216
val	8	8	16

Figure 2: Basic Statistics of Dataset

4 PREDICTIVE TASK

The predictive task based on the dataset is binary classification of pneumonia diagnosis, where the aim is to accurately classify chest X-ray images into two classes: "Pneumonia" and "No Pneumonia." This machine learning task involves training models to identify the presence or absence of pneumonia in chest X-ray images using a set of features such as lung opacity, inflammation, and other clinical factors. To evaluate different models for this task, various performance metrics can be used, including accuracy, precision, recall, F1 score, and area under the curve (AUC). The models can also be validated using techniques such as cross-validation and hyperparameter tuning to optimize model performance. The baseline model for comparison could be convolutional neural networks (CNNs). This model is appropriate because it is widely used in binary classification tasks and has been shown to perform well in medical image analysis. While the baseline model has its strengths, it may have drawbacks such as limited capacity to capture complex patterns and relationships in the data, which may lead to lower accuracy and prediction performance. In contrast, EfficientNet, has shown promise in overcoming these limitations by leveraging hierarchical representations of features, which can lead to higher accuracy and prediction performance. Therefore, we can expect the EfficientNet model to outperform the baseline model for pneumonia diagnosis classification.

5 MODEL

During the model training, we realized that it takes time to train the model since we have 1,575 files (about 829 MB) of normal lung X-Rays and 4,265 files (about 346 MB) of pneumonia lung X-Rays.

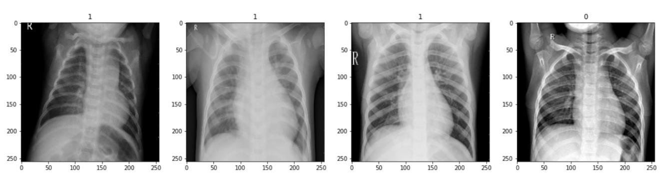


Figure 3: Image Classification for Chest X-Rays (0: Normal, 1: Pneumonia)

5.1 Convolutional Neural Networks

After building the model, we checked the validation accuracy, training accuracy, validation loss, and training loss of the model. On our first trial, it was great to see that the accuracy of the training set and validation set are equally increasing and high. But, we found out that our validation loss is decreasing while train loss is increasing. We figured out that this meant that the model is overfitting the training data. To alleviate this overfitting issue, we tried dropout and L2 Regularization on the convolutional blocks. By doing this, we were able to lower the loss and increase accuracy.

5.2 EfficientNet

We also further improve the CNN model by experimenting with EfficientNet. EfficientNet is a type of convolutional neural network (CNN) that was introduced in 2019 by a team of researchers from Google Brain. The goal of EfficientNet was to create a CNN model that was both more accurate and more efficient than existing models. The researchers used a scaling method that balances the depth, width, and resolution dimensions of the model to optimize performance. They found that by increasing the depth, width, and resolution of the CNN model simultaneously, they could achieve better accuracy with fewer parameters than existing models.

One of the key advantages of EfficientNet is its ability to leverage transfer learning. Because it is pre-trained on large-scale image datasets, it can be fine-tuned on smaller datasets with better results than traditional CNN models.

6 RESULTS

6.1 Purpose

The purpose of this project was to diagnose Pneumonia using a chest X-ray. We got the normal lung chest X-rays and Pneumonia chest X-rays datasets on Kaggle. By using Convolutional Neural Networks(using TensorFlow) and EfficientNet model(using PyTorch) for image classification, we were able to classify chest X-ray photos that are normal or have pneumonia.

6.2 Strengths and Limitations of models

The strength of the CNN Model was we were able to get reliable accuracy after training the model. Generally speaking, accuracy is a method for measuring a model's performance. Since we got an average accuracy of 0.76, we can say that the result of the model is reliable.

During our trials on CNN Model, it was great to see that the accuracy of the training set and validation set are equally increasing and high. But, we found out that our validation loss is decreasing while train loss is increasing at the beginning of the model training. We figured out that the model is overfitting the training data. To alleviate this overfitting issue, we tried dropout and L2 Regularization on the convolutional blocks. As a result, by doing this, we were able to lower the loss and increase accuracy.

We also tried applying the EfficientNet Model in this task and achieved a better accuracy compared to CNN. There are several reasons for the improvement:

- Scale: EfficientNet is designed with a scaling strategy that balances depth, width, and resolution dimensions to optimize performance. This allows it to achieve better accuracy with fewer parameters than traditional CNN models.
- Architecture: The architecture of EfficientNet is carefully designed to maximize the use of computational resources, which leads to a higher accuracy with less computation. This makes EfficientNet more efficient and faster than traditional CNN models.
- Transfer learning: EfficientNet is pre-trained on large-scale image datasets, allowing it to leverage transfer learning. This means that it can be fine-tuned on smaller datasets with better results than traditional CNN models.
- Flexibility: EfficientNet can be used for a variety of computer vision tasks such as image classification, object detection, and segmentation, making it a versatile tool for researchers and practitioners.

Overall, EfficientNet is considered better than traditional CNN models because it achieves higher accuracy with fewer parameters and less computation, while being more flexible and easier to use for a variety of computer vision tasks.

On the other hand, the limitation of our project was, as data scientists, we were not sure whether the given datasets are reliable or not. Since we do not have any medical knowledge to recognize normal lung chest X-rays and abnormal lung chest X-rays, we should trust the categorized data sets are trustworthy.

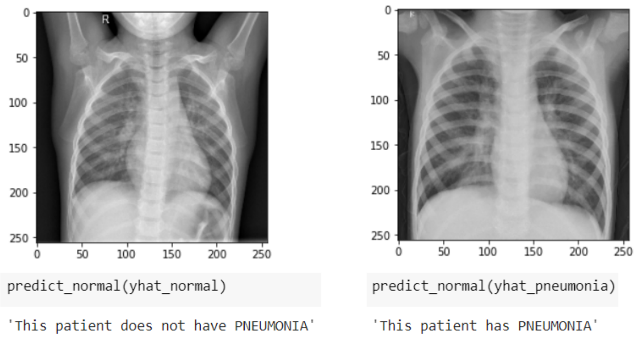


Figure 4: Result for Image Classification (left input-output: Normal-Normal, right input-output: Pneumonia-Pneumonia)

6.3 Conclusions

We tested our models using chest X-Rays that we did not use during the training-validation process. Both of our models were able to identify most of the chest X-Rays correctly and give us appropriate text output, saying "This patient has PNEUMONIA" or "This patient does not have PNEUMONIA".

The CNN model got the accuracy of 0.76 and and the EfficientNet model got the accuracy of 0.88. Therefore, Lung Image Classifications using Convolutional Neural Networking and EfficientNet work fine as we wanted.

6.4 Future Works

For the future project, we want to try building an image-classification model which can detect other lung diseases, such as lung cancer, tuberculosis, sarcoidosis, emphysema, or fibrosis. Moreover, we want to build a model with good time efficiency and good space efficiency that can handle big datasets without breaking the model.

7 REFERENCES

1. Tan, Mingxing, and Quoc Le. "Efficientnet: Rethinking model scaling for convolutional neural networks." International conference on machine learning. PMLR, 2019.
2. https://www.coursera.org/projects/pneumonia-classification-using-pytorch
3. https://medium.com/@himadrisankarchatterjee/a-basic-introduction-to-convolutional-neural-network-8e39019b27c4
4. https://www.youtube.com/watch?v=jztwpsIzEGc
5. https://github.com/nicknochnack/ImageClassification
6. https://www.dropbox.com/sh/4zxgua4g0ozll76/AAAoRMHgI9tG5TAGkAzR
7. https://deepai.org/machine-learning-glossary-and-terms/convolutional-neural-network
8. https://en.wikipedia.org/wiki/Convolutional$_neural_network$

Received 19 March 2023