Satakunnan ammattikorkeakoulu
Satakunta University of Applied Sciences

RUNSHENG SHA

Pneumonia Classification Model by Convolutional Neural Network

DEGREE PROGRAMME IN ARTIFICIAL INTELLIGENCE
2022

Author(s) Sha, Runsheng	Type of Publication Bachelor's thesis	Date December 2022
	Number of pages 32	Language of publication: English

Title of publication
Pneumonia Classification Model by Convolutional Neural Network

Degree Programme
Bachelor of Artificial Intelligence

Abstract

The purpose of this thesis is to train a model to recognize and classify X-ray images of pneumonia patients from those of normal people by approaches of convolutional neural networks in computer vision. This thesis introduces the types of machine learning algorithms, the advantages of deep learning, the composition of artificial neural networks and their hyperparameters, the structure of convolutional neural networks and the points to be noted during training. This thesis also presents previous relevant studies and researches by other authors on similar topics. Finally, with all the background theory mentioned, TensorFlow, Keras, and Sklearn are used on Google Colab platform to build and train the convolutional neural network. A model with an accuracy of 88% was obtained.

There is still room for improvement in this model. For example, continue training with a larger and more diverse set of input data, build more complex convolutional neural networks, different image enhancement schemes, and various hyperparameter adjustments.

Keywords
Machine Learning, Convolutional Neural Network, Image Classification,

CONTENTS

LIST OF SYMBOLS AND TERMS

AI = Artificial Intelligence

ML = Machine Learning

GPU = Graphics Processing Unit

DL = Deep Learning

NN = Artificial Neural Networks

ANN = Artificial Neural Networks

ReLU = Rectified Liner Unit

CNN = Convolutional Neural Network

FC = Fully-connected

1 INTRODUCTION

With the rapid development of Artificial Intelligence (AI) in recent years, AI has been used in a variety of industries. Such as data mining, manufacturing, and financial services. By building the right models, humans can use AI to perform many tasks that once required complex human intervention. One of the most notable applications is the use of AI in healthcare.

Pneumonia as one of the common diseases in the world, can affect the health of 450 million people worldwide in one year. In 2009, pneumonia was ranked as the eighth leading cause of death in the United States(Nair & Niederman, 2011), which means that the number of people suffering from pneumonia is large. Chest X-ray is widely used as a validated way for pneumonia detection, and today's artificial intelligence has a variety of different schemes for successful implementation of image classification. Convolutional neural networks, on the other hand, are considered to be the most effective deep learning approach for solving image classification problems.

The topic of this thesis is to train a model capable of recognizing pneumonia through convolutional neural networks in deep learning. Used in response to a possible or ongoing shortage of medical personnel or a shortage of medical equipment, it aims to provide an effective means for people to see if they have pneumonia.

There have been many researches related to the detection and classification of chest radiographs. Computer vision and deep learning (DL) have made great progress in medical detection, especially in pneumonia chest X-ray images. A few of the more notable examples such as Rajpurkar et al. (2017) have trained a model and developed an algorithm to detect pneumonia and other 13 chest diseases with a good performance from chest X-ray images called CheXNet in 2017. A model to further differentiate viral pneumonia from bacterial pneumonia in pediatric chest X-ray images by implementing a new decision support system in the VGG16 model as well as a visualization strategy has been similarly proposed. And it achieved 96.2% and 93.6% accuracy in differentiating bacterial and viral pneumonia, respectively. (Rajaraman, 2018) Rahman et al. (2020) proposed a transfer learning method based on deep convolutional neural networks (CNN), which can be used to detect different classes of pneumonia. Their DenseNet201 model used can achieve 98%, 95%, and 93.3% accuracy in normal, bacterial, and viral pneumonia, respectively.

The dataset used in this thesis is from the Kaggle website and was provided by user Paul Mooney. The dataset contains 5,863 images from a retrospective cohort of pediatric patients aged 1 to 5 years selected from Guangzhou Women's and Children's Medical Center. The dataset is divided into three folders, Test, Train and Val. Each folder contains two folders, NORMAL and PNEUMONIA, to distinguish the different images.(Mooney, 2017)

2 MACHINE LEARNING

Before dive in, we first need to figure out what machine learning is. Machine learning (ML) is a subfield of artificial intelligence. The term machine learning was first coined by Arthur Samuel, an IBM employee, in 1959. It is noted that machine learning is the ability given to computers to learn without specific programming. (Samuel 2000) Machine learning is considered as learning from labeled data through algorithms and methods that emulate the way humans learn. Machine learning builds models that can predict specific problems by learning from labeled data sets and finding specific patterns in them, builds a mathematical association that applies to the complex training set provided by the trainer. The more complex the data set and training method, the more accurate the prediction accuracy of the model will be.

2.1 Algorithms

Based on the type of input data as well as the type of output, machine learning algorithms can be classified into the following three main categories:

2.1.1 Supervised Learning

Supervised machine learning uses labeled data sets to train algorithms. The computer learns and adjusts the weights repeatedly on the labeled input set to obtain the desired output for prediction. Supervised machine learning can usually be divided into two main categories: regression(Linear regression, Logistic regression) and classification(Decision trees, Random forest). Neural networks are also considered as an algorithm of supervised learning which is the focus of the upcoming discussion in this thesis. Supervised machine learning is considered an excellent training method for text classification, email spam detection, image classification and recognition, and price prediction.(IBM Cloud Education, 2020)

2.1.2 Unsupervised Learning

Unsupervised machine learning uses unlabeled datasets for training. The computer uses algorithms to analyze the relationships between the unlabeled datasets and form clusters that conform to their patterns. Unsupervised machine learning is able to discover similarities and differences in information in the data. Unsupervised machine learning can usually be divided into two main categories: clustering and association. Unsupervised machine learning is considered an excellent training method for data exploration, market segmentation, image compression and market basket analysis-recommendation. (IBM Cloud Education, 2020)

2.1.3 Semi-Supervised Learning

Semi-supervised machine learning combines both supervised machine learning and unsupervised machine learning. During training, it uses a small number of labeled datasets to guide the machine to extract and classify features from a large number of unlabeled datasets. Semi-supervised learning does not require extensive labeling or is used when labeled data is insufficient. (IBM Cloud Education, 2020)

2.1.4 Reinforcement Learning

“The reinforcement machine learning is the problem faced by an agent that learns behavior through trial-and-error interactions with a dynamic environment”(Kaelbling et al., 1996) Reinforcement learning gives rewards or penalties at each particular stage of the learning algorithm's operation to enable the model to produce outcome. It is widely used for AI training in games, and has not yet been used in business.

2.2 Deep Learning

Deep learning (DL) is considered a subset of machine learning, which is driven by multiple layers of neural networks modeled after the human brain. Deep learning allows multiple computational processing layers to learn training data. The deep learning model is eventually obtained by adjusting the weights and optimizing the training layers during the training process. Specifically, the neural network passes data information through the node layers. Each layer that passes data backward builds on the information from the previous layer and selectively passes the information to the next node. There is no certain number of training layers, but usually multiple training layers can optimize and improve the accuracy of the training results.(Oracle, n.d.)

Compared to traditional machine learning, deep learning does not require users to create features of the data during the training process. It learns features from the training process and creates new features on its own. This feature eliminates the need for data categorizing and generalization in traditional machine learning training, which can save a lot of time in the whole training process. And machine learning can only handle less complex tasks, such as predicting routes and cargo detection. Deep learning, however, can handle more complex tasks, such as driverless technology. (Microsoft Azure, 2022)

However, deep learning also requires more computational resources, especially for GUPs, to train its models than traditional machine learning. And the training time is usually longer than machine learning due to its complex layer structure and algorithms. (Talab, 2022)

3 NEURAL NETWORKS

Neural networks (NN), also known as artificial neural networks (ANN) in artificial intelligence, are a core element of deep learning. A neural network is composed of a series of layers made up of neurons (or known as nodes). The layers are interconnected with each other to form a complete training algorithm. Starting from the input layer, these algorithms are computed in different layers of one or more hidden layers and finally output by the output layer. In this process, neural networks produce optimal results by mimicking the operation of neurons in the human brain, working with each other to identify features or specific relationships from the training set. The four training methods mentioned above (supervised learning, unsupervised learning, semi-supervised learning and reinforcement learning) are also applicable to the training methods of neural networks..(Pedamkar, n.d.)

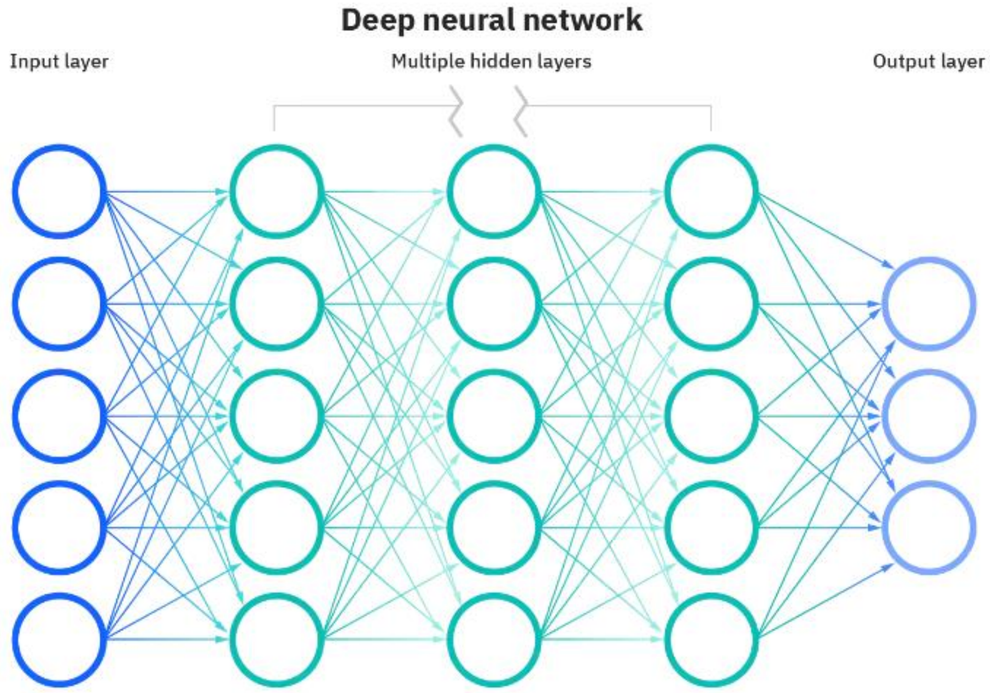

Figure 1. An Image of a Deep Neural Network (IBM Education Cloud, 2020)

3.1 Neuron Model

Neural networks are composed of basic neurons, which are the most fundamental building blocks of neural networks. Figure 2 is a mathematical model of neurons, based on the model first proposed by psychologist McCulloch and mathematician Pitts in 1943.(McCulloch & Pitts, 1943)

As shown in the Figure 2., the values a1, a2, etc. are the input. The connection lines connected between the a and sum have the values w1, w2, etc., which are called weights. The weights help determine the importance of the input data features. The importance of a feature is determined by changing the size of the weights; the larger the data the more important the feature is, and vice versa. Also the weights determine the speed of triggering the activation function. b is the bias. The bias can be used to increase the flexibility (fit) of the model and to shift the left or right of the activation function.(Ganesh, 2020) f is the activation function which gives the final output known as t. These elements form the basic neuronal model.

And the workflow of the neuron is quite simple. Each input value (i.e., a) is multiplied by its respective weight (i.e., w) and summed (i.e., SUM) after adding a bias (i.e., b). The sum is then multiplied by the activation function (i.e., σ) to obtain the final output value. Using the mathematical formula we obtain the following expression:

$$t = f(\sum_{i=1}^{n} w_i a_i + b)$$

This is the basic way in which artificial neurons work. More complex neural networks will contain more such operations.

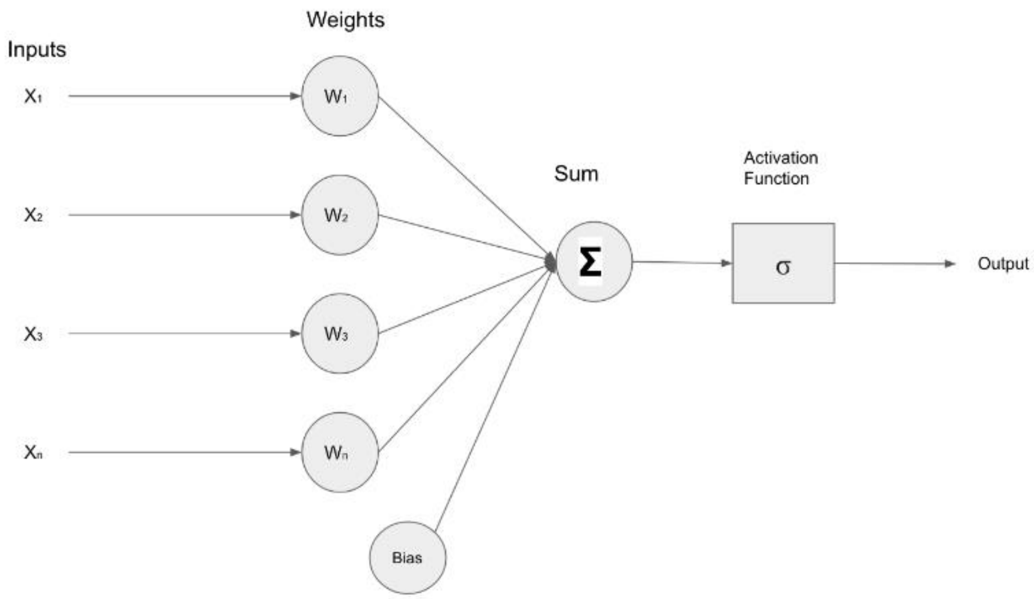

Figure 2. M-P Model(Alake, 2022)

3.2 Activation Function

The activation function as an important component of artificial neural networks, determines whether a neuron should be activated by calculating the sum of the weights and biases. The activation function adds non-linear to the output of the neuron and complicated mappings between the inputs and outputs, so that the neural network can not only solve linear regression problems of duality but to learn and solve more complex problems. Neural networks basically work in a similar way to linear regression models. Without the activation function involved in the work, the neural network can only adapt to the linear way of prediction work and cannot predict more complex situations. (Sharma et al., 2017)

The activation function is likewise considered to be an important factor in determining the prediction accuracy of neural networks. Different types of activation functions have different effects on different data, and there is no guidebook that specifies which activation function has the best results. Therefore, trial-and-error experimentation is a necessary measure. (Sharma et al., 2017)

An important property for activation functions is differentiability. The presence of differentiability allows the implementation of back-propagation and gradient descent. (Sharma et al., 2017)

3.3 Types of Activation Function

There are many kinds of activation functions for neural networks, and there are different activation functions for different problems. In this thesis, we only introduce three commonly used activation functions for solving non-linear problems.

3.3.1 Sigmoid Function

Sigmoid function is the most widely used activation functions in neural network training, especially in output layer. The Sigmoid function converts the output to between 0 and 1 and is usually applied to neural networks using the back propagation algorithm. (Karlik & Olgac, 2011) And the function is calculated as follows:

$$f(x) = \frac{1}{1 + e^{-x}}$$

The function is differentiable but asymmetric to zero, meaning that the sign of the output value is the same for all neurons. However, this problem can be improved by scaling the function. (Sharma et al., 2017)

3.3.2 Hyperbolic Tangent Function

Hyperbolic tangent functions, also known simply as tangent functions, are very similar to Sigmoid functions in that they have an obvious S-shape. Its value interval is distributed between -1 and 1. The function is calculated as follows:

$$f(x) = \frac{2}{1 + e^{-2x}} - 1$$

However, the gradient of the tangent function is unrestricted compared to the Sigmoid function and is symmetric at the origin. (Sharma et al., 2017) The more positive the inputs are, the closer the output in Tanh function to the 1, and vice versa. (Tiwari, 2022)

3.3.3 ReLU Function

ReLU Stands for Rectified liner unit. The ReLU function is also considered to be the most widely used activation function, commonly used in the hidden layer of convolutional neural networks. Its value ranges from 0 to infinity, meaning that any input less than or equal to zero will be output as 0. The function is calculated as follows:

$$f(x) = \max(0, x)$$

However, this also means that the model cannot map any negative values, which reduces the ability to train the model to fit correctly. (Sharma et al., 2017) However, compared to Sigmoid and tangent functions, ReLU has only a few neurons that are activated in specific situations during training, making the whole neural network more sparse and easy to compute efficiently. (Tiwari, 2022)

3.4 Choosing the Activation Functions

However, in general, it is common sense to avoid using Sigmoid and tangent functions in gradient disappearance problems (i.e., these two functions are not suitable for hidden layers), such as ReLU function performs much better than other activation functions and can only be used in the hidden layer. In the training of neural networks and deep learning, we should start with the ReLU function first, and then try other functions if we cannot give satisfactory results. (Sharma et al., 2017)

3.5 Gradient Descent

Gradient descent is an optimization algorithm in machine learning. It is used to help the model measure its accuracy in each iteration and search for parameter values to eventually minimize the cost function to the optimal accuracy.(Alake, 2022)

In neural networks, although the input can get the output value after the series of weights, bias and activation function calculation. But this is not enough for the training of the model. Accuracy is a very important thing for machine learning. Therefore, a value to evaluate the performance of a neural network needs to be introduced. This is the cost function, a value that measures the error gap between the predicted value of a neural network and the actual value of a data sample. The goal of gradient descent is considered to be minimizing the cost function, and minimizing the cost function then allows the model to be most accurate. (Alake, 2022)

And about the way the visualization of gradient descent works. As shown in Figure 3, the steepness of the slope is observed by using a tangent line starting from the starting point and ending at the point of convergence. The slope will update the weights with the bias. As each update is generated, the steepness of the slope will get lower and lower until it reaches the lowest point of the curve on the way, which is the point of convergence, i.e., the model's optimal accuracy.(IBM Cloud Education, 2020)

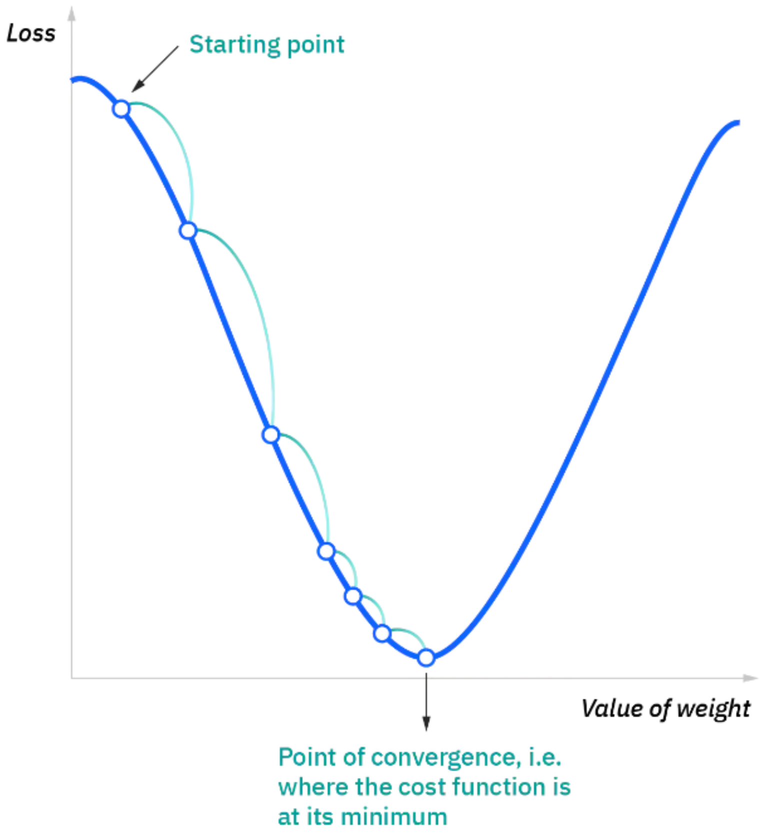

Figure 3. Gradient Descent Model(IBM Cloud Education, 2020)

However, like every algorithm that exists in the world, gradient descent has its own drawbacks. Specifically, in using gradient descent algorithms may experience problems such as vanishing gradient(too small gradient) or exploding gradient explosion (too large gradient).(IBM Cloud Education, 2020) But both types of problems can be effectively dealt with by using the ReLU or Leaky ReLU activation function, reducing the learning rate, batch normalization or changing the neural network architecture, all four approaches.

4 CONVOLUTIONAL NEURAL NETWORK

Convolutional neural network (CNN), as one of the neural networks, is considered to be the best one used in deep learning for image recognition and classification tasks. Convolutional neural networks ultimately distinguish images by assigning weights and biases to the input to them and extracting their best features.(Saha, 2018) The great advantage of a convolutional neural network is that, unlike earlier algorithms or neural networks that require feature extraction and other pre-processing, convolutional neural networks do not require any pre-processing because they can operate directly on the original image. Convolutional neural networks are currently used in a wide range of applications, but can be broadly distinguished into two categories, images (image recognition, image classification) and Speech (natural language processing, speech recognition).(NVIDIA Developer, n.d.)

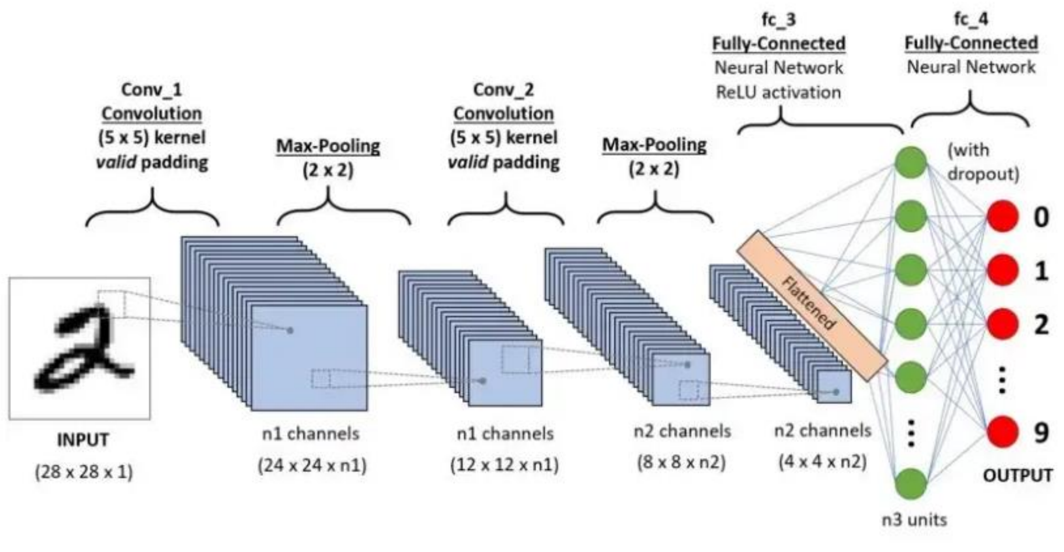

Figure 4. A CNN Sequence to Classify Handwritten Digits.(Saha, 2018)

4.1 Image Channels

Before understanding how convolutional neural networks process images, we first need to understand how the machine represents images. In a computer, pixels are used as a representation of an image and are mapped to numbers between 0 and 255 for human understanding. Each of these numbers represents a specific color, from 0 for white, to 255 for black. (Madhaven, 2021)

According to this representation, the image thus becomes an array of pixels with dimensions in the computer, or can be understood as an array. The computer's image is displayed through a channel composed of these pixel arrays. Grayscale images have only one channel, while color images have three channels, each representing red, green and blue. (Madhaven, 2021)

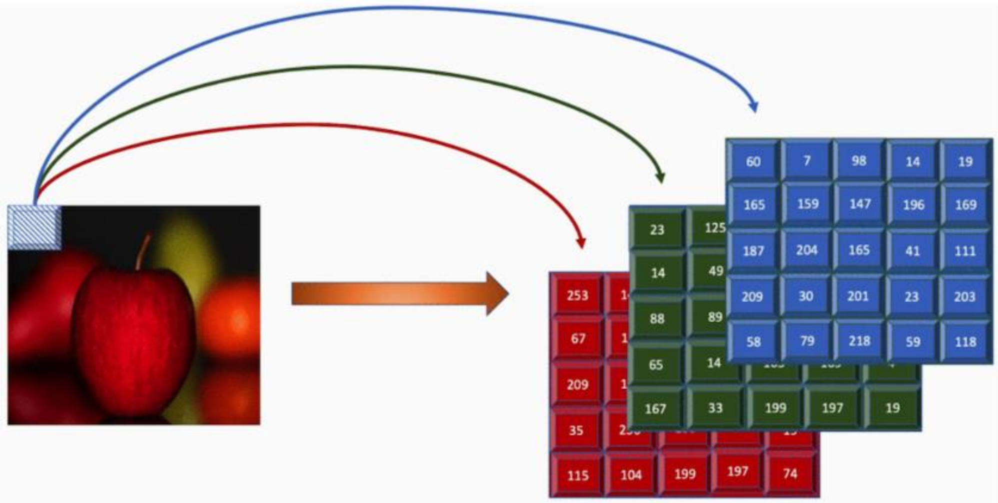

Figure 5. The Channels of an Image. (Madhaven, 2021).

4.2 Design of Convolutional Neural Network

The basic architecture of convolutional neural networks is the same as other neural networks, with input, hidden and output layers. The only difference is that a convolutional neural network is a multilayer feedforward neural network. This means that it is made up of many hidden layers, which is the origin of the word "convolutional" in its name. (Figure 4) Among these layers, there are four main layers, namely convolutional, Non Linearity, pooling, and fully connected layers. (Wood, 2019) Convolutional layers are usually used as the first layer of a convolutional network, after which more convolutional layers or additional pooling layers can be added as needed, but the fully connected layer must be the last layer. (IBM Cloud

Education, 2020) A Non Linearity layer can be added after the convolution to increase the activation function as a way to add nonlinearity. (Wood, 2019) We will describe how convolutional neural networks work after understanding the role of each layer.

4.2.1 Convolutional Layer

The convolutional layer is the core part of a convolutional neural network. Convolutional neural networks extract features on an image by using filters (also known as kernels) in the convolutional layer. The filter in the convolutional layer is usually small, but it should usually be represented as a size of mXm. The filter glides through a matrix of mXm in each region of the image and computes the dot product between the input pixels and the kernel. The dot product is fed to the new output array and learned. Later they are activated when they see the features in that array. Then, the filter will continue gliding over the image in mXm size until all the features are extracted in the input image. (O'shea & Nash, 2015) This final output consisting of dot products is called a feature map, activation map, or convolutional feature.(IBM Cloud Education, 2020)

In the filter of the convolutional layer, the weights are optimized by backpropagation with gradient descent as training proceeds. But some hyperparameters need to be adjusted before training. Which are:

- Depth: defines the number of filters to be applied in the convolution process. By reducing this hyperparameter the total number of neurons in the network can be reduced making the computation easier, but at the same time reducing the recognition ability of the model.

- Stride: is the distance that the filter moves over the input matrix. A larger stride will produce a smaller output. (O'shea & Nash, 2015)

- Zero-padding: is the process of padding the boundaries of the input. It will set all values that fall outside the input matrix to zero. It is an effective way to further control the dimensionality of the output volume. (O'shea & Nash, 2015)

The complexity of the model can be effectively reduced by adjusting these three hyperparameters. (O'shea & Nash, 2015)

4.2.2 Pooling Layer

The pooling layer is used to perform dimensionality reduction of the input image dimensions. By reducing the dimensionality, the computational power required for training can be reduced, and by reducing the dimensionality the network needs to compute fewer weights to prevent overfitting. There are two types of pooling, maximum pooling and average pooling. The difference between the two is that the former returns the maximum value of the image covered by the filter, while the latter returns the average value. Also, the maximum pooling can be used as a noise suppressor. Denoising is performed while dimensionality is reduced. (Saha, 2018)

The pooling layer scales the dimensionality by using the MAX function for each feature map. Due to its destructive nature and a lot of information lost to the original feature map, the stride and filter are usually set to a size of 2x2 when pooling. If this value is exceeded the model performance is significantly degraded. (O'shea & Nash, 2015)

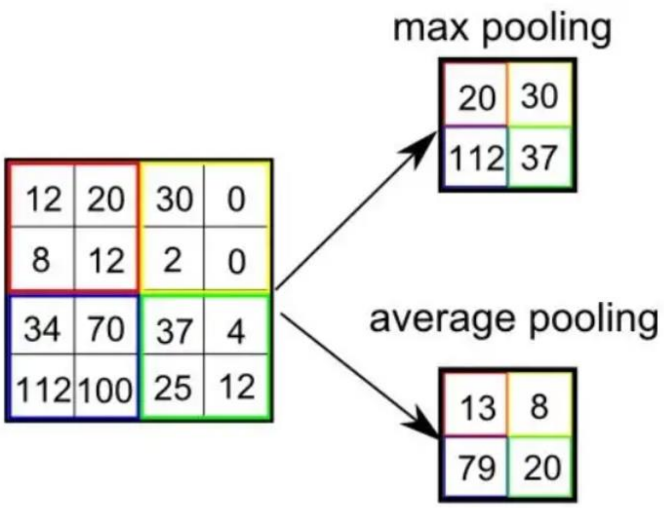

Figure 6. Types of Pooling. (Saha, 2018).

4.2.3 Fully-Connected Layer

The fully-connected (FC) layer is treated as the output layer in the neural network. After working with the previous layers and different filters, the fully-connected layer works by flattening the feature image into a column vector. The column vector is optimized by back propagation in several iterations. The final Softmax activation function is used to perform the final classification of the input. (Saha, 2018)

4.3 Overfitting And Underfitting

Both overfitting and underfitting are obstacles in machine learning that hinder the accuracy of model training. Overfitting usually occurs because the model learns too much noise from the dataset and starts to become fully fitted to the input data causing it to fail to generalize well from the input data. Underfitting, on the other hand, is when the model is unable to find features that can establish relationships from the input data and output, resulting in poor prediction of the model. (IBM Cloud Education, 2021)

There are other reasons for overfitting, such as long training time in a single dataset, insufficient training samples in the training data, and high complexity of the model so that it can learn enough noise. (AWS, n.d.)

There are also many ways to prevent over-fitting and under-fitting. For overfitting, we can prevent the machine from learning noise by stopping training early before it learns it, thus preventing overfitting. In addition, we can increase the training data, perform data augmentation and regularization on the input data, or to reduce the complexity of the model, etc. For underfitting, we can increase the model complexity, remove the noise, extend the training time, etc. (IBM Cloud Education, 2021)

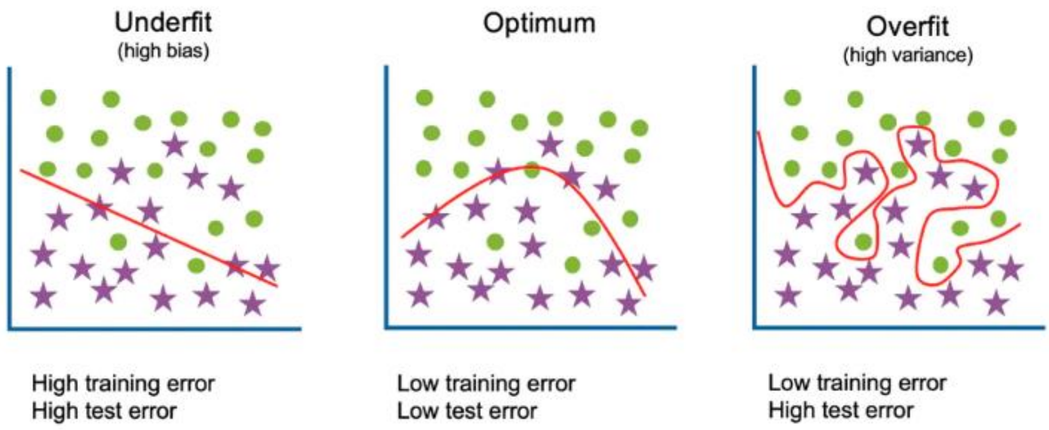

Figure 7. Types of Fitting. (IBM Cloud Education, 2021)

4.4 Data Augmentation

Data augmentation is considered to be an important method to avoid overfitting problems. In the case where the dataset is already established and no new data samples can be added. The use of data augmentation allows the model to be adapted to more possibilities and also saves the trainer's training costs. Data augmentation can make our dataset much larger than before. Common approaches like flipping, color modification, rotation, zooming, and shifting are commonly used. (Ray, 2021)

5 MODEL TRAINING

This section describes the relevant things around model training, including the prerequisite tools, convolutional neural network structure, and model accuracy.

5.1 Prerequisite Tools

The programming language used in this thesis is Python, a widely used interpretative, object-oriented, high-level programming language with dynamic semantics. The advantages and features of Python as a programming language will not be discussed here. For machine learning, Python is considered to be an excellent match for it because of its huge number of libraries and frameworks, especially for machine learning. These libraries and frameworks are maintained by the relevant communities and companies so you don't need to worry about the libraries losing support or having major vulnerabilities. Python is also more readable and easier to learn than other languages. These advantages have made Python an essential tool for data science applications.

Numpy, Pandas, and glob are used as data preprocessing tools in this thesis. Numpy is a basic scientific computing library for Python. It provides support for manipulating arrays and various basic and advanced mathematical operations and manipulations such as mathematics, logic, sorting, linear algebra, I/O etc. Pandas is also a library for Python. It can provide Python with the possibility of data manipulation and analysis. glob is a tool in Python's basic library for finding related files. By using glob you can return a list of all matching file paths. It can be used with Numpy and Pandas to make csv tables.

cv2 and Matplotlib are used in this thesis as graph reading and plotting tools. cv2 is the name of the module imported by OpenCV in Python. It has the same functionality as OpenCV, but is pre-built and can be used quickly. cv2 can read images and do some basic manipulation of them. matplotlib is an important drawing library in Python. It can be used to plot a variety of data such as curves, scatter plots, histograms, etc.

TensorFlow, Keras, and Scikit-Learn are used as the main tools for training convolutional neural networks. Keras is a high-level neural network library API created on the basis of TensorFlow 2. Keras provides tools for users to build their own artificial neural networks like layers, activation functions, and other elements necessary to form artificial neural networks. Scikit-Learn is a machine learning library for Python that provides a variety of algorithms and tools necessary for training artificial neural networks (e.g., calculating class weights).

Ultimately, due to hardware requirements, the model training for this thesis was done in Google Collaboratory (also known as Google Colab), a Jupyter-based hosted notebook service developed by the Google Research team. It is useful for deep learning as you can get free GPU computing resources when using it.

5.2 Data Preprocessing

The first step of model training is not to train the model, but to perform data pre-processing. The dataset for this thesis was taken from the Kaggle website. Information about this dataset has been mentioned in the introduction section of this thesis and will not be repeated here. Regarding the data preprocessing, this thesis starts by making three folders (val, train, test) with classes and pointing to each image file path into dataframes. The end result of train set is as follows:

```
df_train
```

	class	image
0	Normal	/content/drive/My Drive/thesis/chest_xray/trai...
1	Normal	/content/drive/My Drive/thesis/chest_xray/trai...
2	Normal	/content/drive/My Drive/thesis/chest_xray/trai...
3	Normal	/content/drive/My Drive/thesis/chest_xray/trai...
4	Normal	/content/drive/My Drive/thesis/chest_xray/trai...
...	...	...
5211	Pneumonia	/content/drive/My Drive/thesis/chest_xray/trai...
5212	Pneumonia	/content/drive/My Drive/thesis/chest_xray/trai...
5213	Pneumonia	/content/drive/My Drive/thesis/chest_xray/trai...
5214	Pneumonia	/content/drive/My Drive/thesis/chest_xray/trai...
5215	Pneumonia	/content/drive/My Drive/thesis/chest_xray/trai...

5216 rows × 2 columns

Figure 8. Dataframe of Train Set

Then use cv2 with matplotlib to view the image to make sure the file path is pointing to no errors, the result is shown below:

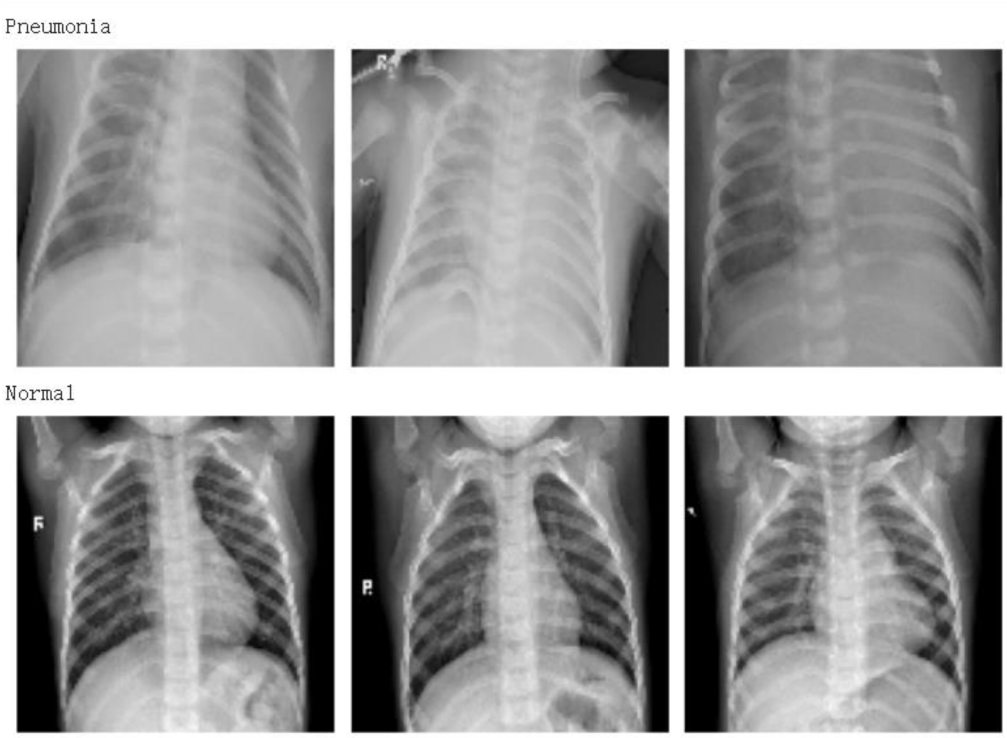

Figure 9. X-Ray Images of Pneumonia Patients and Normal People.

After ensuring that the images are correct, the data augmentation is processed. This thesis uses the ImageDataGenerator tool from the Keras library to perform data augmentation on images. It can generate batches of tensor image data with real-time data augmentation. ImageDataGenerator includes parameters such as rotation_range for random image rotation, vertical_flip for random vertical flipping, etc. After all this is ready, next step is to train the model.

5.3 Build the Convolutional Neural Network

The structure of the convolutional neural network is shown in Figure 5 below. Given the small dataset used in this thesis, the convolutional neural network written does not need to be too complex. The structure of the convolutional neural network is shown in Figure 5 below. It has six convolutional layers, and each pair of convolutional layers is immediately followed by a max pooling layer. The filter values of each convolutional layer are 32, 64, and 128, respectively, and the convolutional and pooling layers are followed by two Dense layers, the former with 128 units and the latter with 1 unit (because only one class of pneumonia needs to be identified). Between the Dense layers there is a dropout layer with 0.2 rate to prevent overfitting. The loss function of the model is chosen as binary_crossentropy, the optimizer is chosen as Adam, and the metrics are chosen as accuracy. And then the model summary table is also shown below:

```
model = Sequential()

model.add(Conv2D(filters=32, kernel_size=(3, 3), input_shape=(110, 110, 3), activation='relu'))
model.add(Conv2D(filters=32, kernel_size=(3, 3), input_shape=(110, 110, 3), activation='relu'))
model.add(MaxPool2D(pool_size=(2, 2)))

model.add(Conv2D(filters=64, kernel_size=(3, 3), activation='relu'))
model.add(Conv2D(filters=64, kernel_size=(3, 3), activation='relu'))
model.add(MaxPool2D(pool_size=(2, 2)))

model.add(Conv2D(filters=128, kernel_size=(3, 3), activation='relu'))
model.add(Conv2D(filters=128, kernel_size=(3, 3), activation='relu'))
model.add(MaxPool2D(pool_size=(2, 2)))

model.add(Flatten())
model.add(Dense(128, activation='relu'))
model.add(Dropout(0.2))

model.add(Dense(1, activation='sigmoid'))

model.compile(loss='binary_crossentropy',
              optimizer='adam',
              metrics=['accuracy'])
```

Figure 10. The Structure of the CNN to be Trained.

```
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 108, 108, 32)      896

 conv2d_1 (Conv2D)           (None, 106, 106, 32)      9248

 max_pooling2d (MaxPooling2D  (None, 53, 53, 32)       0
 )

 conv2d_2 (Conv2D)           (None, 51, 51, 64)        18496

 conv2d_3 (Conv2D)           (None, 49, 49, 64)        36928

 max_pooling2d_1 (MaxPooling  (None, 24, 24, 64)       0
 2D)

 conv2d_4 (Conv2D)           (None, 22, 22, 128)       73856

 conv2d_5 (Conv2D)           (None, 20, 20, 128)       147584

 max_pooling2d_2 (MaxPooling  (None, 10, 10, 128)      0
 2D)

 flatten (Flatten)           (None, 12800)             0

 dense (Dense)               (None, 128)               1638528

 dropout (Dropout)           (None, 128)               0

 dense_1 (Dense)             (None, 1)                 129

=================================================================
Total params: 1,925,665
Trainable params: 1,925,665
Non-trainable params: 0
```

Figure 11. The Model Summary Table

5.4 Train the Model & Result

In the training phase of the model, the value of batch_size is the default value of 32, the value of epochs is 30, and the values of steps_per_epoch and validation_steps are the number of their sets divided by the value of batch_size. The overall performance of both loss value and accuracy is rapid, but then both are relatively flat. The train accuracy is 97.39% and validation is 96.93%, which are pretty close. The loss value tends to decrease while the accuracy tends to increase and exceeds the 90% threshold. This means that the model is learning very well. The loss in the validation set experienced a period of increase but then fell off quickly and therefore had little impact on the learning of the model. Overall, the model performs well. The trained loss plot and accuracy plot are shown in the following Figure 12.

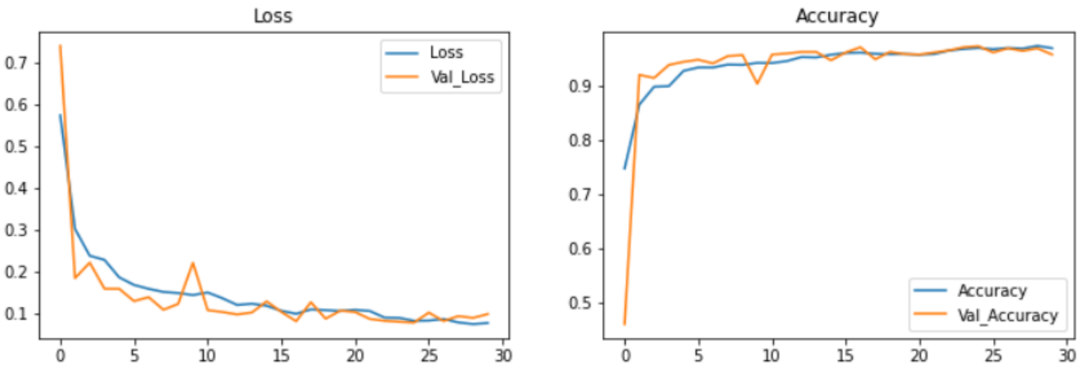

Figure 12. Loss & Accuracy Plot of The Model.

The accuracy of the results obtained after validation on the test set using this model is 88.30%. The accuracy of the validation results on the training set is 97.58%. Because of the limitations of computer hardware, training time, and the knowledge available to the authors.. This model was finally chosen as the final result after seven models were tried in this thesis.

6 CONCLUSION

The focus of this thesis is on the recognition and classification of pneumonia X-ray images and normal human X-ray images using existing image classification techniques. This thesis introduces a sub-category of machine learning, namely deep learning, by introducing the kinds of algorithms used in machine learning and deep learning. The advantages of artificial neural networks in machine learning and their characteristics are introduced by presenting the advantages of deep learning. This is followed by an introduction to the activation function and gradient descent that must be noted when training artificial neural networks. Finally, after introducing the structure and working of convolutional neural networks, a final model is obtained by designing a convolutional neural network in Google Colab for the dataset used in the thesis.

The results showed that the model can achieve an accuracy of about 88% after 3 hours of training in a convolutional neural network with a limited dataset. Patients in most developing countries are suffering from a shortage of medical resources. This can be a way for patients to make a preliminary diagnosis in the case of shortage of medical resources.

However, the model in this paper still has a lot of room for improvement. For medical applications, the accuracy rate must be 99% accurate before it can be put into use. It is believed that the accuracy of the model can be higher if it is trained by more pneumonia X-ray image datasets, or by trying different image augmentation strategies and by trying different more convolutional neural networks. The use of computer vision in healthcare, especially for chest X-ray images, is a topic worth discussing in the future.

REFERENCES

Nair, G. B., & Niederman, M. S. (2011). Community-acquired pneumonia: an unfinished battle. *Medical Clinics*, *95*(6), 1143-1161. https://doi.org/10.1016/j.mcna.2011.08.007

Mooney, P. (2017). *Chest X-Ray Images (Pneumonia).* Kaggle. https://www.kaggle.com/datasets/paultimothymooney/chest-xray-pneumonia

Samuel, A. L. (2000). Some studies in machine learning using the game of checkers. *IBM Journal of Research and Development*, *44*(1-2), 207-219 https://doi.org/10.1147/rd.441.0206

IBM Cloud Education. (2020, July 15). *Machine Learning.* https://www.ibm.com/cloud/learn/machine-learning

Kaelbling, L. P., Littman, M. L., & Moore, A. W. (1996). Reinforcement learning: A survey. *Journal of Artificial Intelligence Research*, *4*, 237-285.

Oracle. (n.d.). *What is deep learning?* https://www.oracle.com/artificial-intelligence/machine-learning/what-is-deep-learning/

Microsoft Azure. (2022, May 11). *Deep learning vs. machine learning in Azure Machine Learning.* https://learn.microsoft.com/en-us/azure/machine-learning/concept-deep-learning-vs-machine-learning

Talab, Z. (2021, September 23). *Machine Learning vs Deep Learning: What's the Difference?* Developer. https://www.developer.com/guides/machine-learning-vs-deep-learning/

Pedamkar, P. (n.d.). *What is Neural Networks?* EDUCBA.

https://www.educba.com/what-is-neural-networks/

McCulloch, W. S., & Pitts, W.(1943). A logical calculus of the ideas immanent in nervous activity. *The Bulletin of Mathematical Biophysics, 5*(4), 115-133. https://doi.org/10.1007/BF02478259

Ganesh, S. (2020, July 25). *What's The Role of Weights and Bias in A Neural Network?* towards data science. https://towardsdatascience.com/whats-the-role-of-weights-and-bias-in-a-neural-network-4cf7e9888a0f

Sharma, S., Sharma, S., & Athaiya, A. (2017). Activation functions in neural networks. *towards data science, 6*(12), 310-316. https://doi.org/10.33564/ijeast.2020.v04i12.054

Karlik, B., & Olgac, A. V. (2011). Performance analysis of various activation functions in generalized MLP architectures of neural networks. *International Journal of Artificial Intelligence and Expert Systems, 1*(4), 111-122.

Tiwari, S. (2022, November 01). *Activation functions in Neural Networks.* geeksforgeeks. https://www.geeksforgeeks.org/activation-functions-neural-networks/

Alake, R. (2022, March 18). An introduction *To Gradient Descent and Backpropagation In Machine Learning Algorithms.* towards data science. https://towardsdatascience.com/an-introduction-to-gradient-descent-and-backpropagation-in-machine-learning-algorithms-a14727be70e9

IBM Cloud Education. (2020, October 27). *Gradient Descent.* https://www.ibm.com/cloud/learn/gradient-descent

Saha, S. (2018, December 16). *A Comprehensive Guide to Convolutional Neural Networks – the ELI5 Way.* towards data science.

https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53

NVIDIA Developer. (n.d.). *Convolutional Neural Network (CNN).* https://developer.nvidia.com/discover/convolutional-neural-network

Wood, T. (2019, n.d.). *Convolutional Neural Network.* DeepAI. https://deepai.org/machine-learning-glossary-and-terms/convolutional-neural-network

IBM Cloud Education. (2020, October 20). *Convolutional Neural Network.* https://www.ibm.com/cloud/learn/convolutional-neural-networks

Madhaven, S. (2021, July 12). *Introduction to convolutional neural networks.* IBM Developer. https://developer.ibm.com/articles/introduction-to-convolutional-neural-networks/

O'shea, K., & Nash, R. (2015). An introduction to convolutional neural networks. *arXiv preprint arXiv:1511.08458.*

IBM Cloud Education. (2021, March 3). *Overfitting.* https://www.ibm.com/cloud/learn/overfitting

AWS. (n.d.). *What is Overfitting?* https://aws.amazon.com/what-is/overfitting/

Ray, S. (2021, November 27). *What Is Data Augmentation?* towards data science. https://medium.com/lansaar/what-is-data-augmentation-3da1373e3fa1

IBM Cloud Education. (2020, August 17). *Neural Networks.* https://www.ibm.com/cloud/learn/neural-networks

Rajpurkar, P., Irvin, J., Zhu, K., Yang, B., Mehta, H., Duan, T., ... & Ng, A. Y. (2017). Chexnet: Radiologist-level pneumonia detection on chest x-rays with deep learning. *arXiv preprint arXiv:1711.05225.*

Rahman, T., Chowdhury, M. E., Khandakar, A., Islam, K. R., Islam, K. F., Mahbub, Z. B., ... & Kashem, S. (2020). Transfer learning with deep convolutional neural network (CNN) for pneumonia detection using chest X-ray. *Applied Sciences, 10*(9), 3233.
https://doi.org/10.3390/app10093233

Rajaraman, S., Candemir, S., Kim, I., Thoma, G., & Antani, S. (2018). Visualization and interpretation of convolutional neural network predictions in detecting pneumonia in pediatric chest radiographs. *Applied Sciences, 8*(10), 1715.
https://doi.org/10.3390/app8101715