|  |  |  |
| --- | --- | --- |
|  | **T.C.**  **MANİSA CELAL BAYAR ÜNİVERSİTESİ**  **MÜHENDİSLİK FAKÜLTESİ**  **BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ** |  |

**Pneumonia Prediction with Chest X-ray Image**

**Graduation Project II**

**HAZIRLAYANLAR**

**160315039 – Mehmet Sarper KESER**

**DANIŞMAN**

**Doç. Dr. Bora CANBULA**

**MANİSA 2021**

**T.C.**



**MANİSA 2021**

**T.C.**

**MANİSA CELAL BAYAR ÜNİVERSİTESİ**

**MÜHENDİSLİK FAKÜLTESİ**

**BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**

**Tasarım Projesi / Lisans Bitirme Tezi**

**KABUL VE ONAY BELGESİ**

**------------------------------------------------------------------------------------------------------------’ın “--------------------------------------------------------------------------------------------------------------” isimli lisans projesi çalışması, aşağıda oluşturulan jüri tarafından değerlendirilmiş ve kabul edilmiştir.**

Danışman : ……………………………………………………..

Üye :

Üye :

Projenin Savunulduğu Tarih : ...........................................

Bilgisayar Mühendisliği Bölüm Başkanı

contents

[ABBREVIATION LIST: 4](#_Toc76039959)

[ABSTRACT 5](#_Toc76039960)

[INTRODUCTION 5](#_Toc76039961)

[CONVOLUTIONAL NEURAL NETWORK I USE IN MY PROJECT 6](#_Toc76039962)

[Convolutional Layer: 6](#_Toc76039963)

[Insert Pixels (Padding) 6](#_Toc76039964)

[Pooling Layer: 7](#_Toc76039965)

[Scroll Step (Stride): 8](#_Toc76039966)

[Flattening Layer: 8](#_Toc76039967)

[Architecture of Convolutional Neural Networks 8](#_Toc76039968)

[DATA INCREASE (DATA AUGMENTATION): 9](#_Toc76039969)

[DJANGO 9](#_Toc76039970)

[PROJECT 10](#_Toc76039971)

[CONCLUSION 11](#_Toc76039972)

[REFERENCES 12](#_Toc76039973)

# ABBREVIATION LIST:

CNN: Convolutional Neural Network

ANN: Artificial neural network

# ABSTRACT

Today, the problem of processing this data in association with big data has also arisen. At this point, it became clear that it was necessary to extract different information from the data to calculate billions of parameters. A classic artificial neural network model allows us to get meaningful output from data without the need for any preprocessing, but as the data grew, the data had to be optimized. At this point, Yann LeCun presented Convolutional Neural Networks. In my project, I will use the artificial neural network that I created with CNN. This neural network will be trained with chest x-ray image data and x-ray photos will predict whether the person is pneumonia.

*Keywords: artificial neural network, convolutional neural network, chest x-ray image*

# INTRODUCTION

**Radiography** is an imaging technique that uses X-rays, gamma rays, or similar types of radiation to display the inner form of an object. My data set consists of photos created using x-rays.

The method I would use to predict pneumonia convolutional neural networks. This type of neural network prepares the data for the artificial neural network by passing large data and photos through certain filters. We can also create a prediction by connecting each pixel to the neural network without applying these filters. But given the size and complexity of the data, it takes a long time for models to get close to the right results. That's why we're trying to optimize data. The area of interest of the Convolutional neural network is image classification.

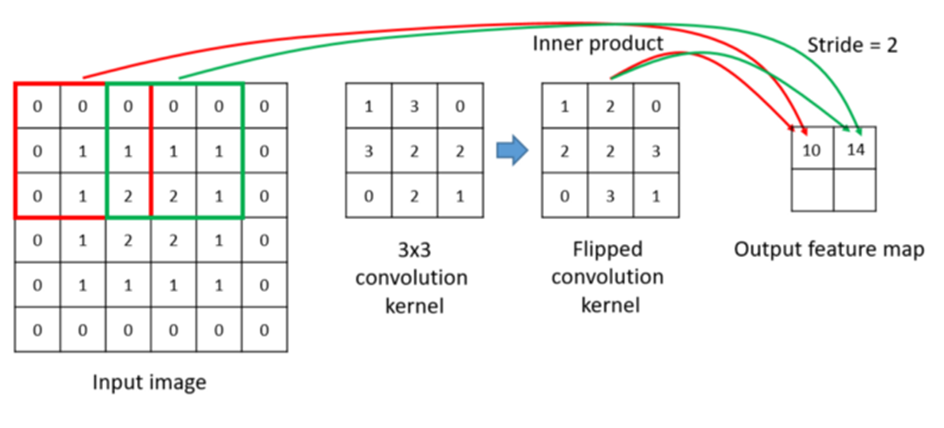
CONSTRUCTION OF CONVOLUTIONAL NEURAL NETWORK:

* Convolutional Layer — Used to determine properties
* Pooling Layer — Reduces weight count and controls suitability
* Flattening Layer — Prepares data for the Classic Neural Network
* Fully-Connected Layer — Standard Neural Network used in classification

# CONVOLUTIONAL NEURAL NETWORK I USE IN MY PROJECT

## Convolutional Layer:

This layer applies some wicks to the image to extract low and high-level features in the image. For example, this filter might be a filter that detects edges. These filters are usually multidimensional and contain pixel values.



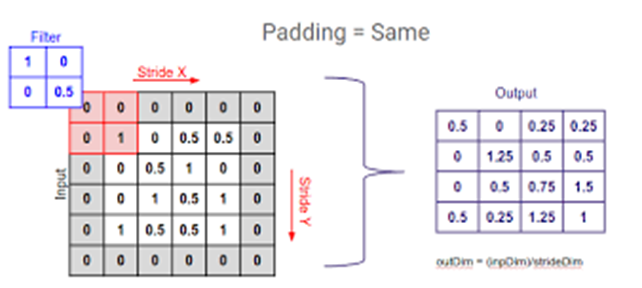
All values are multiplied by element element in matrix, and the sum of all values is recorded as the corresponding element of the output matrix. This process can be done simply when the input data is a single layer. However, input data can be in different formats and number of layers.

First, the filter is positioned in the upper-left corner of the image. Here, the inscriptions between the two matrices (picture and filter) are multiplied by each other and all the results are collected, after which the result is stored in the output matrix. It then moves this filter up to 1 pixel to the right and repeats the operation. 1. After the line ends, 2 lines are moved, and the operations are repeated. After all operations are finished, the output matrix is created.

So, what does the output matrix tell us? by moving the filter through the image and using simple matrix product, we determine our properties. Typically, multiple filters are used to detect multiple properties.

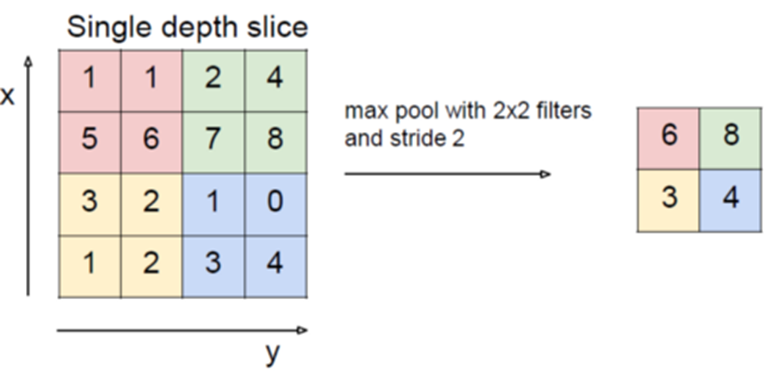
## Insert Pixels (Padding)

Managing the size difference between the input mark and the output mark after the convolution process is a calculation that we have. This is provided with extra pixels to be added to the input matrix. That's exactly what this pixel insertion job (padding) is called. I plan to get the padding value as "same" in the model I use. If the output matrix is desired to be the same size as the input, the input matrix is: (nxn) filter: (fxf), the formula is: (n+2p-f+1)x(n+2p-f+1).



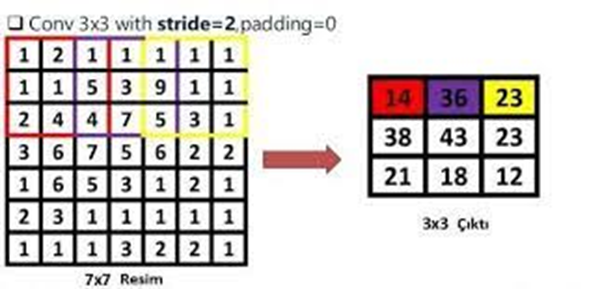
## Pooling Layer:

The task of this layer is to reduce the scroll size of the representation and the parameters and number of calculations within the network. There are many Pooling operations, but the most popular is max pooling. If we have a matrix of 4\*4, it's when you place a 2\*2 filter in the matrix and take the largest of the 4 values. This means taking the values that have the greatest coefficient in photography. By taking the pixels that affect the most, we reduce the data to be processed. Apart from this pooling process, there are also average pooling, and L2-norm pooling algorithms that work on the same principle. However, it compromises my success as it also causes some important information in the data to be lost. That's why many people in the combination prefer not to use this layer. Instead, larger Filter scrolling (Stride) is preferred in the Convolutional layer.



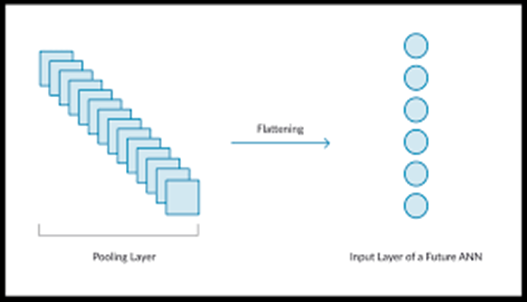
## Scroll Step (Stride):

This value informs you that the filter, which is the weight matrix for the convolution process, will scroll through the image in one-pixel steps or larger steps.



## Flattening Layer:

The task of this layer is simply to prepare the data at the entrance of the last layer, fully connected layer. In general, neural networks retrieve input data from a one-dimensional array. The data in this neural network is a one-dimensional sequence of the matrixes from the Convolutional and Pooling layers.



## Architecture of Convolutional Neural Networks

If we want to create CNN architecture in the simplest way, we need to put a few Convolutional Layers and add relu layers after each. then pooling layers and flattening layers should be added. The Fully Connected layer is then added up to the ReLu layer.

There are some common CNN architectures. Examples LeNet-5, AlexNet, VGG-16, GoogLeNet, ResNet.

# DATA INCREASE (DATA AUGMENTATION):

In small data sets, especially to improve performance, data is increased by exposing it to various distortion effects. In this way, it is ensured that the model also learns different conditions. Replication/increase can be done by creating new pieces of data obtained from the data in many different ways such as taking symmetries of the image according to various axes, cutting and retrieving a random sample part, changing the axes, changing color ratios, and so on. (medium, 2018)



# DJANGO

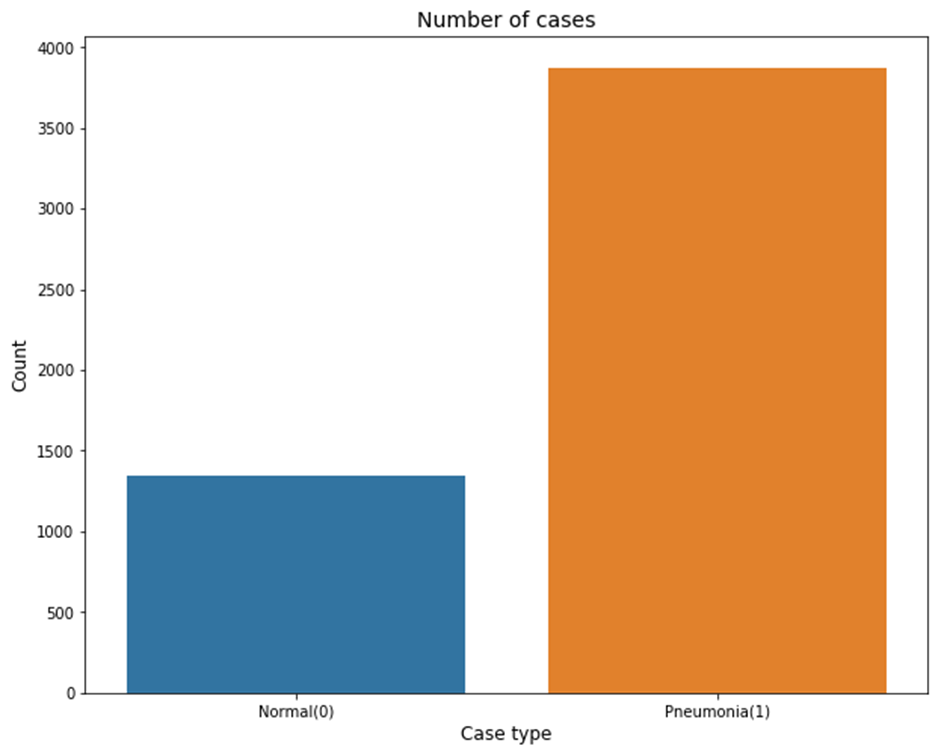
Django is a high-level framework for the Python Programming Language that is licensed with a BSD license. The main purpose of the Django Project is to facilitate the use of web applications that are complex. Django is designed with a policy of reusability, modularity, rapid development process.

So, I thought I'd use Django for my project. I encoded a simple web interface and ran my training there. The website shows whether the person in the photo on the given road is pneumonia. You can develop this simple application and add more comprehensive and beautiful designs.

The biggest advantage of this structure is that it is ideal to make the user realize their mistakes and form ideas.

# PROJECT

In my project, I tried to understand the data at first. There is a total of 5,863 photos in the yield, classified in 2 ways (Pneumonia/Normal). There is a total of 5216 photos in the train data, including 1341 normal 3875 patients. The test data includes a total of 624 photographs, including 234 normal 390 patients.



Due to this distribution in train data, I consider applying data augmentation to normal class photos. In this way, I think it will increase the number of normal photos and provide better learning. However, the resolution of the photos is very high and variable, so I intend to edit each photo in the form of 200X200 or 500X500. Then I plan to create a simple model and train. Right now, I'm taking each of the photos in the form of a path. (Javatpoint, 2021)

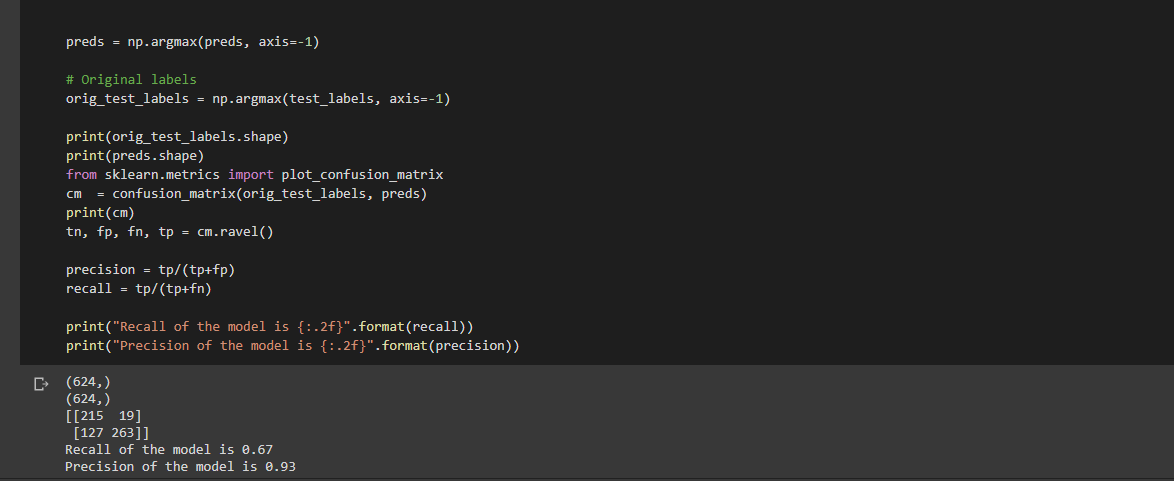
The difficulty in my project was being able to use big data. I had to give you the ways of the photos. However, because the training data was so large, I had to learn and use the generator structure. When I got through this, I tried to save the model from memorizing. First, I increased the number of regular photo photos in the training data with augmentation. With this method, I was trying to prevent the model from calling every tester a patient. But it didn't work. That's what I had the hardest time with. With my research on the Internet, I've seen that VGG16 takes weight values and removes the memorization of education when given with callback attribute. I've tried a lot of methods until I get to this method. here are some of them:

-first, I started my model from scratch. He wasn't memorizing a one-liner, but he wasn't learning properly.

- I tried to generate more data with augmentation.

-I've tried different models.

After a lot of hard work, I've seen that the best way is to use callback. After that, I recorded my model with " load model "so as not to start it while guessing. The test results of the model are as follows.



Then I created a simple web interface with Django. I buried my python code here. In this code I buried, the first head installs the module I recorded. I took the photo from the user's path and edited it. I asked the model to guess if the person in this photo was sick, and I put the estimate on the screen.

# CONCLUSION

Convolutional neural network is also one of the most important elements of data. If the data is incorrect or low, the learning is incorrect or incomplete. We need to pay attention to that when we're playing on data. We need to move towards not distorting and replicating the data. In this project, the data can be called relatively small. Which makes learning weaker. In addition, the weight of neurons is important. The correct weights allow the model to avoid learning speed and memorization. What I've learned in this project is that putting too many filters on CNN systems or adding too many neural networks makes the system memorize and puts too much strain on the system instead of improving it. so constantly adding new layers creates a detrimental situation for the model you create.

# REFERENCES

*medium*. (2018, Mayıs 28). https://ayyucekizrak.medium.com/deri%CC%87ne-daha-deri%CC%87ne-evri%C5%9Fimli-sinir-a%C4%9Flar%C4%B1-2813a2c8b2a9 adresinden alındı