ANKARA YILDIRIM BEYAZIT UNIVERSITY

GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

CLASSIFIYING CT IMAGES OF BRAIN INFARCTION WITH 3D CNN

M.Sc. Thesis by

Nisanur MÜHÜRDAROĞLU

Department of Computer Engineering

May, 2019

ANKARA

CLASSIFIYING CT IMAGES OF BRAIN INFARCTION WITH 3D CNN

A Thesis Submitted to

The Graduate School of Natural and Applied Sciences of

Ankara Yıldırım Beyazıt University

In Partial Fulfillment of the Requirements for the Degree of Master of Science in Computer Engineering, Department of Computer Engineering

by

Nisanur MÜHÜRDAROĞLU

May, 2019

ANKARA

M.Sc. THESIS EXAMINATION RESULT FORM

We have read the thesis entitled **“CLASSIFIYING CT IMAGES OF BRAIN INFARCTION WITH 3D CNN”** completed by **NİSANUR MÜHÜRDAROĞLU** under the supervision of **ASSIST. PROF. DR. ÖZKAN KILIÇ** and we certify that in our opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Assist. Prof. Dr. Özkan KILIÇ

Supervisor

Assist. Prof. Dr. Gül TOKDEMİR	Assist. Prof. Dr. Hilal KAYA
Jury Member	Jury Member

Prof. Dr. Ergün ERASLAN

Director

Graduate School of Natural and Applied Sciences

ETHICAL DECLARATION

I hereby declare that, in this thesis which has been prepared in accordance with the Thesis Writing Manual of Graduate School of Natural and Applied Sciences,

- All data, information and documents are obtained in the framework of academic and ethical rules,
- All information, documents and assessments are presented in accordance with scientific ethics and morals,
- All the materials that have been utilized are fully cited and referenced,
- No change has been made on the utilized materials,
- All the works presented are original,

and in any contrary case of above statements, I accept to renounce all my legal rights.

Date: 2019, 29 May **Signature:**

Name & Surname: Nisanur MÜHÜRDAROĞLU

ACKNOWLEDGMENTS

Firstly, I would like to express my sincere gratitude to my supervisor, Assist. Prof. Dr. Özkan KILIÇ for his tremendous support and motivation during my study. His immense knowledge and precious recommendations constituted the milestones of this study. His guidance assisted me all the time of my research and while writing this thesis.

I also would like to thank Dr. Emirhan TEMEL for dataset which is one of the most important part of our project. He always helped and informed us whenever we need any information.

Finally, I must express my profound appreciations to my family for providing me emotional support, loving care and continuous encouragement throughout my years of study and through the period of writing this thesis. This accomplishment would not have been possible without them. Thank You.

2019, 29 May **Nisanur MÜHÜRDAROĞLU**

CLASSIFICATION OF CEREBNAL INFARCTION WITH 3D CNN

ABSTRACT

Brain infarction occurs as a result of a blockage in the arteries which supply blood and oxygen to the brain. The restricted oxygen causes to stroke that can result in an infarction if the blood flow is not normalized in a short period of time. Approximately, 0.6% of people suffer from stroke every year. About one third is fatal. In addition, stroke is a third leading cause of death. For diagnosis, doctors want to see MRI (Magnetic Resonance Imaging) results to ensure if the patient has infarction or not. However, this process takes a long time while patients require immediate intervention. Losing time while waiting for an exact diagnosis might have fatal consequences. On the other hand, doctors can have CT (Computed Tomography) scan results in a short period of time but is not enough to tell the exact diagnosis for infarction due to uncertainty and the low quality of the imaging technique. This work aims to classify CT scans if the patient has infarction or not. This study uses 3D Convolutional Neural Network (3D-CNN) methodology. It achieves 93% as a maximum accuracy and 74% accuracy after 10-fold testing. We believe that this method could be used as a decision support system to detect the patients with a higher risk of infarction, and prioritize utilization to MRI for them to make the final diagnosis quickly.

Keywords: brain infarction, computed tomography, deep learning, convolutional neural network, classification, image processing

BEYİN ENFARKTÜS CT GÖRÜNTÜLERİNİN 3D CNN METODUYLA SINIFLANDIRILMASI

ÖZ

Beyin enfarktüsü, beyinde kan ve oksijen sağlayan arterlerde tıkanma sonucu oluşur. Kısıtlı oksijen, kan akışı kısa sürede normalleştirilmezse enfarktüs ile sonuçlanabilecek felce neden olur. Yaklaşık her yıl insanların% 0.6's felç geçiriyor. Yaklaşık üçte biri ölümcüldür. Ayrıca, inme üçüncü önde gelen ölüm nedenidir. Teşhis için doktorlar, hastanın enfarktüsü olup olmadığından emin olmak için MRI (Manyetik Rezonans Görüntüleme) sonuçlarını görmek ister. Bununla birlikte, hastalar acil müdahale gerektirirken, bu işlem uzun zaman alır. Kesin tanı için beklerken zaman kaybetmek ölümcül sonuçlara neden olabilir. Öte yandan, doktorlar BT (Bilgisayarlı Tomografi) taramalarına rağmen kısa sürede sonuç alabiliyorlar. Belirsizlik ve görüntüleme tekniğinin kalitesinin düşük olması nedeniyle enfarktüs için kesin tanı konulması yeterli değildir. Bu çalışma, hastanın enfarktüsü olup olmadığını BT taramalarını sınıflandırmayı amaçlamaktadır. Bu çalışma 3D Convolutional Neural Network (3D-CNN) metodolojisini kullanıyor ve en iyi %93 ve 10 test sonrası ortalama olarak %74 doğruluk elde ediyor. Bu yöntemin, enfarktüs riski daha yüksek olan hastaları saptamak için bir karar destek sistemi olarak kullanılabileceğine ve MRI ile hastanın kesin teşhisini hızlı bir şekilde yapmaları için kullanılmasına öncelik verdiğine inanıyoruz.

Anahtar kelimeler: beyin enfarktüsü, bilgisayarlı tomografi, derin öğrenme, evrişimli sinir ağı, sınıflandırma, görüntü işleme

CONTENTS

NOMENCLATURE

Acronyms

CNN	Convolutional Neural Network
DICOM	Digital Imaging and Communications in Medicine
HU	Hounsfield Units
ReLU	Rectified Linear Unit
CT	Computed Tomography
MRI	Magnetic Resonance Imaging

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

A cerebral infarction is an area of necrotic tissue in the brain resulting from a blockage or narrowing in the arteries supplying blood and oxygen to the brain. The restricted oxygen due to the restricted blood supply causes an ischemic stroke that can result in an infarction if the blood flow is not restored within a relatively short period of time. About one third will prove fatal [1].

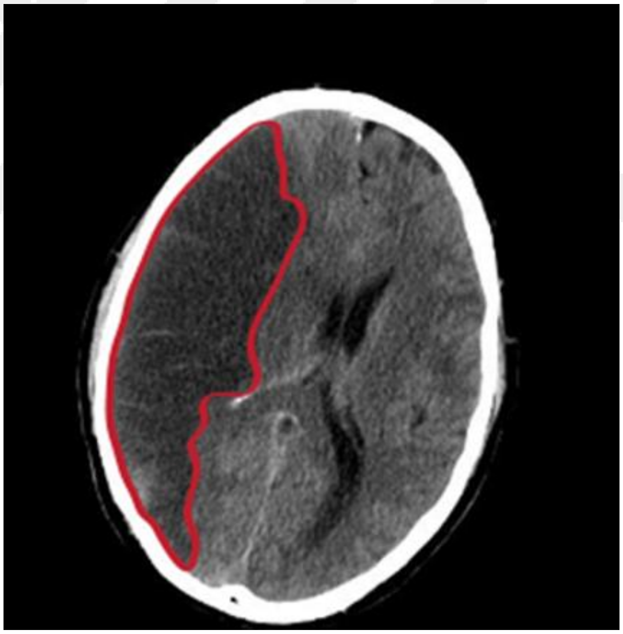


Figure 1.1 CT scan slice of the brain showing a right-hemispheric cerebral infarct (left side of image)

1.1 What is the difference between CT and MRI for Brain Imaging?

CT and MRI are complementary techniques, each with its own strengths and weaknesses. The choice of which method is appropriate depends on some situations such as how quickly it is necessary to obtain the scan, what part of the head is being scanned, age of the patient, etc. [2] The advantages of each modality is listed in the following that doctors use to decide between head CT and MRI.

Advantages of head CT:

- CT is much faster than MRI.
- CT is less costly when compared with MRI, and it is sufficient to diagnose many neurological disorders.
- CT is less sensitive to motion of patients.
- CT may be easier to perform in very heavy patients.
- CT can be obtained at no risk to the patient with implantable devices.

Advantages of head MRI:

- MRI does not use ionizing radiation. Therefore, it is preferred over CT in children.
- MRI has a much greater range of available soft tissue contrast, depicts anatomy in greater detail, and is more sensitive and specific for abnormalities within the brain itself.
- MRI has smaller risk of allergic reaction.
- MRI allows the evaluation of structures that may be obscured by artifacts from bone in CT images

In this study, CT scan images were used. CT images do not provide clear vision as compared with MRI scans. For example, images of patients who have infarction could not be clear to make diagnosis from CT scan. However, to examine MRI scans doctors diagnose in a certain way. As we explain above, they have advantages and disadvantages. Waiting MRI scans could be waste of time for emergency patient. This study aims to classify patients using CT scans to give priority to MRI. Now, we will see the differences between MRI and CT scans.

Figure 1.2 shows us difference between MRI and CT scan differences. We can see clearly the infarcted area by looking MRI image even if we are not a doctor. However, when we look at the right image (CT scan) there is a little color differences and brain folds disappeared. Therefore, it is difficult to classify in CT scans.

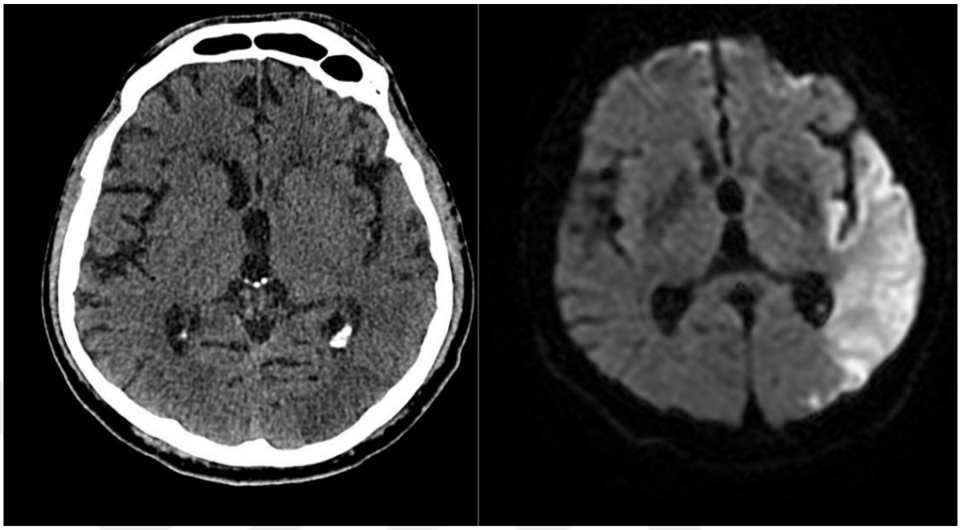


Figure 1.2 An example of infarction with big volume (right one is MRI scan left one is CT scan)

Because CT scan has an ability to show old infarcted area but not MRI, we can see the new infarcted area and old infarcted area in Figure 1.3, Old infarcted area is darker pixels in right side of bottom, the new infarcted area is darker pixel in left side of figure. It is hard to classify whether the patient has infarcted or not like in Figure 1.2.

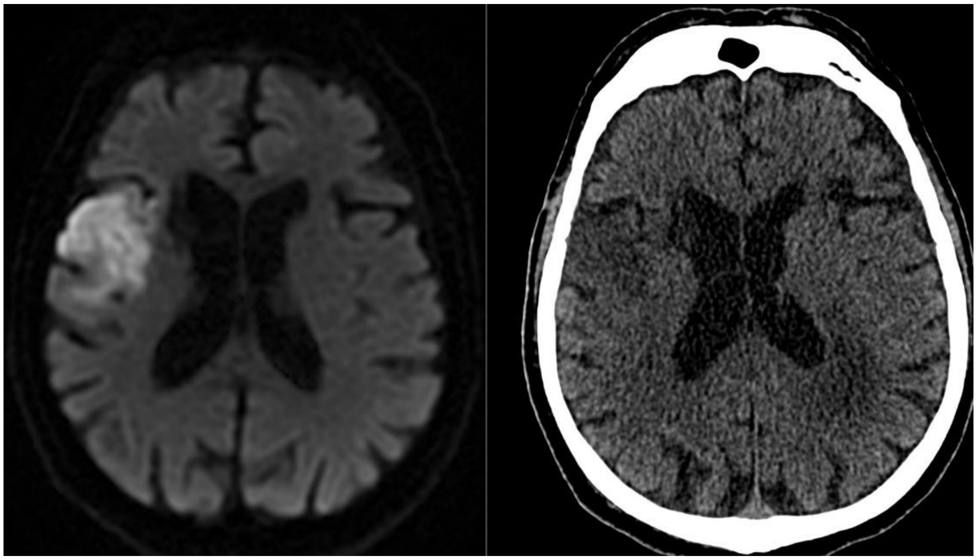


Figure 1.3 An example of old and new infarction

Figure 1.4 shows us it is not possible to say the patient has infarction, since there is small volume of infarction. When there is small infarcted area, it is not possible to detect infarcted area for doctors using CT scan.

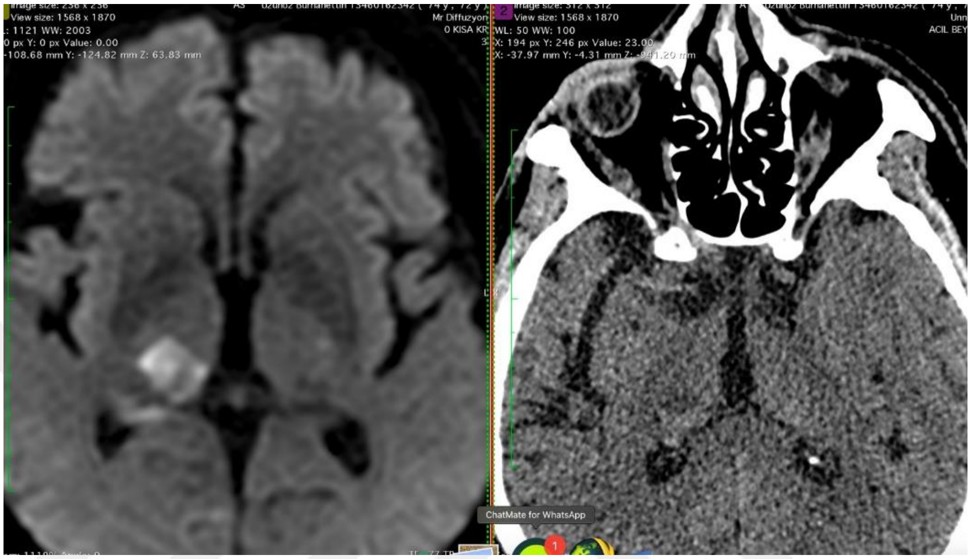


Figure 1.4 Low volume of infarcted area

Figure 1.5 is the other example of small volume of infarction. As we show in Figure 1.4 and Figure 1.5, it is not possible to say there is an infarcted area by looking at the left CT image.

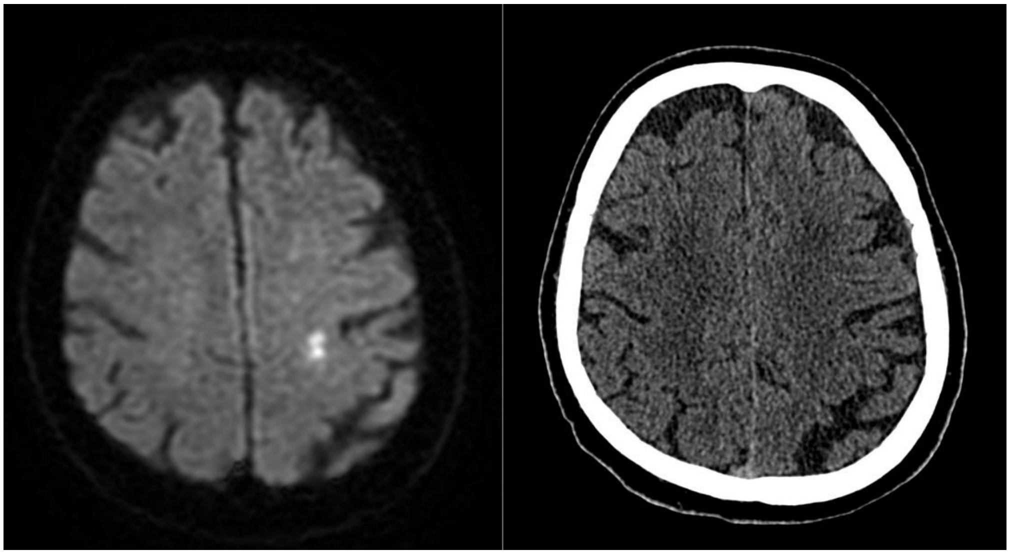


Figure 1.5 Another example of low volume infarcted area

As we see from example figures by looking CT images doctors have difficulty to make diagnosis. MRI scans gives clearness to doctors to say the patient has infarction. However, it is not easy to have MRI scan as CT images because MRI images of the patient can take a long time. During this time the patient may be lost. For this reason, we aim to classify the patient by experimenting CT scans. Thus, priority can be given to urgent patients who are classified as infarcted patient.

1.2 Literature Review

Nowadays, literature has some studies about medical diagnosis using deep learning methodology. This section explains some researches about medical imaging to make a diagnosis.

Hosseini-Asl E. et al. aims to predict Alzheimer disease with 3D-CNN. MRI scans was used as medical imaging type. The MRI dataset has no skull stripping in preprocessing. It claims that the study outperforms several classifiers in terms of accuracy and robustness. In methodology, Theano library was evaluated to implement CNN. The dataset has MRI scans of 210 patients. After measuring seven classification metrics by using ten-fold cross validation, results outperform other state-of art models [27].

Tong P.D. et al. suggest an approach for diagnosis of brain hemorrhage using deep learning methodology. Three types of CNN architecture that are LeNet, GoogLeNet and Inception-ResNet were implemented. The dataset consists of 100 cases from 115 hospitals. Accuracy of 0.997, 0.982 and 0.992 were obtained from LeNet, GoogLeNet, and Inception-ResNet respectively. Through accuracy results, they say that this model can be used in medical image diagnosis, especially in brain hemorrhage diagnosis [28].

Kuan K. et al. developed a computer aided detection (CAD) system for lung cancer diagnosis. Labeled dataset is from Kaggle Data Science Bowl 2017 which provides CT scan images of patients. It provides nodule annotations. However, it does not give information about cancer status. RCNN is the method that is used for nodule and malignancy detector, nodule classifier and patient classifier. This method was validated as 41st out of 1972 teams by the competition [31].

Alakwaa W. et al. aims classifying CT images of lung cancer. The dataset which contains 1397 labeled data is from Kaggle Data Science Bowl 2017. Several segmentation techniques are used. These techniques are threshold and watershed. They proved that watershed method is more successfully. After segmentation, they normalized images using linear scaling. Finally, they made zero-centering by subtraction the mean of images in training set. Test set accuracy is 86.6%. The performance outperforms current CAD systems in literature. They achieve state-of-art CAD system [30].

He et al. implemented deep learning to classify lung cancer using CNN. They made different preprocessing methods. ResNet-50 and Inception-v3 are used to extract features, and then XGBoost is used to classify the data. In addition, a simple 3D CNN was implemented with three layer. They downsized the images to 50 x 50 from 512 x 512 in preprocessing. The best result is obtained by using ResNet and XGBoost without preprocessing [32].

Milletari et al. proposes an approach to 3D image segmentation based on volumetric data. Aim of the study is to segment prostate MRI volumes. This task is suitable during diagnosis and treatment planning. Prostate segmentation could be difficult because of different scanning in terms of intensity changes and deformations. This work uses power of fully convolutional neural network to make 3D segmentation. Firstly, 3D convolutions were proposed instead of processing. Secondly, they tried to maximize a novel objective based on Dice coefficient function for image segmentation. They achieved fast and accurate results on MRI prostate volumes. This work was implemented using Caffe. It can be aimed to make segmentation in multiple regions as a future work [33].

Chon et al. proposes a computer aided diagnosis (CAD) system for classification of lung cancer using CT scans. The dataset is from Kaggle Data Science Bowl 2017. First task is thresholding for segmentation of lung tissue. Initially, they were directly feed into 3D-CNN. However, they proved that it is not enough to have good accuracy value. The vanilla

3D-CNN and Google-net based 3D-CNN architectures was used. This CAD system has three phases as segmentation, nodule candidate detection, and malignancy classification. They preprocessed CT scans with segmentation, down sampling, and zero-centering. Accuracy is 0.705 in Vanilla CNN architecture and 0.751 in 3D Google-net architecture [34].

Mohsen et al. used deep neural network classifier for classifying a dataset of 66 MRI brain images into 4 classes. Evaluation of performance is good after using principal component analysis (PCA). As a methodology DNN was used. Dataset was taken from World Health Organization (WHO). Fuzzy C-means was used in image segmentation. Discrete wavelet transform was used for future extraction. DNN was used for classification. Classification rate is 96.97% using DNN [35].

Işın et al. focuses on deep learning methods instead of traditional methods. Firstly, they make an introduction about brain tumors and methods for segmentation of brain tumor. Then, state-of-the-art algorithms are discussed that focus on recent deep learning methods. Finally, an assessment of the recent state is presented and possible future works to make standard brain tumor segmentation in MRI scans. There are different methodologies for brain image segmentation. First method is manual segmentation. It requires a radiologist information and experience. Procedure requires the radiologist going through slice by slice. Radiologist diagnoses the tumor and manually draws the region of tumor. Although it is so time consuming, it is widely used to reach the results of semi-automatic and fully automatic methods. Second method is semi-automatic method. It requires user interaction. After applying automated algorithms, it is needed to prove. Though semi-automatic brain segmentation is less time consuming than first method, they are still prone to user. Therefore, recent brain segmentation is focused on fully automatic segmentation methods. It does not require user interaction. Main part of it is artificial intelligence and knowledge are combined to solve segmentation problem. The most used deep learning model in segmentation is 3D-CNN and 2D-CNN is possible [36].

Pereira et al. presents an automatic segmentation methods using CNN. They use 3 x3 kernels to allow deeper architecture. Also, it gives positive effect to over-fitting. Intensity normalization was investigated in preprocessing. Their proposal was presented in Brain Tumor Segmentation Challenge 2013 database (BRATS 2013). After preprocessing, data augmentation was implemented which is common procedure in CNN. This method is more useful in relatively small dataset. DSC score is obtained as 0.78 [37].

Havaei et al. presented a fully automatic segmentation of brain tumor using Deep Neural Networks (DNNs). The dataset has both low and high grade in MR images. Healthy brains are made of 3 types of tissues as white matter, gray matter and cerebrospinal fluid. The aim of brain tumor segmentation is to find location and tumor regions. Tumors can see anywhere in brain and have any shape and size. Therefore, they described a 2-phase training procedure that allows to deals with difficulties about imbalance labels of tumors. Finally, they provide a cascaded architecture that the output is used as output of subsequent CNN. Results were offered on 2013 BRATS (Brain Tumor Segmentation Challenge). Their architecture is 30 times faster than published state-of-the-art [38].

Gao et al. developed CNN architecture that combined 2D and 3D architecture. CNN architecture has several layers both linear and non-linear. Therefore, it has ability to learn and it can build high level information from low level features. The dataset was collected from Navy General Hospital, China, which consists of 57, 115 and 113 data respectively AD, lesion and normal. Slices are between 16 and 33 and dimension with either 912 x 912 or 512 x 512 pixels. All slices were cropped and normalized into 200 x 200 pixels. The middle 20 slices were used for processing. Although, lesion dataset is smaller the overall slices are similar to the other two. Segmentation and normalization were made to arrive at a resolution of 200 x 200 x 20 pixels. Then, spatial normalization was performed because of slice thickness to align all 3D CT images into same space. After normalization, each 3D dataset was divided into 40 x 40 x 10 boxes. This study uses architecture of MatConvNet. The network has seven layers. Accuracy rate of classification is 87.62% in average [39].

Fu et al. focuses on machine learning techniques for medical imaging data. Recently, machine learning techniques have been applied in the field of medical imaging. It is assumed that deep learning is the default machine learning technique because it can learn much more special things than the other machine learning techniques. It makes easier to make feature engineering process. Also, some raw data could be applied directly. This property is so important for medical imaging field, since it could take many years to get valuable domain for appropriate future determination. Therefore, this provides to implement easier and faster ideas. Among all deep learning methods, CNNs are more useful for medical image analysis with rectified linear unit. Researchers can provide deeper models to train more efficiently [40].

Bentley et al. applied machine learning methods to acute stroke CT images. The dataset has 116 acute ischemic stroke patients and 106 patients were used for training. CT brain images acted as inputs into SVM. Performance of SVM and established prognostication tools was compared. Performance of the SVM is found as 0.744 as opposed to 0.626 of prognostic tools performance [41].

Zhang et al. was applied image processing to the brain CT images. The symmetric feature based on the characteristics of brain was extracted. See5, inductive learning techniques and RBFNN (Radial Basis Function of Nerve Network) was used to make classification of normal and abnormal brain CT images. Results showed that doctors can assist doctors to make correct classification of human brain. Aim of preprocessing is to increase data quality to reduce the noise and enhance the edges of image. The aim of second step is to quantize features of brain that have abstracted by computer in preprocessing step. Once the features have extracted, classification accuracy can be improved. Size, density or shape of the pathological change regions is so valuable for doctor decisions. The third step is classification. The main methods that are used are ANN, Bayesian, decision tree, SVM, and so on. The dataset has 212 instances, which 103 instances are normal and others abnormal. The data is divided 20% as test and 80% as training data. Accuracy of some features is lower than the references′. Dataset is not good in terms of quality if images. The dataset includes noise and even wrong information [42].

Gong et al. presented a method for classification of computed tomography (CT) brain images of different head trauma types. The first step is segmentation of brain which has hemorrhage using ellipse fitting. Each segmented feature is classified using machine learning algorithm. The medical image classification can be useful for content based image retrieval system. The first step of preprocessing is removing the skull and segment the interior region. Skull region is white color. Therefore, they treated pixels with density 250 and above as the skull. Secondly, the interior area was segmented. Hemorrhage areas are white color but it is not white as skull. Therefore, its density is between 10 and 250. After the preprocessing, thresholds were applied to images. A decision tree classifier was used to train and test the region features. 10-fold cross validation was used. The result of classification is very close to the doctor′s knowledge in classifying potential hemorrhage regions [43].

CHAPTER 2

CONVOLUTIONAL NEURAL NETWORK

Convolutional networks are composed of an input layer, an output layer, and one or more hidden layers. A convolutional network is different than a regular neural. Regular Neural Networks transform an input by putting it through a series of hidden layers. Every layer is made up of a set of neurons, where each layer is fully connected to all neurons in the layer before. Finally, there is a last fully-connected layer (the output layer) that represent the predictions.

Convolutional Neural Networks are a bit different. First of all, the layers are organized in 3 dimensions**:** width, height and depth. The hidden layers are a combination of convolution layers, pooling layers, normalization layers, and fully connected layers. CNNs use multiple convolution layers to filter input volumes to greater levels of abstraction.

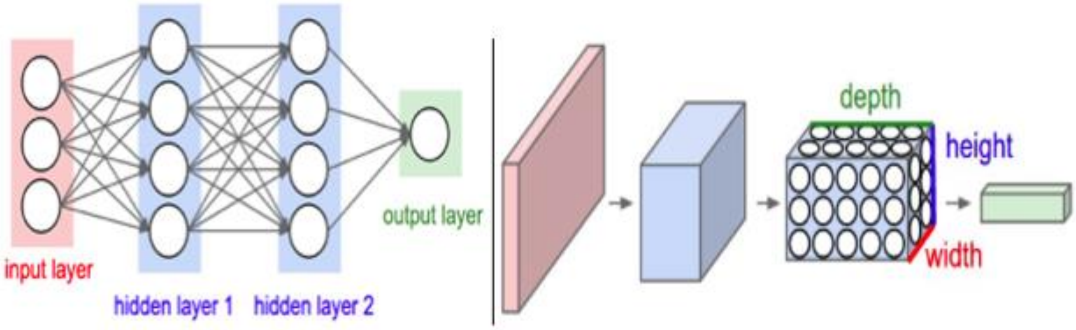


Figure 2.1 Normal NN vs CNN [3]

In Figure 2.1, we see differences between normal Neural Network and Convolutional Neural Network (CNN). Left side of figure is a regular 3-layer Neural Network. Right side of figure is a ConvNet. It arranges its neurons in three dimensions (width, height, depth), as visualized in one of the layers. Each layer of a ConvNet transforms the 3D input volume to a 3D output volume of neuron activations. In this example, the red input layer holds the image, so its width and height would be dimensions of the image, and the depth would be 3 (Red, Green, Blue channels).

CNNs are designed to process data in the form of multiple arrays such as 1D for signals and sequences, including language; 2D for images or audio spectrograms; and 3D for video or volumetric images. There are four key ideas behind CNNs as following: local connections, shared weights, pooling and the use of many layers [4].

CNN is one of the most popular deep learning methodology. Convolutional neural network is useful for image classification, recommender systems, medical image analysis and natural language processing. Also, CNNs can be applied to sound if it is represented visually as a spectrogram. More recently, convolutional networks have been applied directly to text analytics as well as graph data with graph convolutional networks [5].

While CNN may seem like a strange system of biology and computer science, this is a very effective mechanism used for image recognition. CNNs give great results in image recognition. This is the reason of why the world has paid attention to CNNs. It gives many advantages in computer vision applications such as robotics, self-driving cars, drones, medical diagnosis. Images are high-dimensional vectors. Therefore, CNNs process images as tensors. It would take huge amounts of parameters to characterize the network. To solve this problem, convolutional neural network was developed which reduces parameters and adapts the architecture for vision tasks. So, one of the most useful aspects of CNN is reducing number of parameters in ANN. Thus, it was possible to overcome the complex work. In addition, CNN needs little pre-processing compared to other image classification algorithms. Therefore, CNN has great performance in machine learning problems.

2.1 Biology of the idea

The basic idea of the CNN was inspired by biology called as receptive field [6]. Receptive fields are a feature of the animal visual cortex [7]. They act as detectors that are sensitive to certain types of stimulus, for example, edges. They are found across the visual field and overlap each other.

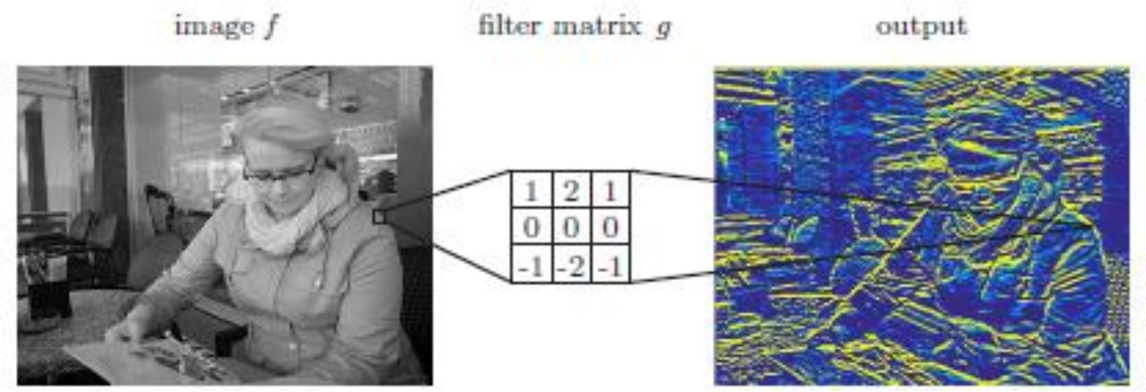


Figure 2.2 Detecting horizontal edges from an image using convolution filtering. [8]

This biological function can be approximated in computers using the convolutional operation [9]. In image processing, images can be filtered using convolution to produce different visible effects.

When we look at an airplane picture, we can define the aircraft by separating the characteristics of two wings, engines and windows. CNN does the same thing, but before they detect lower-level features such as curves and edges, and form them to more abstract concepts. It uses unique features.

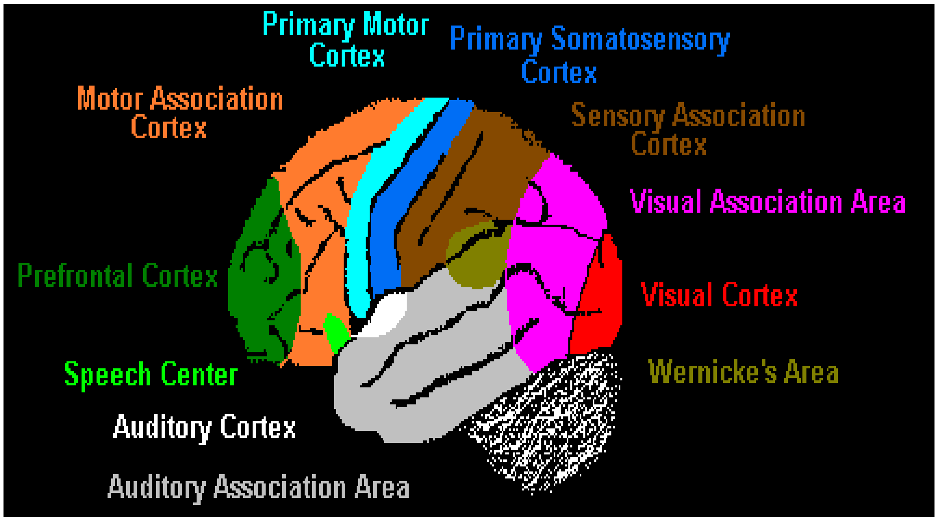


Figure 2.3 Functional Divisions of Brain (Visual Cortex is in Red Color) [29]

In Figure 2.3, we see functional divisions of the brain. Visual Cortex is part of the cerebral cortex of the brain which is interested in visual processes. Visual nerves from eyes run straight to the primary visual cortex. Functional and structural characteristics are divided into different areas as shown in the Figure 2.4.

1. Primary visual area (area 17)
2. Visual association area (area 18 & 19)
3. Higher visual association (area 39)

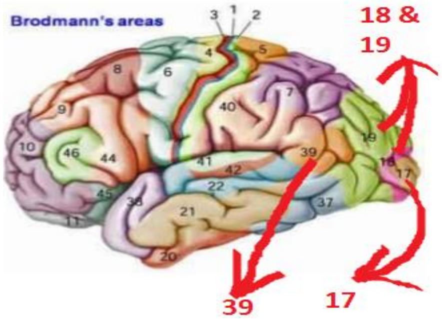


Figure 2.4 Broadmann’s areas

The visual information is passed from one cortical area to another. Each cortical area is more specialized than the last one. The neurons respond to the specific actions in the specific field. Some of fields with their functions are as following:

1. Primary Visual Cortex or V1: It protects location of visual information such as orientation of edges and lines. It receives the signals from eyes what have captured.
2. Secondary Visual Cortex or V2: It receives feed-forward connections from primary visual cortex and sends strong connections to V3, V4 and V5. Also, It sends feedback network strongly to V1. It collects spatial frequency, color, size and shape of the object.
3. Third Visual Cortex or V3: It receives inputs from V2. It helps to process global motion and brings complete visual representation.
4. V4: It receives inputs from V2. It knows simple geometric shapes and forms recognition of object. It is not for complex objects such as human faces.
5. Middle Temporal (MT) Visual Area or V5: It is used to detect direction of moving objects and speed such as motion perception. Also, it detects motion of complex visual features. It receives connections from V1.
6. Dorsomedial (DM) Area or V6: It is used to detect self-motion simulation and wide field. Also, it receives direct connections from V1 like V5. It has highly sharp selection of the orientation of visual contours.

2.2 Structure of convolutional neural networks

As we understand from section 2.1, visual cortex acts as layers of the CNN. It takes scenarios as edge detection, face detection, invariance detection such as rotated face detection, large or small face detection. To achieve the functionality that we mentioned, Convolutional neural networks consist of set of layers which is grouped by their functionalities as hidden layers and classification part. In hidden layer, the network will perform a series of convolutions and pooling operations during which the features are detected**.** If you had a picture of a zebra, this is the part where the network would recognize its stripes, two ears, and four legs.

- Convolutional Layer: This layer is used to detect properties
- Non-Linearity Layer: This layer is used to introduce the non-linearity to system.
- Pooling (Down-sampling) Layer: makes the representations smaller and more manageable

In classification layer, the fully connected layers will serve as a classifier on top of these extracted features. They will assign a probability for the object on the image being what the algorithm predicts it is.

- Fully-Connected Layer: Contains neurons that connect to the entire input volume, as in ordinary Neural Networks

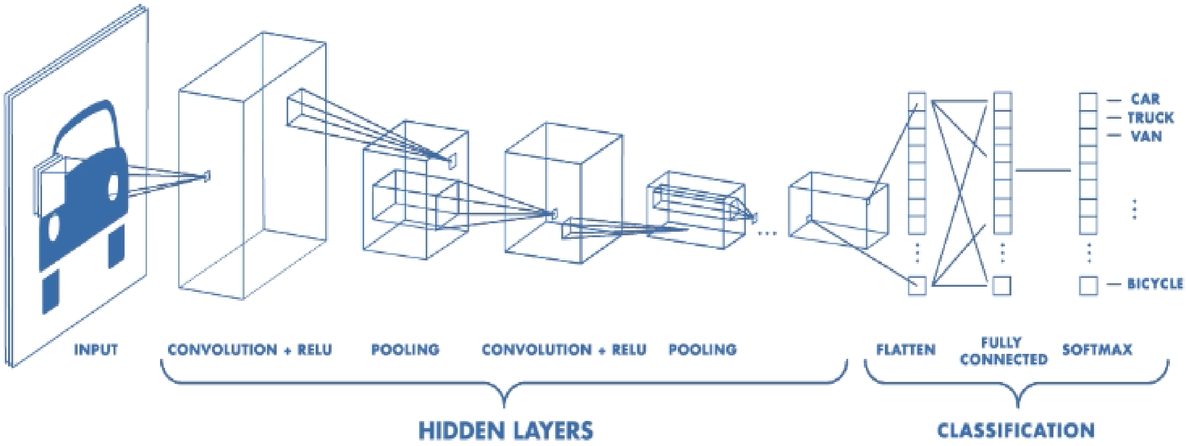
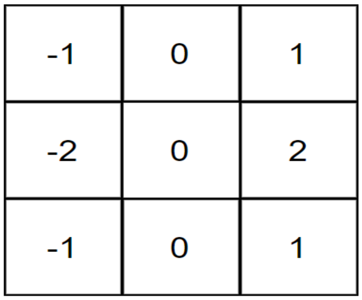


Figure 2.5 Architecture of CNN[10]

2.2.1 Convolutional Layer

This layer is the main building block of CNN. It is responsible for detecting the characteristics of the image. This layer applies some filters to the image. Convolution preserves the relationship between pixels by learning image features using small squares of input data. It is a mathematical operation that takes two inputs such as image matrix and a filter or kernel. Convolutional network takes a normal color image as a rectangular box whose height and width are measured using number of pixels along dimensions, and whose depth is three layers deep, one for each letter in RGB. Depth layers are represented as channels. It is required to pay close attention to the precise attention of each dimension of image volume, because they are foundation of linear algebra operations which are used to process images. Rather than focus on one pixel at a time, convolutional network takes in square patches of pixels and passes them through a filter. Also, the filter is a square matrix smaller than the image itself. It is also called as kernel. In the following Figure 2.6, we see two steps of convolutional operations. To show operations we use a 3x3 filter as Figure 2.6.



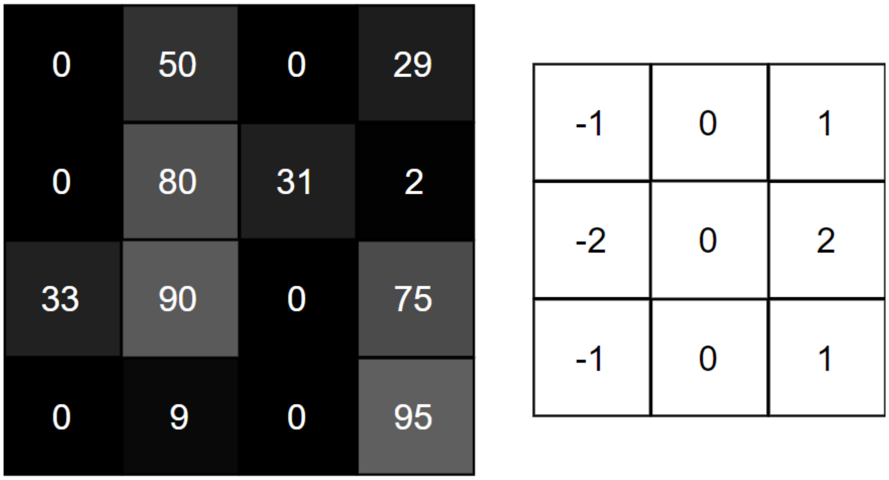
-1	0	1
-2	0	2
-1	0	1

Figure 2.6 A 3x3 filter

We can use an input image and a filter to produce an output image by convolving filter with the input image. This consists of the following steps:

- Covering the filter at some location on top of the image.
- Applying element-wise multiplication between the values in the filter and corresponding values in the image.
- Summing all the element wise products. This sum is the output for destination pixel in the output image.
- Repeating for all matrices.

We describe those processes as an abstract to understand. Like in the Figure 2.7, we use 4x4 gray scale image and 3x3 filter.

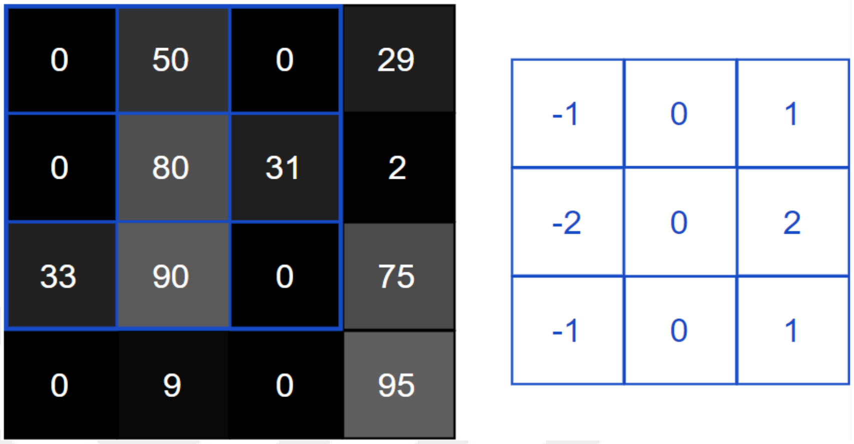


0	50	0	29
0	80	31	2
33	90	0	75
0	9	0	95

-1	0	1
-2	0	2
-1	0	1

Figure 2.7 A 4x4 image (left) and a 3x3 filter (right)

Now, we will convolve the filter like in the Figure 2.8 and the input image to produce a 2x2 output image. To start, we cover the filter in the top left corner of the image.



0	50	0	29
0	80	31	2
33	90	0	75
0	9	0	95

-1	0	1
-2	0	2
-1	0	1

Figure 2.8 Overlap the filter (right) on top of the image (left)

Next, we make matrix multiplication between the image values and filter values. In the following Table 2.1, we find result for an output matrix.

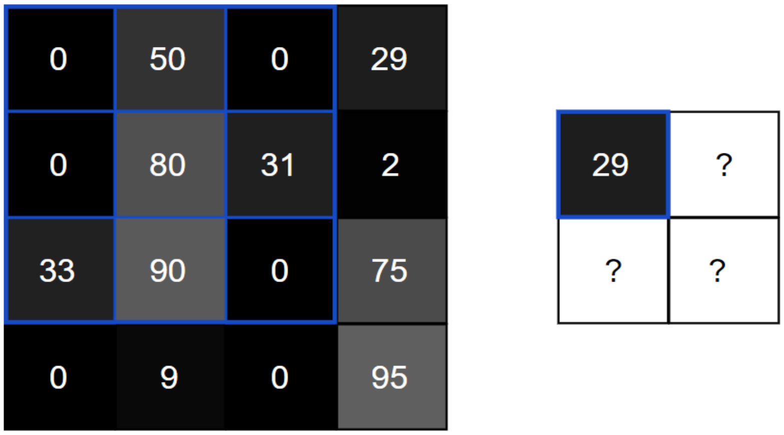
Table 2.1 Performing element-wise multiplication.

Image Value	Filter Value	Result
0	-1	0
50	0	0
0	1	0
0	-2	0
80	0	0
31	2	62
33	-1	-33
90	0	0
0	1	0

Next, we need to sum all the results.

$$62 - 33 = \boxed{29}$$

Finally, we place the result to output matrix. Since out filter is covered in the top left corner of the input image, our output matrix is the top left pixel of the output image.

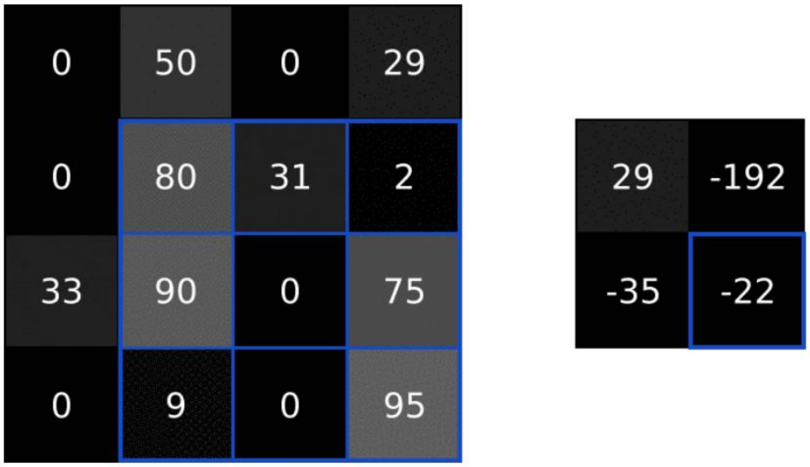


0	50	0	29
0	80	31	2
33	90	0	75
0	9	0	95

29	?
?	?

Figure 2.9 First pixel of output matrix

After doing same operations, we have the output matrix like in Figure 2.10.

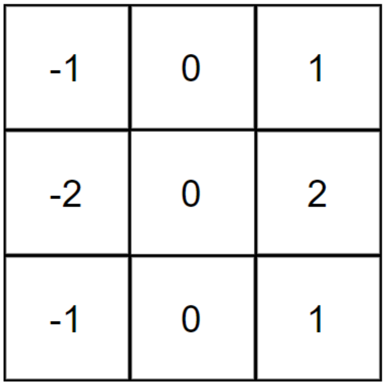


0	50	0	29
0	80	31	2
33	90	0	75
0	9	0	95

29	-192
-35	-22

Figure 2.10 The final output matrix after convolution

Convolution of an image with different filters can perform operations such as edge detection, blurring and sharpen by applying filters. Convolution changes the pixel values. Then, what does convolving to an image using a filter? We can give Sobel filter as an example. The sobel filter values as in the Figure 2.11.



-1	0	1
-2	0	2
-1	0	1

Figure 2.11 The vertical Sobel filter

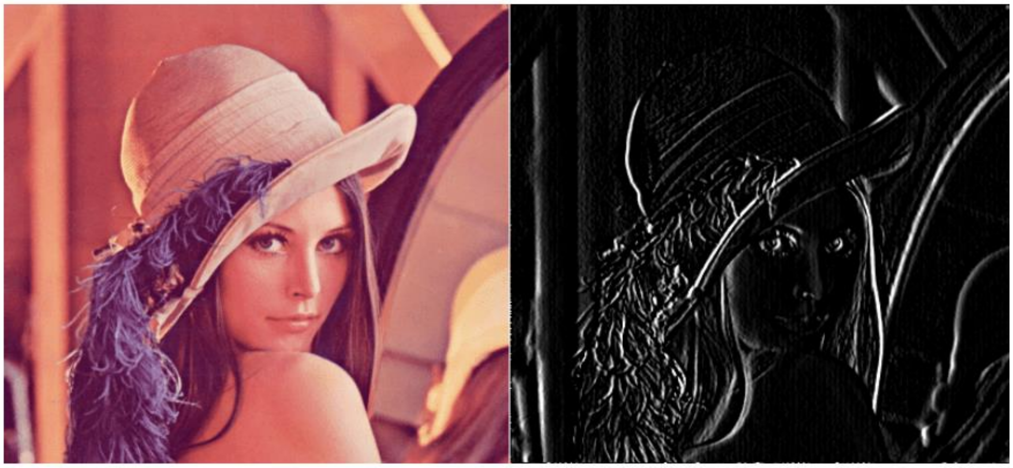
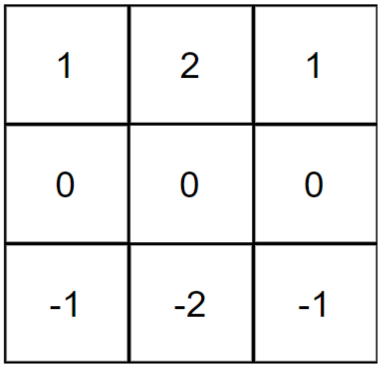


Figure 2.12 An image convolved with vertical Sobel filter

The Figure 2.12 shows us what the vertical Sobel Filter does. Similarly, we have horizontal Sobel filter as Figure 2.13. After applying this filter we obtain the Figure 2.14.



1	2	1
0	0	0
-1	-2	-1

Figure 2.13 The horizontal Sobel filter

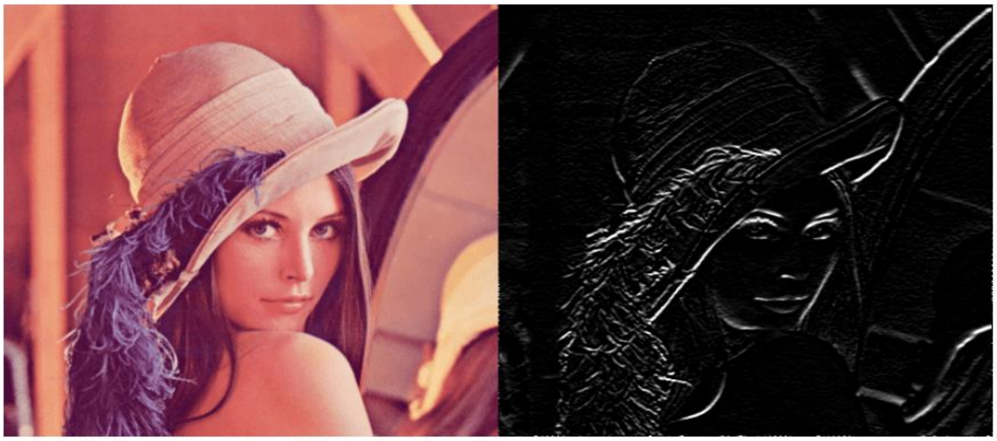


Figure 2.14 An image convolved with the horizontal Sobel filter

As we see from figures, Sobel filters are edge detectors. The horizontal Sobel filter detects horizontal edges, and the vertical Sobel filter detects vertical edges. The bright pixels in the output image indicate that there is a strong edge in the original image.

We can see the importance of edge-detected image from examples. Convolution helps us to look for specific localized image features like edges that we can use later in the network.

2.2.2 Pooling

A pooling layer is a new layer added after the convolutional layer. Specifically, after a non-linearity (e.g. ReLU) has been applied to the feature maps output by a convolutional layer for example, the layers in a model may look as follows:

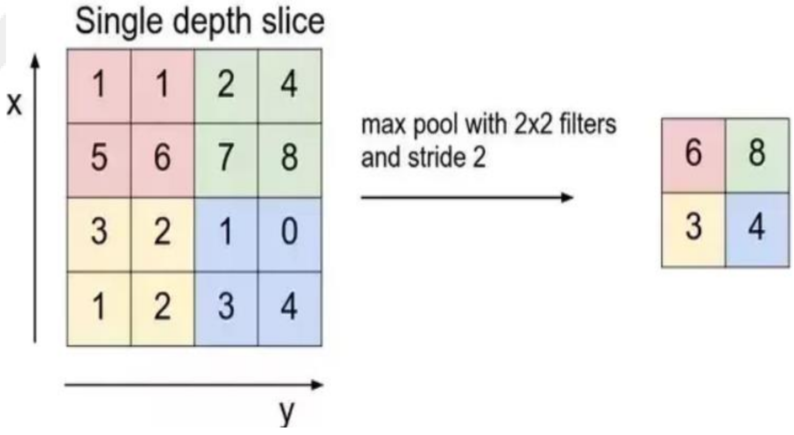
- Input Image
- Convolutional Layer
- Nonlinearity
- Pooling Layer

The addition of a pooling layer after the convolutional layer is a common pattern used for ordering layers within a convolutional neural network that may be repeated one or more times in a given model.

After a convolution layer, it is common to add a pooling layer in between CNN layers. The main idea of pooling is down-sampling to decrease the complexity for further layers. This shortens the training time and controls over fitting. In image processing domain, we can think that it is similar to reducing resolution of the image. Pooling does not affect the number of filters. Pooling also allows for the usage of more convolutional layers by reducing memory consumption. There are different pooling methodologies such as max-pooling, average pooling and min pooling.

2.2.2.1 Max-pooling

Max-pooling is one of the most popular pooling methods. It divides the image to sub-region rectangles, and it only returns the maximum value of the inside of that sub-region. One of the most common sizes used in max-pooling is 2×2.



1	1	2	4
5	6	7	8
3	2	1	0
1	2	3	4

6	8
3	4

Figure 2.15 Max-pooling example

As we can see in Figure 2.15, when pooling is performed in the top-left 2×2 blocks (pink area), it moves 2 and focus on top-right part. This means that stride 2 is used in pooling. To avoid down-sampling, stride 1 can be used, which is not common. It should be considered that down-sampling does not preserve the position of the information. Therefore, it should be applied only when the presence of information is important (rather than spatial information). Moreover, pooling can be used with non-equal filters and strides to improve the efficiency. For example, a 3x3 max-pooling with stride 2 keeps some overlaps between the areas [11].

2.2.2.2 Minimum pooling

Unlike max-pooling, minimum pooling returns the minimum value of sub-region. This method can be used when the darker pixels more important for the project.

2.2.2.3 Average pooling

Average pooling returns takes the average values of sub-region. This method can be used for blurring of an image. In Figure 2.16, we can see which pixel values take for every pooling method. Max-pooling takes max value of sub-region, min-pooling takes min pooling of sub-region and average pooling takes average of sub-region values. Striding methodology is same for all pooling methods.

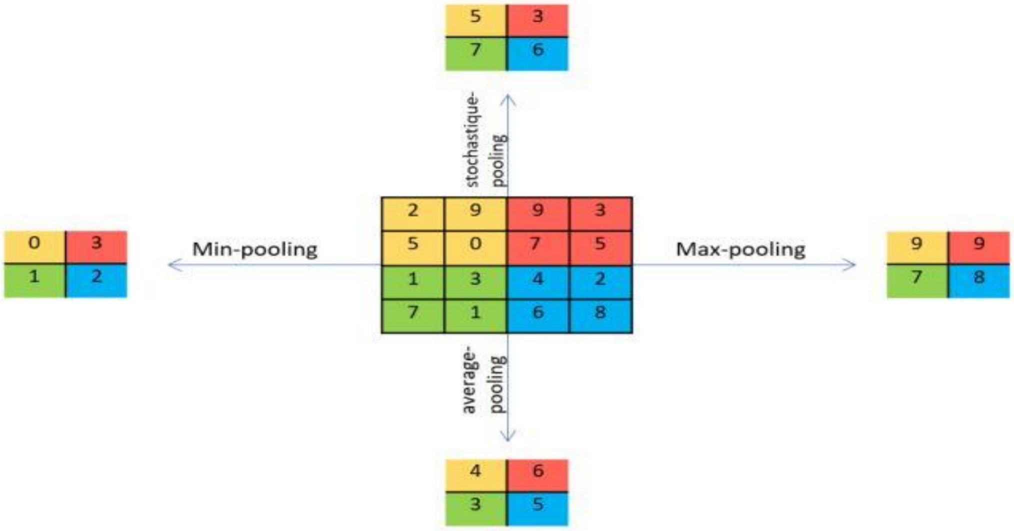
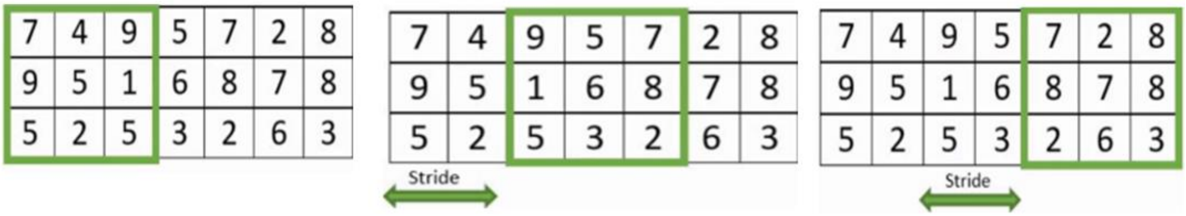


Figure 2.16 An example of each pooling method [12]

2.2.3 Padding and Stride

In convolutional networks, we have a choice to either decrease the data size as we go from one network to another, or keep it to the same size. Both padding and stride affects the data size. The requirement to keep the data size depends on the type of task, and it is part of architecture.

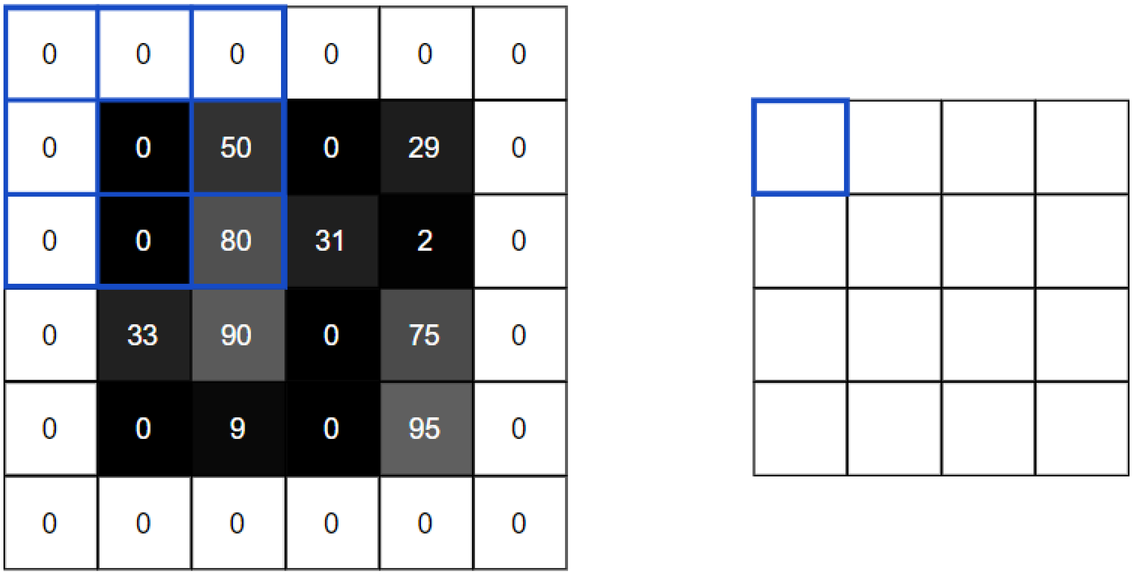
Stride: Pixel number jump at each displacement of the receptive field which evolve within the image.



7	4	9	5	7	2	8
9	5	1	6	8	7	8
5	2	5	3	2	6	3

Figure 2.17 Stride [12]

Padding: Artificial extra pixels added to the image to avoid to lost information on image edge. There are two types of padding as ‘valid’ and ‘same’. The default value is ‘valid’ which means that the filter is applied only to valid ways to the input. The padding value of ‘same’ calculates and adds the padding required to the input image (feature map) to ensure that the output has the same shape as the input. We would prefer to have the output image be the same size as input image. We add zeros around the image so we can overlay the filter in more places. In Figure 2.18, a 4x4 input convolved with a 3x3 filter to produce a 4x4 output using same padding.



0	0	0	0	0	0
0	0	50	0	29	0
0	0	80	31	2	0
0	33	90	0	75	0
0	0	9	0	95	0
0	0	0	0	0	0

Figure 2.18 Same padding

2.2.4 Non Linearity (ReLU)

ReLU stands for Rectified Linear Unit for a non-linear operation. The output is

$$f(x) = max(0,x) \tag{2.1}$$

The reason of why ReLU is important: ReLU's purpose is to introduce non-linearity in our ConvNet. Since, the real world data would want our ConvNet to learn would be non-negative linear values. When ReLU is applied to Feature Map (middle image), we have the right image as in the Figure 2.19.

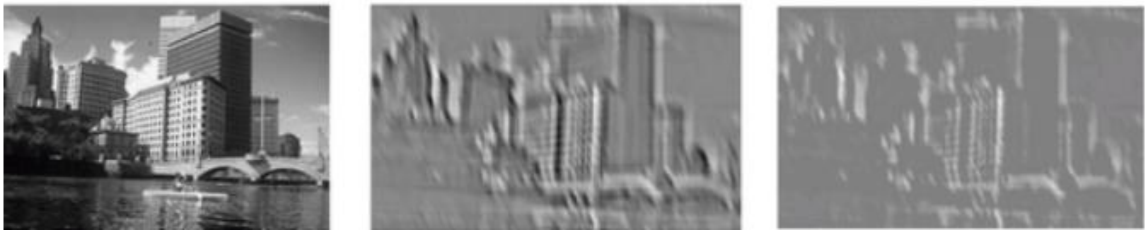
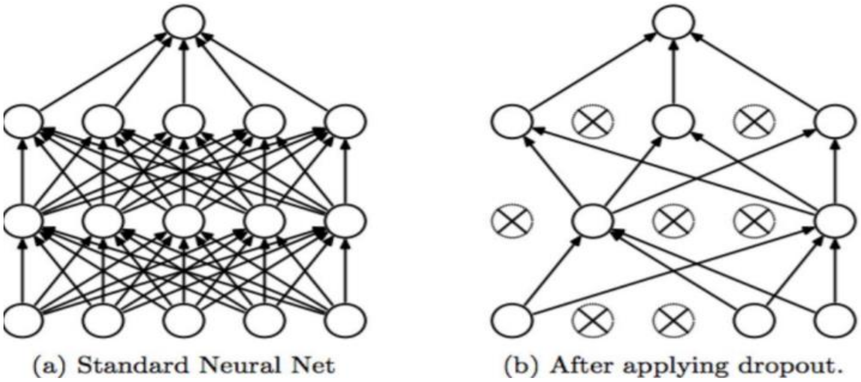


Figure 2.19 ReLU operation

2.2.5 Dropout

At each training stage some nodes are "dropped out" randomly and temporarily. This strategy has the effect of reduce over fitting. The percentage of nodes to dropout is fixed on Keras.



(a) Standard Neural Net

(b) After applying dropout.

Figure 2.20 Standard NN vs. after applying dropout

2.2.6 Flattening Layer

The task of this layer is simply to prepare the data for Fully Connected Layer which is the most important and last layer. In general, neural networks receive input data from a one-dimensional array. The data in this neural network is the one-dimensional array of matrices from the Convolutional and Pooling layers.

2.2.7 Fully Connected Layer

The layer we call as FC layer, we flattened our matrix into vector and feed it into a fully connected layer like neural network.

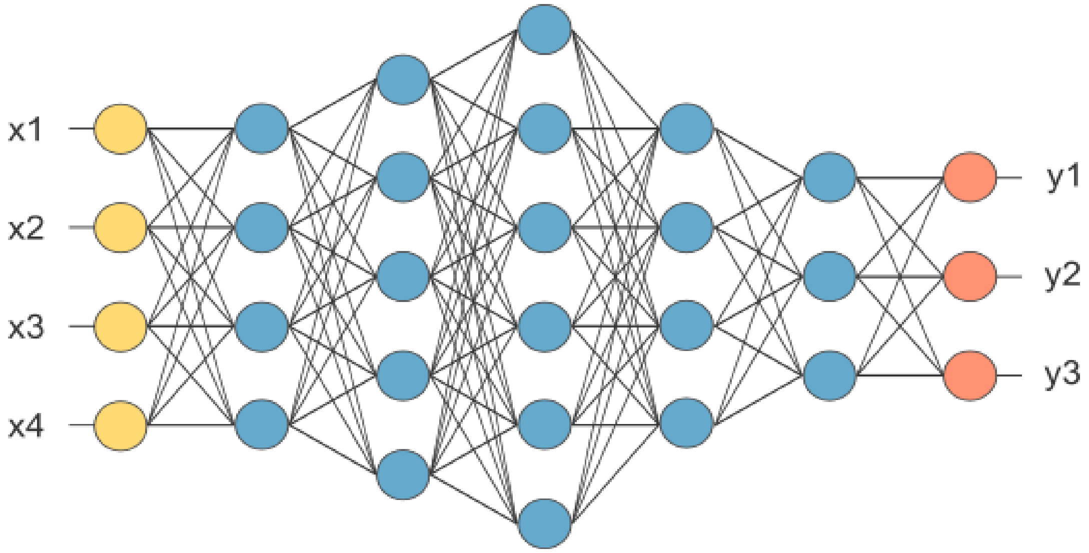


Figure 2.21 After pooling layer, flattened as FC layer

In the above diagram, feature map matrix will be converted as vector (x1, x2, x3, …). With the fully connected layers, we combined these features together to create a model. Finally, we have an activation function such as softmax or sigmoid to classify the outputs as cat, dog, car, truck etc.,

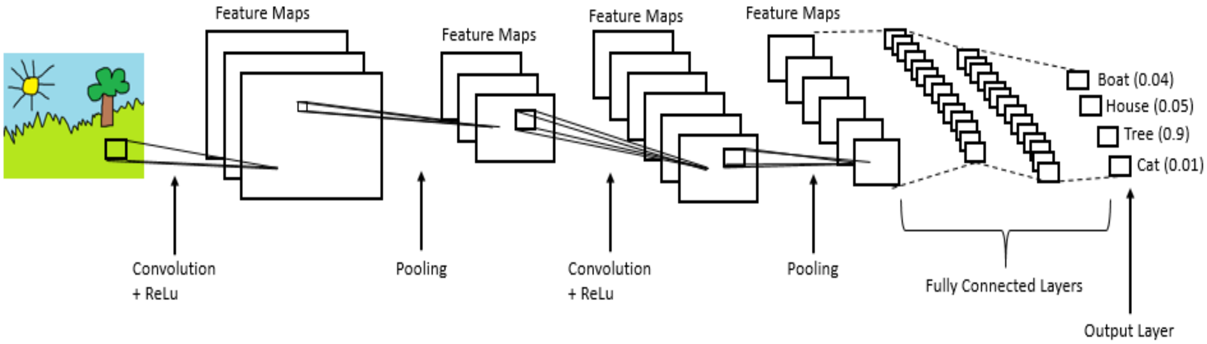


Figure 2.22 Complete CNN architecture

2.2.8 Summary

As a summary, in order to obtain a system with a CNN architecture, the following steps are as follows:

1-) Provide input image into convolution layer.

2-) Choose parameters, apply filters with strides, padding.

3-) Perform convolution on the image and apply ReLU activation to the matrix.

4-) Perform pooling to reduce dimensionality size

5-) Add convolutional layers until satisfied

6-) Flatten the output and feed into a fully connected layer (FC Layer)

7-) Output the class using an activation function (Logistic regression with cost functions) and classifies images.

2.3 3D-CNN

The 3D convolution is achieved by convolving a 3D kernel to the cube formed by stacking multiple contiguous frames together. Traditionally, CNN was developed for RGB images (3 channels). The goal of 3D CNN is to take as input a video or slices (set of images) to extract features from it. When CNN extract the graphical characteristics of a single image and put them in a vector (a low-level representation), 3D CNNs extract the graphical characteristics of a **set** of images. From a set of images, 3D CNNs find a low-level representation of a set of images, and this representation is useful to find the right label of the video or slices. In order to extract such features, 3D convolution uses 3D convolution operations.

In this work, 3D convolutional neural network was built which takes as input a CT scan.

2.4 Difference between 2D-CNN and 3D-CNN?

In 2D convolution, we try to capture the spatial structure across the 2D dimensions at each depth level. Means for an RGB image, we convolute in 2D in each R, G and B channel to capture the spatial features. Since there is no feature structure across the depth, then there is no need to convolute in the 3rd dimension. If there will be, then we will be using 3D convolution. Because 3D convolution provides three dimensional filter.

The filter depth is same as the input layer depth. The 3D filter moves only in 2-direction (height & width of the image). The output of such operation is a 2D image (with 1 channel only).

Naturally, there are 3D convolutions. They are the generalization of the 2D convolution. Here in 3D convolution, the filter depth is smaller than the input layer depth (kernel size < channel size). As a result, the 3D filter can move in all 3-direction (height, width, channel of the image). At each position, the element-wise multiplication and addition provide one number. Since the filter slides through a 3D space, the output numbers are arranged in a 3D space as well. The output is then a 3D data.

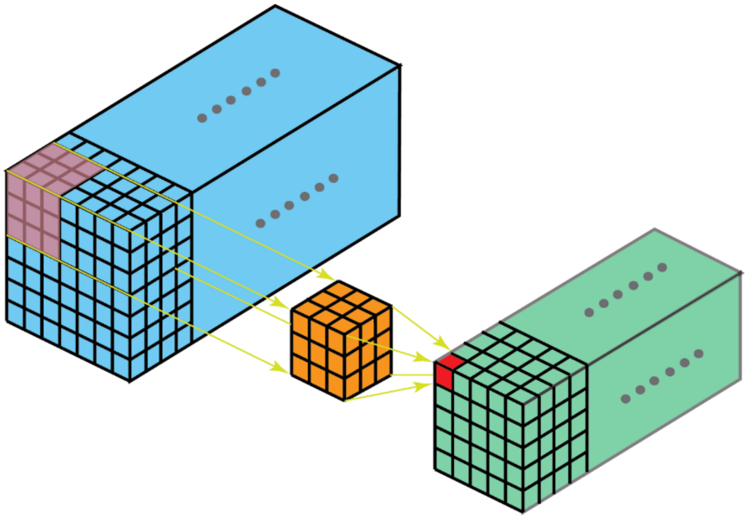


Figure 2.23 Filter of 3D CNN [14]

In 3D convolution, a 3D filter can move in all 3-direction (height, width, channel of the image). At each position, the element-wise multiplication and addition provide one number. Since the filter slides through a 3D space, the output numbers are arranged in a 3D space as well. The output is then a 3D data [14].

Similar as 2D convolutions which encode spatial relationships of objects in a 2D domain, 3D convolutions can describe the spatial relationships of objects in the 3D space. Such 3D relationship is important for some applications, such as in 3D segmentations / reconstructions of biomedical imagining, e.g. CT and MRI where objects such as blood vessels meander around in the 3D space.

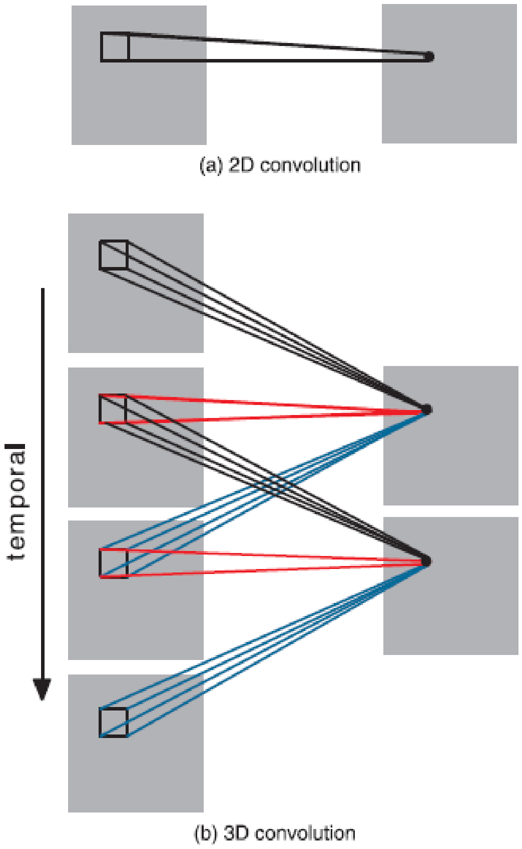


Figure 2.24 Comparison between (a) and (b) 3D convolutions

In (b) the size of the convolution kernel in the temporal dimension is 3 and the sets of connections are color-coded so that the shared weights are in the same color. In 3D convolution, the same 3D kernel is applied to overlapping 3D cubes in the input video to extract motion features [15].

CHAPTER 3

METHODOLOGY

3.1 TECHNOLOGY DECISIONS

In this section, technologies which are used will be explained for this project. Although there are many tools which exist, the following tools were used for the problem that needs to be solved.

3.1.1 Python

Python is a high level interpreted language used for general purpose programming. It is widely used for scientific computing, data mining and machine learning for software development. Python is the main language used for this project. 3.5 version was used. As far as we researched that version is the most proper version for image processing.

3.1.2 Anaconda

Anaconda is a popular data science platform where you can create data science projects and machine learning [18]. Libraries such as NumPy, Pandas, Matplotlib, Tensorflow and etc. come with Anaconda and IDE's such as Jupyter Notebook, Spyder and etc.

3.1.3 Numpy

Numpy is a library in Python that allows for efficient numerical computing in Python. This library is highly optimized to do mathematical tasks. In the project Numpy is heavily used in data pre-processing and preparation One of the main features about Numpy is highly efficient n-dimensional array. Compared to a list in Python a Numpy array can be n-dimensions and has more features associated with the n-dimensional array. Numpy can also perform more efficient mathematical operations compared to the math library in Python.

3.1.4 Pandas

Pandas is also a library in Python, like numpy is also used for data pre-processing and preparation. One of the main features about pandas is the DataFrame and Series data structure. These data structures are optimized and contain fancy indexing that allow a variety of features such as reshaping, slicing, merging, joining and etc to be available. Pandas and Numpy are extremely powerful when used together for manipulating data.

3.1.5 Matplotlib

Matplotlib is a Python plotting library that allows programmers to create a wide variety of graphs and visualizations with ease of use. The great feature about Matplotlib is that it integrates very well with Jupyter Notebook and creating visualizations is simplified. Matplotlib also works very well with pandas and numpy.

3.1.6 OpenCV

OpenCV (Open Source Computer Vision) is well established computer vision library which is written in C/C++ and has been abstracted to interface with C++, Python and Java. This is a powerful tool when working with images and has a myriad of tools regarding image data manipulation, feature extraction and etc.

3.1.7 Tensorflow

Tensorflow is an end-to-end open source platform for machine learning. It has a comprehensive, flexible ecosystem of tools, libraries and community resources that lets researchers push the state-of-the-art in ML and developers easily build and deploy ML powered applications [17].

3.1.7.1 What is tensor?

Convolutional neural networks process images as tensors, and tensors are matrices of numbers with additional dimensions [5]. A vector, in programming, is a type of array that is one dimensional. Vectors are a logical element in programming languages that are used for storing data. Vectors are similar to arrays but their actual implementation and operation differs [16]. So vector is a tensor of order 1. A matrix is a two-dimensional plane. Therefore, matrix is a tensor of order 2. A tensor is a generalization of vectors and matrices to potentially higher dimensions. Internally, Tensors are represented tensors as n-dimensional arrays [17]. When we write a part of code using array of integers, then we can explain it in this way. Suppose each integer represents some kind of differentiable field, then a tensor could be something like:

int a; // This is called a scalar

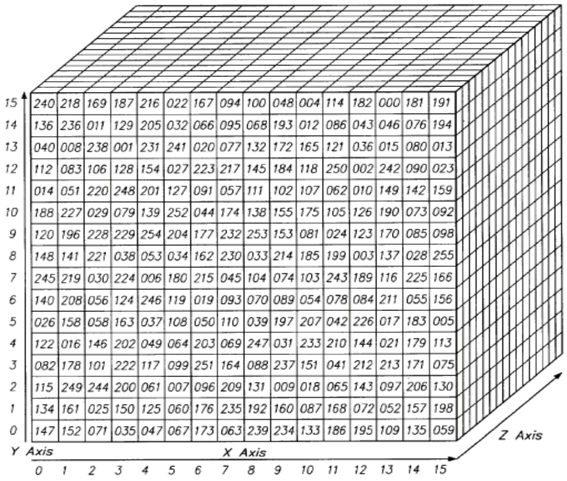
int a[]; // This is called a vector

int a[][]; // This is called a matrix

int a[][][]; // This is a 3D tensor

int a[][][]… n-times … []; // This is an n-D tensor

A tensor encompasses the dimensions beyond that 2-D plane. Here is an example 2x3x3 tensor represented as 3-D tensor. In code, tensor appears like the following: [[[1, 2, 3], [5, 7, 9], [2, 6, 7]], [[8, 7, 5], [5, 8 ,3], [1, 3, 9]]].



15	240	218	169	187	216	022	167	094	100	048	004	114	182	000	181	191
14	136	236	011	129	205	032	066	095	068	193	012	086	043	046	076	194
13	040	008	238	001	231	241	020	077	132	172	165	121	036	015	080	013
12	112	083	106	128	154	027	223	217	145	184	118	250	002	242	090	023
11	014	051	220	248	201	127	091	057	111	102	107	062	010	149	142	159
10	188	227	029	079	139	252	044	174	138	155	175	105	126	190	073	092
9	120	196	228	229	254	204	177	232	253	153	081	024	123	170	085	098
8	148	141	221	038	053	034	162	230	033	214	185	199	003	137	028	255
7	245	219	030	224	006	180	215	045	104	074	103	243	189	116	225	166
6	140	208	056	124	246	119	019	093	070	089	054	078	084	211	055	156
5	026	158	058	163	037	108	050	110	039	197	207	042	226	017	183	005
4	122	016	146	202	049	064	203	069	247	031	233	210	144	021	179	113
3	082	178	101	222	117	099	251	164	088	237	151	041	212	213	171	075
2	115	249	244	200	061	007	096	209	131	009	018	065	143	097	206	130
1	134	161	025	150	125	060	176	235	192	160	087	168	072	052	157	198
0	147	152	071	035	047	067	173	063	239	234	133	186	195	109	135	059
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

Figure 3.1 n-dimensional tensor

In this study, tensor’ s dimensionality is fifth-order. We have five dimensional tensor.

3.1.8 Keras

Keras is also a Deep Learning Framework that abstracts much of the code in the other Frameworks like Tensorflow and Theano. Compared to the other frameworks Keras is more minimalist [19].

3.1.8.1 Difference between Keras and Tensorflow

Keras is easy to use and more user-friendly compared to tensorflow. However, tensorflow is not that easy to use. Keras is a high level API built on tensorflow. If Keras is built on top of TF, what's the difference between the two then? And if Keras is more user-friendly, why should I ever use TF for building deep learning models? The following points will clarify which one you should choose.

- If you want to quickly build a neural network with minimum line of code, choose Keras**.** With Keras, you can build simple or complex neural networks within a few minutes.
- If you want to define something on your own such as cost function, a metric, layer, etc. Tensorflow will be better for you.
- Tensorflow offers more advanced operations as compared to Keras. Using tensorflow you can build more specialized deep learning model.
- Another extra power of TF. With tensorflow, you get a specialized debugger.
- Using tensorflow, you can control whatever you want in your network. Operations on weights or gradients can be done like a charm in tensorflow.

As a summary, if you want more control over the network and want to watch closely what happens with the network, tensorflow is better for you. I would like to more control over my deep learning model. Therefore, I have chosen tensorflow for my research.

3.1.9 CuDNN

Deep learning researchers and framework developers worldwide rely on cuDNN for high-performance GPU acceleration. It allows them to focus on training neural networks and developing software applications rather than spending time on low-level GPU performance tuning. cuDNN accelerates widely used deep learning frameworks, including Caffe,Caffe2, Chainer, Keras, MATLAB, MxNet, Tensorflow, and PyTorch [20].

3.1.10 h5py

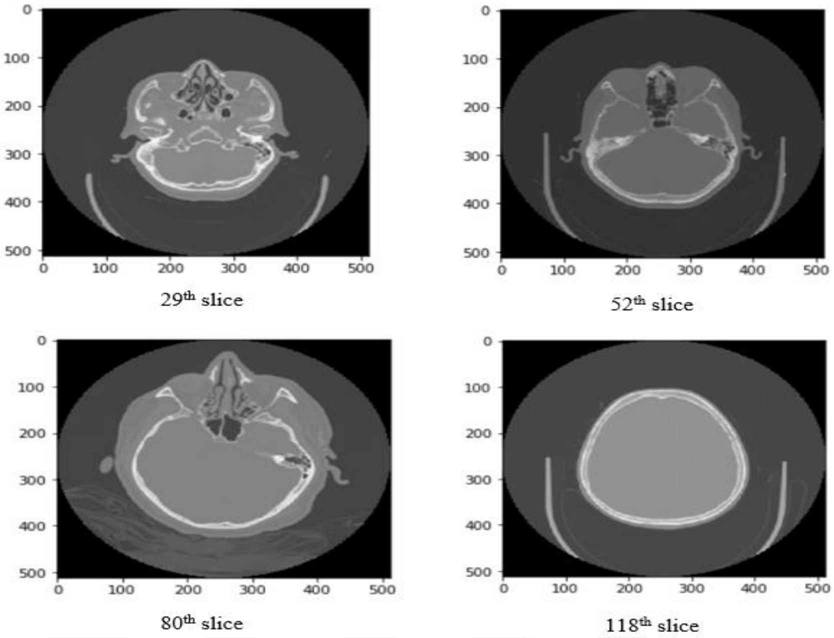
The h5py package is a Pythonic interface to the HDF5 binary data format. It lets you store huge amounts of numerical data, and easily manipulate that data from NumPy. For example, you can slice into multi-terabyte datasets stored on disk, as if they were real NumPy arrays. Thousands of datasets can be stored in a single file, categorized and tagged however you want [21].

3.1.11 Jupyter Notebook IDE

The Anaconda distribution comes with a variety of software that includes Jupyter Notebooks for scientific computing. Jupyter Notebooks [22] is an open source software IDE that allows developers to create and share documents that contain live code and more.

3.2 DATASET

Deep learning models require a lot of data. Prior to coding, I had to ensure that I have a great dataset to build a model. The dataset was provided by Radiology Department of Ankara Numune Training and Research Hospital. The first dataset has 70 patients as 35 patients who have infarction diagnosis and 35 patients who not have infarction diagnosis. This dataset has patients who have big volume of infarction. Each patient data has CT slices between 150 and 400, and dimension size of the images is 512x512. The images are DICOM format. We can see some slices of the first patient in the dataset in Figure 3.2. Moreover, we have second dataset which has low volume of infarcted area. It has 54 patients as 27 infarcted and 27 non-infarcted. The format is same as first dataset. Since the first dataset gave good accuracy, it was used for testing.



29th slice

52th slice

80th slice

118th slice

Figure 3.2 Sample CT slices from one patient

The patients who are infarcted labeled as 1, and the patients who are non-infarcted labeled as 0. This information is kept in a csv file. Therefore, we have a directory which has images all of the patients and a csv file which has labels of the patients. In Table 3.1, you can see the labels of first 8 patients. At this point, we have list of patients, and their associated labels stored in dataframe.

Table 3.1: Part of csv file

Patient	**infarction**
1patient	1
1non-patient	0
2patient	1
2non-patient	0
3patient	1
3non-patient	0
4patient	1
4non-patient	0

3.2.1 DICOM Image Format

Digital Imaging and Communications in Medicine (DICOM) is the standard for the communication and management of medical imaging information and related data [23]. DICOM is most commonly used format for storing and transmitting between several services which are useful in the medical imaging workflow. It has been widely used by hospitals. DICOM supports up to 65,536 (16 bits) shades of gray for monochrome image display, thus capturing the slightest nuances in medical imaging. In comparison, converting DICOM images into JPEGs or bitmaps (BMP), always limited to 256 shades of gray, often makes them impractical for diagnostic reading. DICOM takes advantage of the most current and advanced digital image representation techniques to provide the utmost diagnostic image quality [44]. DICOM format is different from jpeg or png images. The pixel values are not between is not between 0-255 as you can see in Table 3.2.

Table 3.2 Hounsfield Unit (HU) values of some substance in DICOM image format. [46]

Hounsfield Units	
Bone	1000
Liver	40-60
White Matter	46
Gray Matter	43
Blood	40
Muscle	10-40
Kidney	30
Cerebrospinal Fluid	15
Water	0
Fat	-50 - -100
Air	-1000

3.2.2 Hounsfield Scale

The Hounsfield scale, named after Sir Godfrey Hounsfield, is a quantitative scale for describing radio density. It is frequently used in CT scans, where its value is also termed CT number [24]. HU (Hounsfield Unit) scale is obtained the following formula, μ the corresponding HU value is therefore given by:

$$HU = 1000 \times \frac{\mu - \mu_{\text{water}}}{\mu_{\text{water}} - \mu_{\text{air}}} \tag{3.1}$$

where μ_{water} and μ_{air} are respectively the linear attenuation coefficients of water and air. Half of the scanner use 12 bits, so with a pixel intercept value of -1000 (or -1024) the range will be -1000 to +3096 (or -1024 to +3072).

Some scanners produce a 16 bits pixel value. Therefore, the H.U. value could range from -1000 to +64535. (or -32768 to +32767). DICOM header tells everything, but it could be wrong sometime. We can see some DICOM files with pixel value outside the range of max/min values stated in the header [46].

Our images were provided from a 16 bits scanner. We can see HU scale of first patient as an example.

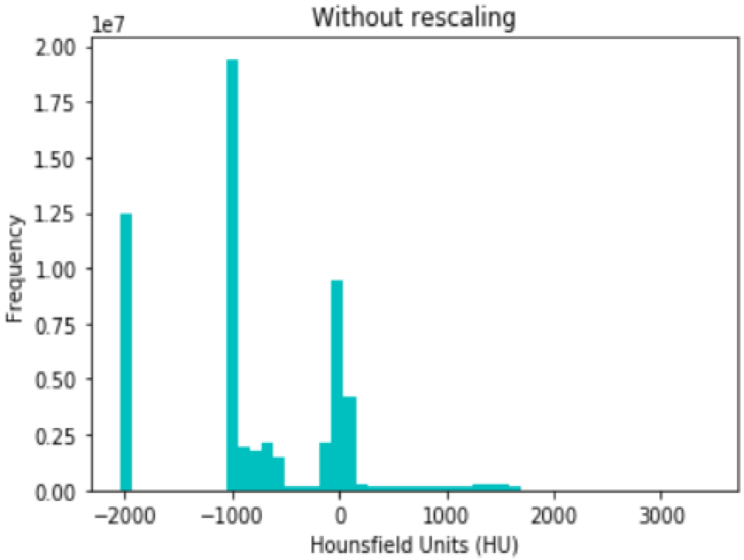


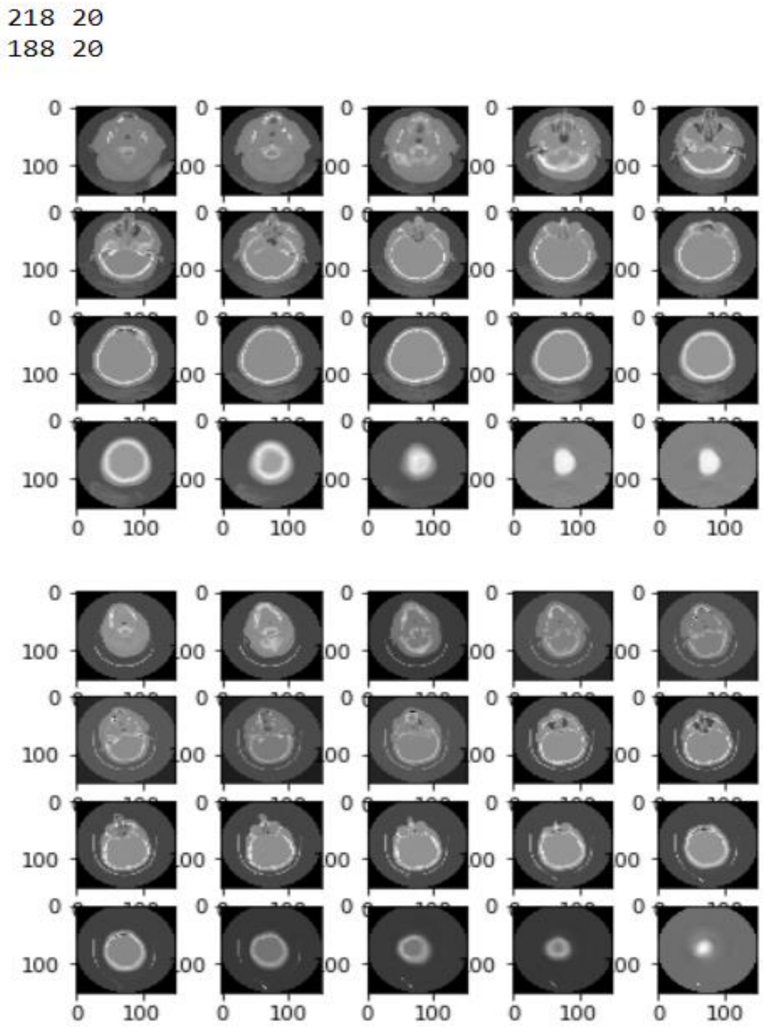
Figure 3.3 HU scale of first patient.

3.3 PREPROCESSING

3.3.1 Dimensionality Reduction

In machine learning classification problems, there are some preprocessing operation on the basis of which the final classification is done. These factors are called as features. High number of features is harder to visualize the training set and then work on it. Most of the features are usually redundant. At this point, dimensionality reduction comes into play. Dimensionality reduction is the process of reducing the number of random variables under consideration by obtaining set of variables. This process can be divided into feature engineering (feature extraction) and feature selection. Feature selection is the process of identifying and selecting relevant features for your sample. Feature engineering is manually generating new features from existing features, by applying some transformation or performing some operation on them. The most common dimensionality reduction methods are PCA (Principal Component Analysis), Factor Analysis and LDA (Linear Discriminant Analysis). Although it reduces computation time and helps remove redundant features, it may lead to some amount of data loss.

Medical images that are in our dataset have dimensionality of 512x512. This size is huge for processing. Therefore, we need to reduce dimensionality. This preprocessing method was adopted from "First pass through Data w/ 3D ConvNet" [25] provided by sentdex. Scans were resized from 512x512 to 50x50 and every patient has approximately 250 slices. In addition, we need to reduce number of slices. Thus, chunking data into the same number of slices is 20. The final shape of data for each patient is 50x50x20 to have short time for training. This dimensionality reduction was done with OpenCV. It is an open source computer vision and machine learning software library. OpenCV was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in the commercial products [45]. resize() method was used to make dimensionality reduction. The function makes bilinear interpolation by default. In Figure 3.4, we can see the reduced scans to 20 slices from 218 and 188 slices.



```
218 20
188 20
```

Figure 3.4 Slices after dimensionality reduction as 50x50x20

3.3.2 Normalization

In image processing, normalization is a process that changes the range of pixel intensity values. Applications include photographs with poor contrast due to glare, for example. Normalization is sometimes called contrast stretching or histogram stretching. In more general fields of data processing, such as digital signal processing, it is referred to as dynamic range expansion [13].

Our pixel values currently range from -2000 to around 2000. This range can be change from scan to scan. Anything above 400 is not interesting to us, as these are simply bone with different radio density. Minimum and maximum bound was determined as -2000 and 400, respectively as we see in Figure 3.3. The range of pixel values is 0-1.

3.4 Modeling

After performing dimensionality reduction and normalization, CNN based classification model was implemented. We build our model using Tensorflow. For more information about tensorflow, you can read the section 3.1.7. Our tensor data is five dimensional.

Firstly, the pixel values and labels of image dataset were saved to numpy array file as matrices to load the file to the CNN model. Then, the CNN architecture was implemented.

Table 3.3: 3D CNN architecture (Dropout with 0.2, learning rate = 0.0001)

Layers	**Parameters**	**Activation**

Conv1	Weights = [3, 3, 3, 1, 32]
	Bias = [32]
		ReLU

Min Pool	Kernel size = [1, 2, 2, 2, 1]
	Stride = [1, 2, 2, 2, 1]
	padding = SAME	

Conv2	Weight = [3, 3, 3, 32, 64]
	Bias = [64]
		ReLU

Min Pool	Kernel size = [1, 2, 2, 2, 1]
	Stride = [1, 2, 2, 2, 1]
	padding = SAME	

Fully Connected Layer	Weight = [54080, 1024]
	Bias = [1024]	

The neural network contains two convolutional layers and the number of total layers is seven. The order of layers is as in Figure 3.5 Convolutional layers are with stride 1 and ReLU as the activation function. Min pooling with 2x2x2 kernel and 2x2x2 strides were applied. The fully connected layer contains 54080x1024 inputs. We used Adam Optimizer and learning rate 0.0001.

As a summary, we applied dimensionality reduction and normalization in preprocessing. Then, 3D CNN architecture has 7 layers as input → conv1 → ReLU → Min pooling → conv2 → ReLU→ Min pooling → Fully Connected Layer → Output like in Figure 3.5. The visualization of our CD CNN architecture is included in Figure 3.6 and described in detail in Table 3.3.

Figure 3.6 shows us the whole 3D Convolutional Neural Network architecture which we have used. In preprocessing section, the images were resized to 50x50 and 20 slices. We have size of 50x50x20 matrices for each patient. Since same size is important for CNN model shape, we need to equalize for each patient. Then, pixel values were normalized between 0 and 1, because Hounsfield Units too big because of CT scan type. After performing preprocessing, Convolutional Neural Network was built. Two convolutional layers were used. When the number of convolutional layers is increased, we did not observe big differences.

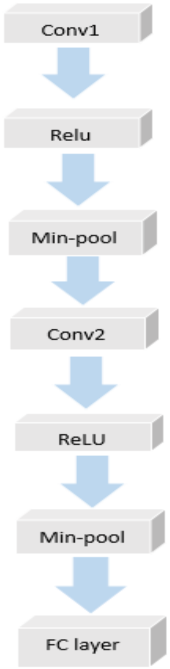


Figure 3.5 Layers of 3D CNN architecture

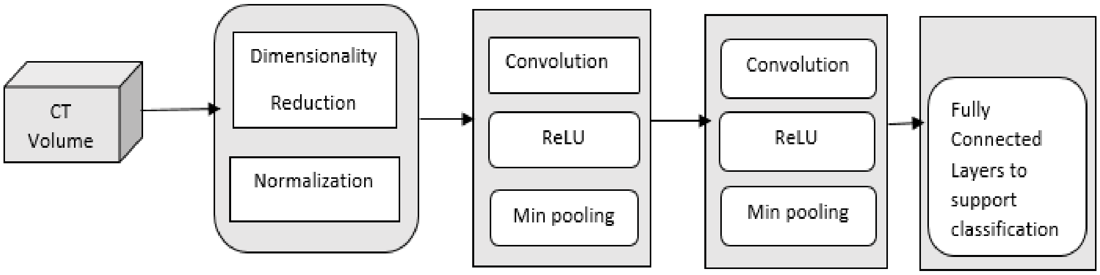


Figure 3.6 3D convolutional neural network architecture. (Dropout with 0.2, learning rate = 0.0001)

Convolutional neural network has convolutional layers which are followed by one or more fully connected layers to make classification and finally output layer. An example of this architecture is illustrated in Figure 3.7.

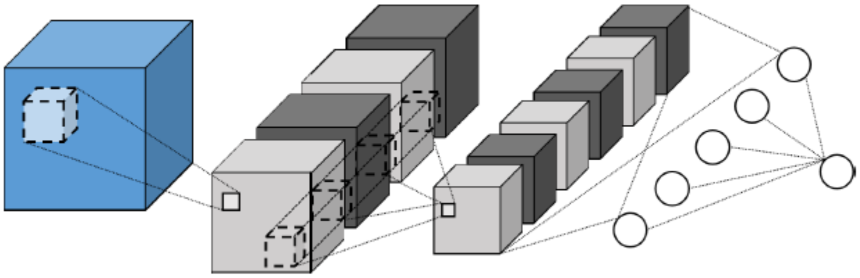


Figure 3.7 Architecture of a 3D convolutional neural network used here. Left one is 3D volume input. The followed ones two convolutional layers, fully connected layer to make classification and an output layer. In convolutional layers, each channel is represented by volume [26]

.

CHAPTER 4

RESULTS AND CONCLUSION

4.1 Results

In the dataset, we have high volume of infarction in average. The database consists of CT of 70 patients. The DICOM images have header about patient and scan parameters like bits, slice thickness. DICOM is de-facto standard in medical imaging.

The experiments were implemented in Python using a computer with CPU i7, 2.3 GHz, 8 GB RAM. Initially, we want to make skull stripping because we don't need those pixels. To do this method, we use a method of filter ellipse. However, slices are not same and some valid pixels were masked. In Figure 4.1, you can see the masked slice. Each slice 512 x 512 pixels and 4096 gray level pixel values in Hounsfield Unit (HU).

The results are compared with min pooling and max pooling. We took better results using minimum pooling. The dark values are infarcted areas. For this reason, these areas should be emphasized. However, minimum pooling emphasizes lighter pixels. For this reason, we have changed pooling methodology as minimum pooling. To do this, pooling methodology 1 subtracted from all pixels, after normalizing the pixel values between 0 and 1. Then, dimensionality reduction was made. The images were resized to 50x50 from 512x512 and slice number was fixed to 20. Finally, the numpy array file which has those pixel values was given to 3D CNN model. This model has two convolutional layer with ReLU layer. After each convolutional layers and ReLU layers, minimum pooling was performed. Totally 3D CNN model has seven layers. More number of layers were tried such as three, four convolutional layers. However, the accuracy value was not change too much. For this reason, convolutional layers were fixed as two.

In addition, we want to differentiate gray matter and white matter to have most valuable pixel values because actual pixels are in white matter. However, as far as I have researched in CT images it is pretty hard, since there is no distinct area. In MRI scans, the colors are more distinguishable. If it can be separated, there is no doubt that the accuracy of classification would increase.

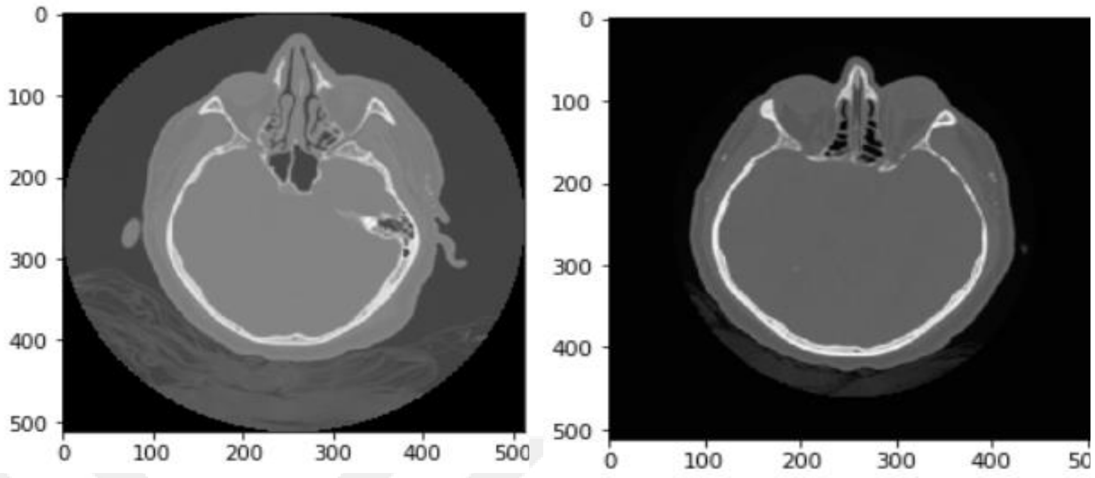


Figure 4.1 The right slice is masked of the left slice.

Experiments were done using 80% training set, 20% test set. So, images of 56 patients were used for training and 14 patients for testing to finally to evaluate the accuracy of the model on the data it has never seen. As we can see in Table 4.1, the maximum accuracy was found as 93% using minimum pooling. After 10 fold-testing, we reach 74% accuracy in the first dataset which has high volume of infarction. We have lower accuracy values when we use minimum pooling methodology. We reached 85% accuracy value as maximum accuracy value using min pooling.

Table 4.1 Results of dataset which has high volume of infarction

Pooling Method	**Best Accuracy**	**Average Accuracy**	**Worst Accuracy**
Min-pooling	0.93	0.74	0.57
Max-pooling	0.85	0.61	0.5

In the second dataset which has low volume of infarction, we have limitations about size of infarcted area. Classifying of these CT images could not be possible with eye. The only solution to classify is to see MRI scan images. The accuracy of this dataset after 10-fold testing is under 50% because of low volume of infarction. The other limitation is dataset size. We have just 54 patients as 27 infarcted patients. If the dataset size is increased, the accuracy could be increased.

4.2 Conclusion

Brain infarction occurs as a result of a blockage in the arteries which supply blood and oxygen to the brain. The restricted oxygen causes to stroke that can result in an infarction if the blood flow is not normalized in a short period of time. Approximately, 0.6% of people suffer from stroke every year. About one third is fatal. In addition, stroke is a third leading cause of death. For diagnosis, doctors want to see MRI (Magnetic Resonance Imaging) results to ensure if the patient has infarction or not. However, this process takes a long time while patients require immediate intervention. Losing time while waiting for an exact diagnosis might have fatal consequences. On the other hand, doctors can have CT (Computed Tomography) scan results in a short period of time but is not enough to tell the exact diagnosis for infarction due to uncertainty and the low quality of the imaging technique. This work aims to classify CT scans if the patient has infarction or not. This study uses 3D Convolutional Neural Network (3D-CNN) methodology. We believe that this method could be used as a decision support system to detect the patients with a higher risk of infarction, and prioritize utilization of MRI for them to make the final diagnosis quickly.

We propose a method to classify CT brain images of infarction. The method consists of two phases: preprocessing and classification. In the preprocessing phase, we reduced dimensionality of images and normalized pixel values between 0 and 1. Also, skull stripping process was made using ellipse filtering. However, it wasn’t applied because of important pixel loses and differences of CT slices. Then, shape of 50x50x20 images and range of 0-1 DICOM images was given to 3D CNN model to make classification. CNN model has two convolutional layers with ReLU activation layers. Firstly, we applied maximum pooling. This pooling method is important for light pixel values. For this reason, the accuracy value was obtained lower when compared with maximum pooling. As mentioned in Results part, the maximum accuracy value is 85%. Infarcted area is marked as darker pixels in the dataset. Since the darker pixels is important for infarcted areas. For this reason, we changed pooling methodology as minimum. After that, the accuracy value increased as 93% in maximum. After 10-fold testing, the accuracy value was obtained as 74%.

As we understand from other studies in literature, there is not so much work about CT brain images because segmentation phase and image processing is pretty hard. There is no distinct boundary between brain folds to make segmentation. Therefore, classical methodologies could be insufficient for CT scans. Moreover, CT brain images are not qualified to classify. For this reason, MRI is used frequently instead of CT scans.

In future works, the data could be increased because CNN methodology gives better accuracy with more data. Moreover, segmentation of gray matter and white matter which is pretty hard in CT scan would be so useful in classification.

As mentioned in Introduction, CT images can be obtained more easily, less costly and accessible. Although MRI provides better quality images, we want to take advantage of our CT images. For this reason, one of the biggest contribution of this thesis is to use CT brain images as decision support system. Thus, priority to MRI might be given to classified as infarcted patients. We believe that this method could be used as a decision support system to detect the patients with a higher risk of infarction, and prioritize utilization of MRI for them to make the final diagnosis quickly.

REFERENCES

[1] *Cerebral infarction – Wikipedia*. Retrieved March 26, 2019, from https://en.wikipedia.org/wiki/Cerebral_infarction

[2] Hess, C. P. *Exploring the Brain: Is CT or MRI Better for Brain Imaging?* Retrieved March 26, 2019. from https://radiology.ucsf.edu/blog/neuroradiology/exploring-the-brain-is-ct-or-mri-better-for-brain-imaging

[3] Convolutional Networks. Retrieved April 3, 2019, from http://cs231n.github.io/convolutional-networks/

[4] LeChun Y., Bengio Y.& Hinton G.," Deep learning", 27 May 2015

[5] *A Beginner's Guide to Convolutional Neural Networks (CNNs)*. Retrieved May 2, 2019, from https://skymind.ai/wiki/convolutional-network

[6] Fukushima, K. "Neocognitron: A hierarchical neural network capable of visual pattern recognition". Neural networks 1, 2 (1988), 119-130.

[7] Hubel, D. H., and Wiesel, T. N. "Receptive fields and functional architecture of monkey striate cortex". The Journal of Physiology 195, 1 (1968), 215-243.

[8] Stenroos O., Object detection from images using convolutional neural networks

[9] Marr, D., and Hildreth, E. Theory of edge detection. Proceedings of the Royal Society of London B: Biological Sciences 207, 1167 (1980), 187-217.

[10] Patel S., Pingel J., *Introduction to Deep Learning: What are Convolutional Neural Networks?* Retrieved May 2, 2019, from https://www.mathworks.com/videos/introduction-to-deep-learning-what-are-convolutional-neural-networks--1489512765771.html

[11] Albawi S., Mohammed T. A., Al-Zawi S., *Understanding of a Convolutional Neural Network,* 2017 International Conference on Engineering and Technology (ICET), 2017

[12] *Deep-learning: Rooftop type detection with Keras and TensorFlow*. Retrieved 3, May 2019 from http://fractalytics.io/rooftop-detection-with-keras-tensorflow

[13] Rafael C. González, Richard Eugene Woods (2007). "Digital Image Processing" Prentice Hall. p. 85.

[14] Bai K., A *Comprehensive Introduction to Different Types of Convolutions in Deep Learning*. Retrieved 3 May, 2019, from https://towardsdatascience.com/a-comprehensive-introduction-to-different-types-of-convolutions-in-deep-learning-669281e58215

[15] Ji S., Xu W., Yang M. & Yu K., *3D Convolutional Neural Networks for Human Action Recognition,* IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 35, no. 1, January 2013

[16] *Vector*. Retrieved 3 May, 2019 from https://www.techopedia.com/definition/22817/vector-programming

[17] *Tensors*. Retrieved 3 May, 2019 from https://www.tensorflow.org/guide/tensors

[18] *Welcome to Python*. Retrieved 3, May 2019, from https://www.python.org/.

[19] Chollet F., *Keras as a simplified interface to Tensorflow: tutorial*. Retrieved 3 May 2019, from https://blog.keras.io/ keras-as-a-simplified-interface-to-tensorflow-tutorial.html.

[20] *NVIDIA cuDNN | NVIDIA Developer*. Retrieved 5, May 2019, from https://developer.nvidia.com/cudnn

[21] *HDF5 for Python*. Retrieved 5, May 2019, from https://www.h5py.org/

[22] *Project Jupyter*. Retrieved 5, May 2019, from http://www.jupyter.org.

[23] "*Scope and Field of Application"*. Retrieved 5, May 2019, from dicom.nema.org.

[24] *Hounsfield scale*. Retrieved 20, April 2019, from https://en.wikipedia.org/wiki/Hounsfield_scale

[25] sentdex, *First pass through Data w/ 3D ConvNet.* Retrieved 15, October 2019 from https://www.kaggle.com/sentdex/first-pass-through-data-w-3d-convnet/notebook

[26] W.Alakwaa, M. Nassef, A. Badr, Lung Cancer Detection and Classification with 3D-CNN, (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 8, No. 8, 2017.

[27] Hosseini-Asl E., Gimel' farb G., & El-Baz A., "Alzheimer's disease Diagnostics by a Deeply Supervised Adaptable 3D Convloutional Network", 2016.

[28] Tong P.D., Hieu D.N., Nguyen H., Nguyen T. T., "Brain Hemorrhage Diagnosis by Using Deep Learning", January 2017S

[29] Kalpande G., *Biological Inspiration of Convolutional Neural Network (CNN).* Retrieved May 26, 2019 from https://medium.com/@gopalkalpande/biological-inspiration-of-convolutional-neural-network-cnn-9419668898ac

[30] Alakwaa W., Nassef M., Badr A., "Lung Cancer Detection and Classification with 3D-CNN", (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 8, No. 8, 2017.

[31] Kuan K., Ravaut M., Manek G., Chen H. , Lin J., Nazir B., et al. "Deep Learning for Lung Cancer Detection: Tackling the Kaggle Data Science Bowl 2017 Challenge", 26 May 2017

[32] He J., Qiu Y., "Lung Nodule Classification using Convolutional Neural Networks", 2016

[33] Milletari F., Navab N., Ahmadi S.A., "V-Net: Fully Convolutional Neural Networks for Volumetric Medical Image Segmentation", 15 Jun 2016

[34] Chon A., Balachandar N., Lu P., “Deep Convolutional Neural Network”

[35] Mohsen H., El-Dahshan S., Horbaty S. M., Salem B. M., “Classification using deep learning neural networks for brain tumors”, Future Computing and Informatics Journal Volume 3, Issue 1, June 2018, Pages 68-71

[36] Işin A., Direkoğlu C., Şah M., “Review of MRI-based brain tumor image segmentation using deep learning methods”, Procedia Computer Science Volume 102, 2016, Pages 317-324

[37] Pereira S., Pinto A., Alves V., Silva C. A., “Brain Tumor Segmentation Using Convolutional Neural Networks in MRI Images”, IEEE Transactions on Medical Imaging, vol. 35, no 5, May 2016

[38] Havaei M., Davy A., Warde-Farley D., Biard A., Courville A., Bengio Y., et al. “Brain Tumor Segmentation with Deep Neural Networks”, Medical Image Analysis 35 (2017) 18–31

[39] Gao X. W., Hui R., Tian Z., “Classification of CT brain images based on deep learning networks”, computer methods and programs in biomedicine 138 (2017) 49–56

[40] Fu G.S, Levin-Schwartz Y., Lin Q. H., Zhang D., “Machine Learning for Medical Imaging”, Journal of Healthcare Engineering, Volume 2019, Article ID 9874591, 2 pages,https://doi.org/10.1155/2019/9874591

[41] Bentley P., Ganesalingam J., Jones A. L. C., Mahady K., Epton S., Rinne P., et al. “Prediction of stroke thrombolysis outcome using CT brain machine learning”, NeuroImage: Clinical 4 (2014) 635–640

[42] Zang W. L., Wang X. Z., “Future Extraction and Classification for Human Brain CT Images“, Proceeding of the Sixth International Conference on Machine Learning and Cybernetics, Hong Kong, 19-22 August 2007

[43] Gong T., Liu R., Tan C. L., Farzad N., Lee C. K., Pang B. C., Tian S., et al. "Classification of CT Brain Images of Head Trauma"

[44] (2008) What Is DICOM?. In: Digital Imaging and Communications in Medicine (DICOM). Springer, Berlin, Heidelberg, DOI, https://doi.org/10.1007/978-3-540-74571-6_1

[45] *About OpenCV*. Retrieved 21 April, 2019 from https://opencv.org/about/

[46] *Hounsfield Unit*. Retrieved April 22, 2019 from http://gdcm.sourceforge.net/wiki/index.php/Hounsfield_Unit

CURRICULUM VITAE

PERSONAL INFORMATION

Name Surname : Nisanur MÜHÜRDAROĞLU

E-mail : muhurdaroglu.nisanur@gmail.com

EDUCATION

High School : Muradiye Nene Hatun Anatolian High School, Ankara, Turkey

Bachelor : Ankara Yıldırım Beyazıt University (2011 - 2016)
Computer Engineering

WORK EXPERIENCE

Research Assistant: Ankara Yıldırım Beyazıt University (February 2019 -)

TOPICS OF INTEREST

- Machine Learning
- Image Processing
- Computational Linguistics
- Data Mining