Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук
(повна назва)

Кафедра Штучного інтелекту
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти другий (магістерський)

Сегментація зображень пневмонії та пухлини за допомогою згорткових нейронних мереж з набором даних MNIST
(тема)

Виконав:
студент 2 курсу, групи СШІм-20-3
Арутюнов Е.Р.
(прізвище, ініціали)

Спеціальність 122 Комп'ютерні науки
(код і повна назва спеціальності)

Тип програми освітньо-наукова
(освітньо-професійна або освітньо-наукова)

Освітня програма Системи штучного інтелекту
(повна назва спеціалізації)

Керівник д.т.н. зав. каф. Філатов В.О.
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри _____ В.О. Філатов
(підпис) (прізвище, ініціали)

2022 р.

Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук
(повна назва)

Кафедра Штучного інтелекту
(повна назва)

Рівень вищої освіти другий (магістерський)

Спеціальність 122 Комп'ютерні науки
(код і повна назва)

Тип програми освітньо-наукова
(освітньо-професійна або освітньо-наукова)

Освітня програма Системи штучного інтелекту (СШІ)
(повна назва)

ЗАТВЕРДЖУЮ:
Зав. кафедри _____
(підпис)
«_____» _____ 20 ___ р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові Арутюнова Еміля Радимовича
(прізвище, ім'я, по батькові)

1. Тема роботи Сегментація зображень пневмонії та пухлини за допомогою згорткових нейронних мереж з набором даних MNIST

затверджена наказом університету від 24 травня 20 22 р. № 414 Ст

2. Термін подання студентом роботи до екзаменаційної комісії 11 05 20 22 р.

3. Вихідні дані до роботи Науково-технічні публікації, дані Інтернет-джерел та відомих наукових проектів щодо розробки та дослідження методів згорткових нейронних мереж, глибинного навчання для рішення нестандартних медичних проблем у 2D та 3D середовищах.

4. Перелік питань, що потрібно опрацювати в роботі

1 Machine learning concepts
2 Convolutional Neural Networks
3 MNIST Dataset
4 Medical imaging
5 Pneumonia classification
6 Hepatocellular carcinoma
7 Program development

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри) Figure 1.1 – Common experimental system; Figure 1.2 – Line Plot of the Increase Square Error with Predictions; Figure 1.3 – Line Plot of the Increase Absolute Error with Predictions; Figure 2.1 – Elementary constituents of CNN; Figure 2.2 – Receptive field of particular neuron in the next layer; Figure 2.3 – Pooling operation performed by choosing a 2 x 2 window; Figure 2.4 – Architecture of LeNet5 (each box represents a different feature map); Figure 2.5 – Convolution operation on 14 x 14 input image by sliding 5 x 5 kernel which yields 10 x 10 feature map; Figure 4.1 – Simple projection radiology mechanism; Figure 4.3 – CT principle mock-up; Figure 4.4 – Examples of human's brain MRIs; Figure 4.5 – From left to right: an MRI image of an ankle in sagittal orientation; an MRI off the chest in axial orientation; an examination of the brain that revealed a metastasis; Figure 5.1 – Example of an X-ray scan of a patient with pneumonia (marked with yellow arrows); Figure 6.1 – Example of a CT scan of a patient with diagnosed liver cancer (tumour is in red circle); Figure 7.1 – Dimensions of a tensor with their names; Figure 7.2 – Output of successfully loaded MNIST dataset

6. Консультанти розділів роботи (п.6 включається до завдання за наявності консультантів згідно з наказом, зазначеним у п.1)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Отримання завдання	28.03.2022	Виконано
2	Аналіз предметної області	29.03.2022	Виконано
3	Постановка задачі	31.03.2022	Виконано
4	Вибір та підготовка вхідних даних	02.04.2022	Виконано
5	Вибір моделей для програмної реалізації	05.04.2022	Виконано
6	Вибір технологій для програмної реалізації	10.04.2022	Виконано
7	Проектування програмного застосунка	15.04.2022	Виконано
8	Програмна реалізація моделі	20.04.2022	Виконано
9	Підготовка пояснювальної записки	25.04.2022	Виконано
10	Надання пояснювальної записки на перевірку	30.04.2022	Виконано
11	Захист роботи	11.05.2022	Виконано

Дата видачі завдання 28 березня 20 22 р.

Студент _____
(підпис)

Керівник роботи _____ _____
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ

Записка пояснювальна: 103 с., 1 табл., 32 рис., 2 дод., 17 формул, 16 джерел.

ЗГОРТКОВА НЕЙРОННАЯ МЕРЕЖА, ПУХЛИНА, ПНЕВМОНІЯ, PYTORCH, MNIST

Об'єкт дослідження – сегментація ракових пухлин та класифікація пневмонії.

Предмет дослідження – сегментація пневмонії та пухлин за допомогою конволюційних нейронних мереж з використанням набору даних MNIST.

Мета роботи – знайти більш ефективний та оптимальний спосіб виявлення ракових пухлин та пневмонії за допомогою машинного навчання.

Методи дослідження – аналіз технічної літератури в галузі машинного навчання на основі нейронних мереж, вивчення датасету MNIST, вивчення медичних джерел у галузі класифікації пневмонії та ракових пухлин.

Було проведено теоретичний аналіз вибірки навчальних даних, архітектури нейронної мережі, методів класифікації та інших параметрів. Практичні дослідження являли собою аналіз зображень та класифікацію діагнозів з описом точності результатів кожного дослідження. Практичні дослідження проводились на тестовому наборі даних MNIST, запропонованому Національним інститутом стандартів та технологій США. У дослідженні використовувалося 60 000 навчальних зображень та 10 000 тестових зображень. На основі отриманих результатів були розроблені моделі, які здатні з достатньою точністю визначати наявність раку або пневмонії у пацієнта за зображеннями.

ABSTRACT

Research practice report: 103 p., 1 tab., 32 fig., 17 formulas, 16 sources.

CONVOLUTIONAL NEURAL NETWORK, MNIST, TUMOUR, PNEUMONIA, PYTORCH

Object of the study – segmentation of cancerous tumours and classification of pneumonia

Subject of study – pneumonia and tumour segmentation using convolutional neural networks with MNIST dataset.

The aim of the work is to find a more efficient and optimal way to detect cancerous tumours and pneumonia using machine learning.

Research methods – analysis of technical literature in the field of machine learning based on neural networks, study of MNIST dataset, study of medical sources in the field of classification of pneumonia and cancerous tumours.

A theoretical analysis of training data sampling, neural network architecture, classification methods, and other parameters was performed. Practical research represented image analysis and classification of diagnoses with a description of the accuracy of the results of each study. Hands-on studies were conducted on the MNIST test data set proposed by the U.S. National Institute of Standards and Technology. The study used 60,000 training images and 10,000 test images. Based on the results, models have been developed that can detect the presence of cancer or pneumonia in a patient using images with sufficient accuracy.

CONTENT

LIST OF SYMBOLS, SYMBOLS, UNITS, ABBREVIATIONS AND TERMS

Adam – an optimization algorithm that can be used instead of the classical stochastic gradient descent procedure to update network weights iterative based in training data.

CNN – a class of artificial neural network, most commonly applied to analyze visual imagery.

Convolutional layers – the major building blocks used in convolutional neural networks.

CT – a medical imaging technique used in radiology (x-ray) to obtain detailed internal images of the body noninvasively for diagnostic purposes.

MNIST a large database of handwritten digits that is commonly used for training various image processing systems.

MRI – a medical imaging technique used in radiology to form pictures of the anatomy and the physiological processes of the body. MRI scanners use strong magnetic fields, magnetic field gradients, and radio waves to generate images of the organs in the body.

NIfTI – is an open file format commonly used to store brain imaging data obtained using Magnetic Resonance Imaging methods.

Tensors – mathematical objects that can be used to describe physical properties, just like scalars and vectors.

U-Net – a convolutional neural network that was developed for biomedical image segmentation.

X-ray – a quick, painless test that produces images of the structures inside your body – particularly your bones.

INTRODUCTION

The development of artificial intelligence as a scientific field became possible only after the creation of the computer. This happened in the 1940s. Research in the 1960s and 1970s produced the first expert system, which is known as DENDRAL. While it was developed for use in organic chemistry, it served as the basis for the subsequent MYCIN system, which is considered one of the most significant early applications of artificial intelligence in medicine. The 1980s and 1990s saw the proliferation of microcomputers and the creation of global networks. There was a recognition by researchers and developers that AI systems in health care needed to be developed. Researchers argued that programs should be designed for the absence of perfect information and should rely on the experience of physicians. New approaches involving fuzzy set theory, Bayesian networks, and artificial neural networks were created to reflect the evolving needs of healthcare for intelligent computing systems.

Since 2002, however, technology has made great strides, and both IT giants and entire nations have become involved in programs to bring artificial intelligence into medicine. Today, scientists hope that with the help of artificial intelligence, it will already be possible in the near future to come to ultra-precise medicine, in which it will be possible to prescribe individual treatment to each individual person, taking into account their unique genetic and other characteristics. The United States has already announced the launch of pilot projects for the development of precision medicine.

The medical-technological advances that have taken place during this half-century have brought healthcare to a new level. New AI-related applications and systems have a number of undeniable advantages:

– increased computing power leads to faster data collection and processing.

– increased volume and availability of health-related data that is derived from personal and medical devices of doctors and patients;

– growth of genomic sequencing databases;

– widespread adoption of electronic medical data recording systems.

AI-enabled tools identify meaningful relationships in raw data and can be applied to all areas of medicine, including drug development, treatment decision-making, patient care, and financial transactions and decisions.

Healthcare professionals can use AI to solve very complex and time-consuming problems. AI can prove to be a valuable resource for medical professionals, helping them realize their full expertise and potential throughout the healthcare ecosystem.

Before artificial intelligence began to be used to process medical information in the 2000s, predictive models in healthcare could only account for a limited number of variables in well-prepared medical data. Today's machine learning tools, which use artificial neural networks to learn extremely complex relationships or deep learning technologies, often outperform human capabilities in medical tasks. Artificial intelligence-equipped systems are capable of solving the complex problems that characterize modern clinical care.

What is the value of artificial intelligence in medicine?

– eliminating information noise;

– providing contextual correspondence;

– reducing errors associated with human fatigue;

– easier disease detection;

– improved physician-patient interaction;

– reduced cost of care.

AI technologies such as IBM's Watson are helping healthcare providers, executives and researchers leverage millions of medical reports, patient records, clinical studies and medical journals to extract valuable information.

In 2013, IBM joins with health insurer WellPoint Inc. and Memorial Sloan-Kettering Cancer Centre to announce two Watson computer-based applications – one to help diagnose and treat lung cancer and one to help manage health insurance decisions and claims. Both applications take advantage of the speed, language skills

and analytical capabilities of IBM's Watson technology, which famously defeated the best "Jeopardy!" television game show champions in 2011.

The goal of IBM Watson Health is to create intelligent health ecosystems. This means simpler processes, more efficient care, faster discoveries, and improved quality of care for people around the world.

This work will explore the key concepts of machine learning. A crucial topic of medical imaging will be intruded as a key component to the estimation of the diagnosis. Next, using Convolutional Neural Networks algorithms, the model will be trained on the MNIST dataset. Conclusions will be drawn on how effectively CNN can perform such tasks as pneumonia classification and cancer segmentation.

1 MACHINE LEARNING CONCEPTS

Before diving deep into the neural networks, deep learning and PyTorch, a few fundamental theory and concepts regarding machine learning have to be described. The difference between supervised learning tasks and unsupervised learning has to be defined.

Machine learning is a method of data analysis that automates analytical model building, and the keyword here is “automates” using algorithms that iteratively learn from the data. Machine learning allows computers to find hidden insights without being explicitly programmed. In classical programming, you would tell the computer what to do and what to look for in your data. With machine learning, you're just following a general set of rules, and then the program or algorithm itself will be able to find where these hidden insights are within the studied data.

So what is machine learning actually used for? It is utilized in a wide variety of tasks, like fraud detection, web search results, prediction of equipment failures, pattern image recognition, email spam filtering, recommendation engines and much more. Neural networks can be used to solve tasks that many other types of algorithms cannot. So for tasks like image classification, classical machine learning or statistical learning, algorithms are actually not able to perform those types of tasks very well. Things like language translation and image classification, it's only neural networks are actually able to solve those sort of tasks, and it will be proved on practice. There's actually specific ways you can structure a neural network to solve those kind of things. Deep learning simply refers to neural networks with more than one hidden layer. The topic of hidden layers will be described later.

There are two main types of machine learning tasks that are going to be focused on during the work:

- supervised learning;
- unsupervised learning.

1.1. Supervised and unsupervised learning

Supervised learning algorithms are trained using labelled examples, and that's a keyword label such as an input or the desired output is known. That means within your dataset, you're going to have some historical features with historical labels. So you already have that information, such as a segment of text, could have a category label. So you take a bunch of previous emails and someone has already gone by and classified them using the correct label. So they read the email and classified it as spam versus legitimate. So the way this works is for neural networks, the networks send or receive a set of input data along with the corresponding correct outputs, and then the algorithm or network will learn by comparing its actual output with correct outputs to find errors. Then it will modify the model accordingly, such as adjusting the weights and biased values in the network.

This experiment is a special case of cybernetic experiment with feedback. The setup of this experiment involves an experimental system, a learning method, and a method for testing the system or measuring characteristics.

The experimental system, in turn, consists of the system being tested (used), the space of stimuli received from the external environment, and the reinforcement control system (regulator of internal parameters). The reinforcement control system can be an automatic regulating device (e.g., thermostat) or a human operator (teacher) capable of responding to the reactions of the tested system and the stimuli of the external environment by applying specific reinforcement rules that change the memory state of the system.

Two variations are distinguished: (1) when the response of the tested system does not change the state of the external environment, and (2) when the response of the system changes the stimuli of the external environment. These schemes indicate the fundamental similarity of such a general kind of system to the biological nervous system.

Types of input data:

– the feature description is the most common case. Each object is described by a set of its own characteristics, called attributes. Traits can be numeric or non-numeric;

– a matrix of distances between objects. Each object is described by distances to all other objects in the training set. Few methods work with this type of input data, such as the k nearest neighbour method, the Parzen window method, and the potential function method;

– a time series or signal is a sequence of measurements in time. Each measurement can be represented by a number, a vector, and in general by a feature description of the object under study at a given point in time.

– image or video sequence;

– there are more complicated cases when input data are represented as graphs, texts, results of database queries, etc. As a rule, they are reduced to the first or second case by preliminary data processing and feature extraction. See figure 1.1 for the schema of a common experimental system.

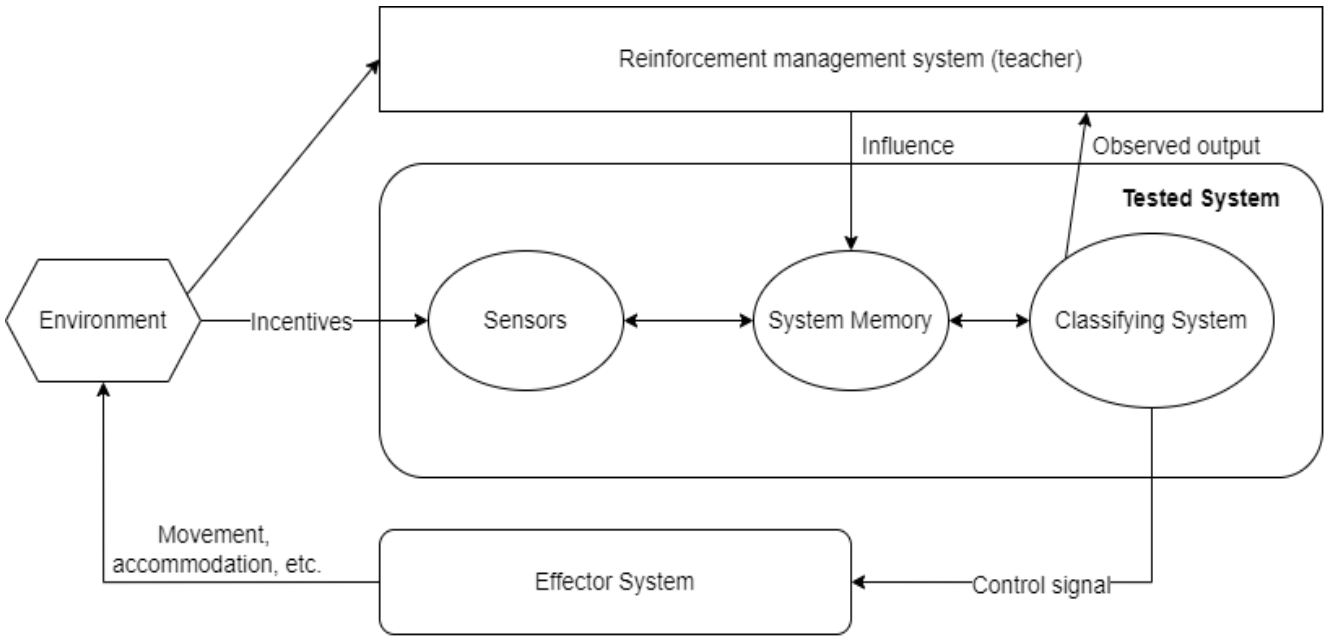

Figure 1.1 – Common experimental system

Perfectly marked and clean data is not easy to get. So sometimes the algorithm is challenged to find answers that are not known beforehand. This is where unsupervised learning is needed.

In unsupervised learning, the model has a data set and no explicit instructions on what to do with it. The neural network tries to find correlations in the data on its own, extracting useful features and analysing them.

Depending on the task, the model organizes the data in different ways.

Clustering. Even without the expertise of an ornithological expert, you can look at a collection of photos and categorize them by bird species based on feather colour, size, or beak shape. This is what clustering is all about, the most common task for unsupervised learning. The algorithm picks up similar data, finding common features, and groups them together.

Anomaly detection. Banks can detect fraudulent transactions by detecting unusual actions in customers' buying behaviour. For example, it is suspicious if the same credit card is used in California and Denmark on the same day. Similarly, unsupervised learning is used to find outliers in data.

Associations. Choose a diaper, applesauce, and baby mug from an online store and the site will recommend that you add a bib and a baby monitor to your order. This is an example of associations: some characteristics of an object correlate with other attributes. By looking at a couple of key attributes of an object, the model can predict others with which there is a correlation.

Autoencoders. Autoencoders take input data, encode it, and then try to reconstruct the initial data from the resulting code. There aren't many real-world situations where a simple autoencoder is used. But add layers and the possibilities expand: using noisy and raw versions of images for training, autoencoders can remove noise from video data, images, or medical scans to improve data quality.

In unsupervised learning, it is difficult to calculate the accuracy of an algorithm because there are no "right answers" or labels in the data. But labelled data

is often unreliable or too expensive to obtain. In such cases, giving the model freedom to look for dependencies can produce good results.

1.2. Overfitting

In supervised machine learning, there's an un-detouring issue. Model does not generalize well from observed data to unseen data, which is called overfitting. Because of existence of overfitting, the model performs perfectly on training set, while fitting poorly on testing set. This is due to that over-fitted model has difficulty coping with pieces of the information in the testing set, which may be different from those in the training set. On the other hand, over-fitted models tend to memorize all the data, including unavoidable noise on the training set, instead of learning the discipline hidden behind the data. The causes of this phenomenon might be complicated. Generally, we can categorize them into three kinds: 1) noise learning on the training set: when the training set is too small in size, or has less representative data or too many noises. This situation makes the noises have great chances to be learned, and later act as a basis of predictions. So, a well-functioning algorithm should be able to distinguish representative data from noises; 2) hypothesis complexity: the trade-off in complexity, a key concept in statistic and machining learning, is a compromise between Variance and Bias. It refers to a balance between accuracy and consistency. When the algorithms have too many hypothesis (too many inputs), the model becomes more accurate on average with lower consistency. This situation means that the models can be drastically different on different datasets; and 3) multiple comparisons procedures which are ubiquitous in induction algorithms, as well as in other Artificial Intelligence (AI) algorithms. During these processes, we always compare multiple items based on scores from an evaluation function and select the item with the maximum score. However, this process will probably choose some items which will not improve, or even reduce classification accuracy. In order to reduce the effect of overfitting, multiple solutions based on different strategies are

proposed to inhibit the different triggers. Nevertheless, most of them perform poorly when dealing with real-world issues, because of the great amount of hypothesis. However, none of the hypothesis sets can cover all the application fields.

1.3. Evaluating Performance – Classification and Regression Error Metrics

After a machine learning process is complete, we're going to be using performance metrics to evaluate how our model actually did. What classification metrics are we going to be using? The key classification metrics we should be understanding are accuracy, recall, precision and F1 score. First, the reasoning behind these metrics and how they will actually work in the real world must be understood. Any classification task the model can achieve two results – either it was correct in its prediction or the model was incorrect in its prediction, and all classification metrics stemmed from this idea.

Now, fortunately, incorrect versus correct also expands the situations where there are multiple classes, such as trying to predict categories of more than two. For example, while having categories A, B, C, D, you can either be correct in predicting the correct category or incorrect in predicting the right category. Let’s simplify this to a binary classification situation. So there's only two available classes and this idea is going to expand to multiple classes as well.

But for simplification, let's imagine just a binary classification situation. We're going to attempt to predict if an image is a dog or a cat and will actually perform this task later on. Now, since it's a supervised learning problem, what we're going to need to do first is fit or train a model on trained data. That means we're going to have images that someone's already gone ahead and labelled dog or cat, so we know the correct answer on these images. Then we're going to test that model on the testing data. So we're going to show new images that the model hasn't seen before. Get the model's prediction and then compare the results of the model prediction to the correct

answer that we already know. So once we have the model's predictions from X test data, we compare it to the true Y values, the correct labels.

Accuracy is one of the most common classification metrics. It's just really intuitive. It measures the number of correct predictions made by the model divided by the number of or the total number of predictions. It’s the number of correct predictions divided by the total number of predictions. Accuracy is really useful when the target classes are well balanced. It means the actual labels themselves are roughly equally represented in the dataset. What if there is an unbalanced class situation in this case, accuracy is actually not a good metric to use. That’s why we must be aware of the downside of accuracy that may occur with an unbalanced class situation.

Recall is the ability of a model to find all the relevant cases within a dataset and the precise definition of recall is the number of what's known as true positives, and we'll kind of hone in on that later when we see the confusion matrix. It's the number of true positives divided by the number of true positives, plus the number of false negatives. The precision is the ability of a classification model to identify only the relevant data points where precision is defined as the number of true positives divided by the number of true positives, plus the number of false positives. There is often a trade-off between recall and precision, while recall expresses the ability to find all the relevant instances in a data set, precision expresses the proportion of the data points the model says was relevant to actually were relevant. Hence, F1 score is essentially a combination of these two. In cases where we want to find the optimal blend of precision and recall, we can combine the two metrics using what is known as the F1 score. The F1 score is the harmonica mean of precision and recall taking both metrics into account in the following equation. So this isn't just taking the average of record precision, it's taking the harmonic mean of them.

$$F1\ Score = 2 * \frac{recall * precision}{recall + precision}, \tag{1.1}$$

How to evaluate performance for regression tests? It doesn't really make sense to calculate the accuracy or recall of a regression task for classifying things. Instead, we'll be predicting a continuous value. For example, while attempting to predict the price of a house given its features that would be a regression task or attempting to predict the country a house is in. Given its features, that would be a classification task. The focus right now on how to evaluate the regression task or a label is continuous and it's not separate categories. The most common evaluation metrics for regression, which are mean absolute error, mean squared error and then root mean squared error are going to be described in this paragraph.

Mean Squared Error, or MSE for short, is a popular error metric for regression problems. It is also an important loss function for algorithms fit or optimized using the least squares framing of a regression problem. Here "least squares" refers to minimizing the mean squared error between predictions and expected values. The MSE is calculated as the mean or average of the squared differences between predicted and expected target values in a dataset.

The squaring also has the effect of inflating or magnifying large errors. That is, the larger the difference between the predicted and expected values, the larger the resulting squared positive error. This has the effect of "punishing" models more for larger errors when MSE is used as a loss function. It also has the effect of "punishing" models by inflating the average error score when used as a metric. We can create a plot to get a feeling for how the change in prediction error impacts the squared error. The graph is displayed below in figure 1.2.

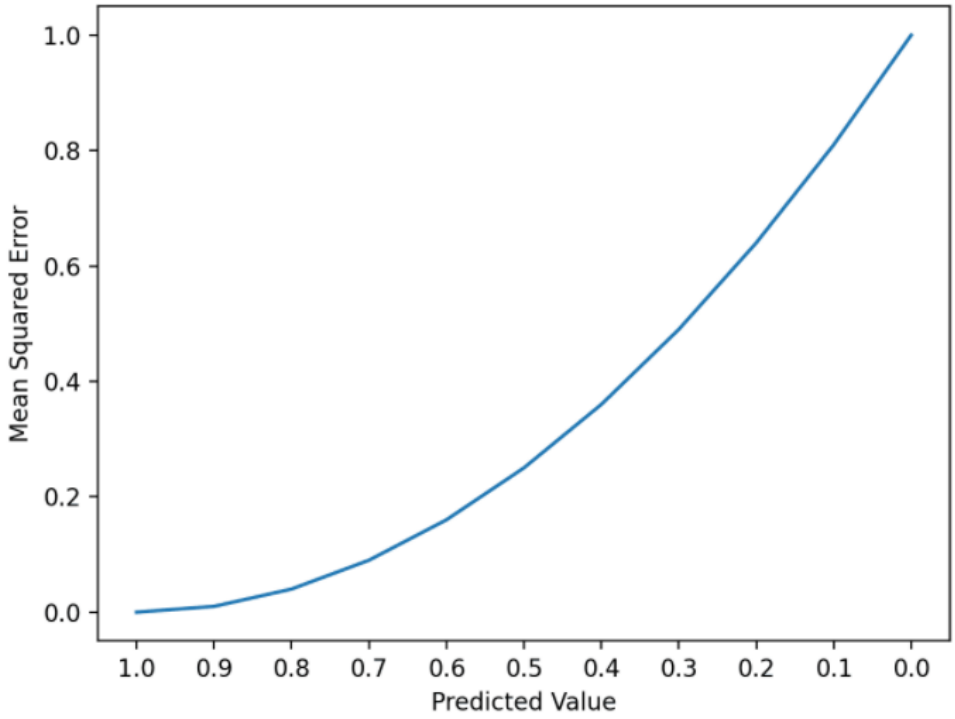

Figure 1.2 – Line Plot of the Increase Square Error with Predictions

The individual error terms are averaged so that we can report the performance of a model with regard to how much error the model makes generally when making predictions, rather than specifically for a given example. The units of the MSE are squared units.

It is a good idea to first establish a baseline MSE for your dataset using a naive predictive model, such as predicting the mean target value from the training dataset. A model that achieves an MSE better than the MSE for the naive model has skill.

As RMSE is clear by the name itself, that it is a simple square root of mean squared error. Note that the RMSE cannot be calculated as the average of the square root of the mean squared error values. This is a common error made by beginners and is an example of Jensen's inequality.

Mean Absolute Error, or MAE, is a popular metric because, like RMSE, the units of the error score match the units of the target value that is being predicted.

Unlike the RMSE, the changes in MAE are linear and therefore intuitive.

That is, MSE and RMSE punish larger errors more than smaller errors, inflating or magnifying the mean error score. This is due to the square of the error value. The MAE does not give more or less weight to different types of errors and instead the scores increase linearly with increases in error. The graph is displayed below in figure 1.3.

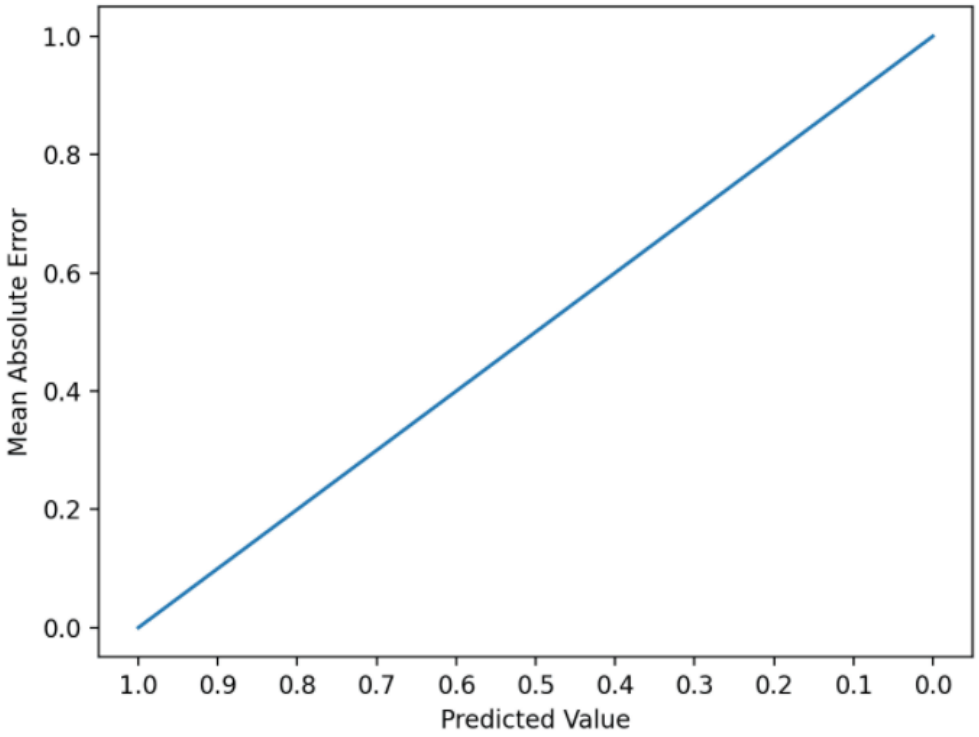

Figure 1.3 – Line Plot of the Increase Absolute Error with Predictions

As its name suggests, the MAE score is calculated as the average of the absolute error values. Absolute or abs() is a mathematical function that simply makes a number positive. Therefore, the difference between an expected and predicted value may be positive or negative and is forced to be positive when calculating the MAE.

2 CONVOLUTIONAL NEURAL NETWORKS

With the rapidly growing demand for learnable machines for solving many complex problems, deep learning has evolved itself as an area of interest to the researchers in the past few years. As researchers tend to mimic human behaviour, a major question arises that how do the humans acquire knowledge? The answer to this question is an essential ability of humans i.e. learning, which needs to be incorporated in machines, hence the term machine learning was coined. Machine learning promises to reduce the efforts by making the machines to learn themselves through past experiences [2] using three approaches of learning namely, learning under supervision, without supervision and semi-supervised learning. The conventional machine learning techniques need feature extraction as the prerequisite, and this requires a domain expert [16]. Furthermore, selection of appropriate features for a given problem is a challenging task. Deep learning techniques overcome the problem of feature selection by not requiring pre-selected features but extracting the significant features from raw input automatically for a problem in hand. Deep learning model consists of a collection of processing layers that can learn various features of data through multiple levels of abstraction [15]. Multiple levels allow the network to learn distinct features. Deep learning has emerged as an approach for achieving promising results in various applications like image recognition, speech recognition, topic classification, sentiment analysis, language translation, natural language understanding, signal processing, face recognition, prediction of bioactivity of small molecules etc. There are different deep learning architectures such as deep belief networks, recurrent neural networks, convolution neural networks etc.

Convolution Neural Network (CNN), often called ConvNet, has deep feed-forward architecture and has astonishing ability to generalize in a better way as compared to networks with fully connected layers [5]. CNN is usually described as the concept of hierarchical feature detectors in biologically inspired manner. It can

learn highly abstract features and can identify objects efficiently. The considerable reasons why CNN is considered above other classical models are as follows. First, the key interest for applying CNN lies in the idea of using concept of weight sharing, due to which the number of parameters that needs training is substantially reduced, resulting in improved generalization. Due to lesser parameters, CNN can be trained smoothly and does not suffer overfitting. Secondly, the classification stage is incorporated with feature extraction stage, both uses learning process. Thirdly, it is much difficult to implement large networks using general models of artificial neural network (ANN) than implementing in CNN. They are widely being used in various domains due to their remarkable performance such as image classification, object detection, face detection, speech recognition, vehicle recognition, diabetic retinopathy, facial expression recognition and many more. The motivation of this study is to establish a theoretical framework while adding to the knowledge and understanding about CNN.

In a usual perceptron, which is a fully connected neural network, each neuron is connected with all neurons of the previous layer, and each connection has its own personal weight coefficient. The convolutional neural network uses in the convolution operation only a limited matrix of small weights, which is "moved" through the whole processed layer (in the beginning - directly along the input image), forming after each shift an activation signal for a neuron of the next layer with the same position. That is, the same matrix of weights is used for different neurons of the output layer, which is also called the convolution kernel. It is interpreted as a graphical coding of some feature, for example, the presence of a slanted line at a certain angle. The next layer resulting from the convolution operation with such a matrix of weights shows the presence of the given feature in the processed layer and its coordinates, forming the so-called feature map. In a convolutional neural network, the set of weights is of course not one, but the whole gamma, which encodes the elements of the image (such as lines and arcs at different angles). In this case, such convolutional kernels are not laid down by the researcher beforehand, but

are formed independently by training the network by the classical method of backward error propagation. Passing by each set of weights forms its own instance of a feature map, making the neural network multi-channel (many independent feature maps on a single layer). It should also be noted that when a matrix of weights tries to search a layer, it is usually moved not by a full step (the size of this matrix), but by a small distance. So, for example, if a weight matrix is 5×5, it is shifted by one or two neurons (pixels) instead of five, so as not to "overstep" the desired feature.

Subsampling operation (also translated as "subsampling operation" or pooling operation), reduces the dimensionality of formed feature maps. The given architecture of the network assumes that the information about the fact of presence of the sought attribute is more important than the exact knowledge of its coordinates, so from several neighbouring neurons of the attribute map the maximal one is selected and taken as one neuron of the compacted attribute map of smaller size. Due to this operation, in addition to speeding up further calculations, the network becomes more invariant to the scale of the input image.

Consider the typical structure of convolutional neural network in more detail. The network consists of a large number of layers. After the initial layer (input image) the signal passes through a series of convolutional layers, in which the actual convolution and subsampling (pooling) alternate. The alternation of layers allows to make "feature maps" from feature maps, at each successive layer the map decreases in size, but the number of channels increases. In practice, this means the ability to recognize complex feature hierarchies. Usually, after passing through several layers, the feature map degenerates into a vector or even a scalar, but such feature maps become hundreds. At the output of the convolutional layers of the network, several layers of a fully connected neural network (perceptron) are additionally installed, which is fed with terminal feature maps.

2.1. General model

The typical model of ANN has single input and output layer along with multiple hidden layers [13]. A particular neuron takes input vector X and produces output Y by performing some function F on it, represented by general equation 2.1 shown below.

$$F\,(X, W) = Y\,, \tag{2.1}$$

where, W denotes the weight vector which represents the strength of interconnection between neurons of two adjacent layers.

The obtained weight vector can be now used to perform image classification. A significant amount of literature exists related to pixel based classification of images. However, contextual information like shape of the image produces better results or outperforms. CNN is a model that is gaining attention because of its classification capability based on contextual information. The general model of CNN has been described below in figure 1. A general model of CNN consists of four components namely (a) convolution layer, (b) pooling layer, (c) activation function, and (d) fully connected layer. Functionality of each component has been illustrated below in figure 2.1.

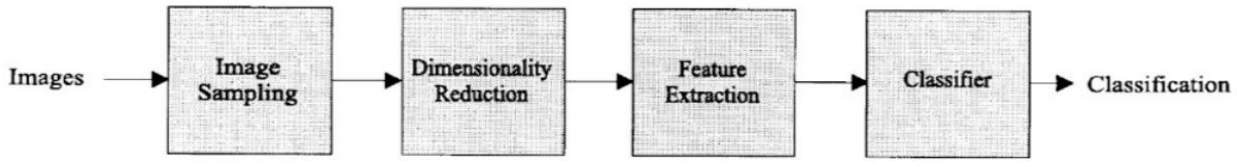

Figure 2.1 – Elementary constituents of CNN

Talking about the convolutional layer there are some certain things to mention. An image to be classified is provided to the input layer and output is the predicted class label computed using extracted features from image. An individual neuron in the next layer is connected to some neurons in the previous layer, this local correlation is termed as receptive field. The local features from the input image are extracted using receptive field. The receptive field of a neuron associated to particular region in previous layer forms a weight vector, which remains equal at all points on the plane, where plane refers to the neurons in the next layer. As the neurons in plane share same weights, thus the similar features occurring at different locations in the input data can be detected [11]. This has been depicted in figure 2.2 shown below.

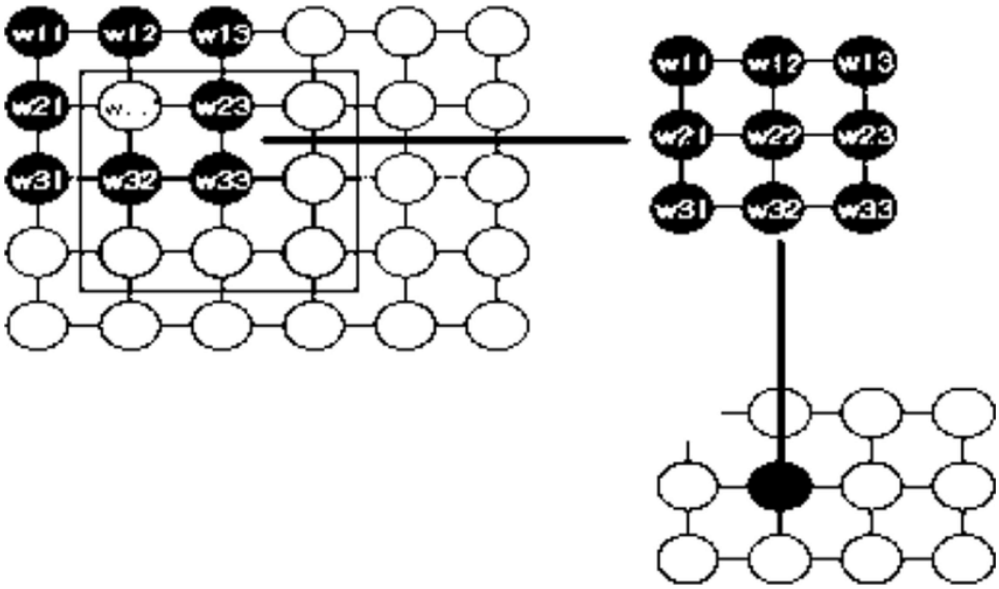

Figure 2.2 – Receptive field of particular neuron in the next layer

The weight vector, also known as filter or kernel, slides over the input vector to generate the feature map. This method of sliding the filter horizontally as well as vertically is called convolution operation. This operation extracts N number of features from the input image in a single layer representing distinct features, leading to N filters and N feature maps. Due to the phenomenon of local receptive field,

number of trainable parameters is significantly reduced. The output a_{ij} in the next layer for location (i,j), is computed after applying convolution operation using formula 2.2 as shown below:

$$a_{ij} = \sigma\big((W * X)_{ij} + b\big), \tag{2.2}$$

where, X is the input provided to the layer, W is filter or kernel which slides over input, b is the bias, * representing the convolution operation, and σ is non-linearity introduced in the network.

Touching upon the topic of the pooling layer, the exact location of a feature becomes less significant once it has been detected. Hence, the convolution layer is followed by pooling or sub-sampling layer. The major advantage of using pooling technique is that it remarkably reduces number of trainable parameters and introduces translation invariance. To perform pooling operation, a window is selected and the input elements lying in that window are passed through a pooling function as shown in figure 2.3.

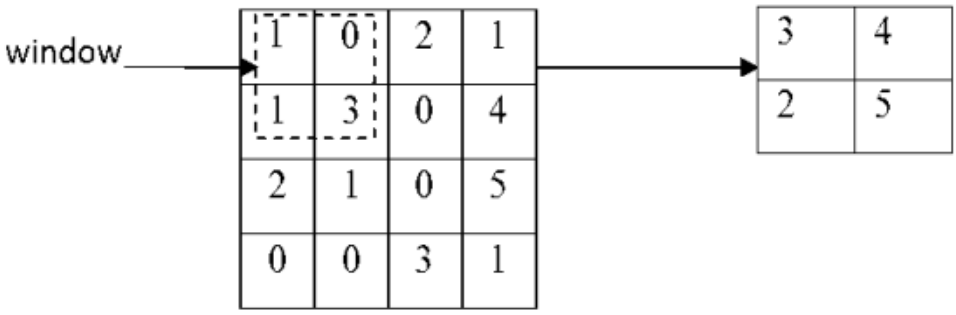

Figure 2.3 – Pooling operation performed by choosing a 2 x 2 window

The pooling function generates another output vector. There exist few pooling techniques like average pooling and max-pooling, out of which max-pooling is the most commonly used technique which reduces map-size very significantly. While

computing errors, the error is not back-propagated to winning unit because it does not take part forward flow.

Fully connected layer is similar to the fully connected network in the conventional models. The output of the first phase (includes convolution and pooling repetitively) is fed into the fully connected layer, and dot product of weight vector and input vector is computed in order to obtain final output [12]. Gradient descent, also known as batch mode learning or offline algorithm, reduces the cost function by estimating the cost over an entire training dataset, and updates the parameters only after one epoch, where an epoch corresponds to traversing the entire dataset. It yields global minima but if the size of training dataset is large, the time required to train the network substantially increases. This approach of reducing the cost function was replaced by stochastic gradient descent.

2.2. Architecture

Various architectures have been developed and implemented in CNN. The one that was used in this experiment will be described briefly in the scope of this paragraph.

The multi-layer networks are suitable for image recognition task because of the ability to learn from highly complex and high dimensional data. In 1998, [2] proposed an architecture called as LeNet architecture which uses dataset, has been summarized in the following paragraph. It has eight layers constituting five convolutional layers and three fully connected layers. Every unit in a plane has 25 inputs. Units in first hidden layer receives input from 5×5 area, which is a part of an entire image thus a very small region of the input image is passed to first hidden layer. This local area of input image is called receptive field of the unit. Every unit in a plane shares same weight vector. The output of unit is stored at the same location in the feature map. You can see architecture of LeNet5 in a figure 2.4 below

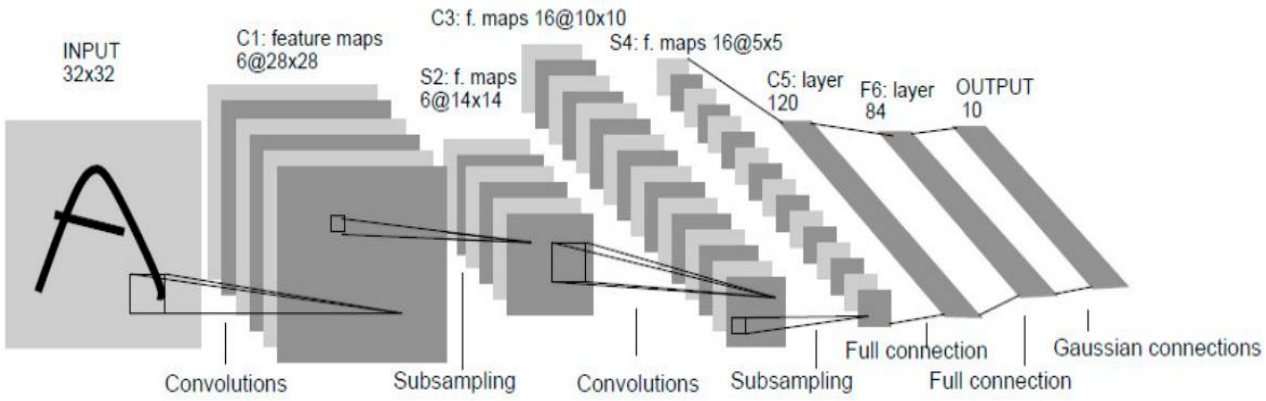

Figure 2.4 – Architecture of LeNet5 (each box represents a different feature map)

The neighbouring units in the feature map are the result of the neighbouring units in the previous layer. Thus, the result is overlapping of contiguous receptive fields. The first layer is convolution layer that consists of neurons which outputs sigmoid activation applied on the weighted sum [3]. As shown in figure 2.4, while computing contiguous units in the feature map, if 5 × 5 area is chosen as an input and a horizontal shift on the area is performed, it will result to overlapping of four rows and five columns. Various feature maps are generated which are the result of different weight vectors applied to the same input image. Different features can be extracted from the feature maps obtained. Essential property of CNN is that slight shift in the input does not affect feature map.

The precision of the position of the feature in an image is not crucial, thus to reduce the precision value subsampling is performed. As shown in figure 2.4, sub-sampling has been depicted in the second layer. Number of feature maps obtained after sub-sampling are same as those obtained after convolution. Here, in sub-sampling layer 2 × 2 area has been taken as input and compute average of the four inputs, multiply it by trainable coefficient and add the trainable bias, pass it to sigmoid function. Increase in number of feature maps can be observed as the spatial resolution decreases layer by layer. The learning is accomplished with back propagation method.

2.3. Learning algorithm

Learning Algorithms often called optimizing algorithms benefit the network by minimizing the objective function (often called loss function E(x)), dependent on various learnable parameters like weight, bias etc. Majorly the optimization algorithms can be branched into two categories i.e. first order optimization algorithms and second order optimization algorithms. The First Order Optimization include the computation of gradient represented by Jacobian matrix, the widely used technique is Gradient Descent. There exist a number of variants of Gradient Descent like Mini Batch Gradient Descent and Stochastic Gradient Descent. In order to improve results, improvements have done in the variants such as introduction of momentum, Adagrad, AdaDelta. Whereas Second Order Optimization include second order derivative represented by Hessian Matrix. One such technique is Adam Optimization.

While training the filters, error backpropagation is the mechanism used to modify the pre-initialized parameters of a network to achieve the optimized network parameters which can produce outputs close to target outputs. In CNN, such network can be achieved using error-backpropagation [5]. As the overall network is a feed forward network, the process starts by computing the outputs at every layer one by one and calculate the error component introduced on the last layer. Now, in order to obtain an optimized network, the computed gradients are backpropagated. Perform the same steps until the efficacy is observed. Initially input vector is provided to the network. Now, perform convolution operation on the input vector using equation 2.3 similar and detailed equation is depicted in equation 2.4.

$$C_q^1 = (\sum_{p=1}^{n} S_p^{l-1} * k_{p,q}^l + b_q^l), \tag{2.3}$$

$$C_q^1 = \left(\sum_{p=1}^{n} \sum_{u=-x}^{x} \sum_{v=-x}^{x} S_p^{l-1}(i-u, j-v) * k_{p,q}^l(u,v) + b_q^l\right), \tag{2.4}$$

where, n represents number of feature maps in last layer, p and q denotes feature map indices of current layer and previous layer respectively, ϕ denotes the activation function applied for example, ReLU and sigmoid, l denotes the layer, * denotes convolution operation, b and x represents the bias and size of filter respectively. Initially S_p^0represents the input image on which first convolution is to be performed $S_p^1$and represents the input on which second convolution is to be performed, which can be obtained after applying pooling on S_p^0. Refer to figure 2.5 below.

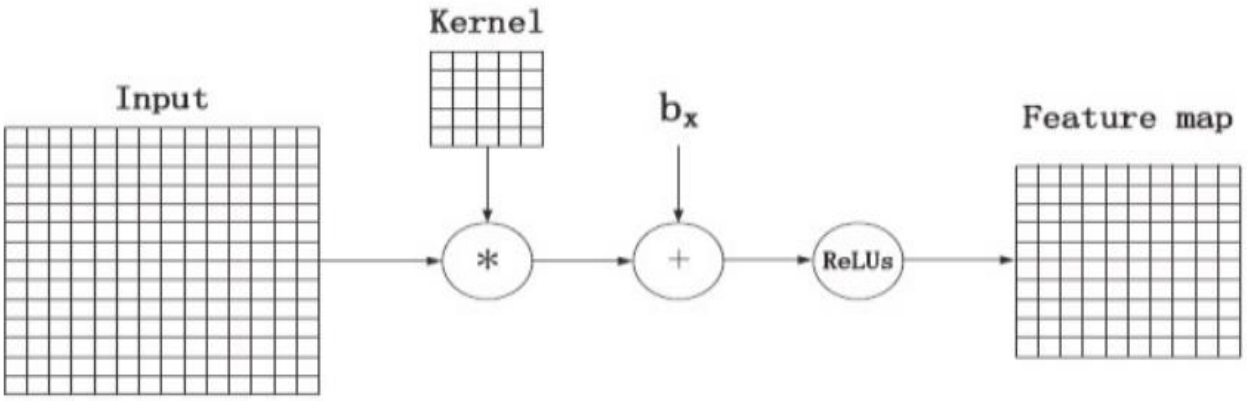

Figure 2.5 – Convolution operation on 14 x 14 input image by sliding 5 x 5 kernel which yields 10 x 10 feature map.

After convolution operation, pooling is applied in order to introduce translation invariance using equation 2.5.

$$s_q^l(i,j) = \frac{1}{4}\sum_{u=0}^{z}\sum_{v=0}^{z} C_q^l(2i-u, 2j-v), \tag{2.5}$$

where, z represents the pool window size. Perform significant number of iterations for convolution and pooling according to the requirement. Pass the result obtained in the last layer of the pooling through fully connected layer for classification and compute the output produced using equation 2.6.

$$\widehat{output} = \sigma(W \times f + b), \tag{2.6}$$

where, f is the final output vector obtained after last pooling operation, W is the weight vector of fully connected layer. A classifier can be used on the last layer. A well-known classifier softmax, depicted by equation 2.7 can be used. Where, label represents number of class labels.

$$\hat{y}(i) = \frac{e^{\widehat{output}}}{\sum_1^{labels} e^{\widehat{output}}}, \tag{2.7}$$

Compute the loss function using equation 2.8 which shows the error component introduced in the network.

$$L = \frac{1}{2}\sum_{i=1}^{noof\ training\ patterns}(\hat{y}(i) - y(i))^2, \tag{2.8}$$

where $\hat{y}(i)$ is the target output and $y(i)$ is the obtained output. Now, the error component needs to be backpropagated so that every neuron gets the penalty. The first order derivative wrt weight and bias are computed using equations 2.9, 2.10, 2.11.

$$\Delta W = \frac{\partial L}{\partial W(i,j)}, \tag{2.9}$$

$$= \frac{\partial L}{\partial \hat{y}} \cdot \frac{\widehat{\partial y}}{\partial W(i,j)}, \tag{2.10}$$

$$= \frac{\partial(\frac{1}{2}\sum_{i=1}^{p*}(\hat{y}(i) - y(i))^2)}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial W(i,j)}, \tag{2.11}$$

where p* denotes no of training pattern. See formula 2.12 below.

$$= (\hat{y}(i) - y(i)) \times \frac{\partial}{\partial W(i,j)} (\sigma(\sum_{j=1}^{noof nodes} W(i,j) \times f(j) + b(i))), \quad (2.12)$$

For sigmoid activation function equation 10 will be the result – formula 2.13 below.

$$\Delta\hat{y}(i) = \big(\hat{y}(i) - y(i)\big) \cdot \hat{y}(i)\big(1 - \hat{y}(i)\big)\,, \quad (2.13)$$

Calculation of ΔW see formula 2.14 below.

$$\Delta W(i,j) = \Delta\hat{y}(i) \times f(j)\,, \quad (2.14)$$

Calculation of Δb see formula 2.15 below.

$$\Delta b = \frac{\partial L}{\partial b(i)}, \quad (2.15)$$

The summarized algorithm for minimizing objective function using error-backpropagation has been shown below:

– step 1: Provide input vector to the network.

– step 2: Perform convolution using filter to produce a feature map.

– step 3: Pass the obtained feature map through ReLU to introduce non-linearity.

– step 4: Apply pooling operation on obtained feature map, which introduces translation invariance.

– step 5: Repeat Steps 2 to 4 for repetition of layers.

– step 6: The obtained feature maps are passed to fully connected layer for classification.

– step 7: Pass the output to a classifier such as softmax.

– step 8: Computer loss at the final layer and calculate gradient w.r.t. all the learnable parameters.

– step 9: Backpropagate the error component and update the parameters.

– step10: Perform the forward pass and repeat Steps 2 to 9 using updated parameters until network converges.

3 MNIST DATASET

In this work the MNIST dataset is going to be used. First off let's defined what it is and how to use it. The MNIST database of handwritten digits has a training set of 60,000 examples, and a test set of 10,000 examples. It is a subset of a larger set available from NIST. The digits have been size-normalized and cantered in a fixed-size image. See the example of the dataset below in figure 3.1.

Picture 3.1 – Example of a handwritten data taken from the dataset

It is a good database for people who want to try learning techniques and pattern recognition methods on real-world data while spending minimal efforts on preprocessing and formatting. The original black and white (bi-level) images from NIST were size normalized to fit in a 20x20 pixel box while preserving their aspect ratio. The resulting images contain grey levels as a result of the anti-aliasing technique used by the normalization algorithm. the images were cantered in a 28x28 image by computing the centre of mass of the pixels, and translating the image so as to position this point at the centre of the 28x28 field. With some classification methods (particularly template-based methods, such as SVM and K-nearest

neighbours), the error rate improves when the digits are cantered by bounding box rather than centre of mass. See the classifiers by error rates below in table 3.1.

Table 3.1 – Error rate comparison of Convolutional nets tested with MNIST dataset.

Classifier	**Preprocessing**	**Error rate (%)**
Convolutional net LeNet-1	16×16 pixels	1.7
Convolutional net LeNet-4	none	1.1
Convolutional net LeNet-4 with K-NN instead of last layer	none	1.1
Convolutional net LeNet-4 with local learning instead of last layer	none	1.1
Convolutional net LeNet-5, [no distortions]	none	0.95
Convolutional net LeNet-5, [huge distortions]	none	0.85
Convolutional net LeNet-5, [distortions]	none	0.8
Convolutional net Boosted LeNet-4, [distortions]	none	0.7
Convolutional net, cross-entropy [affine distortions]	none	0.6
Convolutional net, cross-entropy [elastic distortions]	none	0.4

Many methods have been tested with this training set and test set. Some of those experiments used a version of the database where the input images where deskewed (by computing the principal axis of the shape that is closest to the vertical, and shifting the lines so as to make it vertical). In some other experiments, the training set was augmented with artificially distorted versions of the original training samples. The distortions are random combinations of shifts, scaling, skewing, and compression.

4 MEDICAL IMAGING

The discoveries of seminal physical phenomena such as X-rays, ultrasound, radioactivity, and magnetic resonance, and the development of imaging instruments that harness them have provided some of the most effective diagnostic tools in medicine. The medical imaging community is now able to probe into the structure, function, and pathology of the human body with a diversity of imaging systems. These systems are also used for planning treatment and surgery, as well as for imaging in biology. Data sets in two, three, or more dimensions convey increasingly vast and detailed information for clinical or research applications. This information has to be interpreted in a timely and accurate manner to benefit health care. The examination is qualitative in some cases, quantitative in others; some images need to be registered with each other or with templates, many must be compressed and archived. To assist visual interpretation of medical images, the international imaging community has developed numerous automated techniques which have their merits, limitations, and realm of application.

Medical imaging refers to several different technologies that are used to view the human body in order to diagnose, monitor, or treat medical conditions. Each type of technology gives different information about the area of the body being studied or treated, related to possible disease, injury, or the effectiveness of medical treatment.

As a discipline, it is part of biological imaging and includes radiology, which uses imaging techniques such as radiography, magnetic resonance imaging (MRI), ultrasound (USG), endoscopy, elastography, tactile imaging, thermography, medical photography, and nuclear medicine techniques such as positron emission tomography (PET) and single photon emission computed tomography (SPECT). Measurement and recording is done by non-imaging techniques such as electroencephalography (EEG), magnetoencephalography (MEG), and electrocardiography (ECG), and is a technology that produces data presented as a graph/time function or map that contains data about measurement locations.

Up until 2010, 5 billion medical imaging studies had been performed worldwide. Radiation exposure from medical imaging accounted for about half of total exposure to ionizing radiation in the United States in 2006.

Medical imaging is often thought of as a set of techniques that noninvasively (without introducing instruments into the patient's body) produce images of an internal aspect of the body. In this narrow sense, medical imaging can be thought of as solving mathematical inverse problems. This means that the cause (properties of living tissue) is derived from the effect (observed signal). In the case of ultrasound, the probe consists of ultrasound waves and the echo that comes from the tissue. In the case of projection radiography, the probe is X-rays that are absorbed in different types of tissue, such as bone, muscle and fat.

4.1. Projection radiography

The principle of operation of the X-ray machine is based on bringing voltage to the control panel, from where, after adjustment, the voltage is transmitted to the main transformer. The increased voltage then reaches the X-ray tube and the radiation occurs. The rays pass through the skin and are absorbed to varying degrees by muscle and bone tissue. Calcium, which is part of the bones, absorbs the X-rays the most. That is why bones are bright white in the picture. Connective tissues, muscles, fat and fluid do not absorb the rays as intensively, so they have shades of grey in the image. Air absorbs the least amount of X-rays. Therefore, the cavities containing it will be the darkest in the image.

Bones and internal organs (sometimes for better visualization the organs are pre-filled with a contrasting substance) are clearly visible on the image obtained with the device that converts X-rays into a ready image, which allows to precisely identify various pathologies. The image in figure 4.1 below illustrates how the projection radiology works.

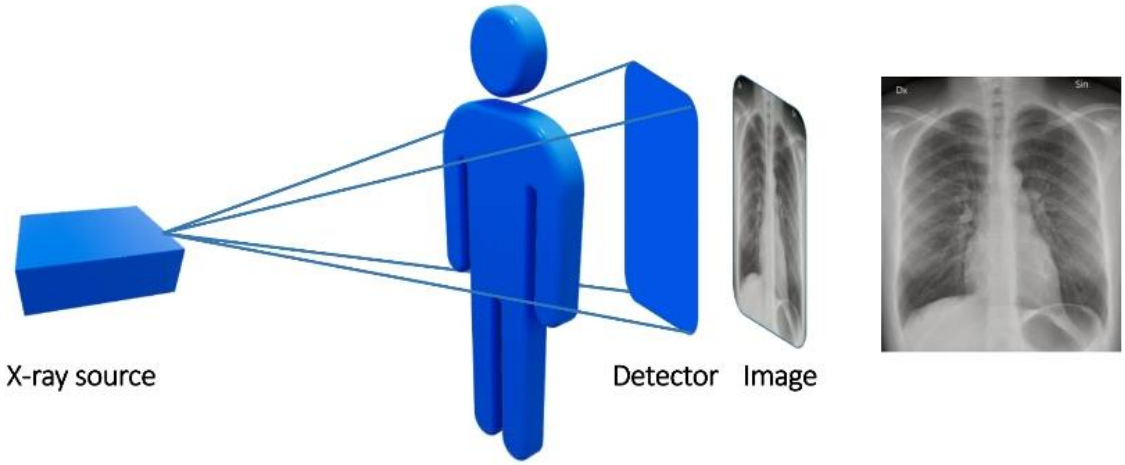

Figure 4.1 – Simple projection radiology mechanism

In this modality, images are generated using X-rays. These X-rays penetrate the human body and are partially absorbed by anatomical structures. The differently attenuated the beams are then detected by a dedicated detector and subsequently converted to an image, depending on the target to the body region. A different picture emerges. In this case, the thorax. The image contrast in protection, radiography comes from the different degrees of X-ray absorption between the X-ray source and the detector. This means that very dense or thick objects result in an X-ray shadow, with less X-rays reaching the detector on the chest X-ray. For example, bones are dense and thus have a different image intensity than the lungs, which are filled with air. It is important to mention that those intensities are relative. Usually, you can assume that high X-ray absorption is depicted as high image pixel intensities, where low absorption is depicted as low intensity on X-ray images. However, there are situations where the images are inverted. When a radiologist diagnosis an X-ray image, he or she often wants to measure some distances. Those distances are measured as the number of pixels on the detector. For us, humans, however, it is close to impossible to interpret such a pixel distance. We can obtain such a measurement by calibration of the full setup that allows us to approximately convert

or translate pixel distances to physical ones. Here are some more examples of X-ray images presented in figure 4.2.

Figure 4.2 – From left to right: A thorough radiograph that revealed bronchial carcinoma within the left lung; the examination of a hand without any abnormal findings; the skin of an ankle during surgical procedure.

In modern X-ray machines, the output radiation can be recorded on a special cassette with a film or on an electronic matrix. Devices with an electronic sensitive matrix are much more expensive than analog devices. The films are printed only if needed, and the diagnostic image is displayed on the monitor and, in some systems, stored in a database along with other patient data.

In diagnostic radiography, it is advisable to take pictures in at least two projections. This is due to the fact that a radiograph is a flat image of a three-dimensional object. And as a consequence, the localization of the detected pathological focus can be established only with the help of two projections.

The quality of the obtained X-ray image is determined by three main parameters: the voltage applied to the X-ray tube, the current intensity, and the exposure time (duration of the X-ray radiation). Depending on the anatomical formations under study and the anthropometry of the patient, these parameters can vary significantly. There are average values for different organs and tissues, but it

should be kept in mind that the actual values will differ depending on the machine where the examination is carried out and the patient to whom the radiography is carried out. An individual table of values is compiled for each machine. These values are not absolute and are adjusted as the examination progresses. The quality of the images produced depends largely on the ability of the radiographer to adequately adapt the table of mean values to the individual patient. To reduce the dynamic unsharpness of the images caused by the nonabsolute immobility of the examined organ or the patient himself, the required exposure should be created with a short exposure time and high peak power of the X-ray tube.

Key advantages of projection radiography:

– wide availability of the method and ease of examination;

– most examinations do not require special preparation of the patient;

– relatively low cost of the study.

Disadvantages of projection radiography:

– stativity of the image - difficulty in assessing organ function;

– the presence of ionizing radiation, which can have a harmful effect on the patient;

– informative value of classic radiography is much lower than such modern methods of medical imaging;

– without the use of contrast agents, radiography is insufficiently informative to analyse changes in soft tissues that differ little in density.

4.2. Computed tomography

A CT scanner emits a series of narrow beams through the human body as it moves through an arc. This is different from an X-ray machine, which sends just one radiation beam. The CT scan produces a more detailed final picture than an X-ray image. The CT scanner’s X-ray detector can see hundreds of different levels of density. It can see tissues within a solid organ. This data is transmitted to a computer,

which builds up a 3-D cross-sectional picture of the part of the body and displays it on the screen. Sometimes, a contrast dye is used because it can help show certain structures more clearly. For instance, if a 3-D image of the abdomen is required, the patient may have to drink a barium meal. The barium appears white on the scan as it travels through the digestive system. If blood vessel images are the target, a contrast agent will be injected into the veins. The accuracy and speed of CT scans may be improved with the application of spiral CT, a relatively new technology. The beam takes a spiral path during the scanning, so it gathers continuous data with no gaps between images. CT is a useful tool for assisting diagnosis in medicine, but it is a source of ionizing radiation, and it can potentially cause cancer.

The physical basis of the method is the exponential law of attenuation of radiation, which is true for purely absorbing media. In the X-ray radiation range, the exponential law is fulfilled with a high degree of accuracy, so the developed mathematical algorithms were first applied specifically to X-ray computed tomography. Computed tomography uses the same principle, but in a more sophisticated way. In a CT scanner, the X-ray source and corresponding detector rotate around the patient and acquire images from multiple different angles. Using this information, three dimensional scans from inside the body can be reconstructed. See the principle in figure 4.3.

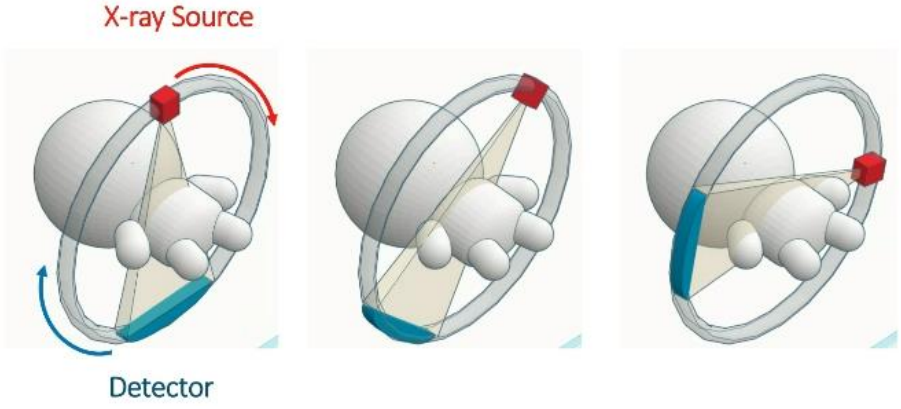

Figure 4.3 – CT principle mock-up

In this figure, you can see how such a scanner looks like the patient is placed on the examination table, which then moves through the scanner to produce three dimensional images of different body parts such as brain or lungs. As one acquisition only takes a few seconds, computed tomography is also heavily used in emergency medicine to detect, as an example, a stroke – figure 4.4.

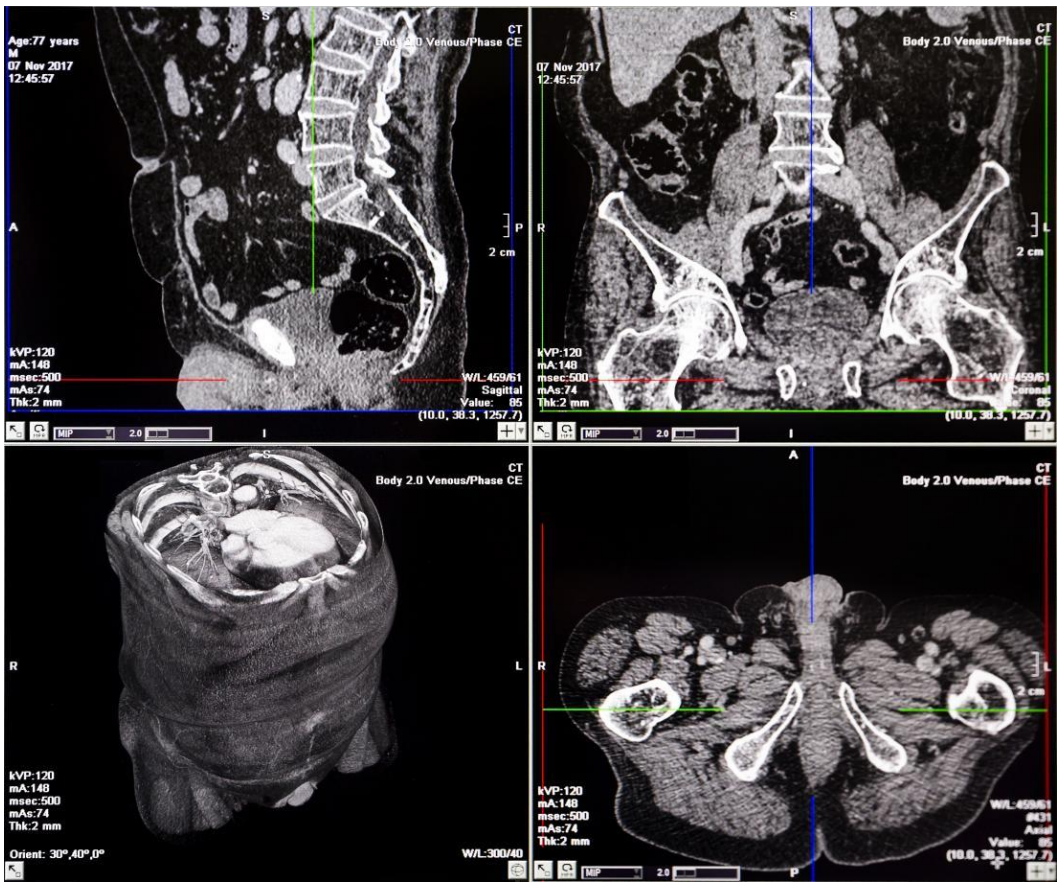

Figure 4.4 – Example of a CT scan

Similar to projection radiography, the image contrast in CT scans is generated by different degrees of X-ray absorption. In contrast to normal X-ray images, the degree of X-ray absorption and thus the image intensity can be quantified in absolute values called Hounsfield units. As an example, air has minus 1000 HU and water Hounsfield unit of 0 HU. Usually, high X-ray absorption leads to high image

intensity, whereas low absorption leads to low intensity. Typically, CT scans are never inverted. Another difference to projection radiography is that distances between single voxels are what is a pixel in the three-dimensional space can be exactly quantified and directly correspond to the physical distances as shown in the example below in figure 4.4.

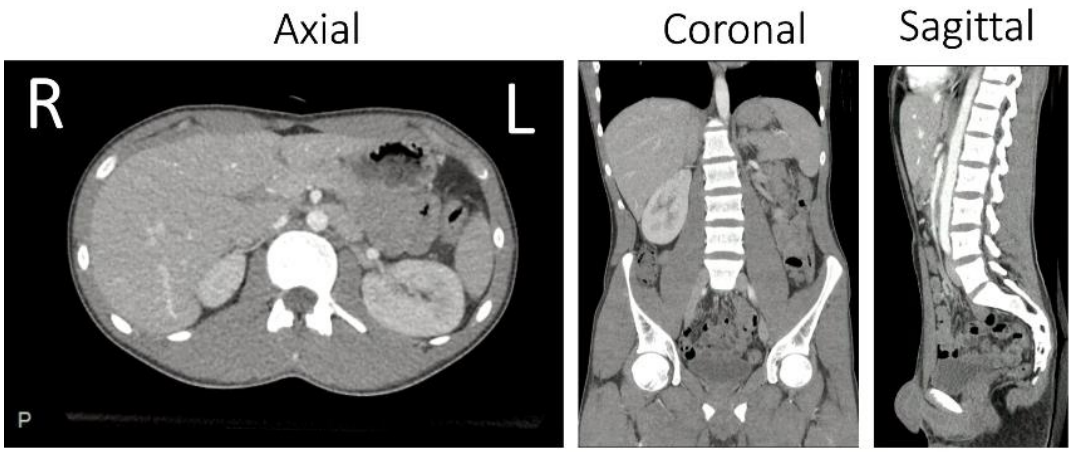

Figure 4.5 – Example of different cross-sections through a CT scan.

As a conclusion to this paragraph, I'd like to pay attention to the fact that CTs can provide better diagnostic confidence compared to projection radiography. An X-ray is built to examine dense tissues, while a CT scan is better able to capture bones, soft tissues, and blood vessels all at the same time. X-ray equipment is much smaller and less complex than a CT scan since a CT scanner needs to rotate around the patient being scanned.

4.3. Magnetic resonance imaging

The method of nuclear magnetic resonance allows studying the human's organism based on the saturation of the organism tissues with hydrogen and

peculiarities of their magnetic properties related to being surrounded by different atoms and molecules. The hydrogen nucleus consists of a single proton, which has a spin and changes its spatial orientation in a powerful magnetic field, as well as under the influence of additional fields, called gradient fields, and external radiofrequency pulses applied at a resonance frequency specific to the proton in a given magnetic field. Based on the proton parameters (spins) and their vector directions, which can only be in two opposite phases, as well as their binding to the proton's magnetic moment, it is possible to determine in which tissues a particular hydrogen atom is located.

If we place a proton in an external magnetic field, its magnetic moment will be either co-directional or opposite to the magnetic field, and in the second case its energy will be higher. When electromagnetic radiation of a certain frequency is applied to the investigated area, some of the protons will reverse their magnetic moment and then return to their original position. At the same time the data acquisition system of the tomograph registers the energy release during the relaxation of pre-excited protons.

MRI examinations take several minutes, two hours in extreme cases. Therefore, they are not suitable for emergency medicine. As already mentioned, the image generation process in MRIs is highly complex, but in short, images from within the body are generated by measuring magnetic properties of tissues in a strong magnetic field. The great advantage to X-ray imaging is that no ionizing radiation is used. In contrast to standard radiography, many different image contrasts can be generated in an MRI due to the physical mechanisms for image generation. Usually, image pixels or voxels that have high MRI signal intensity have high intensity and vice versa. The image intensities are usually relative values and therefore a unique for each patient and skin.

Below, you can see two different contrasts of brain images called T1 and T2 in figure 4.6. In T1 the ventricles are located at the centre of the brain are dark, whereas in T2 weighted memories, they are bright. Those ventricles are filled with

fluid which means we can summarize that in T1 weighted MRIs water is shown dark, whereas in T2 weighted memorize it is shown bright. Figure 4.7 represents different MRIs.

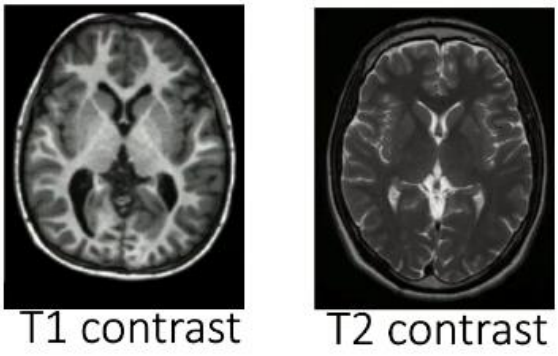

Figure 4.6 – Examples of human's brain MRIs

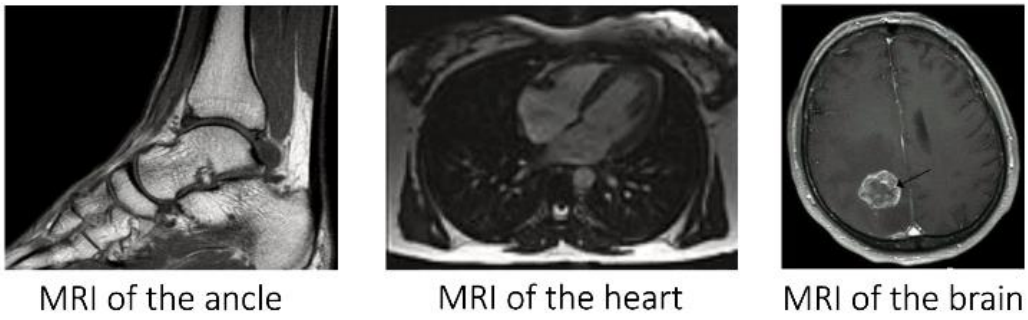

Figure 4.7 – From left to right: an MRI image of an ankle in sagittal orientation; an MRI off the chest in axial orientation; an examination of the brain that revealed a metastasis

Both MRIs and CT scans can view internal body structures. However, a CT scan is faster and can provide pictures of tissues, organs, and skeletal structure. An MRI is highly adept at capturing images that help doctors determine if there are abnormal tissues within the body. MRIs are more detailed in their images.

5 PNEUMONIA CLASSIFICATION

Pneumonia is a common infectious disease that is responsible for over one million cases and tens of thousands of deaths annually in the US alone. As an example, in 2017, 1.3 million cases were reported, of which over 50000 died, resulting in a death rate of nearly four percent. The incidence of pneumonia is stable for 30 years and in European countries is 14 per 1,000 people, including non-specific lung diseases, accounting for 40% of cases. The disease is characterized by severe pathomorphological change, etiology and symptoms of acute pneumonia, changed views on some key issues of diagnosis and treatment of disease. Among patients with pneumonia dominated by men - 55%. The incidence of pneumonia increases with age. The highest mortality occurs among people over 55 years. It is absolutely crucial to recognize if pneumonia in time as otherwise it can be life threatening.

The lack of accepted, widely understood and commonly used definition(s) for pneumonia causes a fundamental problem where related but heterogeneous pathologies and clinical phenotypes are poorly classified. The lack of clear classification results in difficulty with clinical decision making and a potential for poorly formulated research. The magnitude of this problem is most evident in the common inability to identify the infectious organism(s) causing lung infection, necessitating empiric antibiotic therapy. If a specific diagnosis could be made, specific therapy could be provided which would be of similar efficacy to empiric wide spectrum therapy [4] and avoid millions of prescriptions of broad-spectrum antibiotics and the associated risks of antibiotic resistance.

The magnitude of the problem is less evident in the field of pneumonia research. In a qualitative sense, the problem may be distilled to a lack of homogeneity in clinical and pathological phenotypes under investigation. In studies of heterogeneous groups, the research problems that may arise include an inability to determine aetiology due to a limited range of methods; pathology or microbiology with disparate patterns; and conflicting results between studies that investigate risk factors, diagnostic methods or treatments. Heterogeneous groups may result in

disparate and unfocused studies, which fail to target the most important types of pneumonia and the most important questions, and make limited contributions. In epidemiologic terms, investigation of heterogeneous groups will, to a lesser or greater extent, threaten the internal validity of studies. When heterogeneous groups are studied, invalid estimates of effect occur due to misclassification bias [4]. In the field of pneumonia research, determining aetiology is a common difficulty. For example, in the absence of specimens from the lung, studies of aetiology may misclassify causality to organisms detected in nasopharyngeal or sputum samples—in this situation, misclassification bias occurs due to the difficulty in accurately determining the aetiology of lung infection. The example can be see below in figure 5.1.

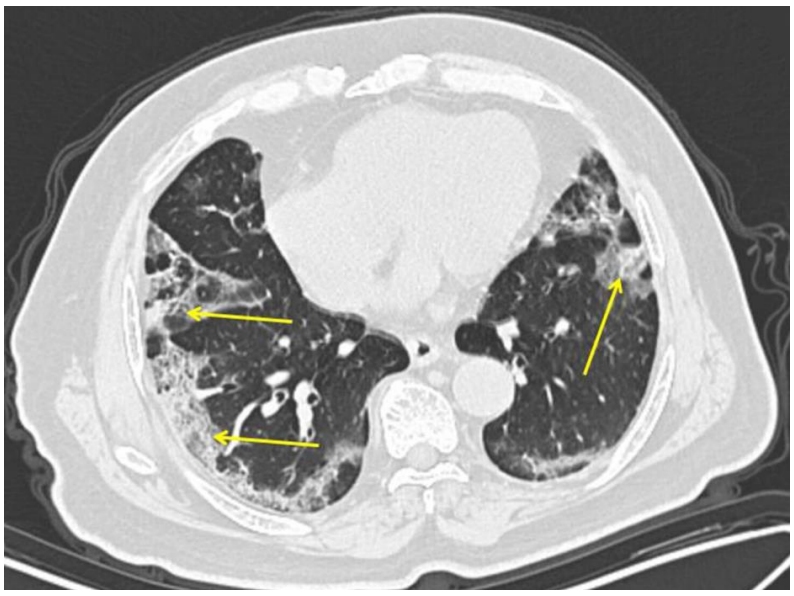

Figure 5.1 – Example of an X-ray scan of a patient with pneumonia (marked with yellow arrows)

A chest radiograph is frequently used in diagnosis. In people with mild disease, imaging is needed only in those with potential complications, those not having improved with treatment, or those in which the cause is uncertain. If a person is sufficiently sick to require hospitalization, a chest radiograph is recommended.

Findings do not always match the severity of disease and do not reliably separate between bacterial and viral infection. X-ray presentations of pneumonia may be classified as lobar pneumonia, bronchopneumonia, lobular pneumonia, and interstitial pneumonia. Bacterial, community-acquired pneumonia classically show lung consolidation of one lung segmental lobe, which is known as lobar pneumonia. However, findings may vary, and other patterns are common in other types of pneumonia. Aspiration pneumonia may present with bilateral opacities primarily in the bases of the lungs and on the right side. Radiographs of viral pneumonia may appear normal, appear hyper-inflated, have bilateral patchy areas, or present similar to bacterial pneumonia with lobar consolidation. Radiologic findings may not be present in the early stages of the disease, especially in the presence of dehydration, or may be difficult to interpret in the obese or those with a history of lung disease. Complications such as pleural effusion may also be found on chest radiographs. Laterolateral chest radiographs can increase the diagnostic accuracy of lung consolidation and pleural effusion. A CT scan can give additional information in indeterminate cases. CT scans can also provide more details in those with an unclear chest radiograph (for example occult pneumonia in chronic obstructive pulmonary disease) and can exclude pulmonary embolism and fungal pneumonia and detect lung abscess in those who are not responding to treatments. However, CT scans are more expensive, have a higher dose of radiation, and cannot be done at bedside. Lung ultrasound may also be useful in helping to make the diagnosis. Ultrasound is radiation free and can be done at bedside. However, ultrasound requires specific skills to operate the machine and interpret the findings. It may be more accurate than chest X-ray.

In summary, refining the definition and classification of pneumonia is a formidable task as multiple terms are used in multiple fields of medical practice and research. The dangers of poor classification of pneumonia are widespread empiric antibiotic therapy and heterogeneous groups in research, which have a tendency to influence the construction of research questions and studies. As a result, these research questions and studies may not provide clear answers.

6 HEPATOCELLULAR CARCINOMA

Primary liver cancer is one of the most severe liver cancers, occurring between 1.08 and 50.6% of all malignant neoplasms in the world. Hepatocellular cancer accounts for about 85% of all malignant liver tumours. Cholangiocellular cancer accounts for about 5-10% of primary liver cancer, and the remainder is accounted for by rarer neoplasms: hemangiosarcoma, hepatoblastoma, and mesenchymal tumours. In most countries of the world there is an increase of morbidity and mortality of primary liver cancer. More than 80% of hepatocellular cancers occur against a background of cirrhosis. The most common cause of cirrhosis is hepatitis B and C. Surgical treatment is currently the mainstay of treatment for primary liver cancer. However, there is an acute problem of the possibility of large liver resections, liver resections in the presence of concomitant pathological processes, and repeated operative interventions in case of the diagnosis of metachronous liver tumours, recurrences, and metastases in the rest of the liver. If surgery is not possible due to significant comorbidities or the extent of the tumour process, radiofrequency chemoablation, cryodestruction, arterial chemoembolization and polychemotherapy are used.

Liver cancer is the sixth most common oncopathology in the world, with a high rate of complications and deaths. It is manifested by pain in the right subcostal area and digestive disorders. Liver tumours are treated by resection, minimally invasive techniques, chemotherapy, and immunotherapy.

Liver cancer is the world's fifth most common malignancy in men, and ninth most common in women. More than 800 thousand new cases are diagnosed in the world annually, and the share of this disease among the causes of cancer mortality is 8.3%.

Cancer is divided into primary cancer, which starts in the cells of the liver, and secondary cancer, which metastasises from neighbouring organs. The pathology has few nonspecific signs and is therefore often detected when it is already advanced.

The course of the disease and its consequences depend on the timeliness of diagnosis, the histological type, the treatment strategy and the general health of the patient.

The TNM-based staging classification is the most important in determining treatment options and prognosis, where T is the primary tumour, N is metastases to regional lymph nodes, and M is distant metastases:

– stage IA: -T1aN0M0;

– stage IB: -T1bN0M0;

– stage II: -T2N0M0;

– stage IIIA: -T3N0M0;

– stage IIIB: -T4N0M0;

– stage IVA: -T(any)N1M0;

– stage IVB: -T(any)N(any)M1.

The most common primary cancer is hepatocellular carcinoma (HCC), which forms when liver cells (hepatocytes) mutate. Rarer forms of the disease include:

– cholangiocellular cancer (cholangiocarcinoma, CC), a neoplasm from cells in the intrahepatic bile ducts;

– mixed hepatocholangiocarcinoma;

– fibrolamellar carcinoma (FLC).

Secondary cancers are more common than primary hepatic oncopathology. About ⅓ of all malignancies can metastasise to the liver, which is associated with a good blood supply to the organ. Most commonly metastatic neoplasms arise as a complication of tumours of abdominal organs, blood from which flows through the hepatic parenchyma via the portal vein system. More than 100,000 people have been diagnosed with metastatic liver cancer in Russia. Metastatic cancer is characterized by multiple tumor masses, and consequently, involvement of a considerable volume of the parenchyma in the pathological process. Signs of total affection are registered in 75% of cases, single metastases - in 16%, and solitary - only in 9% of patients. It

has been noted that with multiple metastases, the average life expectancy of patients decreases by a factor of 1.4.

Most patients are initially asymptomatic with liver oncopathology. There may be minor symptoms such as bloating, loss of appetite or discomfort in the right side of the belly which patients do not pay much attention to. As the tumour grows, the liver capsule becomes stretched and neighbouring anatomical structures become compressed, the following clinical signs occur:

– pain and distention in the right side of the abdomen;

– nausea and vomiting;

– lack of appetite;

– stool disorders;

– symptoms of intoxication – malaise, impaired performance, increased body temperature, weight loss.

When the cancer reaches a large size, it squeezes the bile ducts and causes mechanical jaundice. Signs include jaundice of the skin, sclerae and mucous membranes, discolouration of the stool and urine, and itching of the skin. In the advanced stage of the disease anaemia, gastrointestinal bleeding and fluid accumulation in the peritoneal cavity (ascites) develop. In hepatocellular carcinoma there is often local invasive growth, with malignant cells sprouting into the diaphragm.

The most common cause of hepatocellular carcinoma (see example in figure 6.1) is chronic viral hepatitis B and C, which cause cirrhosis and malignant fibrotic nodular degeneration if prolonged. The following can cause cancer:

– genetic predisposition;

– alcohol abuse and a long history of smoking;

– rare metabolic diseases;

– metabolic syndrome;

– exposure to toxins (aflatoxin, vinyl chloride, arsenic).

Figure 6.1 – Example of a CT scan of a patient with diagnosed liver cancer (tumour is in red circle)

On the basis of localization, size, spread and clinical symptoms, oncology distinguishes 4 stages of malignant liver nodules:

– stage I - Cancer presents as a single tumour of any size, which does not invade blood vessels and surrounding tissues, and has no metastases in lymph nodes. There are no subjective and objective signs of the disease;

– stage II - A single neoplasm that has invaded the blood vessels, or several tumours up to 5 cm in size are present. The cancer does not form regional or distant metastases. Symptoms may occur at this stage;

– stage III - Diagnosed when multiple lesions larger than 5 cm, which have not metastasised, or when several tumours smaller than 5 cm have invaded the portal or hepatic vein;

– stage IV - Cancer has metastasised to regional lymph nodes and/or other internal organs, and the size of the primary focus is irrelevant to the diagnosis.

If there are symptoms of liver damage and a malignancy is suspected, the patient should undergo a full examination by an oncologist. Diagnosis begins with a

history and physical examination. To confirm or rule out hepatocellular cancer, special methods of examination are prescribed:

– CT angiography of the hepatic vessels;

– dynamic liver scintigraphy;

– liver biopsy, followed by histological examination;

– alpha-fetoprotein assay;

– serological reactions for hepatitis B, C, D virus antigens and antibodies.

If there are signs of metastasis, an X-ray of the tubular bones and spine, MRI of the brain, and chest CT scan are added to the examination programme. Patients who are diagnosed with cancer and have symptoms of other diseases may need to consult a hepatologist, a gastroenterologist, and a pulmonologist.

The surgical approach (resection of malignant lesions) is the gold standard of treatment in oncology, providing the best prognosis for long-term survival. Retrospective studies demonstrate a 5-year survival rate after surgery in patients with HCC and preserved liver function of 50-70%. Cancer is an indication for anatomical or multiple resection. Possible options for anatomic liver resections include:

– hemihepatectomy (removal of the right or left lobe of the organ);

– sector resection (removal of two segments);

– segmental resection.

In order to safely perform radical surgery, the two-stage technique (including ALPPS) is used at the Russian Ministry of Health for patients in whom the remaining volume of the parenchyma is insufficient or a decrease in the functional reserves of the organ has been identified. In order to make a final decision on the possibility of surgical treatment in a specific patient, a comprehensive examination is carried out. Liver transplantation is the method of choice for patients with hepatocellular carcinoma and when the cancer is accompanied by severe cirrhosis. Replacing the diseased organ with a healthy donor transplant shows good functional results and

can prolong life and improve its quality. In modern oncology, liver cancer can be treated with minimally invasive techniques, such as:

– radioembolization – cancer cells are destroyed using radioactive microspheres. The emboli block the arterial vessels and deprive the malignant tissue of oxygen and nutrients. Radioembolization is safer than classic radiation therapy;

– local destruction – radiofrequency ablation, microwave, laser, cryoablation and other ablation methods are used when surgery is impossible or not feasible, in combination with resection, or when there is recurrence after surgical treatment. Stereotactic radiotherapy for HCC can be considered as an ablative option for single (no more than 3) tumour nodules;

– HIFU therapy – to destroy the cancer, local exposure to high-intensity focused ultrasound is used. The thermal and mechanical energy generated at the focus point destroys the tumour cells within seconds.

Liver cancer is resistant to chemotherapy drugs, so they are used in a limited fashion, mainly as part of palliative treatment to treat the signs and symptoms of the disease. A promising area of cancer therapy is immunotherapy - the use of new biologically active substances that affect a person's own immune system and activate it to fight the tumour. All cancer patients are monitored regularly and undergo regular examinations, including instrumental and laboratory tests, as recommended by the supervising oncologist. In the event of a sudden worsening of well-being or the appearance of atypical symptoms, a doctor should be consulted as soon as possible.

7 PROGRAM DEVELOPMENT

There are a few important things to define before starting to develop the program solution for the described issue. First of all, I'd like to cover the topic of Numpy library and what role it plays in the project. What's really interesting about Numpy is that pretty much almost every data science library that will be present in the project: Typekit Learn, Pandas or Seaborn all built off the base of Numpy. To begin with, let's define why it's important to actually learn about Numpy. NumPy is an open-source module for python which provides common mathematical and numerical operations as pre-compiled, fast functions. They are combined into high-level packages. They provide functionality that can be compared to that of MatLab. NumPy (Numeric Python) provides basic methods for manipulating large arrays and matrices. The program solution will be dealing with either a one-dimensional vector or a two-dimensional array of data. However, Numpy is expandable to any number of dimensions. What's really important about Numpy is its ability to quickly broadcast functions, and it also has a ton of built in features for optimizing work with data, including things like linear algebra, statistical distributions, trigonometric functions and random number capabilities. Why should they be used? While no structures at first look very similar to standard Python lists, they're actually much more efficient than just a Python list or even a nested python list. The key to all of this is the broadcasting capabilities are also extremely useful for quickly applying functions to an entire dataset very quickly.

The following paragraph will bring more understanding about the importance of tensors and their use in the project. Tensors are multi-dimensional arrays with a uniform type. In other words, a tensor is a generalized matrix. All tensors are immutable like Python numbers and strings: you can never update the contents of a tensor, only create a new one. Below, there is an image that describes the types of tensor dimensions in figure 7.1.

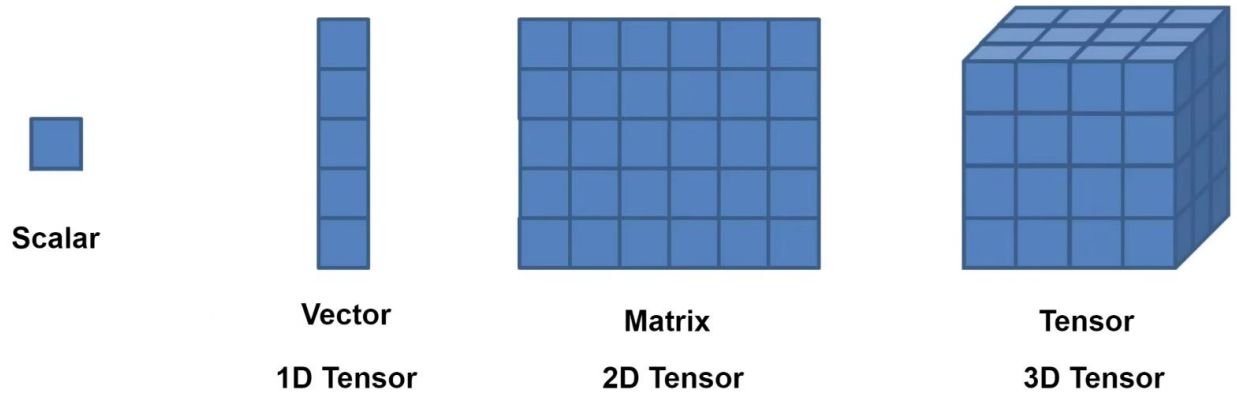

Figure 7.1 – Dimensions of a tensor with their names

The main two reasons for using tensors in the code of this project are: it's often easiest to arrange the data sets as tensors and dealing with image data is easier with 3D tensors. It will allow to have more complex layers like pulling layers and convolutional layers.

Finally, before proceeding to the creation of a convolutional neural network to classify the MNIST dataset, I want to revisit how the MNIST dataset is going to be changed for the CNN. Recall the flattening out that endless data will cause the loss to the information. It's possible to feed convolutional neural networks with the data as an array of two-dimensional images. We can think of the entire group of those 60000 images as a tensor itself. That means when we think of a batch of images, that batch itself is going to be a tensor object or an end dimensional array. It means that there will be twenty-eight by twenty-eight images organized as an array. So for the labels, a One-Hot encoding will be incorporated. This means that instead of having labels such as one, two, three four, etc., there will be a single array for each image, because it is necessarily that each neuron and the output layer to line up for a class. It is also important to bear in mind that when dealing with tensors of image data, we end up with four dimensions: number of images, height, width, colour channels.

7.1 MNIST with CNN

First standard imports must be performed. Then load the MNIST dataset. PyTorch makes the MNIST train and test datasets available through torchvision. The first time they're called, the datasets will be downloaded onto your computer to the path specified. From that point, torchvision will always look for a local copy before attempting another download. See figure 7.2 of successfully loaded MNIST dataset.

```
[3]: train_data

[3]: Dataset MNIST
     ,    Number of datapoints: 60000
     ,    Root location: ../Data
     ,    Split: Train
     ,    StandardTransform
     ,Transform: ToTensor()

[4]: test_data

[4]: Dataset MNIST
     ,    Number of datapoints: 10000
     ,    Root location: ../Data
     ,    Split: Test
     ,    StandardTransform
     ,Transform: ToTensor()
```

Figure 7.2 – Output of successfully loaded MNIST dataset

Then the loaders will be created. When working with images, relatively small batches are required; a batch size of 4 is not uncommon. The next step is defining a convolutional model. Two convolutional layers and two pooling layers will be employed before feeding data through fully connected hidden layers to the output. The model follows CONV/RELU/POOL/CONV/RELU/POOL/FC/RELU/FC. This operation will consist of the substeps that are: extending the base Module class;

setting up the convolutional layers with torch.nn.Conv2d(); setting up the fully connected layers with torch.nn.Linear(). The input size of (5x5x16) is determined by the effect of our kernels on the input image size. A 3x3 filter applied to a 28x28 image leaves a 1-pixel edge on all four sides. In one layer the size changes from 28x28 to 26×26. We could address this with zero-padding, but since an MNIST image is mostly black at the edges, we should be safe ignoring these pixels. We'll apply the kernel twice, and apply pooling layers twice, so our resulting output will be $(((28 - 2) / 2) - 2) / 2 = 5.5$ which rounds down to 5 pixels per side. Activations can be applied to the convolutions in one line using F.relu() and pooling is done using F.max_pool2d(). Flatten the data for the fully connected layers. Including the bias terms for each layer, the total number of parameters being trained is: (1×6×3×3)+6+(6×16×3×3)+16+(400×120)+120+(120×84)+84+(84×10)+10=54+6 +864+16+48000+120+10080+84+840+10=60,074. After that define loss function & optimizer and, finally move to the model training step.

The data will be fed to the model without flattening it first. Run the training batches, apply the model, tally the number of correct predictions, update parameters, print the interim results. There is an output for each training cycle that is called epoch. You can see the output of three training cycles in the figure 7.3 below.

```
epoch:  0  batch:  600 [  6000/60000]  loss: 0.09493287  accuracy:  96.717%
epoch:  0  batch: 1200 [ 12000/60000]  loss: 0.08817653  accuracy:  96.758%
epoch:  0  batch: 1800 [ 18000/60000]  loss: 0.01354485  accuracy:  96.761%
epoch:  0  batch: 2400 [ 24000/60000]  loss: 0.07290515  accuracy:  96.892%
epoch:  0  batch: 3000 [ 30000/60000]  loss: 0.34648952  accuracy:  96.910%
epoch:  0  batch: 3600 [ 36000/60000]  loss: 0.50707448  accuracy:  97.028%
epoch:  0  batch: 4200 [ 42000/60000]  loss: 0.00697857  accuracy:  97.121%
epoch:  0  batch: 4800 [ 48000/60000]  loss: 0.08758115  accuracy:  97.175%
epoch:  0  batch: 5400 [ 54000/60000]  loss: 0.00251424  accuracy:  97.246%
epoch:  0  batch: 6000 [ 60000/60000]  loss: 0.03734301  accuracy:  97.307%
epoch:  1  batch:  600 [  6000/60000]  loss: 0.13037242  accuracy:  98.233%
epoch:  1  batch: 1200 [ 12000/60000]  loss: 0.00371105  accuracy:  98.350%
epoch:  1  batch: 1800 [ 18000/60000]  loss: 0.02641156  accuracy:  98.294%
epoch:  1  batch: 2400 [ 24000/60000]  loss: 0.21039827  accuracy:  98.237%
epoch:  1  batch: 3000 [ 30000/60000]  loss: 0.00519724  accuracy:  98.250%
epoch:  1  batch: 3600 [ 36000/60000]  loss: 0.24456143  accuracy:  98.178%
epoch:  1  batch: 4200 [ 42000/60000]  loss: 0.00304838  accuracy:  98.217%
epoch:  1  batch: 4800 [ 48000/60000]  loss: 0.08488221  accuracy:  98.279%
epoch:  1  batch: 5400 [ 54000/60000]  loss: 0.01072581  accuracy:  98.270%
epoch:  1  batch: 6000 [ 60000/60000]  loss: 0.00575854  accuracy:  98.275%
epoch:  2  batch:  600 [  6000/60000]  loss: 0.00154493  accuracy:  98.800%
epoch:  2  batch: 1200 [ 12000/60000]  loss: 0.00118856  accuracy:  98.675%
epoch:  2  batch: 1800 [ 18000/60000]  loss: 0.00001560  accuracy:  98.739%
epoch:  2  batch: 2400 [ 24000/60000]  loss: 0.04269046  accuracy:  98.696%
epoch:  2  batch: 3000 [ 30000/60000]  loss: 0.00336559  accuracy:  98.633%
epoch:  2  batch: 3600 [ 36000/60000]  loss: 0.00882289  accuracy:  98.633%
epoch:  2  batch: 4200 [ 42000/60000]  loss: 0.00024180  accuracy:  98.614%
epoch:  2  batch: 4800 [ 48000/60000]  loss: 0.10380044  accuracy:  98.638%
epoch:  2  batch: 5400 [ 54000/60000]  loss: 0.00190454  accuracy:  98.661%
epoch:  2  batch: 6000 [ 60000/60000]  loss: 0.00029711  accuracy:  98.670%
```

Figure 7.3 – Example of output for three training cycles

Now let’s plot the loss and accuracy comparisons. See the figure 7.4.

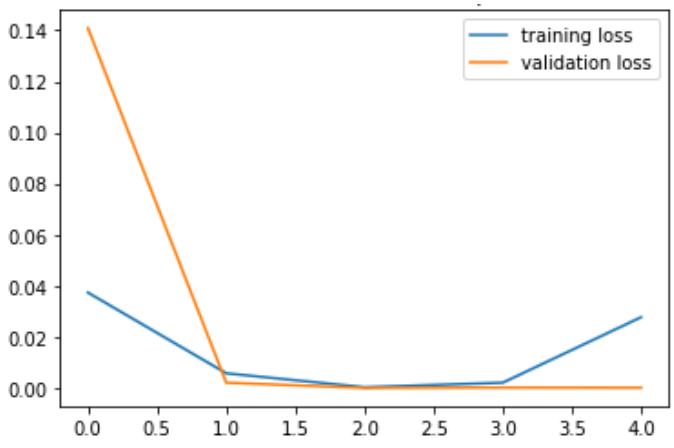

Figure 7.4 – Loss and accuracy comparisons graph

Also let’s plot graph of accuracy at the end of each epoch in figure 7.5.

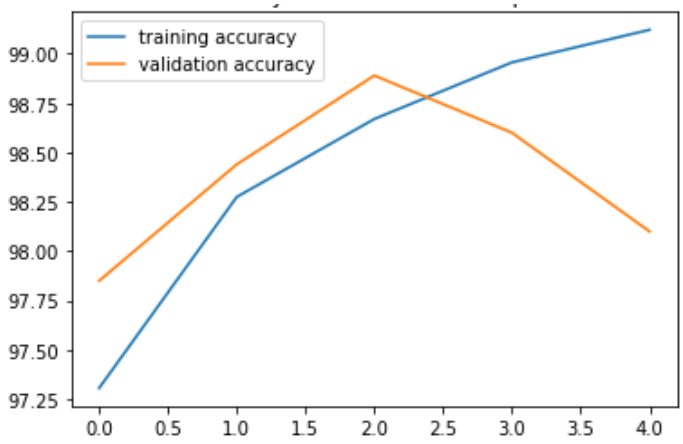

Figure 7.5 – Graph of accuracy at the end of each epoch

As we can see, 98,67% of accuracy is impressive. The results of this experiment will assist in building a model for pneumonia classification and tumour segmentation. All the gathered results and mistakes will be considered for the next models.

7.2 Pneumonia classification

In this paragraph a classifier will be trained to predict whether an X-Ray of a patient shows signs of pneumonia or not. The following steps for pneumonia classification will be described: preprocessing, training, interpretability.

First, let's download the data for analysis. The training data is provided as a set of patientIds and bounding boxes. Bounding boxes are defined as follows: x-min y-min width height There is also a binary target column, Target, indicating pneumonia or non-pneumonia. There may be multiple rows per patientId. All provided images are in DICOM format. Samples without bounding boxes are negative and contain no definitive evidence of pneumonia. Samples with bounding boxes indicate evidence of pneumonia.

File descriptions:

– stage_2_train.csv - the training set. Contains patientIds and bounding box / target information;

– stage_2_sample_submission.csv - a sample submission file in the correct format. Contains patientIds for the test set. Note that the sample submission contains one box per image, but there is no limit to the number of bounding boxes that can be assigned to a given image;

– stage_2_detailed_class_info.csv - provides detailed information about the type of positive or negative class for each image.

```
from pathlib import Path #for convenient path handling
import pydicom #for reading dicom files
import numpy as np #for storing the actual images
import cv2 #for directly resizing the images
import pandas as pd #to read the provided labels
import matplotlib.pyplot as plt #for visualizing some images
from tqdm.notebook import tqdm #for nice progress bar
```

Figure 7.6 – The list of default imports

Then, the csv file containing the labels must be read. Note that subjects may occur multiple times in the dataset because different pneumonia spots are handled individually. For our classification task, we can remove those duplicates as we are only interested in the binary label. After that let's define the path to the dicom files and also the path were we want to store our processed npy files.

In order to efficiently handle our data in the Dataloader, we convert the X-Ray images stored in the DICOM format to numpy arrays. Afterwards let’s compute the overall mean and standard deviation of the pixels of the whole dataset, for the

purpose of normalization. Then the created numpy images are stored in two separate folders according to their binary label:

- 0: All X-Rays which do not show signs of pneumonia;
- 1: All X-Rays which show signs of pneumonia.

To do so, patient ids and concat the patient ID with the ROOT_PATH must be iterated over. Directly save the standardized and resized files into the corresponding directory (0 for healthy, 1 for pneumonia). This allows to take advantage of the ready-to-use torchvision DatasetFolder for simple file reading. Standardize all images by the maximum pixel value in the provided dataset, 255. All images are resized to 224x224. To compute dataset mean and standard deviation, let's compute the sum of the pixel values as well as the sum of the squared pixel values for each subject. This allows to compute the overall mean and standard deviation without keeping the whole dataset in memory. The example of the dataset is presented in the figure 7.7 below.

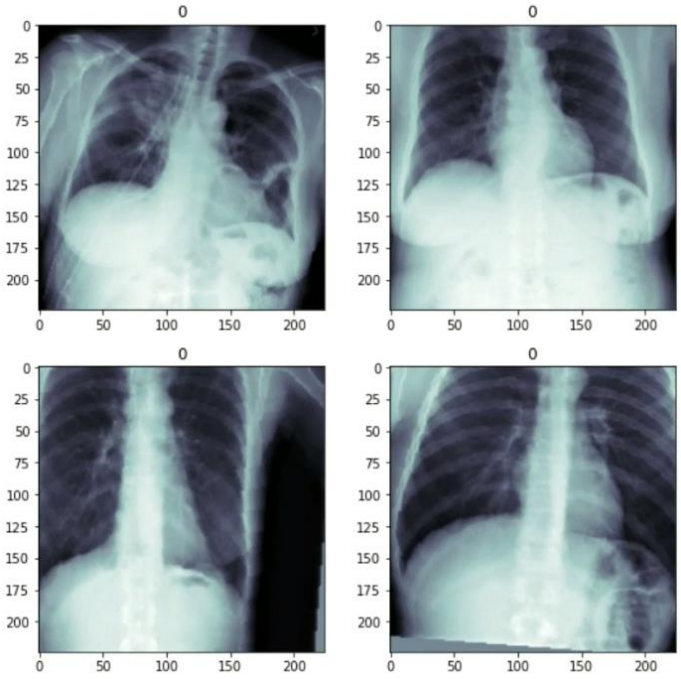

Figure 7.7 – An example of loaded data

Now, we'll proceed to the training step. The list of imports is in figure 7.8.

```
import torch #for model and dataloader creation
import torchvision #transforms from torchvision for Data Augmentation and Normalization
from torchvision import transforms
import torchmetrics # for easy metric computation
import pytorch_lightning as pl #for efficient and easy training implementation
from pytorch_lightning.callbacks import ModelCheckpoint
from pytorch_lightning.loggers import TensorBoardLogger
from tqdm.notebook import tqdm #for progress par when validating the model
import numpy as np
import matplotlib.pyplot as plt #for visualizing some images
```

Figure 7.8 – The list of default imports

It is now possible to leverage the DatasetFolder from torchvision: It allows to simply pass a root directory and return a dataset object with access to all files within the directory and the directory name as class label. It is only needed to define a loader function, load_file, which defines how the files shall be loaded. This is very comfortable as we only have to load our previously stored numpy files. Additionally, we need to define a list of file extensions (just "npy" in our case). Then let's pass a transformation sequence for Data Augmentation and Normalization. The following properties are used:

– randomResizedCrops which applies a random crop of the image and resizes it to the original image size (224x224);

– random Rotations between -5 and 5 degrees;

– random Translation (max 5%);

– random Scaling (0.9-1.1 of original image size).

Then the train dataset and val dataset and the corresponding data loaders are created. Batch size and num_workers must be adapted according to the hardware resources. If the classes are imbalanced: There are more images without signs of

pneumonia than with pneumonia. There are multiple ways to deal with imbalanced datasets: Weighted Loss; Oversampling; Doing nothing.

Each pytorch lightning model is defined by at least an initialization method, a forward function which defines the forward pass/prediction, a training_step which yields the loss and configure_optimizers to specify the optimization algorithm. Additionally, training_epoch_end callback is used to compute overall dataset statistics and metrics such as accuracy. Subsequently, the validation_step is defined. The validation step performs more or less the same steps as the training step, however, on the validation data. In this case, pytorch lightning doesn't update the weights. Validation_epoch_end can be used to compute overall dataset metrics. No loops or manual weight updates are needed!

Now it is time to create the model - the ResNet18 network architecture will be used. As most of the torchvision models, the original ResNet expects a three channel input in conv1. However, our X-Ray image data has only one channel. The in_channel parameter from 3 to 1 must be changed. The Adam Optimizer with a learning rate of 0.0001 and the BinaryCrossEntropy Loss function. (In fact BCEWithLogitsLoss is used, which directly accepts the raw unprocessed predicted values and computes the sigmoid activation function before applying Cross Entropy). Then create a checkpoint callback which only stores the 10 best models based on the validation accuracy.

Let's evaluate the model. At first, the latest checkpoint is loaded and the model is sent to the GPU, if possible. Compute prediction on the complete validation set and store predictions and labels. Compute metrics: It is observable that the overall result is already decent with the simple model. However, it suffers from a large amount of False Negatives due to the data imbalance. This is of particular importance in to avoid in medical imaging as missing findings might be fatal. An alternative to retraining with a weighted loss is to reduce the classification threshold from 0.5 to e.g 0.25. It produces way less false negatives but increases the number of False positives. This is called the precision-recall trade-off.

Finally, let's use Class Acvitation Maps. Then let's define the list of default imports (you can see it in figure 7.9):

```
%matplotlib notebook
import torch #for tensor manipulation
import torchvision #for resnet18
from torchvision import transforms #for Normalization
import pytorch_lightning as pl #for model creation
import numpy as np #for data loading
import matplotlib.pyplot as plt #for plotting
```

Figure 7.9 – The list of default imports

The key idea of CAM is to multiply the output of the last convolutional layer (BasicBlock 1 of layer 4) $\boldsymbol{A_k}$ (consisting of k channels) with the parameters $\boldsymbol{w}$ of the subsequent fully connected layer to compute an activation map $\boldsymbol{M}$. See formula 7.1:

$$M = \sum_k \omega_k A_k \,, \tag{7.1}$$

The network to a generator using the children() function can be converted which means that the list function is used to convert it into a list. The convolutional part of the network comprises all layers up to the AdaptiveAvgPool2d layer. Using Sequential from pytorch, the list of layers is converted back to a Sequential Model. Let's add an additional output to the forward function of our pneumonia model, to return the feature maps of the last convolutional layer *(A)*. The feature map is extracted in the forward pass, followed by global average pooling and flattening. Finally, we use the fully connected layer to compute the final class prediction. After that the CAM function is defined by using the formula from above.

Below, in figure 7.10 the image shows the heatmap focuses on the area which shows signs of pneumonia.

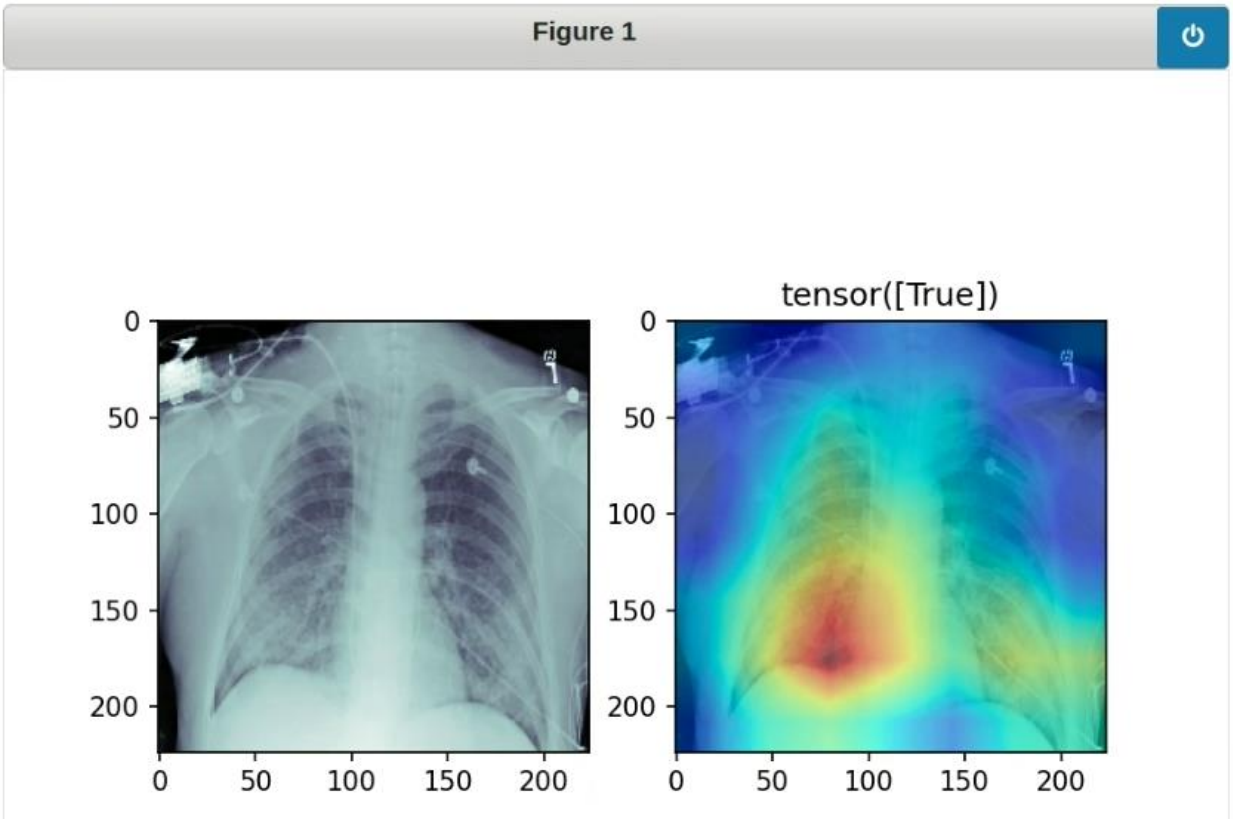

Figure 7.10 – Model successfully defined that the scan has a presence of pneumonia

7.3 Liver and tumour segmentation

The next model will be trained how to segment liver and liver tumour in full body CT scans. This unit will not work with two-dimensional slices as the previous but use 3D images. The dataset data set was designed to explore the axis of difficulties typically encountered when dealing with medical images, such as small data sets, unbalanced labels, multi-site data and small objects. It consists of 131 full body CTs of varying shape. The example is in figure 7.11.

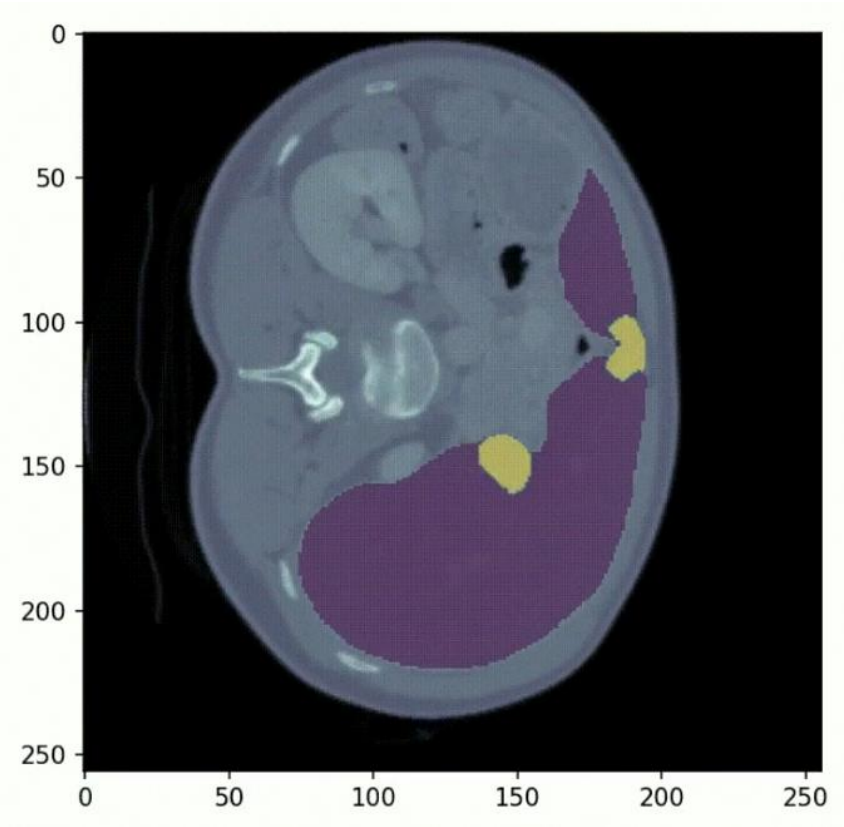

Figure 7.11 – Single screenshot from a full body CT scan: The liver is purple. Deletions are segmented in yellow.

Since this model will be working with a CT scan, instead of handling a simple 2-D image of e.g., shape (256×256) we have a volume of shape (256×256×256). It increased the information content as the network sees multiple slices at once (3D convolutions). In consequence, the problem of memory must be tackled. A volume of size 256×256×256 needs 64Mb in memory. However, the main problem arises when computing a parallelized forward pass or training the network. It may take up to 20-50 gigabytes during the forward pass. The solution is to use smaller patches – a volume of shape (256×256×256) yields 8 128x128x128 patches. The number 8 comes from the fact that it's possible to fit two of those patches along the x-axis, two along the y-axis, two along the z-axis, resulting in 2^3 that equals 8. In conclusion,

forward pass will require only 2-3 gigabytes. Then the patches will be aggregated together to the original volume.

The steps described above will be implemented using TorchIO library which is efficient for loading, preprocessing, augmentation, and patch-based sampling of 3D medical images in deep learning, following the design of PyTorch. It also provides functionalities for all aspects of three-dimensional imaging. The implementation flow will look the following way:

– convert all niftis to subject and store them in a list;

– create a tio.SubjectsDataset based on the subject list;

– define a sampler to sample patches;

– create tio.Queue for efficient data loading;

– create pytorch DataLoader based on Queue and proceed as usual;

– define GridSampler to split the subject into patches;

– define GridAggregator which merges the predicted segmentations back together;

– compute predictions and perform aggregations.

Now let’s move to the data visualization step. As this dataset has over 26GB we provide a resampled version of it. The new scans are of shape (256×256×Z), where Z is varying and reduce the size of the dataset to 2.5GB. It is not necessarily to preprocess this dataset as the necessary steps are directly performed by torchio during training. First the helper function is implemented that automatically replaces "imagesTr" with "labelsTr" in the filepaths so that we can easily switch between CT images and label masks. Then let’s load NIfTI and extract image data. Extracted data is presented in figure 7.12.

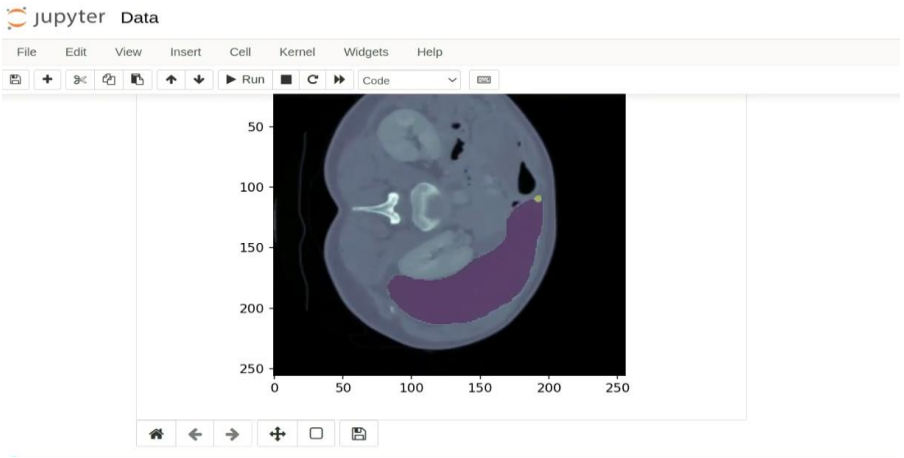

Figure 7.12 – Animated data loaded to the environment

The previously defined 2D-UNET architecture with some small changes is used:

– Conv2d -> Conv3d;

– MaxPool2d -> MaxPool3d;

– "trilinear" upsampling method;

– three Output Channels instead of One to model background, liver and tumour.

Additionally, the filters used in the convolutions are reduced to shrink the network size. Below is the list of imports used for training the model:

– pathlib for easy path handling;

– HTML for visualizing volume videos;

– torchio for dataset creation;

– torch for DataLoaders, optimizer and loss;

– pytorch-lightning for training;

– numpy for masking;

– matplotlib for visualization;

– the 3D model.

We can loop over all available scans and add them to the subject list. Regarding the processing, the CropOrPad functionality is used, which crops or pads all images and masks to the same shape (256×256×200). Now let's define the train and validation dataset. 105 subjects will be used for training and 13 for testing. In order to help the segmentation network learn, the LabelSampler with p=0.2 is used for background, p=0.3 for liver and p=0.5 for liver tumours with a patch size of (96×96×96). The queue to draw patches from is created. The tio.Queue accepts a SubjectsDataset, a max_length argument describing the number of patches that can be stored, the number of patches to draw from each subject, a sampler and the number of workers. Then let's define train and val loader. As the dataloaders only pop patches from the queue 0 num workers are used!

Finally, the Segmentation model is created. The Adam optimizer with a learning rate of 1e-4 and a weighted cross-entropy loss is used, which assigns a threefold increased loss to tumorous voxels.

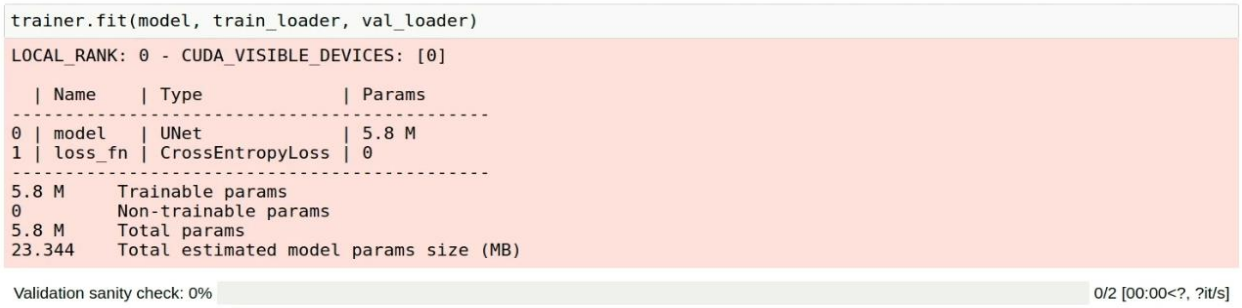

Figure 7.13 – The beginning of the training process

The model was trained in a patch wise manner as the full volumes are too large to be placed on a typical GPU. However, it is required to get a result for the whole volume. Torchio helps us doing so by performing Patch Aggregation. The goal of patch aggregation is to split the image into patches, then compute the

segmentation for each patch and finally merge the predictions into the prediction for the full volume. The pipeline is as follows:

– Define the GridSampler(subject, patch_size, patch_overlap) responsible for dividing the volume into patches. Each patch is defined by its location accesible via tio.LOCATION;

– Define the GridAggregator(grid_sampler) which merges the predicted patches back together;

– Compute the prediction on the patches and aggregate them via aggregator.add_batch(pred, location);

– Extract the full prediction via aggregator.get_output_tensor().

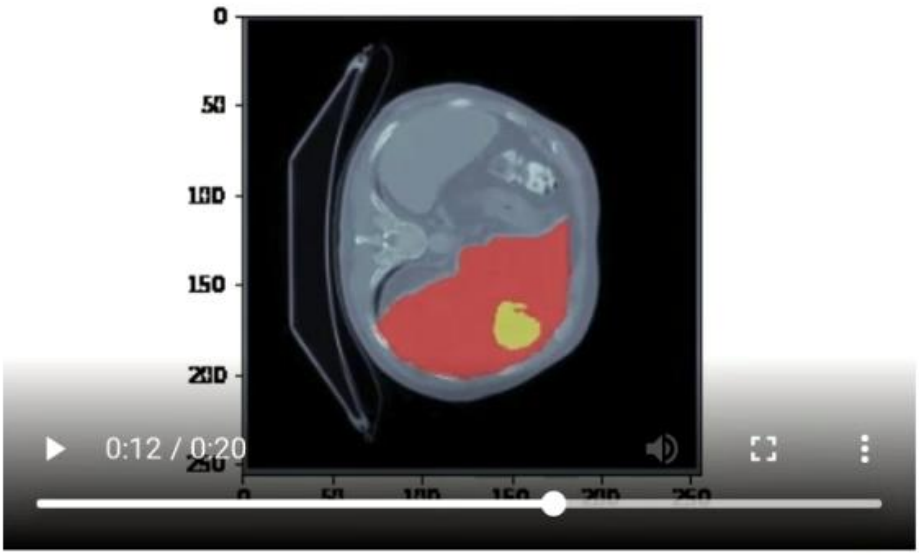

Figure 7.13 – Video of successful liver (red) and tumour (yellow) segmentation by the model

Additionally, DataLoader from pytorch is leveraged to perform the prediction in a batch wise manner for a nice speed up. The visualized prediction is shown above in figure 7.13.

SUMMARY

This paper looked at the prospect of using machine vision using convolutional neural networks to solve complex and non-standard medical tasks like classifying pneumonia and liver tumour segmentation. After learning the foundations of Medical Imaging and data formats, the model was built that automatically detects and segments cancer using 2D and 3D scans of organs. The precision of the model has been successfully boosted up to 98,67% which is impressive.

Artificial intelligence has drastically broadened the scope of influence in recent years, allowing people to utilize its opportunities in serious spheres like healthcare, where a mistake can lead to a fatal outcome. Healthcare is at an inflection point. Through a combination of technology and expertise AI helps organizations build a more resilient future. For example, Watson Health provides industry-leading data, analytics and AI solutions to help providers, payers, governments and life science companies modernize operations and get more value from ever-expanding health data.

Accurate cancer classification is significant in saving the lives of many humans. Despite the use of known diagnostic tools, many researchers are currently interested in using AI classification techniques to classify cancer. The difficulty of integrating AI into radiologists' workflow leads to envisage lung cancer diagnosis becoming more automated in a series of small, gradual steps. A computer system could initially do a baseline reading of each lung scan and present them in the order the doctor prefers — from easiest to hardest to interpret, for instance, or from the highest to the lowest likelihood of lung cancer.

The advantage of adopting AI incrementally is that radiologists will not have to suddenly change the way they work. Deep-learning tools can be incorporated into the existing computer-aided diagnosis systems. As clinicians and engineers feed ever more lung CT scans into deep-learning systems, privacy must remain paramount.

That will result in better long-term outcomes. Using AI to find tumours early can effectively double the amount of time oncologists have to treat a patient, giving them much more opportunity to keep the cancer from spreading. That deep-learning systems can outperform humans on some diagnostic tasks does not mean that they will take over radiologists' jobs.

None of the researchers expects AI to replace physicians, radiologists or pathologists. But with an ageing population, increased availability of diagnostic tests and growing emphasis on precision medicine, machine learning could help them to do their jobs by identifying the high-risk cases they should focus on and helping them to make decisions about uncertain diagnoses.

The potential of AI for various types of cancer prognosis and diagnosis is reported in this paper. It is expected that AI-based clinical cancer research will result in a paradigm shift in cancer treatment, thereby resulting in dramatic improvement in patient survival due to enhanced prediction rates. Thus, it is logical to expect that the challenges of cancer prognosis and diagnosis will be solved by advances in AI in the foreseeable future.

LIST OF SOURCES

1. Paris G., Robilliard D., Fonlupt C. (2004) Exploring Overfitting in Genetic Programming. Artificial Evolution, International Conference, Evolution Artificielle, Ea 2003, Marseilles, France, October. DBLP p. 125-132.
2. Jensen D D, Cohen P R. (2000) Multiple Comparisons in Induction Algorithms. Machine Learning. p. 4-11.
3. Zhou J , Cao Y, Wang X, Li P, Xu W. Deep recurrent models with fast-forward connections for neural machine translation. p. 10-17.
4. Shoeb AH , Guttag J. Application of machine learning to epileptic seizure detection. Proceedings of the 27th International Conference on Machine Learning (ICML-10), 2010 p. 20.
5. An introduction to machine learning with scikit-learn. 2016 [cited 2022; Available from: http://scikit-learn.org/stable/tutorial/basic/tutorial.html.
6. Carbonell, J. G., Michalski, R. S., and Mitchell, T. M. (1983) “An overview of machine learning. In Machine learning.” Springer Berlin Heidelberg. p. 3-23.
7. Rothman KJ, Greenland S, Lash TL. Validity in Epidemiologic Studies, Ch 9. In: Rothman KJ, Greenland S, Lash TL, editors. Modern epidemiology. 3rd ed. Philadelphia: Lippincott Williams & Wilkins; 2008. p.98
8. LeCun, Y., Bengio, Y., and Hinton, G. (2015) “Deep learning.” Nature 521 (7553): p436-444.
9. LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998) “Gradient-based learning applied to document recognition.” Proceedings of the IEEE 86 (11): p.2278-2324.
10. Lee, K. B., Cheon, S., & Kim, C. O. (2017) “A convolutional neural network for fault classification and diagnosis in semiconductor manufacturing processes.” IEEE Transactions on Semiconductor Manufacturing 30 (2): 135-142.Lima, E., Sun, X., p.57-90

11. Nebauer, C. (1998) “Evaluation of convolutional neural networks for visual recognition.” IEEE Transactions on Neural Networks 9 (4): p. 685- 696.
12. Tivive, F. H. C., and Bouzerdoum, A. (2005) “Efficient training algorithms for a class of shunting inhibitory convolutional neural networks.” IEEE Transactions on Neural Networks 16 (3): p. 541-556.
13. Zhou, Y., Wang, H., Xu, F., and Jin, Y. Q. (2016) “Polarimetric SAR image classification using deep convolutional neural networks.” IEEE Geoscience and Remote Sensing Letters 13 (12): p. 1935-1989.
14. Angeles Marcos, M., Camps, M., Pumarola, T., Antonio Martinez, J., Martinez, E., Mensa, J., Garcia, E., Peñarroja, G., Dambrava, P., Casas, I., Jiménez de Anta, M. T., & Torres, A. (2006). The role of viruses in the aetiology of community-acquired pneumonia in adults. Antiviral therapy, 11(3), p. 351–359.
15. Bewick, T., Simmonds, M., Chikhani, M., Meyer, J., & Lim, W. S. (2008). Pneumonia in the context of severe sepsis: a significant diagnostic problem. The European respiratory journal, 32(5), p. 1417–1418.
16. Huijskens, E., Koopmans, M., Palmen, F., van Erkel, A., Mulder, P., & Rossen, J. (2014). The value of signs and symptoms in differentiating between bacterial, viral, and mixed etiology in patients with community-acquired pneumonia. Journal of medical microbiology, 63(Pt 3), p. 441–452.