BRNO UNIVERSITY OF TECHNOLOGY

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

FAKULTA ELEKTROTECHNIKY
A KOMUNIKAČNÍCH TECHNOLOGIÍ

DEPARTMENT OF TELECOMMUNICATIONS

ÚSTAV TELEKOMUNIKACÍ

CLASSIFICATION OF THORAX DISEASES ON CHEST X-RAY IMAGES USING ARTIFICIAL INTELLIGENCE

KLASIFIKACE NEMOCI NA RENTGENOVÝCH SNÍMCÍCH POMOCÍ UMĚLÉ INTELIGENCE

BACHELOR'S THESIS

BAKALÁŘSKÁ PRÁCE

AUTHOR | Štěpán Pijáček
AUTOR PRÁCE

SUPERVISOR | Ing. Anzhelika Mezina
VEDOUCÍ PRÁCE

BRNO 2022

Bachelor's Thesis

Bachelor's study program **Telecommunication and Information Systems**

Department of Telecommunications

Student: Štěpán Pijáček — ***ID:*** 223313

Year of study: 3 — ***Academic year:*** 2021/22

TITLE OF THESIS:

Classification of thorax diseases on chest X-ray images using artificial intelligence

INSTRUCTION:

Study the neural networks and the Python environment. Study the problem of disease classification on X-rays of the lungs and summarize the current state of science and technology in this area. Choose a suitable dataset for experiments.

Test existing methods, which classify the thorax disease in lung X-ray images. Design a neural network that will be able to classify an image basing on the thorax disease. Train the model in a Python / Tensorflow environment and compare the accuracy with existing methods. Show the measured values in a table and compare them. Discuss the achieved results.

RECOMMENDED LITERATURE:

[1] Chen, Bingzhi, et al. "DualCheXNet: dual asymmetric feature learning for thoracic disease classification in chest X-rays." Biomedical Signal Processing and Control 53 (2019): 101554.

[2] Guan, Qingji, et al. "Thorax disease classification with attention guided convolutional neural network." Pattern Recognition Letters 131 (2020): 38-45.

Date of project specification: 7.2.2022 — ***Deadline for submission:*** 31.5.2022

Supervisor: Ing. Anzhelika Mezina

prof. Ing. Jiří Mišurec, CSc.
Chair of study program board

ABSTRACT

This thesis is researching workable solutions to the problem of classification of thorax disease on chest x-ray images using artificial intelligence. For a better understanding of the problem, the first chapters explain the basic convolutional neural network and its advantages and disadvantages. Based on these first explanations, two neural networks which are expanding on the concept of the convolutional neural network are chosen. Those are capsulated network and residual network both explained further in their respective sections with their advantages and disadvantages. Residual network and Capsulated network are implemented using programming language python and framework TensorFlow with Keras library, both with their respective chapters. At the end of this thesis, you can find results and conclusion.

KEYWORDS

Neural Networks, CapsNet, ResNet, CNN, Thorax, CXR8, CXR14, Classification, Multi Classification, Diseases

ABSTRAKT

Tato práce se zabývá výzkumem použitelných řešení pro problém klasifikace onemocnění hrudníku na rentgenových snímcích hrudníku pomocí umělé inteligence. Pro lepší pochopení problému jsou v prvních kapitolách vysvětleny základní konvoluční neuronové sítě a jejich výhody a nevýhody. Na základě těchto prvních vysvětlení jsou vybrány dvě neuronové sítě, které rozšiřují koncept konvoluční neuronové sítě. Těmito sítěmi jsou kapslová síť a reziduální síť, obě jsou dále vysvětleny v příslušných kapitolách s jejich výhodami a nevýhodami. Reziduální síť a kapslová síť jsou poté implementovány pomocí programovacího jazyka python a frameworku TensorFlow s knihovnou Keras, obě se svými příslušnými kapitolami. Na konci práce jsou uvedeny výsledky a závěr.

KLÍČOVÁ SLOVA

Neuronové sítě, CapsNet, ResNet, CNN, hrudní koš, CXR8, CXR14, klasifikace, multiklasifikace, nemoci

1 Rozšířený abstrakt

1.1 Úvod

Informační technologie a samotný výpočetní výkon v posledních letech zaznamenaly nárůst schopnosti provádět velmi složité úlohy pomocí strojového učení. Jednou z oblastí, kde by tento technologický pokrok mohl přinést mnoho výhod pro všechny, je zdravotnictví.

Výzkum této práce je zaměřen na nalézání potenciálních technologií a implementací asistované klinické diagnostiky, konkrétně diagnostiky na rentgenových snímcích oblasti hrudníku.

Rentgenové vyšetření hrudníku je jedním z nejběžnějších způsobů diagnostiky stavu pacientů. Je to proto, že tento typ vyšetření je relativně levný, rychlý a je schopen ukázat mnoho informací. Tato metoda má však svá úskalí, jejichž překonání není vždy snadné překonat. Hlavním problémem je možnost chybné interpretace toho, co je na snímku vidět. To může vést k chybné léčbě pacienta, k dalšímu psychickému tlaku na pacienta a zpětně ke zhoršení jeho stavu, i když k tomu objektivně nemusí existovat žádný důvod. Na opačné straně je zde možnost, že se pacientovi vůbec nedostane adekvátní péče, protože radiolog údaje neinterpretoval správně. Tím pádem se mylně domnívá, že pacient je zdravý, což může zpětně vést k pacientově smrti.

Z výše uvedených důvodů bylo navrženo mnoho experimentů, metod a architektur, které mají pomoci s diagnostikou. Tyto experimenty mají různou míru úspěšnosti.

Hlavním přínosem této práce je výzkum možných řešení problému klasifikace rentgenových snímků. Na základě tohoto výzkumu je pak navrženo možné řešení. Druhým přínosem této práce je výzkum možnosti využití kapslové sítě pro medicínské aplikace a její rozšíření o příslušné podpůrné vrstvy a sítě, tak aby dosáhla co možná nejlepších výsledků. Tato síť byla zvolená na základě výzkumu možných řešení. Byla zvolena i navzdory tomu, že její aplikace na takto komplexní datovou množinu nebyla plně prověřena. Dalšími důvody pro volbu této sítě je její preferovaná chování a přistupování k datům, toto chování z části minimalizuje a v některých případech eliminuje nežádoucí jevy vyskytující se v jiných populárních řešeních.

Tato práce je rozdělena do dvou hlavních kapitol. První kapitola je teoretická, v níž lze nalézt důkladné seznámení s kapslovou sítí spolu s představením reziduální sítě a problémů, které doprovází výzkum umělé inteligence. Tyto problémy jsou zaměřeny zejména na pole zpracování obrazových dat a jejich následnou klasifikaci tak, aby se minimalizovaly nežádoucí jevy vznikající při klasifikaci a zpracování.

Druhá kapitola této práce se zabývá návrhem vlastního řešení pro daný problém, experimenty s tímto řešením a následným srovnáním s řešeními, která byla již vytvořena nebo architekturami, které pro tento specifický problém nebyly primárně určeny. Z tohoto porovnání pak vzniká horní a dolní hranice pro srovnání architektury navržené v této práci.

1.2 Popis řešení

Po úvodním výzkumu je v této bakalářské práci předloženo řešení obsahující dvě uvedené architektury. Toto řešení tedy vytváří architekturu reziduální a kapslové sítě, spojených v jednu tak, aby bylo dosaženo optimálních výsledků.

Architektura samotná obsahuje celkem čtyři hlavní větve, tři reziduální a jednu kapslovou. Popsané rozložení bylo zvoleno z důvodu zvýšení celkové citlivosti architektury na drobné detaily v obraze, kdy ideálně reziduální větev provádí důkladný pohled na obraz, tak aby byla maximalizováno detailní zpracování. Tento přístup je pak podpořeno hustě propojenými vrstvami, které vypočítávají pravděpodobnost již v rámci reziduální větve.

Kapslová síť pak provádí globální pohled na celý obraz, aby dodala potřebný kontext ke konečnému vyhodnocení předložených dat. Tato větev je doplněna o konvoluční bloky pro předzpracování obrazu a hustě propojenými bloky pro výpočet pravděpodobnosti výskytu nemoci v obraze.

Pravděpodobnosti získané z těchto větví jsou posléze spojeny dohromady a představeny finálnímu bloku sítě, který provádí vyhodnocení získaných dat. Toto vyhodnocení provádějí hustě propojené bloky, které vypočítají finální pravděpodobnost existence jedné nebo více ze čtrnácti nemocí v obrázku.

1.3 Závěr

Na začátku této práce byly uvedeny tři otázky vycházející ze zadání.

První z nich se týkala již existujících metod, které byly vyvinuty výzkumníky po celém světě. Ze všech možných řešení a metod byly vybrány dvě, které byly vybrány na základě vynikajících výsledků, které vykazují. Jsou to KGZ Net [23] a ChesXnet[24], jejich výsledky lze vidět v tabulce 5.1. Obě metody si vedly velmi dobře a jejich výsledky se s drobnými rozdíly v podstatě shodují s výsledky prezentovanými původními autory.

Druhá otázka, která byla položena, se týkala možnosti jiných architektur. V návaznosti na tuto otázku byla představena možná architektura kombinující ResNet a CapsNet.

Třetí a poslední otázka se týkala implementace těchto architektur pomocí dostupných frameworků. Pro tuto úlohu byl zvolen TensorFlow, protože se jedná o velmi známý framework pro tyto aplikace v kombinaci s populární knihovnou Keras, která obsahuje několik předtrénovaných sítí. Nicméně i s těmito frameworky se tato úloha ukázala jako velmi náročná. A pro jejich úspěšnou implementaci je třeba pochopit mnoho nových konceptů matematiky, informatiky a programování obecně.

Jak již bylo zmíněno, výsledky již existujících architektur jsou skutečně nejmodernější, coý potvrzuje i tabulka s výsledky 5.1. Z této tabulky je jasně vidět, že jak KGZ Net, tak ChesXnet jsou výrazně lepší než samostatné sítě nebo navrhovaná architektura. Tento výsledek se však dal očekávat, protože obě architektury jsou velmi dobře promyšlené a propracované do posledního detailu. A samozřejmě obě používají více dílčích modulů, z nichž každý má jiný účel pro splnění úlohy klasifikace rentgenových snímků hrudníku.

Přestože prezentovaná architektura nevykazuje příliš slibné výsledky, pouze překonává VGG, stále se jedná o velký úspěch. Vzhledem k tomu, že tato architektura ukazuje, že s dalším časem na výzkum a implementaci lze tuto architekturu úspěšně použít a možná i překonat již existující řešení.

I velmi základní architektura CapsNet v kombinaci s předtrénovaným ResNetem je schopna pracovat s tak velkým a komplexním souborem dat. Z toho plyne nezamýšlený přínos této práce. CapsNet i ve velmi základním stavu dokáže poněkud rozumně pracovat s rozsáhlými a velmi složitými soubory dat.

Vzhledem k těmto výsledkům a výzkumu se domnívám, že existuje velmi velká šance, že by navrhovaná architektura mohla být schopna dosáhnout přesnosti srovnatelné nebo dokonce vyšší než již existující metody. To však může potvrdit nebo vyvrátit pouze následný výzkum.

PIJÁČEK, Štěpán. *Classification of thorax diseases on chest X-ray images using artificial intelligence*. Brno: Brno University of Technology, Faculty of Electrical Engineering and Communication, Department of Telecommunications, 2022, 46 p. Bachelor's Thesis. Advised by Ing. Anzhelika Mezina

Author's Declaration

Author:	Štěpán Pijáček
Author's ID:	223313
Paper type:	Bachelor's Thesis
Academic year:	2021/22
Topic:	Classification of thorax diseases on chest X-ray images using artificial intelligence

I declare that I have written this paper independently, under the guidance of the advisor and using exclusively the technical references and other sources of information cited in the paper and listed in the comprehensive bibliography at the end of the paper.

As the author, I furthermore declare that, with respect to the creation of this paper, I have not infringed any copyright or violated anyone's personal and/or ownership rights. In this context, I am fully aware of the consequences of breaking Regulation § 11 of the Copyright Act No. 121/2000 Coll. of the Czech Republic, as amended, and of any breach of rights related to intellectual property or introduced within amendments to relevant Acts such as the Intellectual Property Act or the Criminal Code, Act No. 40/2009 Coll. of the Czech Republic, Section 2, Head VI, Part 4.

Brno ....

author's signature*

*The author signs only in the printed version.

ACKNOWLEDGEMENT

I would like to thank the advisor of my thesis, Ing. Anzhelika Mezina. for her valuable comments, advises, and most importantly fact that she gave me opportunity to work on this extremely interesting thesis. For that I can not thank her enough.

Contents

List of Figures

Introduction

In recent years information technologies and computing power itself have seen rise in its ability to perform very complex tasks using machine learning. One of the fields where this technological gain, might bring a lot of benefits for everyone is healthcare.

Research of this thesis aims to find potential technologies and implementations of assisted clinical diagnosis, namely diagnosis on x-ray images of chest region.

Chest x-ray exam is one of the most common ways to diagnose patients state. This is because this type of exam is relatively cheap, fast and can show a lot of information. However this method has its own challenges which are hard to overcome. The main issue is potential for miss interpretation of what is seen in the image. This can then lead to miss treatment of the patient, put extra mental pressure on the patient and in return worsen his state even though there might not be a reason for it. On the opposite side there is potential for the patient to not receive adequate care at all, because radiologist did not interpret the data correctly. Thus believing that the patient is healthy, this can in return lead to patients death.

For the reasons stated above, many experiments, method and architectures have been proposed to help with the diagnosis. Those experiments have different levels of success.

The main contribution of this thesis is the research of possible solutions to the problem of classification of the X-Ray images. Based on that research, and possible solution is then proposed. This solution utilizes a novel neural network called a capsule network in combination with residual neural network. This is the second contribution of this thesis, which is utilization of the capsula network on large and complex datasets

This thesis is divided into two main parts. First part is theoretical, in which in depth introduction to the Capsule Network can be found, along side with introduction to the Residual Network and different problems that need to be solved in order for the network to work properly.

Second part of this thesis is about experiment itself. This chapter contains proposed solution to the problem, explanation of the experiment, results and final conclusion with discussion.

With this thesis I would also like to answer some questions. What are the already existing methods for this problem? What are possible architectures? How to implement those solutions using programming language python and framework TensorFlow?

2 Deep learning

2.1 Introduction

Deep learning is a type of Machine learning(ML), that has a goal to imitate the human approach to learning certain tasks [1].

At its core, deep learning is a way to automate predictive analytics on certain data sets, in our case X-Ray images of lungs. This automation of predictive analytics is base for the main difference between traditional artificial intelligence (AI) and deep learning, where AI algorithm is linear in its nature, and deep learning algorithms are using the method of stacking layers upon each other, each layer has an increasing complexity and abstraction. By this hierarchical approach, we are able to create abstract layers based on the knowledge from the previous layer.

2.2 Capsule Neural Network

2.2.1 Description

Capsule Neural Network (CapsNet) one of the latest breakthrough in the field of neural networks architecture. This new architecture is able to achieve better results on Modified National Institute of Standards and Technology (MNIST), database than conventional Convolutional Neural Network (CNN). Error results on this dataset were set to be 0.39% until Caps Net was introduced which achieved a score of 0.25% without any data augmentation [2]. To achieve such performance Caps Net uses concept of capsules as proposed by Hinton [5]. By this method each capsule creates its own output, by combining outputs from all the capsules we are able to create much more stable and precise output.

Caps Net was created and proposed to address two big issues that were introduced with the usage of CNN, which saw an increase in application in recent years. CNN's failure to account for spatial hierarchies between features and lack rotational invariance [5]. Since CNN classifies the test data as the object and disregards any spatial relative features, this approach causes an increase in false positive cases. The second issue is lack of rotational invariance causing the network to assign an incorrect label to the object increasing false negative cases.

Earlier in this section, we have stated the reasons that brought to life Caps Net architecture. That being said, let's turn our attention to the history of Caps Net. The first predecessor of Caps Net was proposed by Geoffrey Hinton in 2000. In the document named Learning to Parse Images [3], Hinton described a new imaging system that would use a combination of segmentation and recognition. This combination would then be put into a single interface process using the parse tree. In the year 2011, Hinton published a second document expanding on this very basic idea of the parse tree [4]. In this document, he introduced the concept of the capsules to CNN architecture. The basic idea is that we add these capsule structures to CNN and use the output from several of those capsules to form a more stable representation of higher capsules. This will create an output vector that can be further processed. After this document, Hinton and his team published a third document on the topic of Caps Net [5]. In this document, Hinton and his team proposed the utilization of dynamic routing between capsules. As stated in the document, there are many ways to implement capsule architecture, but not all of them are as straightforward as proposed by Hinton. This approach is further supported by the utilization of dynamic routing.

2.2.2 Architecture

As an example of Caps Net architecture,architecture proposed in the Dynamic routing between capsules will be used [5].

The CapsNet can be divided into two main parts, encoder, and decoder. Each utilizes different layers and has a different and important role. The encoder can be found within the first three layers of the architecture. The decoder then on the last three layers in the architecture.

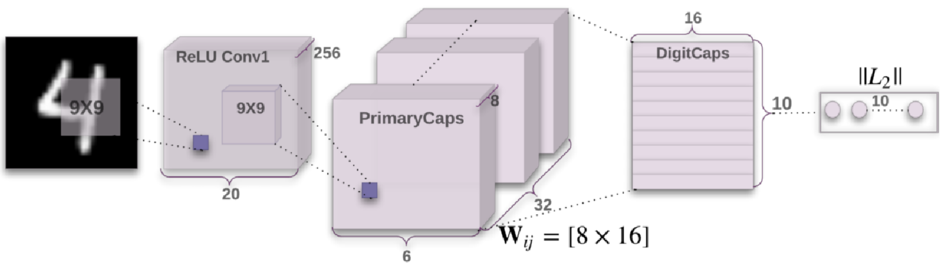

Fig. 2.1: basic Caps Net architecture. Source [5]

Capsule

To describe the architecture of the CapsNet, explanation of the capsules need to be done. This will be done by using first document in which capsules were introduced [4].

In this paper, Hinton and his team proposed a novel idea to solve one of the issues of standard networks. Instead of aiming for the viewpoint invariance in the neurons that use a single scalar output to summarize the activation. Networks should instead use local capsules to perform complex internal computation on the inputs and then enclose this output into a small yet information-dense vector. By this method, capsules learn to identify a specific entity in the image over a limited space.

This is in contrary to the way that the CNN performs this task. The CNN, as the name suggests, uses a convolution layer that takes weights from the kernel, and replicates them over the entire input, and then outputs a 2D matrix. After this process, all 2D matricies are taken and stacked on top of each other to produce the output of the convolution layer. From stack of this matrices the viewpoint invariance needs to be achieved. This is done by introducing the max-pooling layer into our network. This pooling layer then looks through regions in the 2D matrix and selects the largest number in each region. By this, the desired behaviour is achieved but in return, the details in the images are lost, and the pooling layer leads to the loss of information about the spatial relationship. This aproach is not very ideal. Therefore we should instead use capsules that take that important information and encapsulate them into the output vector.

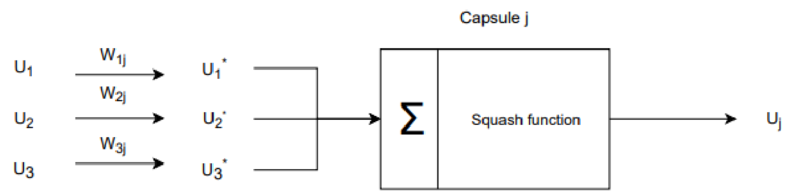

Fig. 2.2: Internal process in capsule.

To summarize, it can be said that the capsule is somewhat similar to the neurons in its design, but expands it into vector form to allow better and more precise representational capabilities.

Loss function

As stated in the previous section, the output of the digit capsule is ten sixteen-dimensional vectors, which represent weather the entity exists or not.

For each of those vector loss value will be calculated according to the formula. This calculation is performed for all the digit caps L_k:

$$L_k = T_k max(0, m^+ - ||v_k||)^2 + \lambda(1 - T_k) max(0, ||v_k|| - m^-)^2 \tag{2.1}$$

Where $T_k = 1$ if a entity of the class k is present and $m^+ = 0.9$ and $m^- = 0.1$. λ down weighting of the loss for absent entity classes stops the initial learning from lowering the lengths of the activity vectors. Value of the $\lambda = 0.5$ [5].

Training of the CapsNet

The CapsNet is supervised learning. This means that the training group has data correctly labeled.

The correct label in the vector also changes the value of the T_k in the loss function. This means that the correct label changes the value of the T_k to one if the label corresponds to the DigitCaps and zero otherwise.

For better understanding of the training process, the task of identification of the hand written numbers from MNIST dataset will be used.

Assume that the number given on the input is 1, with that we can tell the first capsule of DigitCap is responsible for correctly labelling the input. That also tells us that for the first capsule T_k will be 1 and for the rest, it is 0. Now we can calculate the loss, in order to do that we take the output vector from our capsule and subtract it from m^+. After the calculation is done there needs to be decision, if the results should be kept or not if we want to keep the result or not. This is done based on the simple is a statement, if the result is bigger than zero it is kept and squared, otherwise, the zero is than returned.

From the picture above we can see that if the mismatching of labels occurs the loss will be zero.

Encoder

In this example of Caps Net architecture, the encoder takes an MNIST digit image which has 28×28 px size and learns to encode it into a 16-dimensional vector of instantiation parameters. The network in this stage is 10-dimensional vectors that have the length of the digit capsule outputs.

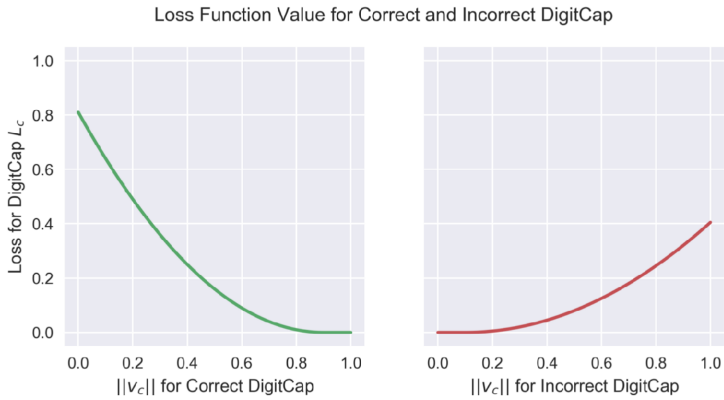

Fig. 2.3: Loss function output for correct and incorrect outputs [8]

Layer 1. Convolution layer

The convolution layer is used to detect basic features in the 2D images. This layer takes 28×28 px images as an input and turns them into 20×20×256 tensor output.

In the Caps Net architecture the convolution layer has 256 kernels which each has a size of 9×9×1 and one stride. This can be seen in the image of the Caps Net architecture2.1. Where the image is divided into 9×9×1 sections, which then are individually scanned for basic features.

This section is then followed by the Rectified Linear Unit (ReLU) [6]. That is a linear activation function. This function does a simple calculation if the input is less than zero, the entire input is then returned as zero. And vice versa, where if the input is greater than zero then input is returned. This can be seen on the graph of the ReLU function, where activation doesn't occur until input is bigger than zero.

We can also simply define this function mathematically using max function

$$\texttt{ReLu}(x) = max(0, x) \tag{2.2}$$

This behaviour combined with higher sensitivity to the sum input and not being easily saturated makes it a great activation function for Deep learning.

Layer 2. Primary capsule layer

The primary capsule layer consists of 32 primary capsules, their purpose in the Caps Net is to take basic features from the first layer and produce combinations of the features. The 32 capsules are similar to the convolution layer.

That being said, each capsule applies 9×9×256 convolutional kernels with stride 2 to the input of the size 20×20×256 and therefore produces a 6x6x8 output tensor.

And as mentioned earlier in this section, there are 32 such capsules therefore output tensor is 6×6×8×32. And as stated earlier in the description of the Caps Net, multiple outputs give us an advantage of the stable output.

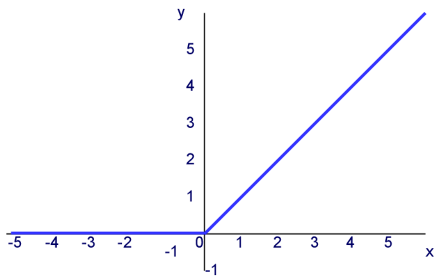

Fig. 2.4: ReLU function [7]

Layer 3. Digit capsule layer

In this architecture, this digit layer has 10 capsules corresponding to the 10 possible digits that might occur in the MNIST dataset. Every one of the capsules takes as an input a 6×6×8×32 tensor. This tensor can be also described as a 6×6×32 8-dimensional vector.

As explained earlier in the section explaining capsules 2.2.2, each of those input vectors is assigned an 8x16 weight matrix that translates 8-dimensional input space into the 16-dimensional capsule output. That gives us a 16×10 matrix as output. This output gives 1152 matrices for each capsule and the same number of c and b coefficients for dynamic routing.

Decoder

Task of the decoder is to take a 16-dimensional vector that came from the previous layer and learn to decode it into an image of the digit. What is worth mentioning, is that the decoder only uses the correct Digit cap vector during the training and ignores the incorrect one. From which we can see that every incorrect output from the DigitCaps is masked with zero and ignored. After this processing of the vector is done, we move to other parts of the network, those are 3 fully connected layers.

In the Caps net, the decoder is used as a regularising layer. The correct vector from the previous layer is used as input and from that decoder learns to recreate 28×28 image that was at the beginning of the entire process. With loss function included and being the Euclidian distance between reconstructed and original image. Decoder forces capsule to learn features that can be used for reconstruction purposes later on. From this sentence we can derive very simple logic, the closer the reconstructed image is to the original the better.

And as stated earlier, the decoder consists of the three last layers of the network. This can be seen in the image below. As we can see the Decoder is composed of three fully connected layers with two using ReLu as the activation function and the last one using a sigmoid function.

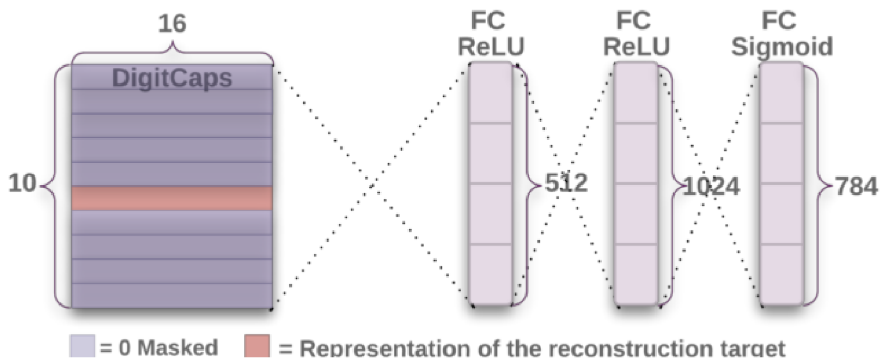

Fig. 2.5: Decoder layers [5]

Layer 4., 5. and 6. Fully connected layers

Fully connected layer is a type of layer in neural network where every input from one layer is connected to the activation unit of the next layer. Fully connected layers can be found in the CNN and Caps Net, to mention a few of the architecture that uses them. Those fully connected layers are usually the last layers in the architecture that compiles extracted data by previous layers to form the final output of the network.

In this architecture, the Fully connected layer fills the same role as described earlier. The layer takes extracted data from the encoder and compiles them or decodes them into the output of our Caps Net. What is also worth mentioning in regards to the last layer, fourth and fifth layer uses as the activation function ReLU 2.2.2.

The last layer is very much the same as the previous two but instead of the ReLU the sigmoid function is used. The Sigmoid is classified as a logistic function. Which means that whatever is inputted into the function is transformed into the output in range from one to zero. This is useful for application in the last layer of the architecture where there is a need for probability of the digit existing in the picture.

2.2.3 Comaprison of CapsNet and CNN

As stated earlier, CNN has multiple issues that have risen with its extended use. To name the main that has been presented so far is max-pooling which is causing a loss of details and inability to retain the spatial relationship between the features and a few more. From those problems has risen the need for a better network, that will overcome those issues, and thus Caps Net has been created.

Research done by the team at the Malaysian Multimedia University[9] shows that there is not much difference between Caps Net and CNN in the way of spatial relationship. Both networks are capable of performing such tasks, and counterintuitively CNN even outperforms Caps Net, even though theoretically it should be another way around. And as stated earlier in this thesis 2.2.1, Caps Net outperforms most neural networks on MNIST dataset, so the outcome of the research[9] is very surprising. But as stated by the researchers themselves, Caps Net is a very novel idea that has not been researched well enough to draw any final conclusions. Datasets that have been used in the research are very simple and therefore it is possible that the Caps Net can outperform CNN on much more complex datasets.

2.3 Residual Neural Network

2.3.1 Description

The residual neural network (ResNet) is neural network based on the construct of the pyramidal cell in the cerebral cortex. This is done by utilizing skip connections or shortcuts to skip some layers.

Popularity of the ResNet comes from the framework that the ResNet presented, that made it possible to train deep neural networks more effectively with less issues. This means that the network can contain hundreds or thousands of layers and still achieve great performance.

Typical ResNet is implemented with double or triple layer skips that contain ReLU function 2.4 and batch normalization[10]. We may also include weight matrix to learn the skip weights, which helps network to learn the most optimal skips between the layers. ResNets equipped with such matrices are known as HighwayNets. Parallel skips can be also used to further improve the output, networks which use this are known as DenseNets.

2.3.2 Biological analogy

As mentioned earlier, ResNet is based around the construct of pyramidical cells found in the cerebral cortex. This can be translated into analogy where cortical layer 6 gets input from the layer 1, skipping intermediary layers in the process [12].

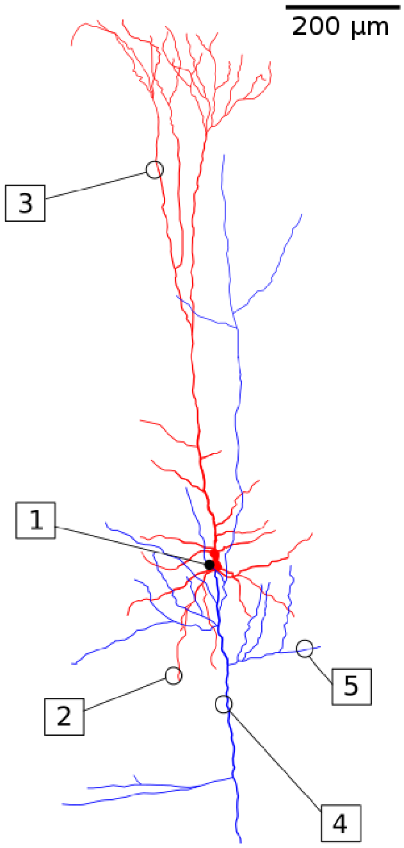

Fig. 2.6: Reconstruction of the pyramidal cell. Soma and dendrites are red, axon arbor in blue. (1) Soma, (2) Basal dendrite, (3) Apical dendrite, (4) Axon, (5) Collateral axon. Source [11]

In the figure 2.6, this behaviour compares to the apical dendrite skipping over layers, while the basal dendrite collects signal from the same layer or previous one [13]. This statement is rather simplified since there is a research suggesting that there are more complex structures.

There is uncertainty about how many layers are built on this construct in cerebral cortex or if all the layers exhibit the same behaviour, but if we look at the cerebral cortex in global scope, it does exhibits similar behaviour to the ResNet.

However there is no evidence of global teaching signal, or some sort of backpropagation being present in the brain, or iterative optimization, nothing of this sort has been proven by neurophysiological evidence in human or animal brain.

2.3.3 Architecture

As an introductory example the ResNet50 architecture is chosen since it is one of the most popular architectures of the type. This model is consisting of 48 convolutional layers 2.2.2, one Max Pooling layer and one Averaging layer.

Skip connections

Skip connections were first introduced with the ResNet and is the key part of the architecture. In simple words skip connection includes the input in the output of the next convolution layer. Layer that will include this input is decided based on the number of allowed skips, which can be two or three skips.

Skip connections in the ResNet are based around very simple idea, use identity function to backpropagate through by just using vector addition. By this process we are able to mitigate and, in some cases, even avoid problem of the vanishing gradient. Because gradient at the earlier layers is multiplied by one, and therefore its value will be maintained in the earlier layers. This was huge problem since until skip connection gradient of the current layer would be multiplied by the gradient of the previous one and therefore would fall to extremely low numbers, and never reach desired value.

In ResNet we stack building blocks called skip residual blocks, as seen in the image below 2.7, to form the network.

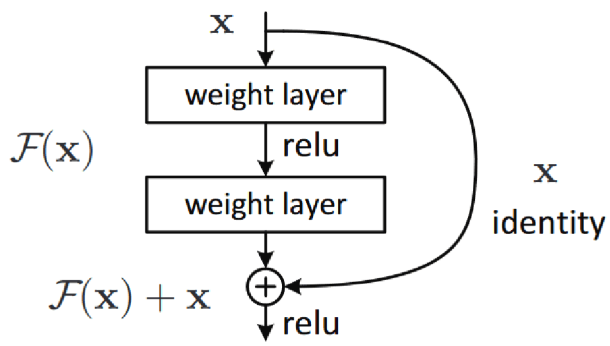

Fig. 2.7: Skip residual block. Source [10]

Layers of the ResNet50

First step has a kernel size of $7 \times 7, 64$ different kernels all with stride of size two giving us one layer.

Second step consists of $1 \times 1, 64$ kernel that is followed by $3 \times 3, 64$ kernel and one $1 \times 1, 256$ kernel. Together these three layers are repeated three times in total giving us nine layers.

Third step consists of $1 \times 1, 128$ kernel, $3 \times 3, 128$ kernel, and one $1 \times 1, 512 kernel$. This configuration is repeated four times giving us 12 layers in this step.

Fourth step consists of $1 \times 1, 256$ kernel, $3 \times 3, 256$ kernel, and one $1 \times 1, 1024$. This configuration is repeated 6 times giving us 18 layers in total.

Fifth step consist of $1 \times 1, 512$ kernel, $3 \times 3, 512$ kernel, and one $1 \times 1, 2048$. This configuration is repeated three times giving us nine layers in total.

And final step in which we do average pooling layer and end it with fully connected layer containing 1000 nodes. Giving us 1 layer.

In total we are given 50 layers of deep convolutional network.

2.4 Medical application

Advances in the computation over the years and massive data generated by the healthcare system created many new problems that are great for AI application. This can be seen in the rise of new systems, architectures and frameworks for medical use.

Prime example of this new era can be seen in this thesis or many other research papers looking into possibility of using AI in process of clinical diagnosis. Idea of assisted diagnosis is to give doctor second opinion on the matter. Because humans are not perfect and might run into some issues, uncertainty, being tired and many other problems. Which can put patient into uncomfortable position of being miss treated or not treated in time, or worse not treated at all. This is not to say that the AI is perfect in this area, but the research proved that it can be done. There are two great examples of implementation of AI in assisted diagnosis.

First is Deep Learning Based Automation Detection (DLAD)[14] created in 2018 by the researchers at Seoul University that can analyse chest radiographs and detect abnormal cell growth that might cause cancer. When put to the testing, algorithm was able to outperform 17 out of 18 doctors in the diagnosis[14].

Second algorithm was created by the Google AI Healthcare in the year 2018. Algorithm named Lymph Node assistant (LYNA)[15] can identify metastatic breast cancer from lymph node biopsies, this algorithm proved to be so advanced that it could detect suspicious regions, normally undistinguishable for the human eye. When put to the test, algorithm shown 99% accuracy[15] in identifying cancerous and noncancerous samples. Second test of this algorithm was done in combination with doctors that would use LYNA for assistance. When LYNA was used, average review time of the sample was halved.

These are just two examples of AI in assisted diagnosis, which shows the potential and worth of researching it. But there are still few issues that are holding back full implementation of this technology in the practical settings.

First being the trust issue between doctor and the AI, and of course between patient and the AI. This key issue is somewhat understandable since there is the doctor that needs to put aside years of his study in order to trust computer. And patients' mistrust to the computer, since the computer is about to decide if he gets the treatment or not.

Second issue is legal matter. There is still no straightforward guideline about implementation of the AI in medical field. Even though the U.S. Food and Drug Administrative has approved two assistive algorithms. Currently every new drug and technology aimed at healthcare needs to go through intense trial process before being even allowed to test on the patients. This process also requires extensive transparency which is not always easy to provide in the context of AI. Many algorithms rely on very complex math that is done inside. Thus making it very hard to explain, some are even referred to as "black box" since nobody can clearly explain inner workings that got us from the input to the output. And then there is an issue of money, since by giving complete transparency during the trials also means exposing proprietary methods or algorithms to the world. In which case algorithm or method might get stolen, improved upon and original creator might lose gains from his invention.

3 Classification of lung diseases in X-Ray images

This chapter shows possible technological challenges in the are of X-Ray and image classification.

This includes image recognition, image segmentation, multi-label classification, and architectures that are used in this field.

3.1 Image segmentation

Image segmentation plays key part in the application of AI in the medical field. This is due to the fact that image segmentation automatizes selection of regions of interest and anatomical structures[16]. Thus making the process of analyzation of the image or the x-ray much easier and faster.

Image segmentation is the process of dividing image into multiple smaller segments. This is done to simplify the representation of an image into segments that are more meaningful and easier to analyse. In simple words, image segmentation is the process of assigning labels to every pixel. Labels are assigned to pixels based on the features that they are sharing[16].

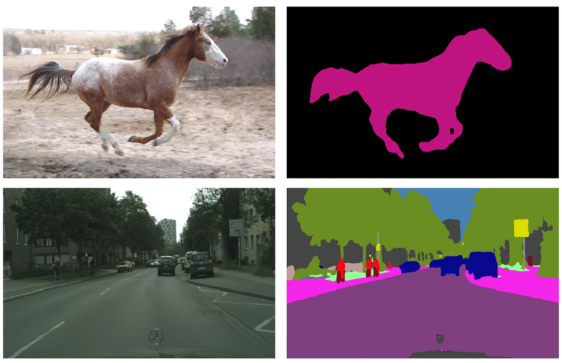

Fig. 3.1: Example of image segmentation. Source [17]

3.1.1 Conventional methods

This subsection introduces three conventional methods of the image segmentation.

Thresholding method. The simplest method of image segmentation. It is based on a threshold value to turn grey-scale image into a binary image[16]. As said earlier this method is very simple yet very effective in the instances where image has multiple distinctive features that can be quantified.

As an example, we can use picture above 3.1. There are two images. On each image we can clearly see very distinctive features. Each feature corresponding to one threshold mode. Threshold method attempts to determine intensity value, called threshold, which separates the desired features. Then the segmentation is achieved by grouping pixels with intensities greater than threshold values into one class.

Region growing. Method of extracting region from the image that is connected based on some predefined criteria[16]. These criteria might be intensity, edges in the image, or both.

In the simple words, this method operation depends on the seep point which is picked manually. Seed point is extremely important for this method. Because all pixels of the region are extracted based on the combination of the seed point and criteria.

Similarly, to the thresholding method, this one is used in the scope of image processing unit. Where it is used to locate simple structures, like tumours and lesions.

Main disadvantage of this method is requirement of manual interaction[16]. Seed point needs to be selected for every region of interest for it to properly work.

Clustering segmentation. Method that performs pattern recognition technique that seeks to partition a feature space derived from the image. Which is similar to the classifier method, Classifier method is pattern recognition technique that seeks to partition a feature space. A feature space is the range space of the image, most common being the image intensity. But Clustering does not need any training data, thus making it unsupervised method[16].

To make for the lack of training data, clustering method is iteratively changing between segmentation and characterizing the properties of each class. This means that the clustering method is using available data to train itself.

As stated earlier in this paragraph, clustering method does not require any training data. But it does require initial segmentation of the image. Meaning that it can ideally, operate within the scope of image processing unit that has thresholding or region growing method before the clustering one.

Disadvantages of this method is that it cannot operate on its own, since it needs to have pre-processed data. Second issue is that the clustering method has no regards for the space modelling, therefore making it sensitive for the noise in the image[16]. This means that it is not exactly suited for x-ray image segmentation. But since method has no regard for the space modelling it also brings one advantage to it, it has very low computation requirements.

3.2 Multi-label classification

Multi-label classification is the problem of supervised learning, it is closely related to the problem of multi-output classification. This type of classification has seen rise since it is applicable for wide range of problems. For example, text classification [31], scene and video classification [30] and bioinformatics [31]. Main idea of this problem is that one instance may be associated with multiple labels. Another problem linked to the multi-label classification is that there are no constraints on how many labels can be assigned to the instance.

A common approach to this problem is to perform problem transformation, where multi-label problem is transformed into one or more single-label problems. This single-label classifier is used, and its calculation of predictions are transformed to the multi-label predictions. Problem of this approach is scalability and flexibility.

Transformation into binary classification problems. This method has been recently overlooked. But as research [18] shows, binary classification is very viable option.

Basic idea of this method is to create one binary classifier for each label. With those classifiers created combined model is able to predict labels for unseen datasets. This binary method might look similar to the One-vs-All method but it is different. Because binary classifier deals with only one label and does not account for the relationship with another label.

There are two main advantages of this method. First is resistance to the overfitting of the label combinations. This is accomplished by the fact that the classifier does not expect any association with the previous label. This spans from the relationship of one to one, one binary classifier to one label. Second advantage of this method is low computational complexity.

Transformation into multi-class classification problem. Another option to solve the problem of multi-label classification is to transform problem into the multi-class problem. This method is known as the label powerset[19]. Key idea for this method, is that combination of the labels results as one class value in one corresponding multi-class dataset.

Opposite to the previous method of binary classifiers, this method takes into the account other label relationship.

This method has one main issue. Class values in the multi-class dataset may be associated with a very small number of instances. Thus making the dataset unbalanced.

3.3 NN architectures for classification of X-Ray

Two architectures are introduced as a possible solution to the problem of the classification of X-Ray images.

3.3.1 Architectures

KGZ Net

First architecture is Knowledge-guided Deep Zoom Neural Network (KGZNet)[23]. As stated by the researcher, this architecture is leveraging previous medical knowledge to guide its training process.

This architecture is based on three classification sub-networks. Those are global, lung and lesion sub networks.

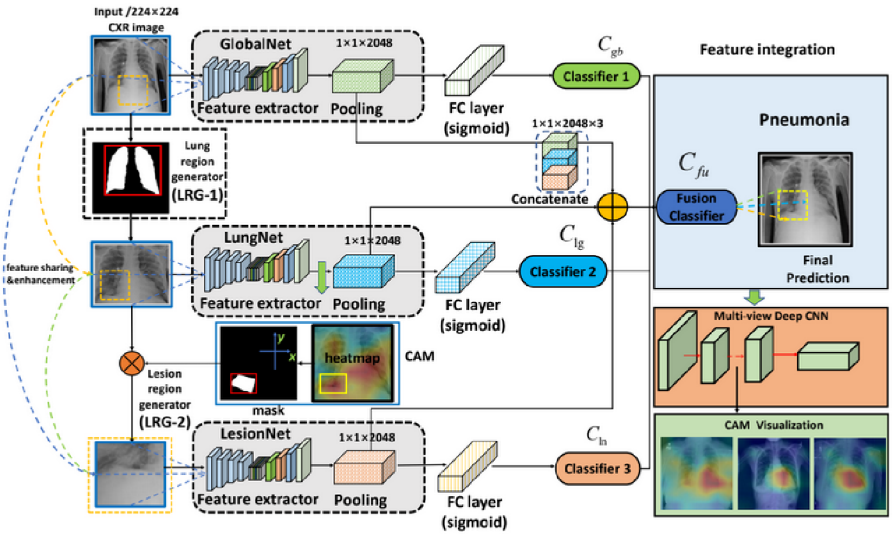

Fig. 3.2: Architecture of the KGZ Net. Source [23]

As mentioned earlier there are multiple components that make KGZ Net architecture. First, we have ResNet sub-networks. From those sub-networks we take the output that is provided on the pooling layers of the sub-networks. That pooling output is then taken and connected into one that is provided to the CNN network, which then makes final classification.

One of the interesting features of this architecture is ability of self-improvement between layers. Because output of the pooling layer is then transferred into fourteen neuron big fully connected layer. Output of the fully connected layer is then shared across all other sub-networks to improve and learn.

This architecture has one main disadvantage, computation power required for the training.

CheXNet

Second architecture that has been chosen is CheXNet [24]. This architecture is very different in comparison to the previous one. It utilizes 121-layer Dense CNN (DenseNet), to detect and classify fourteen common thoracic diseases.

At the beginning CheXNet was meant to detect only pneumonia[24], but later was expanded to detect all diseases from the dataset ChestX-ray14.

As mentioned earlier architecture is based on 121-layer DenseNet, that improves flow of information and gradient [24]. Nature of DenseNet allows even 121-layer large network, to be optimized [24]. Researchers have made one adjustment to the DenseNet. The last layer of the network, was replaced by the layer that has only one output, and on top of that sigmoid function is applied.

4 Experiment description and architecture

In this chapter of thesis, I would like to talk about experiments, that were performed. Those are, experiments with other possible solutions mainly KGZ Net 3.3.1 and ChesXNet 3.3.1, experiments with other neural networks, like ResNet, DenseNet and Vanilla CNN network, and finally experiments with my own architecture, which combines CapsNet 2.2.2 and ResNet 2.3.

Since both KGZ Net and ChesXNet have been explained in their respective subsections they will be skipped for now. What is worth mentioning in the case of both architectures is hardware that was used for their training. Since both architecture were already tested therefore training and validation were skipped. Hardware used for their training:

- Processor: Intel I7-1165G7
- Operational memory: 16GB RAM
- Graphical card: NVIDIA Quadro T500 with 4 GB of memory

For the training and validation of other networks, dedicated server was used with following parameters:

- Processor: 48 × AMD EPYC 7272 12-core processor
- Operational memory: 198 GB RAM
- Graphical card: NVIDIA Tesla V100S with 32,5GB of memory

To maintain stability and to give all the architecture high number of data to train on, the CXR14 data-set[25] has been chosen. This data-set is also one of the prerequisites for successful execution of the training.

As a framework for my experiment Tensorflow and Keras have been chosen, since they offer many user friendly features and also prebuild neural networks, which in return allows faster experimenting. To allow architectures to maximally utilise available GPU performance, CUDA toolkit from NVIDIA in combination with NVIDIA CUDnn library was used.

Competitors architectures however are written using different very popular framework, which is PyTorch. But they were also trained on the GPU using CUDA toolkit and CUDnn library.

For validation of all the network the Area Under the receiver operating characteristic curve (AUC) was chosen. This choice has one main benefit to any other performance measurement. In case of classification for multiple classes neural network or architecture could become biased towards some classes. This means that accuracy for some classes can reach up to 90% and for other as low as 50% this happens because network is positively or negatively biased towards some classes thus making the accuracy value useless as a measurement unit. Therefore AUC value can be used instead to give better unbiased measurements of performance.

4.1 Data-set CXR14

With the rise of interest in the thoracic diseases classification, the need for better and larger data-sets rose as well. That is why the CXR14 has been introduced. This chestX-ray dataset is one of the largest data-sets available, it consists 112 120 frontal view images of 30 805 unique patients.

The CXR14 was introduced in 2017 alongside with the document about Hospital-scale Chest X-ray Database and Benchmarks [25]. Need for this data-set was also proven in this document. In which researchers performed series of different tests on architectures. Those architectures could not perform as well as they could because of lack of large scale data-sets.

At first CXR14 had 8 most common diseases, those are:

- Atelectasis
- Consolidation
- Infiltration
- Pneumothorax
- Edema
- Emphysema
- Fibrosis
- Effusion

Later those eight were expanded to fourteen, which means six more were added, those are:

- Pneumonia
- Pleural_thickening
- Infiltration
- Cardiomegaly
- Nodule
- Mass
- Hernia

Example of data that are in the datas-set can be found in image down bellow.

4.2 Experiments with competitions architectures

In this part of the thesis, experiments with already existing architectures were conducted.

Those experiments consisted of training and evaluation of the architectures.

In case of ChesXNet training part was skipped, because it already has pre-trained values in the git hub repository (`https://github.com/arnoweng/CheXNet`). Which made the process faster.

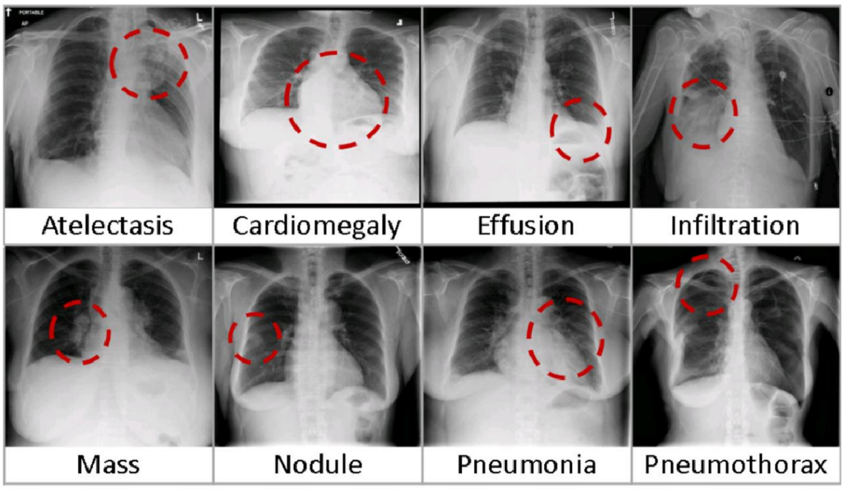

Fig. 4.1: Eight common diseases found in the CXR14 data-set. Source [25]

In case of KGZ Net, both steps were performed. Again with the CXR14 data-set.

4.3 Experiment CapsNet and ResNet architecture and other networks

In this section of thesis I am going to introduce experiments that were performed. First one is experiment with my own architecture which combines CapsNet and ResNet, second experiment is with neural networks provided in the Keras library:

- DenseNet121 from Keras library
- ResNet101V2 from Keras library
- VGG16 from Keras library

4.3.1 Frameworks used in the experiment

As stated, multiple times in this thesis, the deep learning has seen a boom in its research and usage across numerous fields of science and technology. Inevitably that also brings new ideas, algorithms, and complex structures, but also many complications in the implementation. That's why frameworks and libraries have been created to easy the implementation.

Frameworks and libraries do not just bring easier implementations but also better optimization, faster development and deployment.

Frameworks

TensorFlow is the example of those frameworks. TensorFlow is very popular library for deep learning applications. It was created by the Google researchers and engineers[20].

This library was designed for complex numerical computations that are for example performed in the neural networks. Another design choice that made TensorFlow popular option is that it can run numerical operations on both CPU and GPU. And the last design choice, that made TensorFlow popular is its multilanguage abilities, it has a permutation for JAVA, python, C# and many more languages.

PyTorch is another popular choice for the deep learning is PyTorch. PyTorch is library based on Torch, this library is primarily developed by Facebook AI research lab[21].

In comparison to the TensorFlow, PyTorch has a primary language, which is python, that has very polished base, but it also supports C++[21].

PyTorch provides two high-level features, which is tensor computing accelerated by the GPU and deep neural network built on typed-base automatic differentiation system.

Examples of PyTorch usage include Tesla Autopilot, Uber's Pyro and Catalyst.

Keras is second example of library that is used for deep learning application. It is high level Application Programming interface (API) for python that can run on TensorFlow, it used to run o multiple backends such as Cognitive Toolkit and Theano but since version 2.3 of Keras, only TensorFlow is supported[22].

Keras aims to provide easy and fast experimentation with deep neural networks. This is achieved by including many basic building blocks of deep neural networks in the library.

Another huge advantages of Keras is that it provides ability to split training in to the Graphical Processing Unit (GPU) and Tensor Processing Unit (TPU) clusters.

4.3.2 My own architecture

For this part of the experiment I have chosen architecture based on previously mentioned CapsNet in section 2.2.2 and ResNet in section2.3. Combination of these architectures was chosen for their relatively promising results when compared to other networks.

CapsNet

This architecture is chosen as one of the last parts of the architecture for its ability to maintain spatial relationship between features within the picture or data provided by higher layers of the architectures. This approach is inspired by the KGZ Net architecture 3.3.1 which shows great results. Of course this cannot solely be contributed to the architecture, other parameters play important part however architecture and the layout of different neural networks is extremely important.

Implementation of the CapsNet for this solution has been inspired by the implementation of the user sulaimanvesal (`https://github.com/sulaimanvesal/CapsNet`)

ResNet

This architecture is a main portion of the architecture, occupying all the modules in some shape or form. This neural network was chosen for its great ability to allows multiple layers to work together without any mayor issues thanks to its residual connections and ability to skip layers if deemed to be beneficially for the network. This network was chosen based on two factors, first is inspiration by the KGZ Net 3.3.1 which also uses power of the ResNet, second factor was further research which proved great results of the ResNet. To make development of the architecture easier and faster the prebuild neural networks from the Keras 4.3.1 library were chosen.

For optimizer function, Adaptive Moment Estimation (ADAM) was chosen. As it is very well known, and very well performing optimizer. Another improvement over regular CapsNet is that the Asymmetric Loss For Multi-Label Classification [26] was chosen. This function was used for its promising results in the applications that use multi-label classification.

Training

The training of this architecture was performed on images of size 384×384 px. This size was chosen as a compromise between image quality and performance impact since higher resolutions did not prove to have that big of an impact on overall performance of the architecture but maid significant impact in terms of time per each epoch.

As mentioned earlier full data-set was used for the training. This data-set was than splitted into two sub sets. One intended for the training and one for the validation of the model. Furthermore those sub-sets were splitted into batches. For the training sub-set batch size of eight was chosen. For the validation sub-set size of eight was chosen. Batch size was chosen based on the greatest common divider of the splitted data-set.

For the training process the decision was maid that the 100 epochs will be sufficient for the training. Each epoch is divided into 500. Validation process which follows after training is divided into the value that is gained from the number of validation images divided by eight. Which gives roughly 670 validation steps per epoch.

Validation

Validation was performed in a similar manner to the training. Except in this case entire data-set is used for the process. From the validation set, prediction is calculated using built in TensorFlow method. To this method, two overloads are provided. Data-set, and batch size of four.

Prediction is than used to help calculate other parameters used for the validation and analysis of models capabilities.

Final architecture

As mentioned earlier CapsNet and ResNet were chosen for the final proposed solution to the problem of classification. Architecture is mainly composed of ResNet in different configuration of the layers, ranging from 50 to 152, and CapsNet with one primary capsule, two digit capsule and decoder.

This combination on it's own is not very capable therefore multiple auxiliary layers and blocks are need for the best possible results. Those parts are mainly composed of convolutional layers and dense blocks, all with different parameters and activation functions to ensure best possible results.

This can be seen in the diagram above which roughly describes main parts of the architecture. There are three ResNet branches each with its own dense blocks to maximize accuracy of the provided output. Last branch which is CapsNet is the largest, composed of multiple convolutional layers and dense blocks to ensure maximal possible image preprocessing and give the core capsules best possible data to work with.

Processed data from all the branches are then concatenated and passed on to final dense blocks which perform final steps before giving the final results.

4.3.3 Experiments with other architectures

For this part of the experiment three other architectures were chosen 4.3 to give better idea of how other architecture perform on this problem. Main reason for picking those architectures is their interesting approach to the problem of multi layer neural networks.

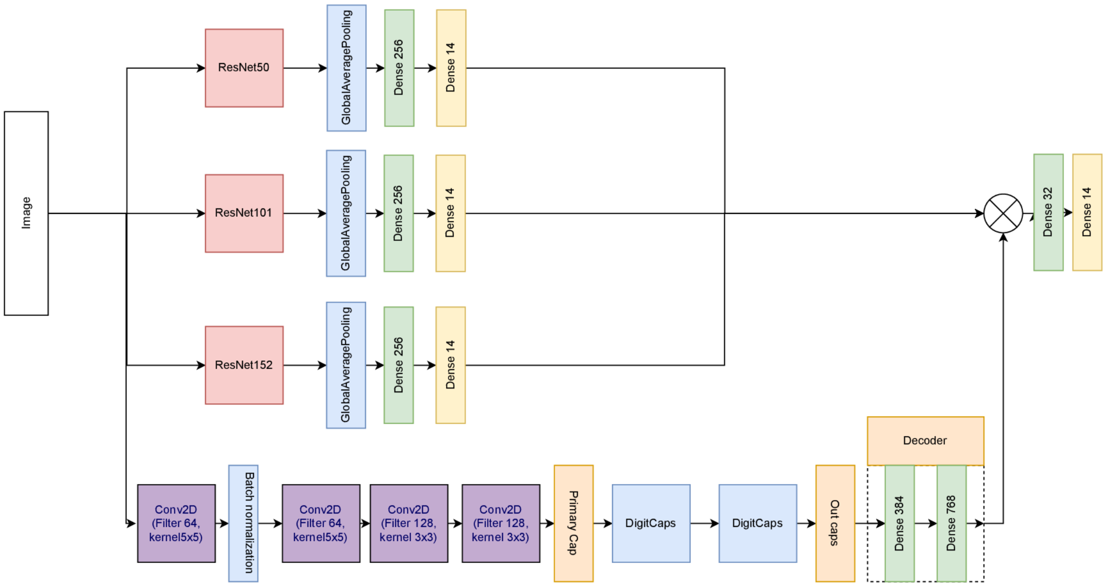

Fig. 4.2: Diagram of architecture used in the experiment.

DenseNet

This is a first of the three neural networks chosen as comparison to my own architecture. DensNet architecture, as any other architecture used in this thesis is basically multilayered CNN which uses different types of connections and algorithms to achieve better performance and mitigate the problem of multilayered solutions. This network utilizes dense connection and dense blocks to connect all layers with matching feature maps directly to each other[27]. This in return allows efficient work and great results even in the case of hundred or more layers.

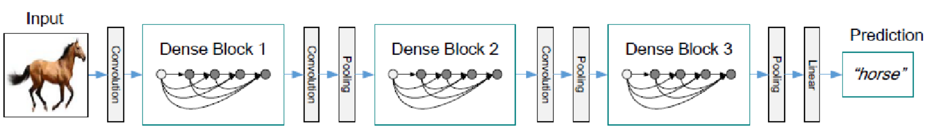

Fig. 4.3: Example of Dense Net architecture. Source [27]

ResNet

In this part of experiment the sole ResNet was chosen as comparison to my own architecture. ResNet was chosen because previous experiments showed that standalone ResNet performs very well on this type of data.

VGG16

In the last part of the experiment VGG16 was chosen to set the minimum baseline for my own architecture. Which is another architecture based on CNN. This architecture's main aim is improving AlexNet by changing kernel size in the first and second layer of the network [28].

This architecture was chosen for its promising results in classification of the ImageNet data-set which contains nearly fourteen million images with over one thousand classes. Therefore assumption was made that this architecture could show promising results on the CXR14 data-set.

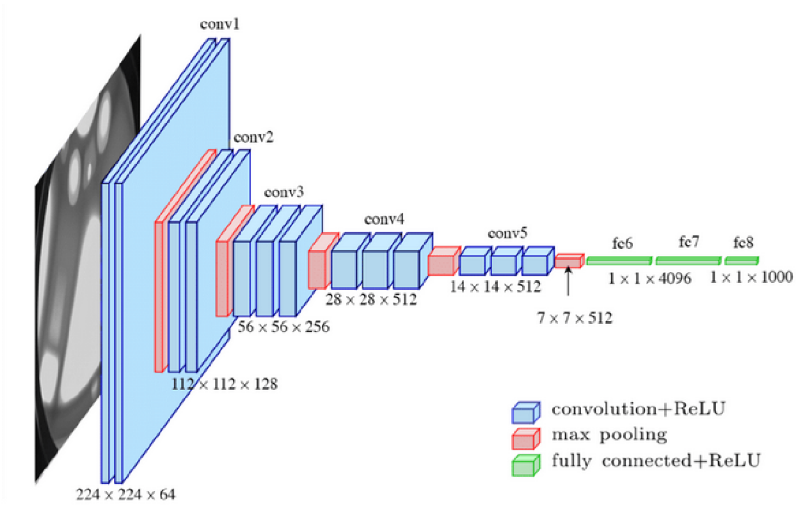

Fig. 4.4: Example of VGG16 architecture. Source [29]

5 Thesis Results and Conclusion

5.1 Results

Values presented are, as mentioned earlier, AUC values. Those values were collected from all the networks on which experiments were performed. I have decided to use AUC value since it is value used by all the documents to measure success of the architecture.

	KGZ Net	ChesXNet	DenseNet	ResNet	VGG16	My arch
Atelectesis	0.839	0.809	0.732	0.744	0.601	0.506
Consolidation	0.843	0.790	0.663	0.666	0.618	0.571
Infiltration	0.760	0.733	0.679	0.676	0.552	0.587
Pneumothorax	0.776	0.888	0.844	0.845	0.463	0.529
Edema	0.918	0.886	0.820	0.812	0.694	0.644
Emphysema	0.9350	0.937	0.831	0.808	0.425	0.439
Fibrosis	0.835	0.803	0.792	0.798	0.522	0.649
Effusion	0.819	0.863	0.788	0.794	0.625	0.537
Pneumonia	0.916	0.768	0.663	0.659	0.579	0.559
Pleural thickening	0.846	0.806	0.701	0.718	0.513	0.482
Cardiomegaly	0.938	0.924	0.851	0.851	0.555	0.535
Nodule	0.830	0.780	0.730	0.742	0.556	0.570
Mass	0.894	0.867	0.779	0.786	0.462	0.531
Hernia	0.910	0.916	0.854	0.622	0.455	0.447

Tab. 5.1: Table of results. Source author

As we can see from the table 5.1, KGZ Net is performing best out of all networks. Only cases where KGZ Net was outperformed by the competition was in the case of Pneumothorax, Emphysema, and Hernia. In those cases KGZ Net was beaten by the ChesXNet, in the case of Pneumothorax the difference between two networks is quite significant. In the second case of Emphysema, the difference is pretty small, and in my opinion insignificant in the scope of the results, same can be said about the third case.

Other tested networks that were intended to set the baseline for further experiments show interesting results, namely ResNet which outperforms DenseNet, VGG, and my architecture in most of the cases. This result was expected from the DenseNet and VGG since the previous experiment showed that ResNet performs very well on this dataset.

The last column presents the results of my architecture which managed to only overcome VGG, and that happened in only a few cases. This result was somewhat expected since the implementation of the CapsNet is not ideal. Another issue that became visible during the training is the problem of bottlenecking, this issue is not easy to fix since the value of the parameters and number of layers can not be increased indefinitely for two main reasons. The first one is the problem of vanishing gradient which was present during the entire process of experimentation with my architecture and the second issue is the limitation of the computing power. Even though a dedicated server is used for the training, its performance is still limited.

5.2 Conclusion

At the beginning of this thesis three questions, coming from the assignment were stated.

The first one was about already existing methods, that were developed by the researchers across the globe. From all the possible solutions and methods two have been chosen, they were chosen based on the excellent results that they are showing. Those are KGZ Net [23] and ChesXnet[24], their results can bee seen in the table 5.1. Both performed very well and their results are pretty much same as the results presented by the original authors, with slight differences.

Second question that has been stated was about the possibility of other architectures. Following this question, possible architecture combining ResNet and CapsNet is presented.

Third and the last question was about implementation of those architectures using available frameworks. For this task TensorFlow was chosen, as it is very well known framework for such applications in combination with popular library Keras which contains multiple pretrained networks. However even with those frameworks, this task proved to be very challenging. And many new concepts of math, computer science and programming in general need to be understood in order to implement them successfully.

As mentioned previously, results of already existing architectures are truly state-of-the art, as can be seen in the result table 5.1. From that table it can be clearly seen that both KGZ Net and ChesXnet are significantly better then standalone networks or proposed architecture. However this result was expected since both architectures are very well thought out, and worked out to the last detail. And of course both are using multiple sub-modules, each with different purpose to accomplish task of classification of the chest x-ray images.

Even though the presented architecture is not showing very promising results, only outperforming VGG, it is still a great accomplishment. Since this architecture shows that with more time for research and implementation, this architecture can be successfully used and perhaps even outperform already existing solutions.

Even very basic CapsNet architecture in combination with pre-trained ResNet is capable of working with such large and complex data-set. From this unintended contribution of the thesis comes. The CapsNet even in its very basic state can perform somewhat reasonably on a large scale and very complex datasets.

With those results and research in my mind, there is a very large chance that the proposed architecture might be able to achieve accuracy comparable to or even higher than already existing methods. But this can only be confirmed or debunked by follow-up research.

Bibliography

[1] BRUSH, Kate, BURNS, Ed, ed. *Deep learning: What is deep learning?* TechTarget [online]. March 2021 [cit. 2021-10-31]. Available from: <https://searchenterpriseai.techtarget.com/definition/deep-learning-deep-neural-network>.

[2] XI, Edgar, Selina BING and Yang JIN. *Capsule Network Performance on Complex Data* [online]. Carnegie Mellon University Pittsburgh, 10 December 2017n. l. [cit. 2021-10-31]. Available from: <https://arxiv.org/abs/1712.03480>. Carnegie Mellon University Pittsburgh.

[3] HINTON, Geoffrey E, Zoubin GHAHRAMANI a Yee Whye TEH. *Learning to Parse Images* [online]. 2000 [cit. 2021-11-01]. Available from: <https://papers.nips.cc/paper/1999/hash/5a142a55461d5fef016acfb927fee0bd-Abstract.html>

[4] HINTON Geoffrey E., Alex KRIZHEVSKY and Side D. Wang. *Transforming Auto-encoders* [online]. 2011 [cit. 2021-11-01]. Available from: <https://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.220.5099>

[5] SABOUR, Sara, Nicholas FROSST a Geoffrey E. HINTON. *Dynamic Routing Between Capsules* [online]. Toronto, 2017 [cit. 2021-10-31]. Available from: <https://arxiv.org/abs/1710.09829>

[6] BROWNLEE, Jason.*A gentle introduction to the Rectified LinearUnit (ReLu).* [cit. 2021-11-06]. Available from: <https://machinelearningmastery.com/rectified-linear-activation-function-for-deep-learning-neural-networks/

[7] HANSEN, Casper. *Activation functions explained* [online]. [cit. 2021-11-07]. Available from: <https://mlfromscratch.com/activation-functions-explained/#/>

[8] PECHYONKIN, Max. *Understanding Hinton's Capsule Networks. Part IV: CapsNet Architecture* [online]. [cit. 2021-11-23]. Available from: <https://medium.com/ai%C2%B3-theory-practice-business/part-iv-capsnet-architecture-6a64422f7dce>

[9] WONG, Ya a Boon NG. *Advances in Intelligent Systems and Computing: CapsNet vs CNN: Analysis of the Effects of Varying Feature Spatial Arrangement* [online]. 25 August 2020 [cit. 2021-11-24]. Available from: <https://link.springer.com/chapter/10.1007/978-3-030-55187-2_1

[10] HE, Kaiming, Xiangyu ZHANG, Shaoqing REN and Jian SUN. *Deep Residual Learning for Image Recognition* [online]. [cit. 2021-11-27]. Available from: <`https://arxiv.org/abs/1512.03385`>

[11] *Residual Neural Networks. Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2021-11-28]. Available from: <`https://en.wikipedia.org/wiki/Residual_neural_network#cite_note-6`>

[12] THOMSON, Alex. *Neocortical Layer 6, A Review* [online]. Online prepublication, 2010 [cit. 2021-11-28]. Dostupné z: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2885865/

[13] WINTERER, Jochen, Nikolaus MAIER, Christian WOZNY, Tiziano D'ALBIS, Richard KEMPTER a Dietmar SCHMITZ. *Excitatory Microcircuits within Superficial Layers of the Medial Entorhinal Cortex* [online]. 09 May 2017 [cit. 2021-11-28]. Available from doi:<`https://doi.org/10.1016/j.celrep.2017.04.041`>

[14] NAM, Ju Gang, Sunggyun PARK, Eui HWANG, Jong LEE, Kwang-Nam JIN, Kun LIM a Huy. *Development and Validation of Deep Learning–based Automatic Detection Algorithm for Malignant Pulmonary Nodules on Chest Radiographs* [online]. Seoul, 25 September 2018n. l. [cit. 2021-11-29]. Available from: <`https://pubs.rsna.org/doi/full/10.1148/radiol.2018180237`>. Research. National University Hospital and College of Medicine.

[15] LIU, Yun, Timo KOHLBERGER, Mohammad NOROUZI, et al. *Artificial Intelligence–Based Breast Cancer Nodal Metastasis Detection: Insights Into the Black Box for Pathologists* [online]. 8 October 2018n. l. [cit. 2021-11-29]. Available From: <`https://meridian.allenpress.com/aplm/article/143/7/859/10038/Artificial-Intelligence-Based-Breast-Cancer-Nodal`>. Reaserch. Arch Pathol Lab Med.

[16] PHAM, Dzung, Chenyang XU a Jerry PRINCE. *Current Methods in Medical Image Segmentation* [online]. Maryland, 2000 [cit. 2021-12-04]. Available from: <`https://www.annualreviews.org/doi/10.1146/annurev.bioeng.2.1.315`>. Reaserch. Department of Electrical and Computer Engineering, The Johns Hopkins University.

[17] ARTACHO, Bruno. *Semantic segmentation examples using WASPnet* [online]. [cit. 2021-12-04]. Available from: <`https://www.researchgate.net/figure/Semantic-segmentation-examples-using-WASPnet_fig1_337771814`>

[18] READ, Jesse, Bernhard PFAHRINGER, Geoff HOLMES a Eibe FRANK. *Classifier chains for multi-label classification* [online]. 2011 [cit. 2021-12-03]. Available From: <https://link.springer.com/article/10.1007%2Fs10994-011-5256-5>

[19] SPOLAÔR, Newton, Everton CHERMANT, Maria MONARD a Huei LEE. *A Comparison of Multi-label Feature Selection Methods using the Problem Transformation Approach* [online]. Sao Paulo, 2013 [cit. 2021-12-03]. Available from: <https://www.sciencedirect.com/science/article/pii/S1571066113000121?via%3Dihub>. Reaserch. University of São Paulo,.

[20] *TensorFlow*. Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2001-, 2015 [cit. 2021-12-03]. Available from: <https://en.wikipedia.org/wiki/TensorFlow>

[21] LORICA, Ben. *Why AI and machine learning researchers are beginning to embrace PyTorch* [online]. [cit. 2021-12-03]. Available from: <https://www.oreilly.com/radar/podcast/why-ai-and-machine-learning-researchers-are-beginning-to-embrace-pytorch/>

[22] *Keras*. Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2001-, 2015 [cit. 2021-12-03]. Available from: <https://en.wikipedia.org/wiki/Keras>

[23] WANG, Kun, Xiaohong ZHANG, Sheng HUANG a Feiyu. *Learning to Recognize Thoracic Disease in Chest X-Rays With Knowledge-Guided Deep Zoom Neural Networks* [online]. 2020 [cit. 2021-12-04]. Available from: <https://ieeexplore.ieee.org/document/9181568>

[24] RAJPURKAR, Pranav, Jeremy IRVIN, Kaylie ZHU, et al. *CheXNet: Radiologist-Level Pneumonia Detection on Chest X-Rays with Deep Learning* [online]. [cit. 2021-12-04]. Available from: <https://paperswithcode.com/paper/chexnet-radiologist-level-pneumonia-detection>

[25] WANG, Xiaosong, Yifan PENG, Le LU and Ronald SUMMERS. *ChestX-ray8: Hospital-scale Chest X-ray Database and Benchmarks on Weakly-Supervised Classification and Localization of Common Thorax Diseases.* 2017 IEEE Conference on Computer Vision and Pattern Recognition(CVPR) [online]. [cit. 2021-12-12].

[26] BEN-BARUCH, Emanuel, Tal RIDNIK, Nadav ZAMIR, Asaf NOY, Itamar FRIEDMAN, Matan PROTTER a Lihi ZELNIK-MANOR. *Asymmetric*

Loss For Multi-Label Classification [online]. [cit. 2021-12-13]. Available from: <https://arxiv.org/abs/2009.14119>

[27] HUANG, Gao, Zhuang LIU a Kilian Q. WEINBERGER. *Densely Connected Convolutional Networks* [online]. [cit. 2022-05-26]. Available from: <http://arxiv.org/abs/1608.06993>

[28] ZHANG, Xiangyu, Jianhua ZOU, Kaiming HE a Jian SUN. *Accelerating Very Deep Convolutional Networks for Classification and Detection* [online]. 18 Nov 2015 [cit. 2022-05-26]. Available from: <https://arxiv.org/abs/1505.06798>

[29] FERGUSON, Max, Ronay AK a Tina Lee YOUNG-TSUN. *Automatic localization of casting defects with convolutional neural networks* [online]. Boston, MA, USA, 2017 [cit. 2022-05-28]. Available from: <https://www.researchgate.net/publication/322512435_Automatic_localization_of_casting_defects_with_convolutional_neural_networks>

[30] HE, Haodi, Yuhui YUAN, Xiangyu YUE a Han HU. *MLSeg: Image and Video Segmentation as Multi-Label Classification and Selected-Label Pixel Classification* [online]. UC Berkeley, 2022 [cit. 2022-05-28]. Available from: <https://arxiv.org/abs/2203.04187>

[31] DU, Jingcheng, Qingyu CHEN, Yifan PENG, Yang XIANG, Cui TAO a Zhiyong LU. *ML-Net: multi-label classification of biomedical texts with deep neural networks* [online]. [cit. 2022-05-28]. Available from: <https://paperswithcode.com/paper/ml-net-multi-label-classification-of>

Symbols and abbreviations

CNN Convolutional Neural Network
Caps Net Capsule Neural Network
Res Net Residual Neural Network
CPU Central Processing Unit
GPU Graphical Processing Unit
TPU Tensor Processing Unit
RAM Random Access Memory
MNIST Modified National Institute of Standards and Technology database
CXR14 Chest X-ray 14
FC Fully Connected Layer
ReLU Rectified Linear Unit
AUC Area Under the receiver operating characteristic curve
ADAM Adaptive Moment Estimation
DenseNet Densly connected network
VGG Visual Geometry Group