BRNO UNIVERSITY OF TECHNOLOGY

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

FACULTY OF MECHANICAL ENGINEERING

FAKULTA STROJNÍHO INŽENÝRSTVÍ

INSTITUTE OF AUTOMATION AND COMPUTER SCIENCE

ÚSTAV AUTOMATIZACE A INFORMATIKY

ARTIFICIAL METHOD IN MEDICAL IMAGE PROCESSING

VYUŽITÍ METOD UMĚLÉ INTELIGENCE PRO ZPRACOVÁNÍ LÉKAŘSKÝCH DAT

BACHELOR'S THESIS

BAKALÁŘSKÁ PRÁCE

AUTHOR AUTOR PRÁCE	Iryna Derevianko
SUPERVISOR VEDOUCÍ PRÁCE	Mgr. Jana Procházková, Ph.D.

BRNO 2023

BRNO FACULTY
UNIVERSITY OF MECHANICAL
OF TECHNOLOGY ENGINEERING

Assignment Bachelor's Thesis

Institut:	Institute of Automation and Computer Science
Student:	**Iryna Derevianko**
Degree programm:	Engineering
Branch:	Applied Computer Science and Control
Supervisor:	**Mgr. Jana Procházková, Ph.D.**
Academic year:	2022/23

As provided for by the Act No. 111/98 Coll. on higher education institutions and the BUT Study and Examination Regulations, the director of the Institute hereby assigns the following topic of Bachelor's Thesis:

Artificial method in medical image processing

Brief Description:

The bachelor thesis will be about the use of artificial intelligence methods for medical image processing. This issue is very progressive nowadays, where a pre–trained network can evaluate even subtle changes in medical images and catch undesirable changes in time. The work will involve learning about the methods that are currently used, as well as implementing our own method and comparing the results.

Bachelor's Thesis goals:

Objectives:
1.To get acquainted with the problems of working with medical images, studying current methods, basics of working with image matrix.
2. Design of an AI detection method for use with medical images. Discussion of the results.
3. Creation of a custom application (web or mobile) where the proposed method will be used.

Recommended bibliography:

GURNEY, Kevin. An Introduction to Neural Networks. Florida, USA: CRC Press, 1997. ISBN 13 978-1857285031.

Medical Image Computing and Computer Assisted Intervention – MICCAI 2021 [online]. 2021. London: Springer, 2021 [cit. 2022-09-08]. ISBN 978-3-030-87234-2. Dostupné z: https://link.springer.com/book/10.1007/978-3-030-87193-2.

Deadline for submission Bachelor's Thesis is given by the Schedule of the Academic year2022/23

In Brno,

L. S.

doc. Ing. Radomil Matoušek, Ph.D. Director of the Institute	doc. Ing. Jiří Hlinka, Ph.D. FME dean

ABSTRAKT

V této práci je implementován systém podpory rozhodování pro diagnostiku abnormalit kolenního kloubu. Navrhujeme diagnostickou metodu založenou na technologii hlubokého učení, jako jsou konvoluční neuronové sítě se specifickým rozložením řezů magnetické rezonance. Bylo implementováno webové rozhraní, které zahrnuje funkce pro stahování souborů magnetické rezonance, možnost prozkoumat libovolný řez dat z libovolného požadovaného pohledu, vytváření animací pro průběžné prohlížení a také diagnostiku procesů pomocí výše uvedených technologií umělé inteligence.

ABSTRACT

In this work, a decision support system for diagnosing knee joint abnormalities is implemented. We propose a diagnostic method based on deep learning technology, such as convolutional neural networks with a specific distribution of magnetic resonance imaging slices. A web-based interface has been implemented that includes functions for downloading MRI files, the ability to examine any data slice from any required view, the creation of animations for continuous viewing, as well as process diagnostics using the aforementioned artificial intelligence technologies.

KEYWORDS

MRI, convolutional neural networks, weighted average ensemble, decision support system

KLÍČOVÁ SLOVA

MRI, konvoluční neuronové sítě, ansámbl váženého průměru, systém podpory rozhodování

Rozšířený abstrakt

Podle statistik trpí každý třetí člověk na planetě patologiemi a poraněními kolenního kloubu, které mohou vést ke ztrátě pohyblivosti, výkonnosti, a dokonce k invaliditě. Diagnostika onemocnění kolenního kloubu je naléhavým problémem. Jeho řešení pomocí metod umělé inteligence významně optimalizuje proces interpretace magnetické rezonance odborníky.

Hlavním tématem této práce je využití umělé inteligence pro zlepšení analýzy a interpretace magnetické rezonance. Cílem je navrhnout a vytvořit systém pro podporu rozhodování při diagnostice patologií na snímcích magnetické rezonance kolenního kloubu pomocí umělých neuronových sítí.

Aplikace urychlí diagnostiku snímků MRI kvalifikovanými radiology a může předejít občasným chybám. Je navržena diagnostická metoda založená na technologii hlubokého učení, konkrétně pak konvoluční neuronové sítě se specifickým rozložením řezů magnetické rezonance. Aplikace může být také využita pro výukové účely studentů medicíny.

V této práci je implementován systém pro podporu rozhodování při diagnostice abnormalit kolenního kloubu. Bylo navrženo rozdělení každého naskenovaného snímku na několik oblastí, aby bylo možné vyšetřit většinové části kolenního kloubu, jejichž poškození je na magnetické rezonanci viditelné.

Inovativnost práce spočívá v tom, že navrhovaný přístup nebyl dosud pro diagnostiku 3D lékařských snímků použit. V práci je předloženo specifické rozdělení rovinných řezů, pomocí něhož je pro každou rovinu natrénováno několik neuronových sítí, které jsou kombinovány pomocí souboru klasifikátorů s váženým průměrem. Vzhledem ke zpracování každého pohledu více neuronovými sítěmi byly sítě natrénovány a integrovány za pomocí klasifikátoru s váženým průměrem. Jako nejvhodnější se ukázaly centrální řez a 10% posun v obou směrech pro axiální, koronální a sagitální rovinu. Během předběžné přípravy dat bylo zjištěno, že zpracovávaný soubor dat je nevyvážený, což komplikuje úlohu trénování modelů strojového učení. Tento problém byl vyřešen rozšířením dat (augmentací) a použitím stochastického gradientního optimalizátoru. Takto bylo natrénováno devět neuronových sítí, které byly integrovány pomocí souboru klasifikátorů s váženým průměrem. Pro objektivní posouzení výkonnosti modelu byly použity další metriky (F-measure a AUC). Kombinací všech předtrénovaných neuronových sítí bylo dosaženo přesnosti 85,83% a váženého průměru F-measure 91,79%, což je výrazné zlepšení oproti výkonnosti jednotlivých modelů. Metrika AUC činila 81,22%, což znamená, že navržený model dosáhl vyššího skóre výkonnosti než konvoluční neuronová síť VGG16 s AUC 71% a téměř dosáhl výkonnosti sítě ResNet18 (84%). Je třeba poznamenat, že počet falešně negativních předpovědí (chyba typu II) je 0. Tato skutečnost je obzvlášť dů-

ležitá při vývoji systému pro podporu rozhodování v oblasti medicíny. Model tedy vykázal lepší výsledek než výše uvedené stávající metody. Model je implementován v programovacím jazyce Python s využitím knihoven TensorFlow, Keras. Webové rozhraní bylo realizováno prostřednictvím frameworku Streamlit a technologií User Experience Design (UX) / User Interface Design (UI). Aplikace umožňuje rychle získat kompletní individuální obraz potřebných řezů (koronální, sagitální, axiální) pro konkrétního pacienta a generovat animace pro průběžné prohlížení. Zahrnuje také automatickou interpretaci s pravděpodobnostními ukazateli přítomnosti abnormality pomocí výše zmíněných technologií umělé inteligence. To pomůže s nalezením nevýrazných rysů při rutinních interpretacích velkého počtu snímků.

ai ÚSTAV AUTOMATIZACE A INFORMATIKY

2023

BIBLIOGRAPHIC CITATION

DEREVIANKO, Iryna. *Artificial method in medical image processing*. Brno, 2023. Dostupné také z: `https://www.vut.cz/studenti/zav-prace/detail/149416`. Bakalářská práce. Vysoké učení technické v Brně, Fakulta strojního inženýrství, Ústav automatizace a informatiky, Vedoucí práce: Mgr. Jana Procházková, Ph.D

AUTHOR’S DECLARATION

I declare that this bachelor thesis is my original work, I have prepared it independently under the supervision of my supervisor and using literature and other sources of information that are all cited in the thesis and listed in the reference list.

As the author of this thesis, I further declare that in connection with the development of this thesis I have not infringed in an unauthorized manner on other’s personal copyright and I am fully aware of the consequences of violating the provisions of § 11 and the following of the Copyright Act No. 121/2000 Sb., including possible criminal consequences.

Brno, 21 May 2023 . . . .

Iryna Derevianko

ACKNOWLEDGEMENTS

I would like to thank my supervisor Mgr. Jana Procházková, Ph.D. for the excellent guidance and collaboration during the work. I also wish to thank everyone, who has supported me.

CONTENTS

1 Introduction

The knee joint is one of the largest and complex joints of the human skeleton. According to statistics, every third person over the age of 30 has suffered a knee joint injury at least once in their life, and the third part of the population of the planet suffers from its pathologies. Diagnosis of diseases of the knee joint by artificial intelligence methods is an urgent problem, the solution of which will significantly optimise the process of magnetic resonance imaging (MRI) interpretation by specialists.

The main idea of the current work is the artificial intelligence(AI) assistance that will improve the analysis and interpretation of MRI. Application will speed up the diagnosing of MRI-scans by qualified radiologists and can prevent an occasional mistake. It also can be used for educational purposes by medical students.

In the current work, a decision support system for diagnosing knee joint abnormalities is implemented. We suggest a diagnostic method based on deep learning technology, such as convolutional neural networks with specific distribution of magnetic resonance imaging slices. It was proposed to split each scanned view into several areas in order to investigate majority parts of the organ whose damage is visible on the MRI. We are not aware of using similar method for diagnosing 3D medical images, thus the approach can be considered novel. Due to distribution several neural networks for each view will be trained and integrated with use of weighted average ensemble. The model will be implemented in Python programming language with use of TensorFlow, Keras libraries.

A web-based interface has been realised via Streamlit framework and User Experience Design (UX) / User Interface Design (UI) technologies. Application provides an opportunity to quickly obtain a complete individual picture of the necessary sections (coronal, sagittal, axial) and slices for a particular patient, generating animations for continuous viewing. It also includes automated interpretation with probabilistic indicators of the abnormality presence using the aforementioned artificial intelligence technologies. This will help to find non-explicit features in the routine interpretation for a large number of images.

2 State of the art

The idea of using the principle of the visual process in the human brain for the object detection was introduced by Fukushima in the 1980s, [1], and improved by Yann LeCunn, [2]. More precisely, due to their proposition, computer models can extract features of the object and attempt to match them with features on unknown objects.

Artificial intelligent methods have wide range of applications, including engineering, construction and medicine. Medical applications are the main area of interest of the current work. Let us point out some of the literature, that was used during the research. A short overview of recent advances and some associated challenges in machine learning applied to medical image processing and image analysis were demonstrated in [3]. Authors of [4] are concentrated at key applications of deep learning in the fields of translational bioinformatics, medical imaging, pervasive sensing, medical informatics, and public health.

The other demonstrated approach lays in classification of skin lesions using a single convolution neural network (CNN), trained from images directly, using only pixels and disease labels as inputs, [5]. Specific computer-aided detection (CADe) problems, namely thoraco-abdominal lymph node (LN) detection and interstitial lung disease (ILD) classification were studied in [6] also by using Convolutional Neural Networks. The article [7] reviews the existing methods for processing medical images, separately highlighting the CNN methods.

The knee is one of the most frequently injured joints and knee pain is a common problem that can affect all age groups. In the current work the magnetic resonance imaging (MRI) of the knee for diagnosing knee injuries is considered. Advances in MRI technology provide the imaging necessary to obtain high-resolution images to evaluate menisci, ligaments, and tendons, [8], [9], see Fig.1.

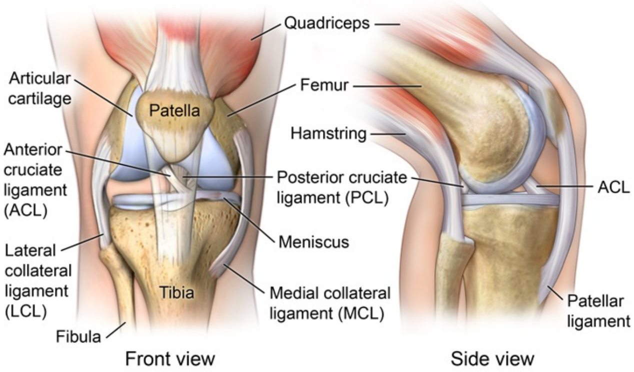

Quadriceps
Articular cartilage
Patella
Femur
Hamstring
Anterior cruciate ligament (ACL)
Posterior cruciate ligament (PCL)
ACL
Meniscus
Lateral collateral ligament (LCL)
Tibia
Medial collateral ligament (MCL)
Patellar ligament
Fibula
Front view
Side view

Fig. 1: The knee anatomy [10]

The MRI-method is based on the phenomenon of nuclear magnetic resonance. This property of hydrogen protons reacts in a specific way to the field induced by the apparatus. Charged particles change their trajectory and return to their original state. A tomograph captures energy bursts, then a computer program converts the data into an image. Thus, MRI series of monochrome images of the anatomical region are obtained in the axial, sagittal and coronal planes, see Fig.2.

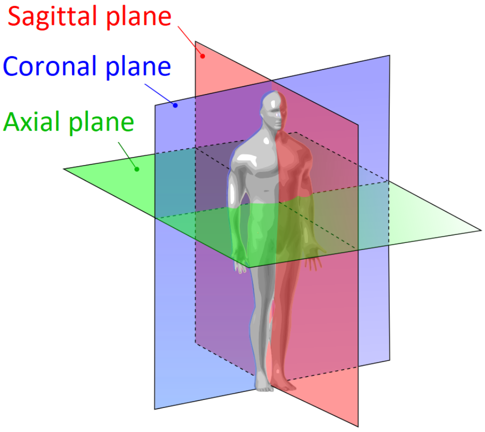


Fig. 2: The anatomical planes[11]

Interpretation an MRI image of the knee joint implies an assessment of the scans in comparison with the norm. MR tomograms are called slices due to the fact that the images are produced in layers, with a step of 1-2 mm.

The result of the research is formed in the form of a medical image, for example, in the format of DICOM (Digital Imaging and Communications in Medicine), NIfTI (Neural Imaging Technology Initiative in Informatics), Analyze, Minc1 (Medical Imaging Network Common Data Format), Minc2 or another suitable format. Files in any of these formats can be easily converted to the NPY extension, i.e. the standard binary file format, which is an universal solution for further processing in data science and machine learning.

Unlike radiography, magnetic resonance imaging generates a 3D projection of an organ, so 3D convolutional neural networks are most often used to work with such images, which require much more computational and memory capacity to train models, see [12].

In [13] there was investigated a 3D deep CNN architecture for automatic MRI glioma brain tumor grading. It was based on employing 3D convolutional filters and benefits a generating more powerful contextual features that deal with large brain tissues' variations.

The challenge of deep learning for medical imaging lies in processing large datasets containing examples of a wide variety of pathologies that can occur during a particular visual examination. The classification task can be defined as the distinguishing the "normal value"from the "abnormal value". By taking into account the fact that the model can learn the range of normality for a given MRI data set, we can hypothesize that any abnormality, no matter how rare, can be detected by the model.

Often, transfer learning technology, which uses pre-trained neural networks, is used to improve model accuracy. The MRI dataset studied in current work was used in [14]. The paper investigates the effect of structural variations, data augmentation and different transfer learning implementations on the performance of a deep neural network in the classification task of knee MRI. AlexNet based CNNs with layer freezing achieved AUC for abnormal, Anterior cruciate ligament lesion and meniscal lesion classification of 0.913, 0.859 and 0.792, an improvement over no layer freezing which had AUCs of 0.896, 0.842 and 0.773. The ResNet18 based classifier achieved AUCs of 0.843, 0.774 and 0.671. VGG16 based classifier achieved AUCs of 0.728, 0.690 and 0.711.

A deep learning–based cartilage lesion detection system was developed by using segmentation and classification CNN. Fat-suppressed T2-weighted fast spin-echo MRI data sets of the knee of 175 patients with different knee pains were retrospectively analysed by using the deep learning method in [15].

The main task of [16] was to present the findings of a systematic literature review of knee (anterior cruciate ligament, meniscus, and cartilage) injury detection papers using deep learning.

MRI of one part of the human body takes 20 to 40 minutes, but in urgent cases, the result is needed in less time. According to a new study conducted by the Grossman School of Medicine at New York University and Meta-II, artificial intelligence can reconstruct raw images of fast magnetic resonance imaging into high-quality images, that have diagnostic value similar to that of traditional MRI [17]. DL methods for reconstructing MR images from undersampled data can be trained on images of much higher complexity than those used for compressed sensing, and may therefore allow unprecedented levels of speed-up while maintaining high image quality. In the fast developing field of image reconstruction using DL, photorealistic results have been achieved for images of ordinary objects such as faces, animals or flowers. The reconstructed images must also be diagnostically precise. DL methods have not been widely adopted in the clinical setting because of the challenge of designing techniques that can demonstrate their ability to meet the often conflicting goals of high image quality and strict clinical fidelity. Studies have shown progress towards achieving acceptable image quality with small numbers of subjects, but to our knowledge no study has been reported that demonstrates both high quality and high accuracy.

The authors of [17] sought to use DL to accelerate MRI to a level consistent with a 5-minute total knee examination, without affecting image quality or diagnostic accuracy. To achieve this goal, the authors designed a DL model based on a variational network architecture that explicitly trains MRI detector coil sensitivities, incorporates several architectural improvements, and is followed by adaptive image enhancement to improve perceived image quality.

3 Theoretical Aspects of Neural Networks

3.1 Introduction to Neural Networks

The idea of transition from the processing of some formalized knowledge by an algorithm embedded in the computer to the implementation in it of the methods of information processing (mental activity) characteristic of a person led to the artificial neural networks, see [18]. A characteristic feature of biological systems is adaptation, thanks to which such systems develop and acquire new properties in the process of learning. Like biological neural networks, ANNs consist of interconnected elements, artificial neurons, the functionality of which to one degree or another corresponds to the elementary functions of a biological neuron.

Artificial neurons, also called neuron cells, nodes, modules, have the structure and functions similar to biological neurons.

The structure of an artificial neuron is shown in Fig.3

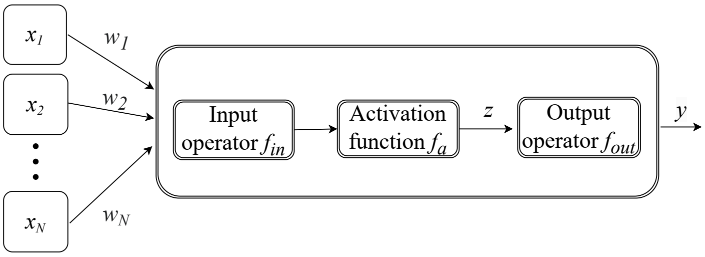


Fig. 3: The structure of an artificial neuron

The input signals of an artificial neuron $x_i(i = 1 \dots N)$ are the output signals of other neurons, each of which is taken with its weight $w_i(i = 1 \dots N)$.

The input operator f_{in} transforms the weighted inputs and sends them to the activation operator f_a. The output signal of the neuron y is the output signal of the activation operator transformed by the output operator f_{out}. It serves to represent the state of the neuron in the desired range of values. Thus, the nonlinear operator of the transformation of the vector of input signals x into the output signal y can be written as follows:

$$y_i = f_{out}(f_a(f_{in}(x_i, w_i))), \quad i = 1 \dots N \tag{1}$$

The following input functions are commonly used:

- sum of weighted inputs

$$f_{in}(\mathbf{x}, \mathbf{w}) = \sum_{i=1}^{N} x_i * w_i$$

- the maximum value of the weighted inputs

$$f_{in}(\mathbf{x}, \mathbf{w}) = \max\{(x_i * w_i)\}, \quad i = 1 \dots N$$

- product of weighted inputs

$$f_{in}(\mathbf{x}, \mathbf{w}) = \prod_{i=1}^{N} x_i * w_i$$

- the minimum value of the weighted inputs

$$f_{in}(\mathbf{x}, \mathbf{w}) = \min\{(x_i * w_i)\}, \quad i = 1 \dots N$$

The activation function $f_a()$ describes the mapping from the input signal to the output signal, (see [19]).

The simplest activation functions are

- linear $f_a(z) = Kz, \quad K = const$
- linear bipolar with saturation

$$f_a(z) = \begin{cases} 1, \quad if \quad z > \alpha_2 \\ Kz, \quad if \quad -\alpha_1 \leq z \leq \alpha_2 \\ -1, \quad if \quad z < -\alpha_1 \end{cases} . \tag{2}$$

- bipolar threshold function

$$f_a(z) = \begin{cases} 1, \quad if \quad z \geq \alpha \\ -1, \quad if \quad z < \alpha \end{cases} . \tag{3}$$

The application of linear functions is limited mainly to the simplest ANNs, which do not have hidden layers in their composition, and moreover, there is a linear dependence between input and output variables. In case of nonlinear dependences between input and output variables logistic, sigmoidal function is used.

The function is called sigmoidal if it is monotonically increasing, differentiable and satisfies the condition $\lim_{\lambda \to -\infty} f(\lambda) = k_1$, $\lim_{\lambda \to \infty} f(\lambda) = k_2$, $k_1 < k_2$.

Sigmoid function is given by formula:

$$f_a(z) = \frac{1}{1 + e^{-z}} \tag{4}$$

One of the advantages of the sigmoid function is its derivative $f'(x) = f(x) * (1 - f(x))$, see Fig.4

As we can see from the graph below, the derivative has the highest value in the interval [-2; 2], which allows the input values to quickly get close to the boundaries, in other words, to quickly determine how active the neuron is in the current moment. It is suitable for back-propagation during training CNN.

However, approaching the values of the activation function to the boundaries, again, leads to the problem of the vanishing gradient, and as a result, almost does not change weight coefficients.

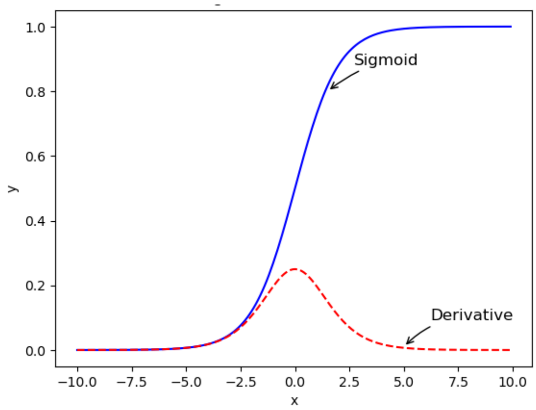


Fig. 4: The Sigmoid function

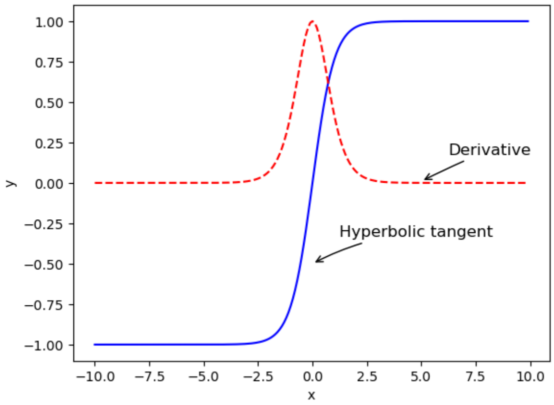


Fig. 5: The hyperbolic tangent

One more popular activation function is the hyperbolic tangent, (see Fig.5).

Hyperbolic tangent provides faster convergence than the standard logistic function due to its symmetry with respect to origin. It is given by:

$$f_a(z) = \tanh(z) = \frac{2}{1 + e^{-2z}} - 1 \tag{5}$$

On the other hand, the simpler option is available. That is a rectified linear unit (ReLU), which is expressed by the formula:

$$f_a(z) = \max(0, z) \tag{6}$$

Among the advantages of using ReLU are: ease of calculating the derivative (for negative values it equals to 0, for positive to 1) and sparsity of activation, i.e. in networks with a very large number of neurons, using the sigmoid function or hyperbolic tangent as an activation function causes almost all neurons to be activated, which can affect the performance of model training, (see Fig.6). If we use ReLU, then the number of included neurons will become less, and the network will become lighter.

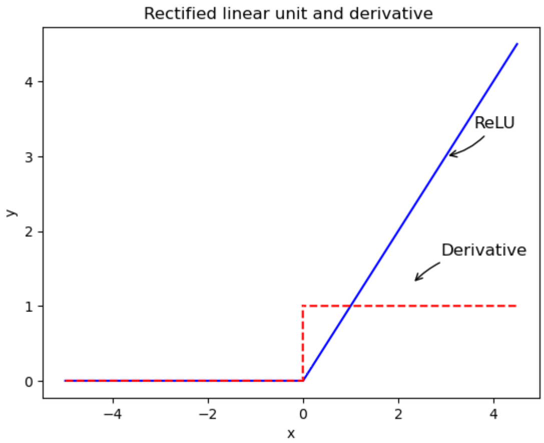


Fig. 6: The ReLU function

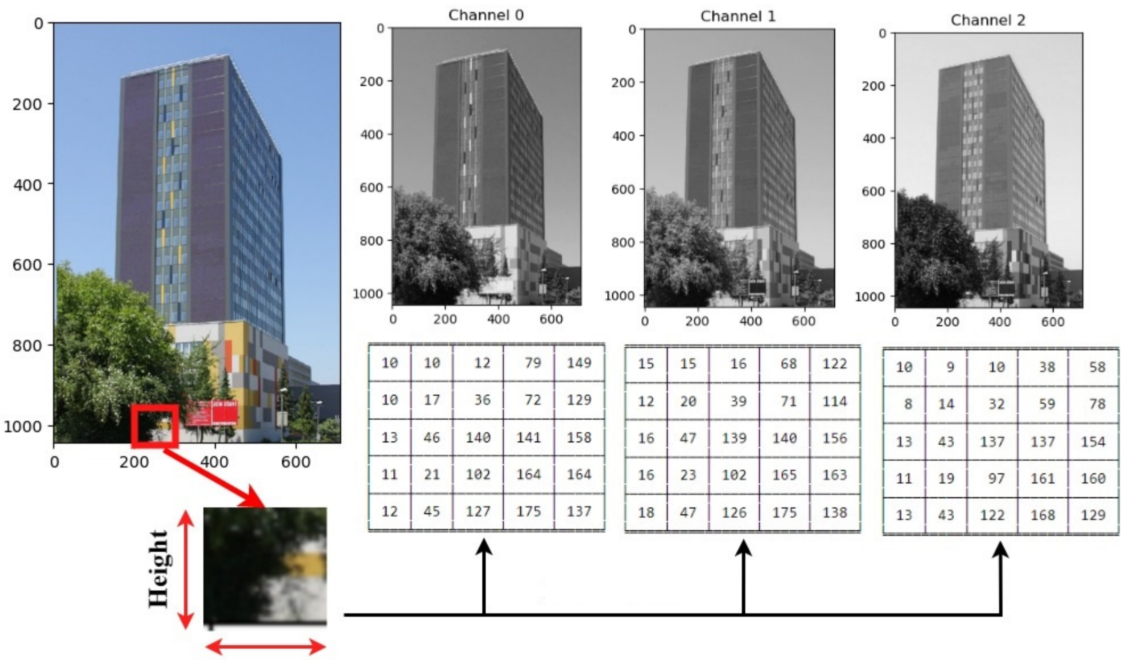
The main disadvantage is called the problem of the dying ReLU. Since part of the derivative of the function is zero, then the gradient for it will be zero, which means that the weights are not changing during the descent and the neural network will stop learning.

3.2 Convolutional Neural Networks

A Convolutional Neural Network, also known as CNN, is a type of neural networks that specializes in processing data with a grid-like structure, such as an image. A digital image is a binary representation of visual data, can be represented as a three-dimensional array of integer values. Usually, first and second dimensions of the array represent the pixel width and the height of the picture respectively, while the third dimension contains the color depth information, see Figure 7. The third dimension can be equal to one in case of grayscale image. In the case of 3-dimension these 3 numbers represent red, green and blue intensity values of pixel.

The Convolution Neural Network is an architecture of artificial neural networks aimed at efficient pattern recognition, which is part of deep learning technologies. It uses some features of the visual cortex, in which so-called simple and complex cells have been discovered. Simple cells respond to straight lines at different angles, and complex cells response is associated with the activation of a certain set of simple cells.

The structure of the CNN is unidirectional, or feed-forward, multilayer. In the feed-forward neural network structure, the signal moves strictly in the direction from the input layer to the output layer. The signal does not move in the opposite direction. Today, the development of this type of ANN is widespread and it successfully solves the problems of pattern recognition, prediction and clustering. The most



10	10	12	79	149
10	17	36	72	129
13	46	140	141	158
11	21	102	164	164
12	45	127	175	137

15	15	16	68	122
12	20	39	71	114
16	47	139	140	156
16	23	102	165	163
18	47	126	175	138

10	9	10	38	58
8	14	32	59	78
13	43	137	137	154
11	19	97	161	160
13	43	122	168	129

Fig. 7: Multichannel image representation

common standard method for training CNN is the back-propagation method. The method is used to effectively train a neural network using a so-called chain rule. In simple terms, after each pass through the network, the error signals are propagated from the network outputs to its inputs in the direction opposite to the forward propagation of the signals in the normal mode of operation. This adjusts the model parameters (weights and biases). A CNN typically has three type of layers (see Fig. 8): convolutional layers, pooling layers, and fully-connected layers.

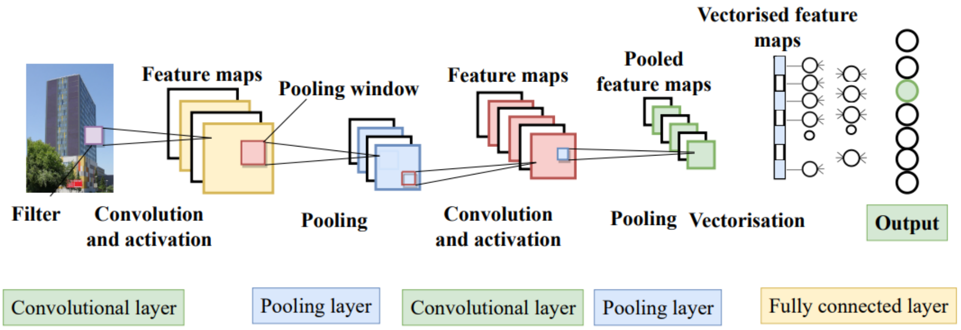


Fig. 8: CNN structure

3.2.1 Convolutional Layer

A distinctive feature of convolutional neural networks is the presence of the so-called convolutional layer. Convolutional layer is used to extract the features from the input images. In this layer, the mathematical operation of convolution is performed between the input image and a filter, also known as kernel.

The main task of the convolutional layer is to detect local conjunctions of features from the previous layer and mapping their appearance to a feature map ([20]). The output is termed as the Feature map which gives information about the image (such as the corners and edges). Later, this feature map is sent to the other layers in order to learn other features of the input image.

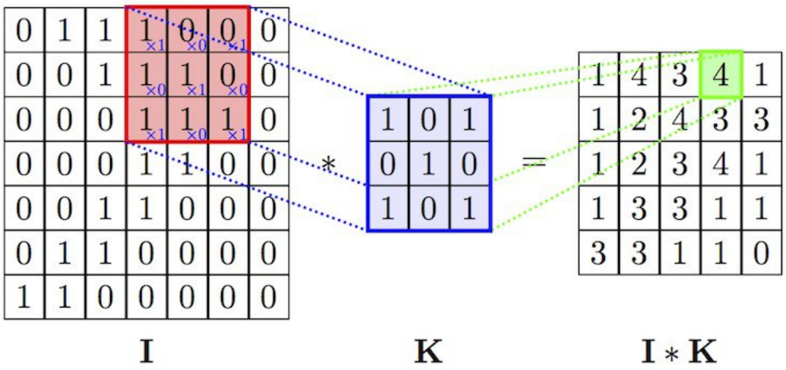
By sliding the filter over the input image (see Fig. 9), the dot product between the filter and the parts of the input image is taken with respect to the size of the filter. This produces a two-dimensional representation of the image known as an activation map that gives the response of the kernel at each spatial position of the image. The sliding size of the kernel is called a stride. The parameter corresponding to extending (or remaining) the area of which a convolutional neural network processes an image is called padding.

By considering a two-dimensional image I as our input and a two-dimensional kernel K, we can calculate convolution by the following formula:

$$S(i,j) = (K * I)(i,j) = \sum_{m=-a}^{a} \sum_{n=-b}^{b} I(i-m, j-n)K(m,n), \tag{7}$$

where

$$a = (K_{width} - 1)/2, \quad b = (K_{height} - 1)/2$$



$$\begin{array}{|c|c|c|c|c|c|c|}\hline 0&1&1&1&0&0&0\\\hline 0&0&1&1&1&0&0\\\hline 0&0&0&1&1&1&0\\\hline 0&0&0&1&1&0&0\\\hline 0&0&1&1&0&0&0\\\hline 0&1&1&0&0&0&0\\\hline 1&1&0&0&0&0&0\\\hline\end{array} * \begin{array}{|c|c|c|}\hline 1&0&1\\\hline 0&1&0\\\hline 1&0&1\\\hline\end{array} = \begin{array}{|c|c|c|c|c|}\hline 1&4&3&4&1\\\hline 1&2&4&3&3\\\hline 1&2&3&4&1\\\hline 1&3&3&1&1\\\hline 3&3&1&1&0\\\hline\end{array}$$

Fig. 9: Convolutional filter, [21]

Let us now consider one convolution layer with input picture of the size $W \times W \times C$, where C is a number of channels. Further C should match the number of channels in filter. In case of RGB color model we will have C equals to 3. Now we apply C_{k+1} filters of the size F with stride S and padding P, therefore the size of output volume can be determined by the following formula:

$$W_{out} = \frac{W - F + 2P}{S} + 1 \tag{8}$$

So the output of the current layer (k) will be the input of the following layer ($k+1$) and will be equal to $W_{out} \times W_{out} \times C_{k+1}$. Number of parameters in considered Convolution layer is equal to $(F * F * C + 1) * C_{k+1}$.

3.2.2 Pooling Layer

In most cases, a Convolutional layer is followed by a Pooling or Subsampling layer. The main goal of this layer is to decrease spatial size of the input image, to speed up the calculations by reducing computational cost, and to make CNN more general. Thus the probability of overfitting(when the network is biased towards particular pixels) decreases. It helps to make some of the features that detects more robust. Mostly there are needed two hyperparameters to perform the subsampling operation, which are the sample or filter size F and the stride S. If we have an input volume of size $H \times W \times C$, then the size of output can be determined by the following formula:

$$(\frac{H-F}{S}+1) \times (\frac{W-F}{S}+1) \times C \tag{9}$$

So the number of input channels (C) is equal to the number of output channels because pooling applies to each of the channels independently,[22]. The primary aim of this layer is decreasing of the size of the convolved feature map to reduce the computational costs. This is usually performed by minimizing the amount of the connections between layers and operating separately on each feature map. Depending upon method used, there are three the mostly used types of Pooling operations, which are Max, Average and Sum Pooling. Basically, it summarises the features previously generated by a convolution layer.

In Max pooling, the largest element is taken from feature map (see Fig. 10). Average Pooling means calculating the average of the elements in a predefined sized image section. The total sum of all elements in the predefined section is computed during the Sum Pooling. The Pooling Layer usually plays the role of a connection bridge between the Convolutional Layer and the FC Layer.

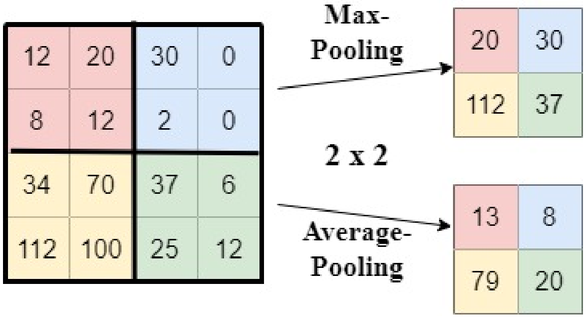


Fig. 10: Average- and Max-Pooling

3.2.3 Fully Connected Layer

The last layer in CNN is a fully-connected or dense layer. Its purpose is making classification, the layer models a complex non-linear function, which will be optimized in order to improve the quality of recognition. Each neuron of previous pooling layer is taken into account by each neuron of the output layer, therefore, as a result, all possible connections layer-to-layer are present.

After transforming the input image into a form suitable for a multilevel perceptron, we must flatten the image into a column vector. The type of output layer activation function affects the choice of loss function. The sigmoid and softmax functions are usually used as the activation function of the output layer of the network to solve the classification problem, i.e. when it is necessary to assign the described object to one of several classes. Softmax is used for a classification problem with $N > 2$ classes and modelling the value of the probability of belonging to a class. This paper considers the binary classification problem, so the sigmoid will be used as the activation function of the output layer.

3.3 Model performance evaluation

While considering different models, the question of the quality of the network occurs. For the classification models such quality criteria use different types of evaluation metrics, that are, for instance, accuracy, precision and recall.

Definition 1. *Accuracy A is the percentage of correct predictions for the test data, defined by the formula:*

$$A = \frac{C}{T}, \tag{10}$$

where C is the number of correct predictions and T is the total number of predictions.

As a performance measure, accuracy is not suitable for imbalanced classification problems. The main reason for this is that the large number of examples from the majority class (or classes) will overwhelm the amount of examples in the minority class, which means that even inefficient models can achieve accuracies up to 95%, depending on how serious the class imbalance is.

Relevance of the data is the base for both precision and recall.

Definition 2. *Precision P can be defined by the formula:*

$$P = \frac{N_{TruePositive}}{N}, \quad N = N_{TruePositive} + N_{FalsePositive}, \tag{11}$$

where $N_{TruePositive}$ is the amount of the relevant predictions and N is the amount of all the examples that by the prediction belong to a certain class. $N_{FalsePositive}$ stands for the case when the observation was predicted to belong to a certain class but in reality it does not.

Precision evaluates the fraction of correct classified instances among the ones classified as positive, [23].

Now let us introduce the recall.

Definition 3. *Recall R is the fraction*

$$R = \frac{N_{TruePositive}}{M}, \quad M = N_{truepositive} + N_{FalseNegative}, \tag{12}$$

where $N_{TruePositive}$ is the amount of elements which were correctly predicted to belong to a class and M is the amount of all elements truly belonging to the class. $N_{FalseNegative}$ stands for the case when the prediction tells that an observation does not belong to a class when in reality it does.

For imbalanced learning, recall is typically used to measure the coverage of the minority class,[24].

In imbalanced datasets, the goal is to improve recall without hurting precision. These goals, however, are often conflicting, since in order to increase the TP for the minority class, the number of FP is also often increased, resulting in reduced precision,[24].

For working on imbalanced data it is better to use the F-Measure, which combines the precision and recall metrics into a single metric.

Definition 4. *F-Measure can be defined by formula:*

$$F = \frac{2 * R * P}{R + P}, \tag{13}$$

where R is recall and P stands for precision.

There exist 3 types of F-Measure, that are macro-averaged, weighted-averaged and micro-averaged F-Measure. The macro-averaged F-Measure (or macro F1 score) is an arithmetic mean (unweighted mean) of all the F-Measures per class.

The weighted-averaged F-Measure is the mean of all F-Measures per class taking in account each class's support. Support is the amount of actual occurrences of the class in the dataset.

Micro-averaged F-Measure is equivalent to the accuracy. It computes a global average F-Measure by counting the sums of the True Positives, False Negatives, and False Positives.

For an imbalanced dataset with assigning the greater contribution to classes with more instances in the dataset, the weighted average is preferred. The reason lays in the contribution of each class to the F average, which is weighted by its size.

The Receiver Operator Characteristic (ROC) curve is the evaluation metric used for binary classification problems. It is a probability curve that plots the TPR with respect to the FPR at different thresholds, essentially separating the 'signal' from the 'noise'. In other words, it shows how a classifier performs at all classifying thresholds. The area under the curve (AUC) is the measure of a binary classifier's ability to distinguish between classes. It is used as a summary of the ROC curve.

The higher the AUC, the better is the performance of the model in distinguishing between the positive and negative classes.

Now let us introduce the loss function. It is function that compares the target and predicted output values and measures the quality of the training. One of the most used loss functions in machine learning is the cross-entropy loss function, also called logistic loss or multinomial logistic loss. The known cross-entropy types are binary and categorical, their use depends on amount of predicted classes in task, see [25]. In current work we will use binary cross-entropy, that is usually mathematically expressed as follows

$$L_i = -(y_i * log(p(x_i)) + (1 - y_i) * log(1 - p(x_i))) \tag{14}$$

for an instance i, where $p(x_i)$ is the predicted probability that instance x_i belongs to the class and y_i is the boolean label for instance i, [26].

Due to the fact that y_i takes values 1 or 0, the formula (14) can be decomposed into two cases. When y equals to 1, the loss equals to the negative logarithm of the predicted probability p_i: $L_i = -\log(p_i)$. Otherwise, $L_i = -\log(1-p_i)$. Therefore, the cross-entropy loss is minimal when the difference between the predicted probability and the true value is small, and gets big values otherwise.

While accuracy demonstrates if the particular prediction is correct, cross-entropy loss tells the level of correctness of the particular prediction.

It may happen, that the machine learning model already has some big amount of pre-built structure and therefore it can not learn from the examples. Such case occurs, for example, when the linear model is trained on a exponential dataset. Such type of situation is called underfitting or the model has high bias.

Underfitting occurs when the model is not able to decrease the error for either the test or training set. The model is not powerful enough for compliance with the complexity of data distribution.

In terms of the loss function, this means that model does not give small error on both train and validation sets.

Therefore the solution for reducing underfitting usually lays in increasing the learning rate, [27]. Often, the CNN architecture should be modified, i.e. the number of layers in the model or the number of neurons in each layer should be increased. Changing the order in which to apply the layers may help.

Other problem that can occur while training the model is called overfitting. It happens in the case when the machine learning model starts to capture not only the signal, but also noise in the data. As the result it causes the fluctuations in the model that do not reflect the true trend. Such problem is also called the high variance in the model. In other words, the machine learning model is not able to generalize well (the generalization error increases) due to the fact that too much attention is paid to the training data without further adaptation to the new data. In this case train set error is sufficiently small while error on the validation set is much more.

Therefore, the main point is finding the balance point between underfitting and overfitting, see Fig. 11.

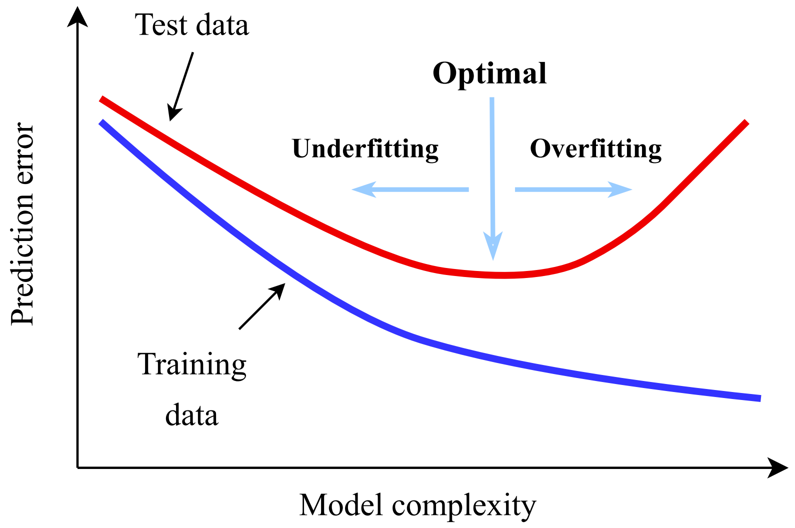


Fig. 11: The balance between underfitting and overfitting

In order to reduce model overfitting, more data is needed. The most commonly used techniques to decrease high variance are using data augmentation, adding regularisation, and reducing architectural complexity.

In order to get the best model generalisation results, it is necessary to have more data, including different variations. That is, it is necessary to increase the size of the original set artificially, and this can be done by data augmentation. The types of data augmentation are the following: creating training examples from natural and from artificial images. Instances for learning from natural images are created from real data. The second approach to creating training data is to generate it artificially. To create the required number of examples for training, several "ideal"examples can be taken and various distortions applied. The most popular methods are horizontal flip, rotation, padding, cropping, adding noise, color jittering etc.

The most common method used to regularise neural networks is called Dropout. In a conventional neural network, the phenomenon of overfitting occurs due to what is known as co-adaptation, i.e. when updating the weights of neurons, the activity of other neurons is taken into account during back-propagation learning in order to minimise the loss function. Therefore, the weights of neurons can change while correcting the errors of other neurons. The dropout method prevents this adaptation.

The main idea of Dropout is that instead of training a single Deep Neural Network (DNN), to train an ensemble of multiple DNNs, and then average the results. Networks for training are obtained by dropping out neurons with probability p, so the probability that a neuron remains in the network is $q = 1 - p$, see Fig. 12. "Excluding"a neuron means that for any input or parameter it returns 0.

Excluded neurons do not contribute to the learning process at any stage of the backpropagation algorithm; therefore, excluding at least one neuron is equivalent to training a new neural network.

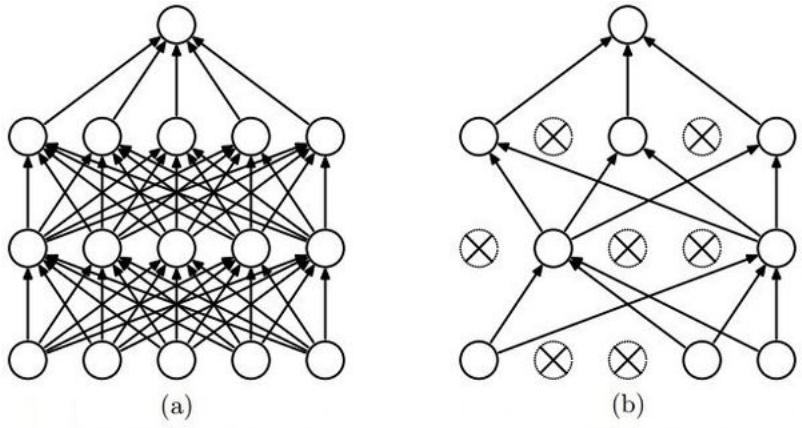


Fig. 12: Dropout Neural Net Model. Left: A standard neural net with 2 hidden layers. Right: An example of a thinned net produced by applying dropout to the network on the left [28]

4 Experimental part

4.1 Technologies

Python was chosen as the programming language because it provides a wide range of libraries, packages and frameworks for machine learning, mathematical computation and data manipulation. The main libraries used in the implementation process are Numpy, Pandas, Matplotlib, OpenCV and Scikit Learn. The library for building a convolutional neural network, Keras, supported by TensorFlow, is an open source library that allows interaction with artificial neural networks.

TensorFlow is good for complex projects such as building multi-layer neural networks. It is used for speech or image recognition, and for text processing applications such as Google Translate. Keras supports a wide range of neural network layers, such as convolutional, recursive, or dense layers, as well as metrics, losses, optimisers, etc. The TensorFlow SIG Addons was used to implement functionality not available in the core TensorFlow. The training procedure ran on an AMD Ryzen 7 5700U with Radeon Graphics.

In order to create the user interface, we used the Streamli library [29], the main goal of which is creating and deploying web-based machine learning applications.

4.2 Data description and preprocessing

The dataset that was chosen for the current research consists of 1,370 knee MRI exams performed at Stanford University Medical Center, [30]. The dataset contains 1,104 (80.6%) abnormal exams, with 319 (23.3%) anterior cruciate ligament (ACL) tears and 508 (37.1%) meniscal tears. Labels were obtained through manual extraction from clinical reports.

The considered dataset contains 3 folders for each view (Coronal, Sagittal and Axial) with .npy files of each of the 1130 patients. This set is used for the training, while the data used for the testing model contains 120 patients.

For further work, the data were converted to .jpg files, 256 × 256 pixels for each slice, see figure 13. Converting images to JPG format does not compromise image quality through compression. It should be noted that the number of slices for each view differs from patient to patient, and, moreover, the number of slices per patient is different for each of the three views, see figure 14. Both of cases happen due to the physical individuality and make the task of the model training more complex. In order to deal with that problem, the data normalization is needed. In the current

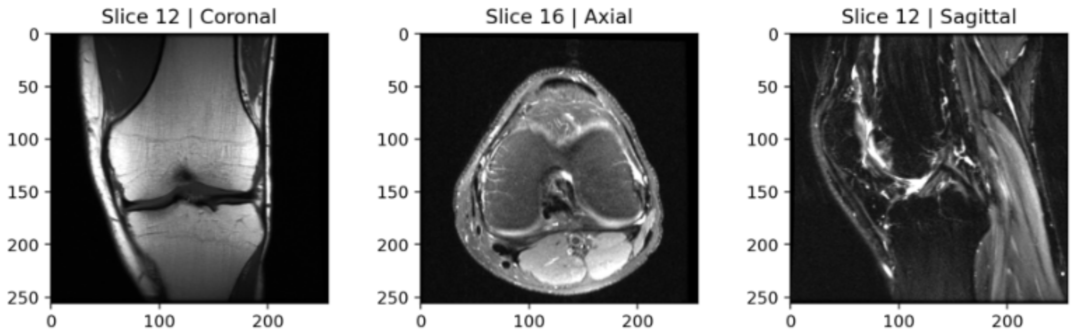
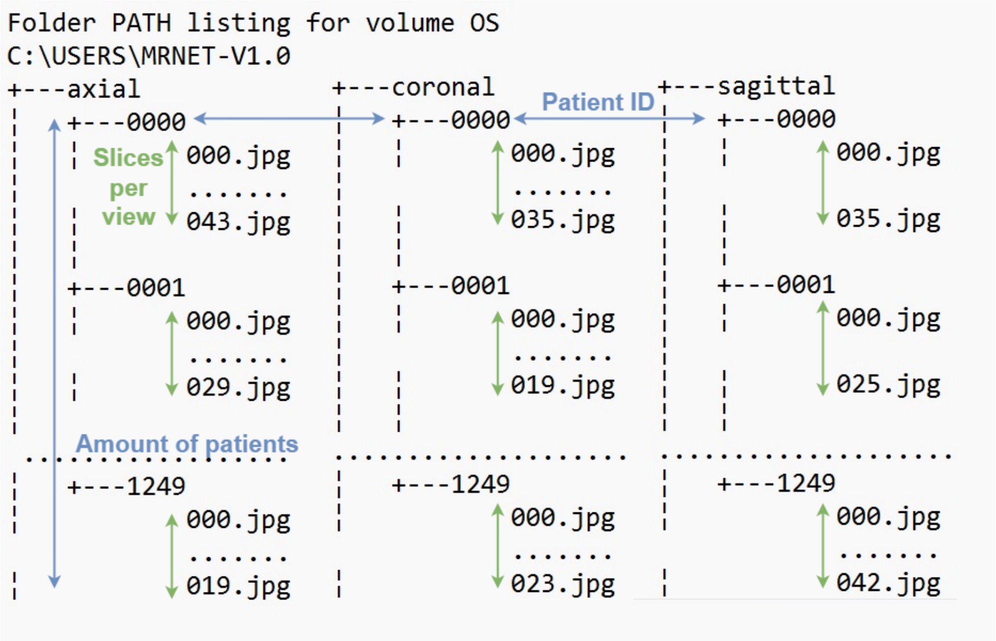


Fig. 13: Central slices of coronal, axial and sagittal views by means of `Matplotlib` library

research abnormality will be used as a target, thus the binary classification problem is being considered.



```
Folder PATH listing for volume OS
C:\USERS\MRNET-V1.0
+---axial                    +---coronal             +---sagittal
¦   +---0000                 ¦   +---0000            ¦   +---0000
¦   ¦       000.jpg          ¦   ¦      000.jpg      ¦   ¦      000.jpg
¦           .......          ¦          .......      ¦
¦   ¦       043.jpg          ¦   ¦      035.jpg      ¦   ¦      035.jpg
¦   ¦                        ¦   ¦                   ¦   ¦
¦   +---0001                 ¦   +---0001            ¦   +---0001
¦   ¦       000.jpg          ¦   ¦      000.jpg      ¦   ¦      000.jpg
¦           .......          ¦          .......      ¦
¦   ¦       029.jpg          ¦   ¦      019.jpg      ¦   ¦      025.jpg
¦                            ¦   ¦                   ¦   ¦
 ...................         ...................     ....................
¦   +---1249                 ¦   +---1249            ¦   +---1249
¦           000.jpg          ¦          000.jpg      ¦          000.jpg
            .......                     .......                 .......
¦           019.jpg          ¦          023.jpg      ¦          042.jpg
```

Fig. 14: Input data structure

Attention also should be paid to the balance in training and testing data. It was observed that positive target value overwhelms in both datasets. The proportion is 80,8% to 19,2% (913/217 cases) for training and 79,1% to 20,9% (95/25 cases) for testing. Unbalanced dataset may cause the overfitting because of a lack of training data. This is an example where it will be necessary to use additional metrics(F-Measure and AUC) to obtain an objective estimate of the model's performance. In some cases the problem of unbalanced data may be solved by using of stratified K-fold cross-validation [31]. The most common ways for solving the problem of unbalanced data in image classification are using of data augmentation and stochastics gradient-based optimizers [32], i.e. Adam. Taking into account the individuality of each person's knee joint and the errors while magnetic resonance imaging process, the following types of augmentation techniques were proposed:

- rotation by 15 degrees
- increase/decrease of scale by 10%
- vertical and horizontal shifts by 0.05%

4.3 Implementation

It is common to use 3d-CNN or U-Net architecture for 3-dimensional learning models like MRI/CT, but they do not always pay attention to the individuality of particular organ. In this work, it was proposed to split each scanned view into several areas in order to investigate majority parts of the organ whose damage is visible on the MRI. The most suitable scans were the central scan and a 10% shift to each side, see Figure 15.

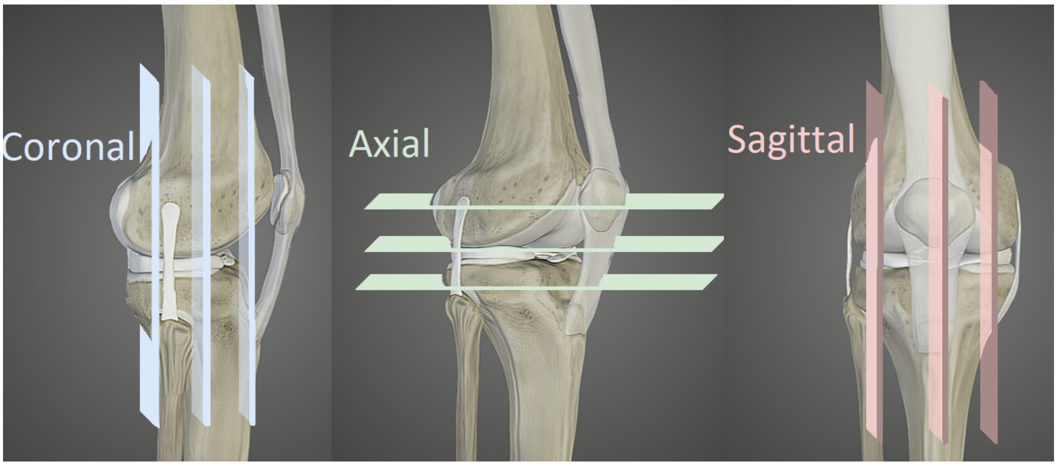
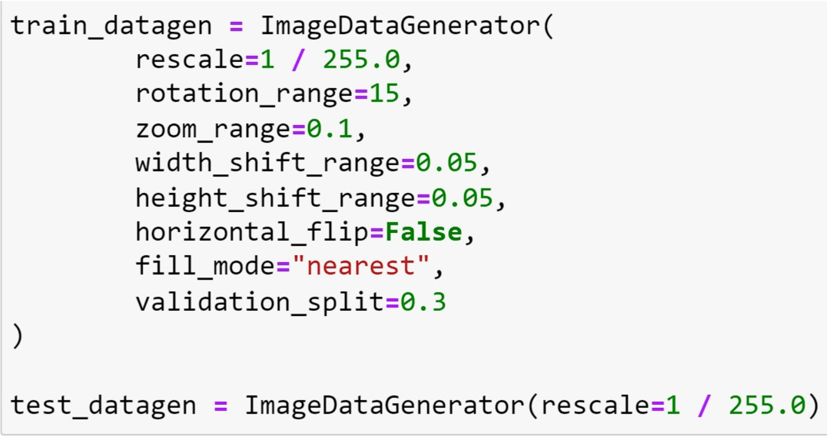


Fig. 15: The proposed deviation of slices for investigation

Thus, 9 models need to be trained, i.e. 3 models for each of the three views. In further studies, another split, depending on the individual characteristics of the organ and its deviation, can be considered.

To structure and separate the data into train/validation/test sets (train/validation proportion 70/30, train/test 90/10) the `flow_from_dataframe()` function of `keras.preprocessing` module was used. The dataset was shuffled and divided with a fixed seed number, what is needed to perform qualitative comparison of model‘s performance. The `flow_from_dataframe()` function allows to directly augment the image by reading its name and target value from the dataframe. Data augmentation was done using `ImageDataGenerator`, see figure 16. When the image is rotated, some pixels are moved outside the image and therefore an empty area is left there.

In order to fix that the area was filled according to the mode `"nearest"`. Original images consist of RGB coefficients 0-255 values, which have been scaled between 0 and 1 by a 1/255 instead.



```
train_datagen = ImageDataGenerator(
        rescale=1 / 255.0,
        rotation_range=15,
        zoom_range=0.1,
        width_shift_range=0.05,
        height_shift_range=0.05,
        horizontal_flip=False,
        fill_mode="nearest",
        validation_split=0.3
)

test_datagen = ImageDataGenerator(rescale=1 / 255.0)
```

Fig. 16: Data augmentation by means of `tensorflow.keras.preprocessing.image`

The structure of the convolutional neural network model is demonstrated in the figure 17. So the proposed sequential model has 6 convolution layers with different number of filters and filter size; the amount of filters increases with the deepness of NN. As an activation function the ReLU was used. Images were resized for decreasing computational complexity during training, so the input image has size of $200 \times 200 \times 3$. After each convolution layer there are Max-pooling layers.

The `Flatten()` method is used then due to the fact that fully-connected layer requires input in single-dimensional matrix. The Dropout layer with rate 0.5 was used to regularise the neural network and prevent overfitting of the model. The last Dense layer was proposed to have the sigmoid activation function due to the binary classification problem.

Networks are trained using the Adam optimizer (`keras.optimizers.Adam`) with the learning rate set to $5 \cdot 10^{-6}$. Such a small learning rate was chosen according to [27], the training took higher number of epochs, but the convergence was guaranteed. As it was mentioned in [33], Adam optimization method is computationally efficient, has little memory requirement, invariant to diagonal rescaling of gradients, and is well suited for problems that are large in terms of data/parameters. The batch size is set to 8. Small batch size was chosen in order to have more regularization effects. The loss-function is set to a binary cross-entropy function. The chosen metrics are accuracy, AUC of the package `keras.metrics` and F-measure of `tensorflow_addons`.

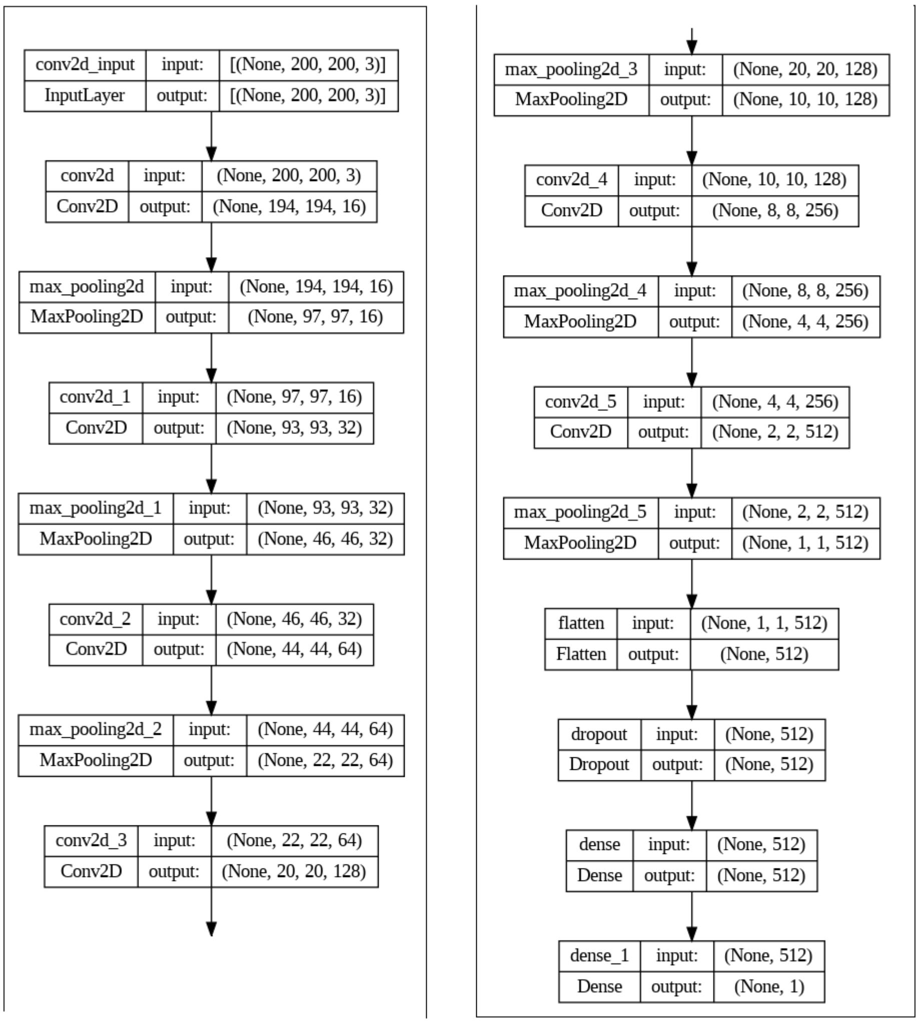

conv2d_input | input: | [(None, 200, 200, 3)]
InputLayer | output: | [(None, 200, 200, 3)]
conv2d | input: | (None, 200, 200, 3)
Conv2D | output: | (None, 194, 194, 16)
max_pooling2d | input: | (None, 194, 194, 16)
MaxPooling2D | output: | (None, 97, 97, 16)
conv2d_1 | input: | (None, 97, 97, 16)
Conv2D | output: | (None, 93, 93, 32)
max_pooling2d_1 | input: | (None, 93, 93, 32)
MaxPooling2D | output: | (None, 46, 46, 32)
conv2d_2 | input: | (None, 46, 46, 32)
Conv2D | output: | (None, 44, 44, 64)
max_pooling2d_2 | input: | (None, 44, 44, 64)
MaxPooling2D | output: | (None, 22, 22, 64)
conv2d_3 | input: | (None, 22, 22, 64)
Conv2D | output: | (None, 20, 20, 128)
max_pooling2d_3 | input: | (None, 20, 20, 128)
MaxPooling2D | output: | (None, 10, 10, 128)
conv2d_4 | input: | (None, 10, 10, 128)
Conv2D | output: | (None, 8, 8, 256)
max_pooling2d_4 | input: | (None, 8, 8, 256)
MaxPooling2D | output: | (None, 4, 4, 256)
conv2d_5 | input: | (None, 4, 4, 256)
Conv2D | output: | (None, 2, 2, 512)
max_pooling2d_5 | input: | (None, 2, 2, 512)
MaxPooling2D | output: | (None, 1, 1, 512)
flatten | input: | (None, 1, 1, 512)
Flatten | output: | (None, 512)
dropout | input: | (None, 512)
Dropout | output: | (None, 512)
dense | input: | (None, 512)
Dense | output: | (None, 512)
dense_1 | input: | (None, 512)
Dense | output: | (None, 1)

Fig. 17: Structure of the input data

4.4 Results

The experiments were started with training models for the central slice. Let us now consider training procedure for the central slices of each view. The training/validation behaviour based on accuracy and loss depending on epochs is illustrated in Figures 18–20.

As it was mentioned above we are dealing with the problem that has skewed data; so for obtaining an objective assessment of model performance addition metrics were used. Loss, accuracy, as well AUC and F-score are illustrated in the Table 1. Macro- and weighted-average of precision, recall and F-score for each target class were also taken into account, see Appendix A. While analysing the results, we should pay attention to F-measure and AUC metrics due to the fact that accuracy will not give realistic evaluation for imbalanced data. We also can notice that model‘s per-

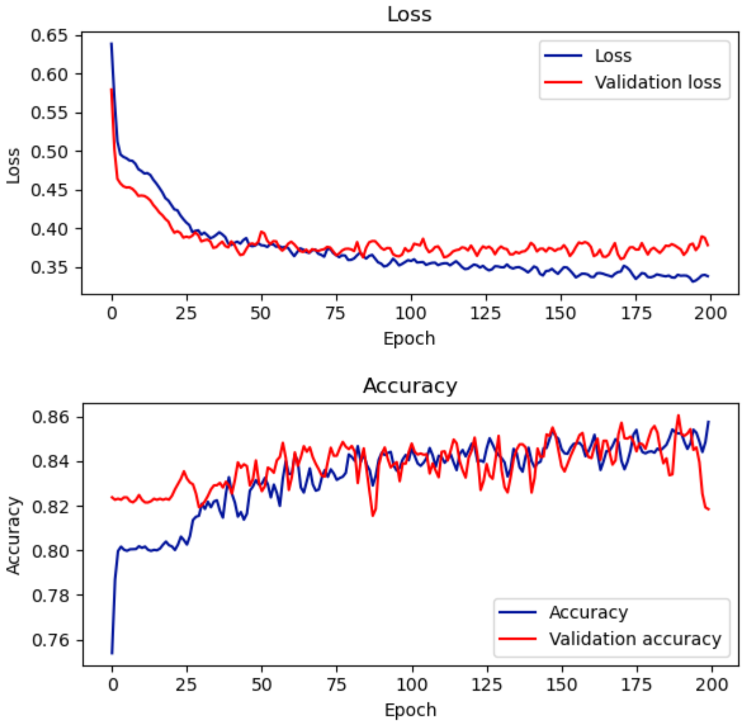


Fig. 18: The training/validation behaviour based on accuracy and loss for Axial view

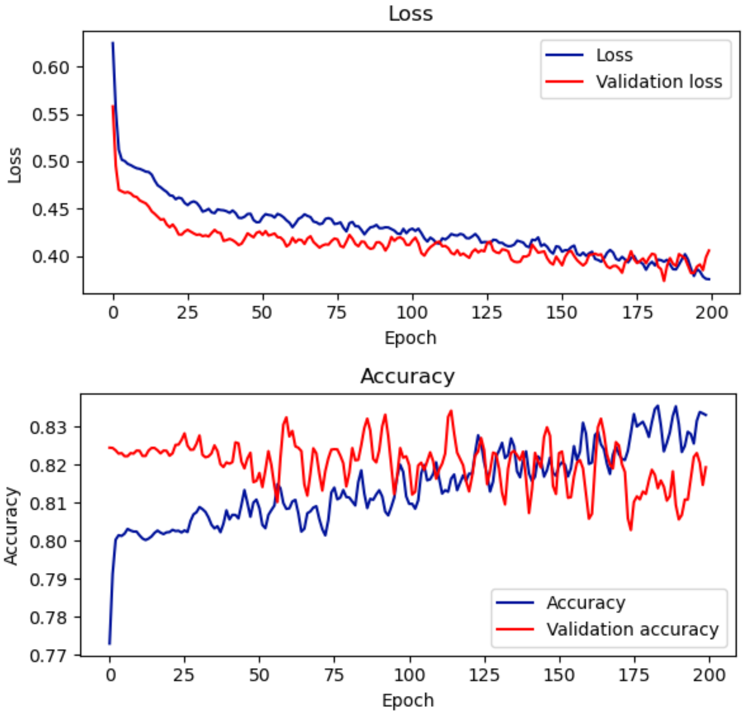


Fig. 19: The training/validation behaviour based on accuracy and loss for Coronal view

formance varies. The reason for that is not only model's ability to find dependencies, but also different input data.

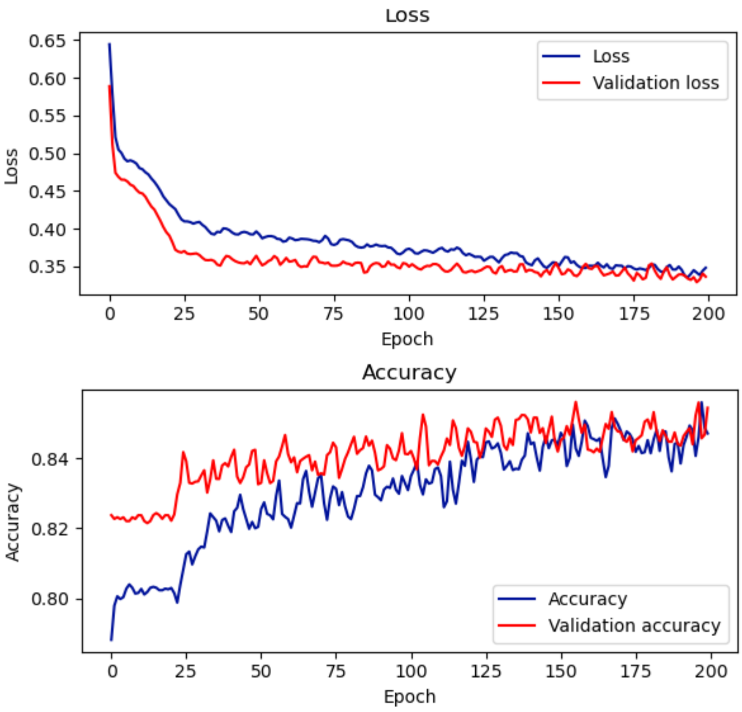


Fig. 20: The training/validation behaviour based on accuracy and loss for Sagittal view

Tab. 1: Metrics results for performance models on central slices

	Axial model	Coronal model	Sagittal model
Test loss	0.352	0.510	0.450
Accuracy	0.866	0.808	0.850
AUC	0.856	0.657	0.807
F-score	0.883	0.884	0.884

Let us now look on confusion matrices, see Figure 21. There are cases where the model ’misses’ the disease when it is present (the predicted value of the target attribute is 0 when the actual value is 1). This is called a false-negative prediction, or type II error. In medical applications, this situation should be minimised.

Similar experiments were carried out on 40% and 60% of the slices in relation to the distribution, mentioned above, see Figure 15. The training/validation behaviour based on accuracy and loss as well as metrics results for performance models and confusion matrices for each model are given in Appendix A.

In order to obtain a result with higher performance for a target attribute, it was proposed to combine all models. The weighted average ensemble was used in this work. It is an ensemble machine learning approach that merges the predictions

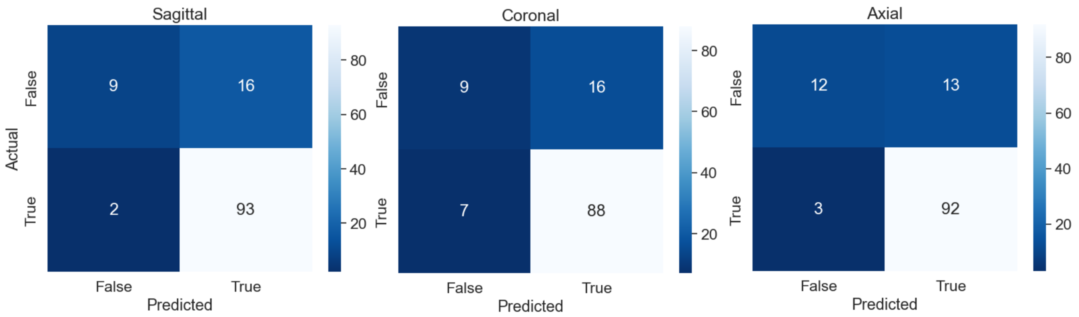


Fig. 21: Confusion matrices received by testing central slices of Sagittal, Coronal and Axial view respectively

of multiple models, with each model's contribution weighted in relation to its performance. The final result is obtained through a dot product of the weight vector and the model predictions vector; after that, the sum or highest probable class is taken [34]. The weighted average AUC of 9 models was normalized and taken as the weights $(0.862, 0.606, 0.829, 0.857, 0.658, 0.808, 0.800, 0.692, 0.844)$. In this way, each model has a different influence on the final result, proportional to its individual performance. The proposed distribution of the MRI slices and the further joining of the models has not been used before.

Tab. 2: Metrics performance results of average-weighted ensemble model

	Ensemble model
Accuracy	85.83%
Average precision score	84.82%
F-score	91.79%

The accuracy of 85.83% and average weighted F-score of 91.79% were achieved through combining all pre-trained models, see table 2. That is a significant improvement in comparison with a single model performance. In turn, AUC became 81.22%, see Figure 22. Moreover, attention should be paid on the amount of false-negative predictions (type II error), which is now equal to 0, see Figure 23. This fact is extremely important, while developing a decision-making support system for medical goals.

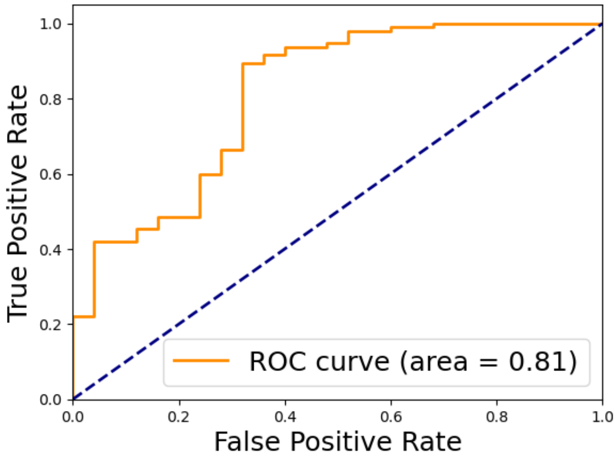


Fig. 22: Receiver operating characteristic curve of ensemble model

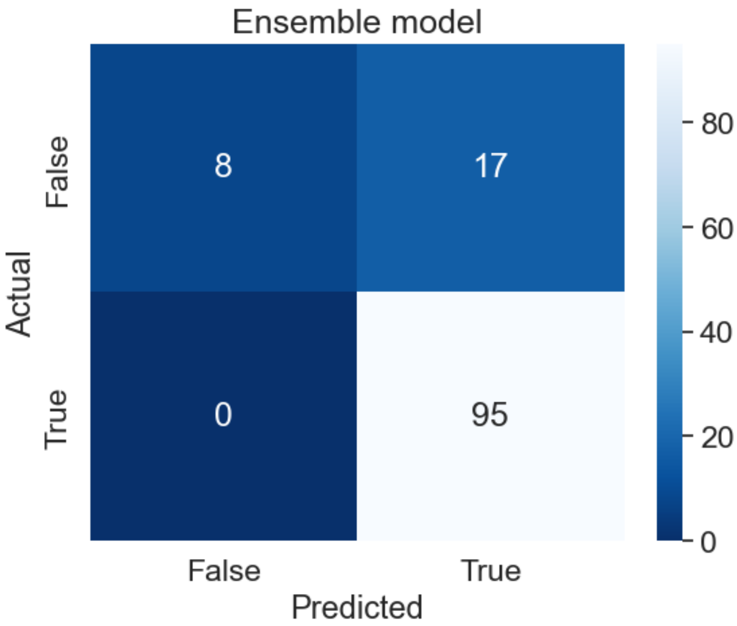


Fig. 23: Confusion matrix received by testing ensemble model

5 User interface design

In addition to the above components of a decision support system, the user interface should also be a focus of attention. Nowadays the effectiveness and flexibility of information technology depends to a large extent on the characteristics of system interface. Based on UI/UX technology, we carry out all stages of interface development, including collection, analysis and classification of incoming information about the designed object, development of structure, forms and layout of interface screens, selection of control elements, use of information graphics and design revision of the interface.

According to the requirements, the following functions have been identified and implemented:

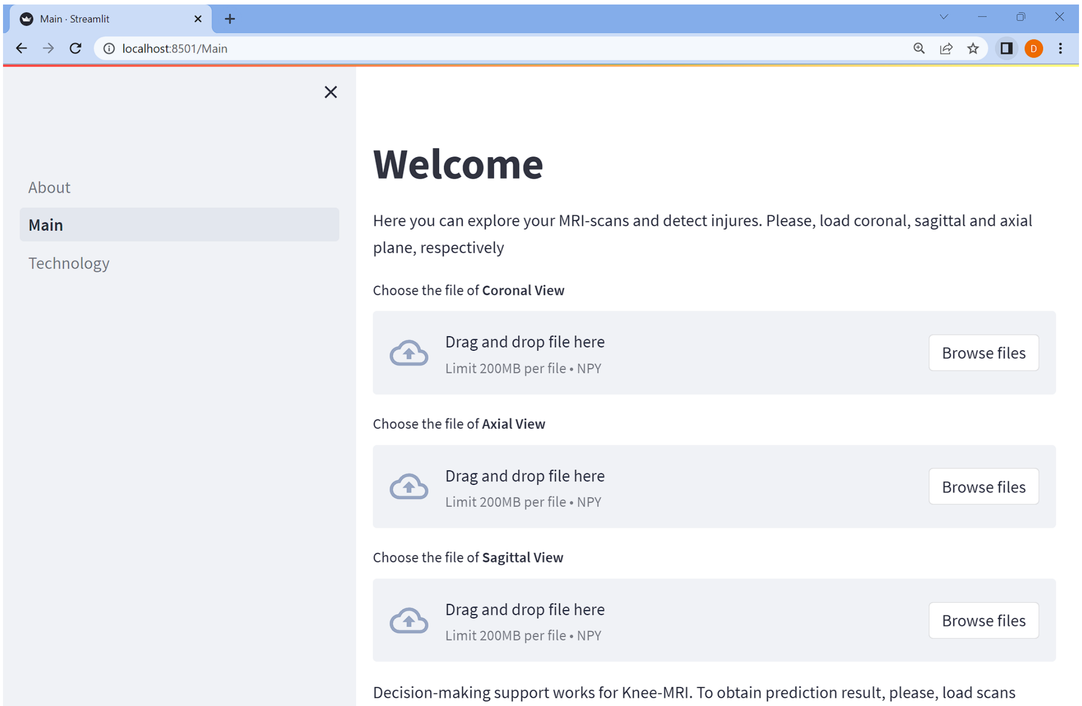
- ability to upload .npy files;
- ability to view files of 3 planes in the form of slices of the .jpg format;
- ability to adjust the required slices using the slider;
- ability to download a .jpg file to a local computer;
- ability to generate a gif file to view all slices simultaneously;
- ability to use a neural network for support decision-making during diagnosis;

A web-based interface has been realised via Streamlit framework, which is one of the most popular and flexible frameworks for developing and deploying web-based machine learning projects. The current application is compatible with the recent versions of the browsers Google Chrome, Firefox, Microsoft Edge and Safari. The content structure of the interface is defined by the following pages: the introduction, main and technology pages.

The introduction page consists short overview of application functionality and goals, for more information check Appendix B. It also has links to the most important information and instructive video.

Using the pop-up navigation menu, user can access to the main page of application. The initial state of the main page allows to upload files of each views, see Figure 24. The system processes and checks that the user has uploaded the correct data in terms of format and size. In case of an error, the system alerts the user and displays a message with instructions on how to correct current issue, see Appendix B.

If at least one of the files has been loaded, new functionality appears, see Figure 25. Sliders, which are illustrated at Point 1 (Figure 25), responsible for changing the slice, that is, when user changes the value of the slider, the image of the corresponding view automatically changes. By clicking zooming arrow at Point 2 (Figure 25) user can enlarge current image. Button (Point 3, Figure 25) corresponds to downloading chosen slice to a local device in jpeg-format. Ability to view all



Welcome

Here you can explore your MRI-scans and detect injures. Please, load coronal, sagittal and axial plane, respectively

Choose the file of **Coronal View**

Drag and drop file here
Limit 200MB per file • NPY

Choose the file of **Axial View**

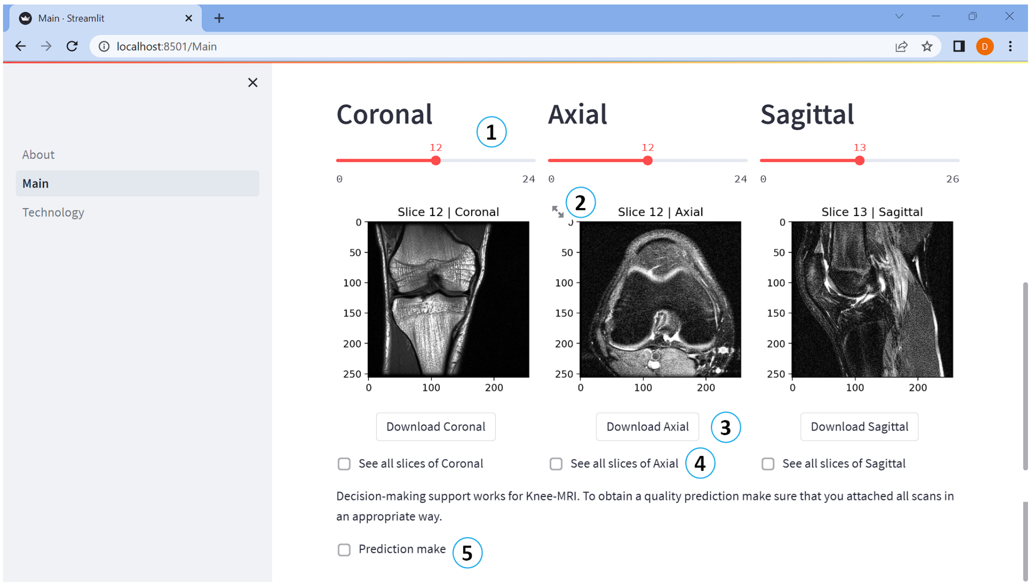
Drag and drop file here
Limit 200MB per file • NPY

Choose the file of **Sagittal View**

Drag and drop file here
Limit 200MB per file • NPY

Decision-making support works for Knee-MRI. To obtain prediction result, please, load scans

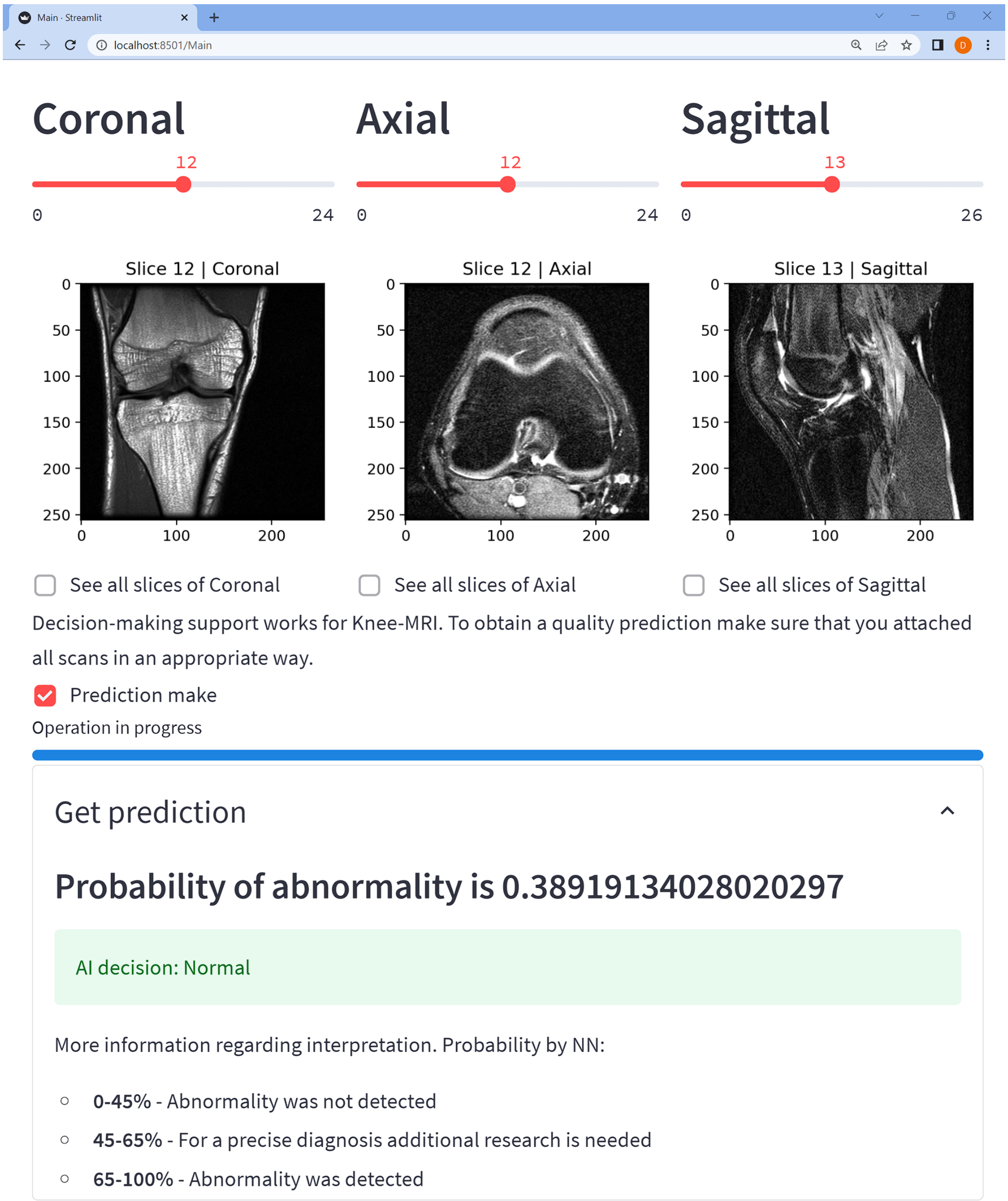
Fig. 24: Initial state of the main page of system



Decision-making support works for Knee-MRI. To obtain a quality prediction make sure that you attached all scans in an appropriate way.

Fig. 25: Main page functionality after uploading

slices simultaneously determines a check-box at Point 4 (Figure 25), it generates appropriate .gif file. By clicking check-box at Point 5 (Figure 25) user can start neural network for detecting abnormalities. The decision support system performs probabilistic indicators of the abnormality presence, see Figure 26. While processing progress of operation is shown; running on a local network takes from 5 to 7 seconds depending of input data. Further, according to given division it prepares final solution and displays it by the following background colours: Normal - green, Abnormal - red; or ask for additional research (blue indicator) in case of probability in range of 0.45-0.65, all cases are illustrated in Appendix B.



□ See all slices of Coronal □ See all slices of Axial □ See all slices of Sagittal

Decision-making support works for Knee-MRI. To obtain a quality prediction make sure that you attached all scans in an appropriate way.

☑ Prediction make

Operation in progress

Get prediction

Probability of abnormality is 0.38919134028020297

AI decision: Normal

More information regarding interpretation. Probability by NN:

- **0-45%** - Abnormality was not detected
- **45-65%** - For a precise diagnosis additional research is needed
- **65-100%** - Abnormality was detected

Fig. 26: Displaying abnormality detection results

Technology page consists information concerning methods that were used during developing AI-system and directions for further system's expansions, for more information check Appendix B.

6 Conclusions and directions of further work

This bachelor's thesis was dedicated to the problems of working with medical images, such as MRI. A decision support system for diagnosing knee joint abnormalities was implemented. We suggested a diagnostic method based on deep learning technology, such as convolutional neural networks with specific distribution of magnetic resonance imaging slices. It was proposed to split each scanned view into several areas in order to investigate majority parts of the organ whose damage is visible on the MRI. We are not aware of using similar method for diagnosing 3D medical images, thus the approach can be considered novel. The most suitable scans were the central scan and a 10% shift to each side for each of Coronal, Sagittal and Axial views.

During data pre-processing, it was found that analyzed dataset contains unbalanced data, what is a challenging task when training machine learning models. This problem was solved by data augmentation and using stochastics gradient-based optimizer. Therefore nine neural networks were trained and integrated with use of weighted average ensemble. For obtaining an objective assessment of a model performance addition metrics (F-measure and Area under the curve) were used. The accuracy of 85.83% and average weighted F-score of 91.79% were achieved through combining all pre-trained models. That is a significant improvement in comparison with a single model performance. In turn, AUC became 81.22%, what is better than performance of CNN VGG16 with AUC of 71% and almost reached performance of ResNet18 (84%).

Moreover, the amount of false-negative predictions(type II error) has been reached to 0. This fact is extremely important, while developing a decision-making support system for medical goals.

The model was implemented in Python programming language with use of TensorFlow, Keras libraries.

A web-based interface for decision support system has been realised via Streamlit framework and User Experience Design (UX) / User Interface Design (UI) technologies. Application provides an opportunity to quickly obtain a complete individual picture of the necessary sections (coronal, sagittal, axial) and slices for a particular patient, generating animations for continuous viewing. It also includes automated interpretation with probabilistic indicators of the abnormality presence using the aforementioned artificial intelligence technologies.

In the future the application can be extended by the following:

- Improving the accuracy of the decision support model. This can be achieved by updating and augmenting the data for training, changing slice distribution, using pre-trained neural networks, using 3D CNN or more complex architecture, adding other AI methods etc.

- Insert additional parameters. Adding parameters such as age, gender, previous results, and results of additional tests will improve the quality of the system's classification and make it more flexible.
- Speeding up the MRI process. MRI of one part of the human body takes 20 to 40 minutes, but in urgent cases, the result is needed in less time. According to a new above mentioned study, artificial intelligence can reconstruct raw images of fast magnetic resonance imaging into high-quality images, that have diagnostic value similar to that of traditional MRI. Combining proposed model with such method will allow a diagnosis to be made while the MRI-process is still ongoing.

BIBLIOGRAPHY

[1] Fukushima, K. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological Cybernetics*. 1980, vol. 36, p. 193–202.

[2] LeCun, Y., Jackel, L. D., Bottou, L., Cortes, C., Denker, J. S. et al. Learning algorithms for classification: A comparison on handwritten digit recognition. In:. 1995.

[3] Lundervold, A. S. and Lundervold, A. An overview of deep learning in medical imaging focusing on MRI. *Zeitschrift für Medizinische Physik*. 2019, vol. 29, no. 2, p. 102–127. DOI: https://doi.org/10.1016/j.zemedi.2018.11.002. ISSN 0939-3889. Special Issue: Deep Learning in Medical Physics.

[4] Ravi, D., Wong, C., Deligianni, F., Berthelot, M., Andreu Perez, J. et al. Deep Learning for Health Informatics. *IEEE Journal of Biomedical and Health Informatics*. 2017, vol. 21, no. 1, p. 4–21. DOI: 10.1109/JBHI.2016.2636665.

[5] Esteva, A., Kuprel, B., Novoa, R. A., Ko, J., Swetter, S. M. et al. Dermatologist-level classification of skin cancer with deep neural networks. *Nature*. Nature Publishing Group. 2017, vol. 542, no. 7639, p. 115–118.

[6] Shin, H.-C., Roth, H. R., Gao, M., Lu, L., Xu, Z. et al. Deep Convolutional Neural Networks for Computer-Aided Detection: CNN Architectures, Dataset Characteristics and Transfer Learning. *IEEE Transactions on Medical Imaging*. 2016, vol. 35, no. 5, p. 1285–1298. DOI: 10.1109/TMI.2016.2528162.

[7] Litjens, G., Kooi, T., Bejnordi, B. E., Setio, A. A. A., Ciompi, F. et al. A survey on deep learning in medical image analysis. *Medical image analysis*. Elsevier. 2017, vol. 42, p. 60–88.

[8] Bien, N., Rajpurkar, P., Ball, R. L., Irvin, J., Park, A. et al. Deep-learning-assisted diagnosis for knee magnetic resonance imaging: Development and retrospective validation of MRNet. *PLOS Medicine*. Public Library of Science. november 2018, vol. 15, p. 1–19. DOI: 10.1371/journal.pmed.1002699. Available at: `https://doi.org/10.1371/journal.pmed.1002699`.

[9] Nacey, N. C., Geeslin, M. G., Miller, G. W. and Pierce, J. L. Magnetic resonance imaging of the knee: an overview and update of conventional and

state of the art imaging. *Journal of Magnetic Resonance Imaging.* Wiley Online Library. 2017, vol. 45, no. 5, p. 1257–1275.

[10] Ortho, C. *Anatomy of the Knee* [https://comportho.com/anatomy/anatomy-of-the-knee/]. Accessed 23/02/23.

[11] YassineMrabet. *File:Human anatomy planes* [https://commons.wikimedia.org/wiki/File:Human_anatomy_planes.svg]. Accessed 23/02/23.

[12] Laiton Bonadiez, C., Sanchez Torres, G. and Branch Bedoya, J. Deep 3d neural network for brain structures segmentation using self-attention modules in mri images. *Sensors.* MDPI. 2022, vol. 22, no. 7, p. 2559.

[13] Bensalah, H., Njeh, I., Wali, A., Slima, M., BenHamida, A. et al. Deep Multi-Scale 3D Convolutional Neural Network (CNN) for MRI Gliomas Brain Tumor Classification. *Journal of Digital Imaging.* may 2020, vol. 33. DOI: 10.1007/s10278-020-00347-9.

[14] Haddadian, J. and Balamurali, M. Transfer Learning and Data Augmentation in the Diagnosis of Knee MRI. In: Long, G., Yu, X. and Wang, S., ed. *AI 2021: Advances in Artificial Intelligence.* Cham: Springer International Publishing, 2022, p. 452–463.

[15] Liu, F., Zhou, Z., Samsonov, A., Blankenbaker, D., Larison, W. et al. Deep learning approach for evaluating knee MR images: achieving high diagnostic performance for cartilage lesion detection. *Radiology.* Radiological Society of North America. 2018, vol. 289, no. 1, p. 160.

[16] Siouras, A., Moustakidis, S., Giannakidis, A., Chalatsis, G., Liampas, I. et al. Knee Injury Detection Using Deep Learning on MRI Studies: A Systematic Review. *Diagnostics.* 2022, vol. 12, no. 2. ISSN 2075-4418. Available at: https://www.mdpi.com/2075-4418/12/2/537.

[17] Johnson, P. M., Lin, D. J., Zbontar, J., Zitnick, C. L., Sriram, A. et al. Deep learning reconstruction enables prospectively accelerated clinical knee MRI. *Radiology.* Radiological Society of North America. 2023, p. 220425.

[18] McLean, D. Kevin Gurney, An Introduction to Neural Networks, University College London (UCL) Press, 1997. ISBN 1-85728-673-1 HB. £14.95. xi+234 pages. *Natural Language Engineering.* june 2000, vol. 6, p. 203–204. DOI: 10.1017/S1351324900002540.

[19] Bodyanskiy, Y. V. and Rudenko, O. Artificial neural networks: architectures, learning, applications. *TELETEX, Kharkov*. 2004.

[20] Abraham, A., Kumar, A., Melin, P. and Gandhi, N. *Intelligent Systems Design and Applications*. Vellore, India: Springer Cham, 2020.

[21] S. Mohamed, I. *Detection and Tracking of Pallets using a Laser Rangefinder and Machine Learning Techniques*. Dissertation.

[22] Ng, A. *COURSERA: Convolutional Neural Networks* [`https://www.coursera.org/learn/convolutional-neural-networks`]. Accessed 17/02/23.

[23] Fernández, A., García, S., Galar, M., Prati, R. C., Krawczyk, B. et al. *Learning from imbalanced data sets*. Springer, 2018.

[24] Ma, Y. and He, H. Imbalanced learning: foundations, algorithms, and applications. John Wiley & Sons. 2013.

[25] Obi, J. A Comparative Study of Several Classification Metrics and Their Performances on Data. *World Journal of Advanced Engineering Technology and Sciences*. february 2023, vol. 8, p. 308–314. DOI: 10.30574/wjaets.2023.8.1.0054.

[26] Wali, R. Xtreme Margin: A Tunable Loss Function for Binary Classification Problems. october 2022. DOI: 10.48550/arXiv.2211.00176.

[27] Smith, L. A disciplined approach to neural network hyper-parameters: Part 1 – learning rate, batch size, momentum, and weight decay. march 2018.

[28] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I. and Salakhutdinov, R. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research*. 2014, vol. 15, no. 56, p. 1929–1958. Available at: `http://jmlr.org/papers/v15/srivastava14a.html`.

[29] *Streamlit Framework* [`https://streamlit.io/`].

[30] StanfordMLGroup. *MRNet Dataset* [`https://stanfordmlgroup.github.io/competitions/mrnet/`].

[31] Derevianko, I. and Bondarev, S. Practical study of Cross-validation algorithms using Scikit-learn toolkit in Python. *Scientific Collection InterConf*. 2022, no. 136, p. 382–390.

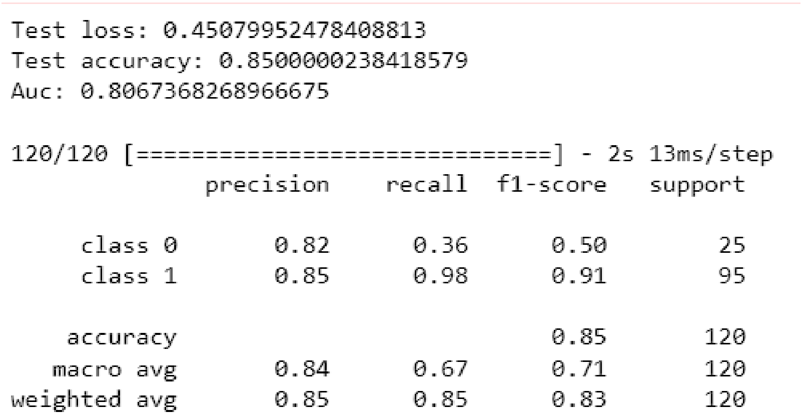
[32] Valova, I., Harris, C., Mai, T. and Gueorguieva, N. Optimization of Convolutional Neural Networks for Imbalanced Set Classification. *Procedia Computer Science*. 2020, vol. 176, p. 660–669. DOI: https://doi.org/10.1016/j.procs.2020.09.038. ISSN 1877-0509. Knowledge-Based and Intelligent Information and Engineering Systems: Proceedings of the 24th International Conference KES2020. Available at: https://www.sciencedirect.com/science/article/pii/S1877050920319335.

[33] Kingma, D. and Ba, J. Adam: A Method for Stochastic Optimization. *International Conference on Learning Representations*. december 2014.

[34] Cao, Y., Geddes, T. A., Yang, J. Y. H. and Yang, P. Ensemble deep learning in bioinformatics. *Nature Machine Intelligence*. Nature Publishing Group UK London. 2020, vol. 2, no. 9, p. 500–508.

LIST OF APPENDICES

A Training and testing of models



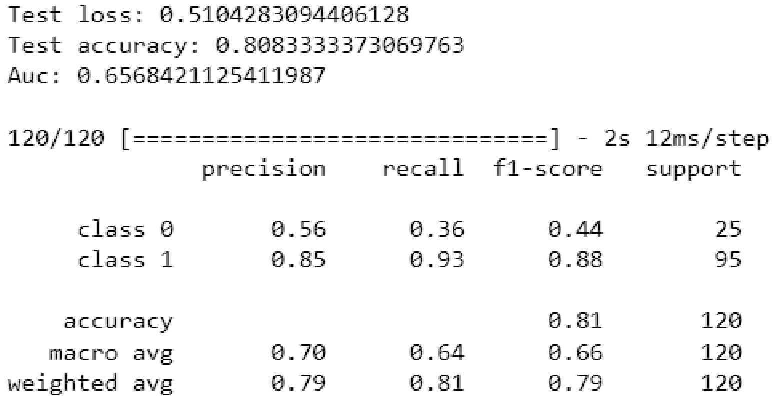
```
Test loss: 0.45079952478408813
Test accuracy: 0.8500000238418579
Auc: 0.8067368268966675

120/120 [==============================] - 2s 13ms/step
              precision    recall  f1-score   support

     class 0       0.82      0.36      0.50        25
     class 1       0.85      0.98      0.91        95

    accuracy                           0.85       120
   macro avg       0.84      0.67      0.71       120
weighted avg       0.85      0.85      0.83       120
```

Fig. A.1: Metrics of Sagittal model performance for central slices



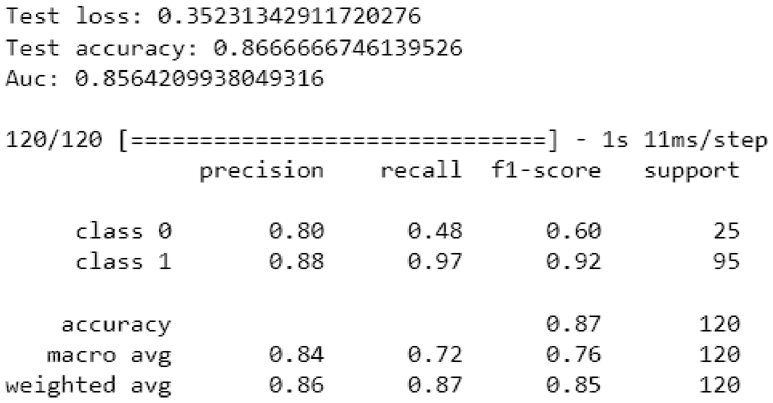
```
Test loss: 0.5104283094406128
Test accuracy: 0.8083333373069763
Auc: 0.6568421125411987

120/120 [==============================] - 2s 12ms/step
              precision    recall  f1-score   support

     class 0       0.56      0.36      0.44        25
     class 1       0.85      0.93      0.88        95

    accuracy                           0.81       120
   macro avg       0.70      0.64      0.66       120
weighted avg       0.79      0.81      0.79       120
```

Fig. A.2: Metrics of Coronal model performance for central slices



```
Test loss: 0.35231342911720276
Test accuracy: 0.8666666746139526
Auc: 0.8564209938049316

120/120 [==============================] - 1s 11ms/step
              precision    recall  f1-score   support

     class 0       0.80      0.48      0.60        25
     class 1       0.88      0.97      0.92        95

    accuracy                           0.87       120
   macro avg       0.84      0.72      0.76       120
weighted avg       0.86      0.87      0.85       120
```

Fig. A.3: Metrics of Axial model performance for central slices

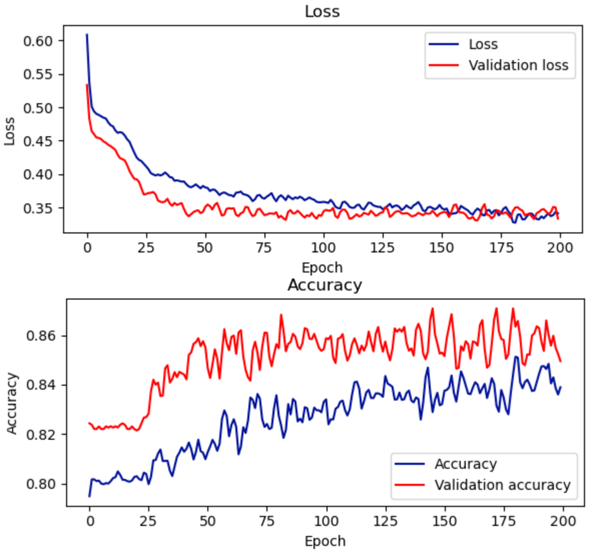


Fig. A.4: The training/validation behaviour based on accuracy and loss for 40% slices of Axial view

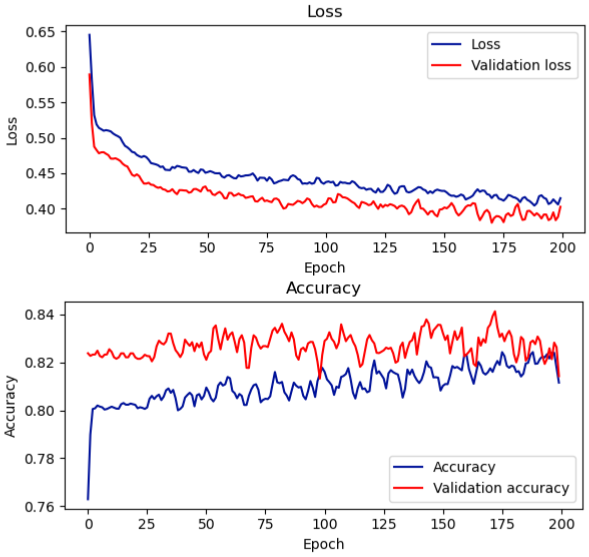


Fig. A.5: The training/validation behaviour based on accuracy and loss for 40% slices of Coronal view

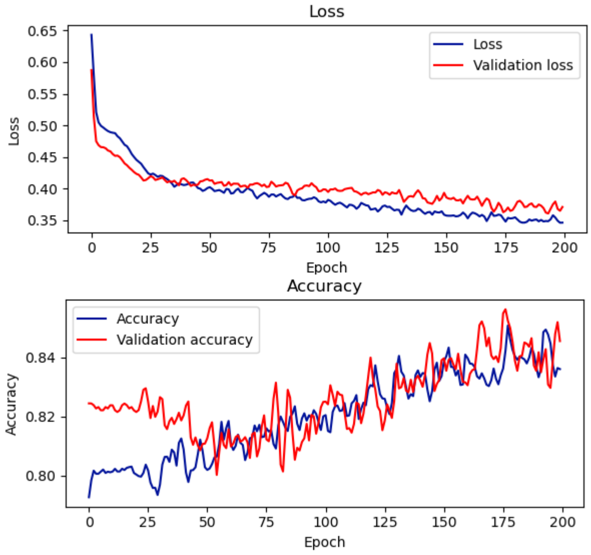


Fig. A.6: The training/validation behaviour based on accuracy and loss for 40% slices of Sagittal view

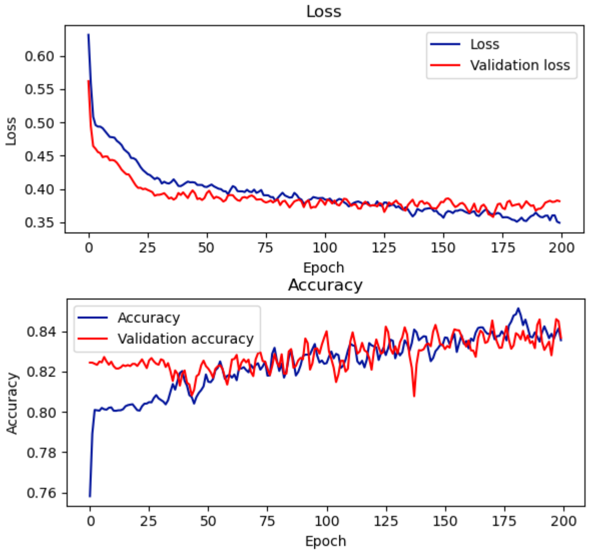


Fig. A.7: The training/validation behaviour based on accuracy and loss for 60% slices of Axial view

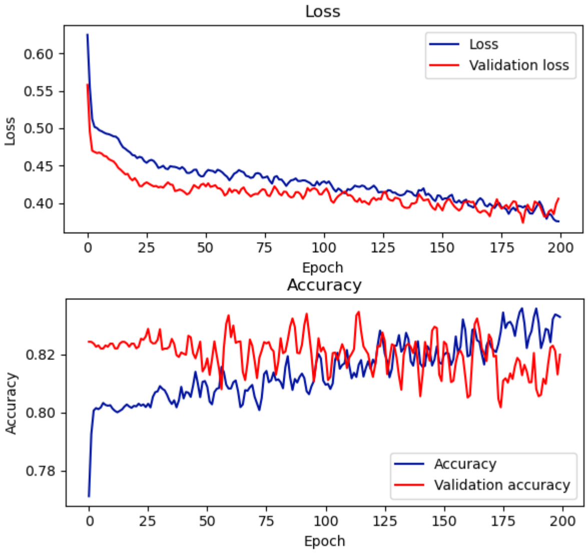


Fig. A.8: The training/validation behaviour based on accuracy and loss for 60% slices of Coronal view

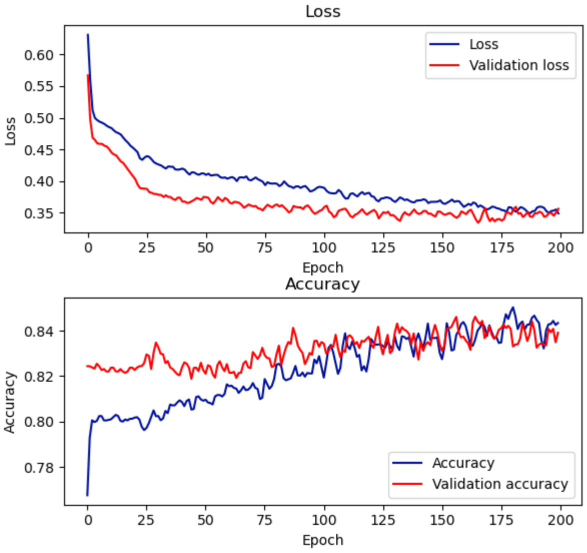


Fig. A.9: The training/validation behaviour based on accuracy and loss for 60% slices of Sagittal view

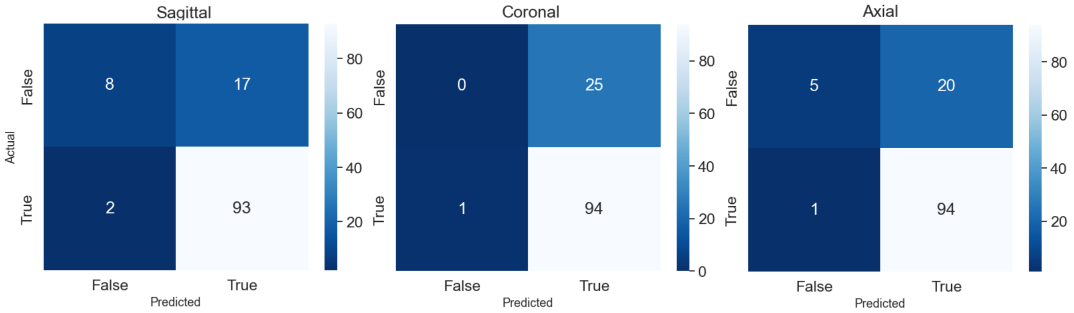


Fig. A.10: Confusion matrices received by testing 40% slices of Sagittal, Coronal and Axial view respectively

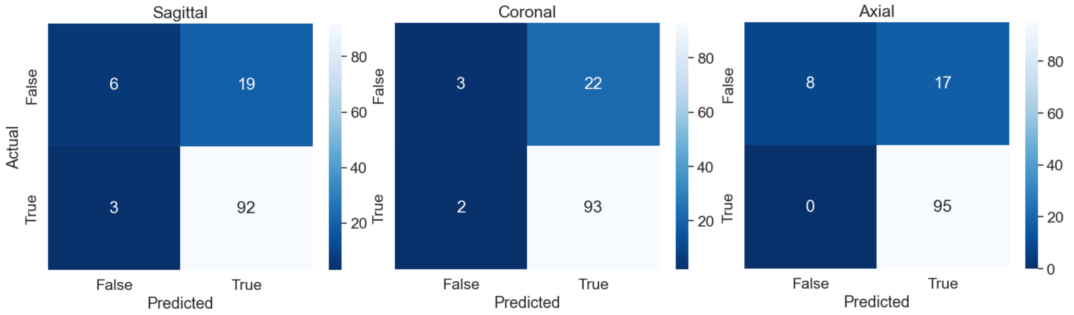
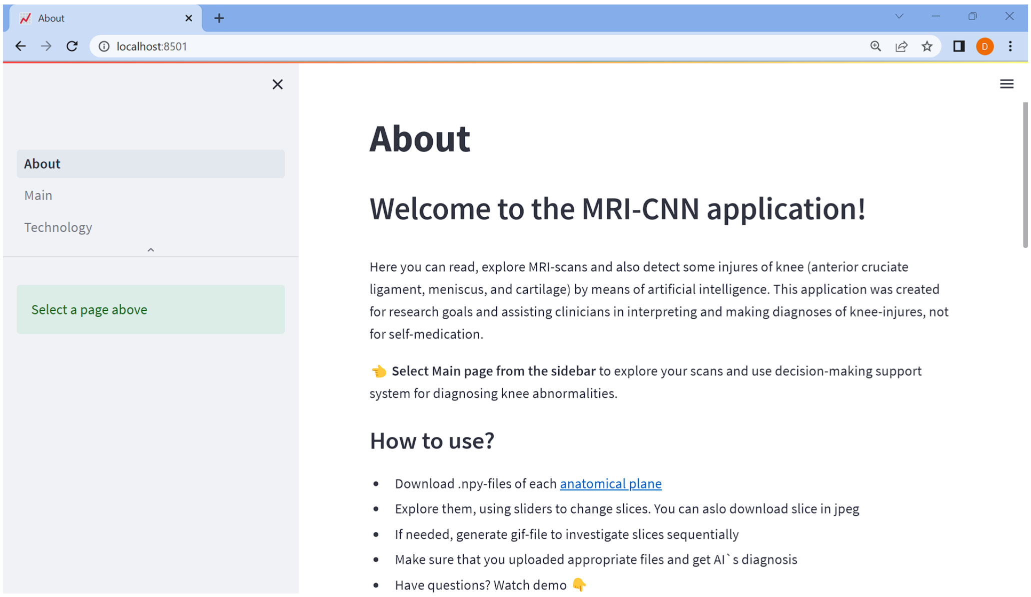


Fig. A.11: Confusion matrices received by testing 60% slices of Sagittal, Coronal and Axial view respectively

B Graphical user interface



Select a page above

About

Welcome to the MRI-CNN application!

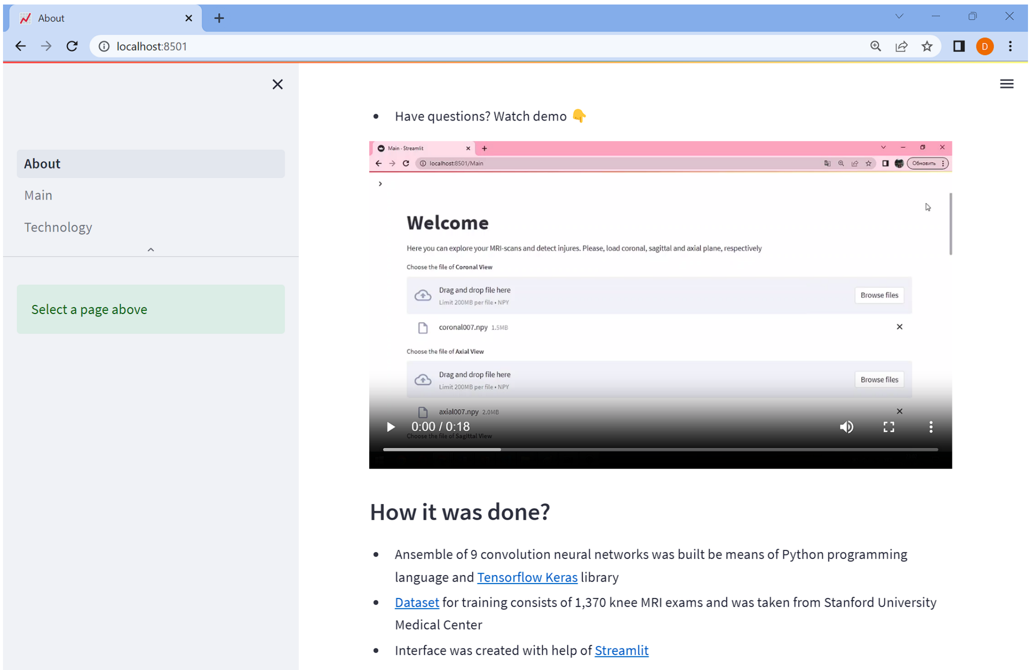
Here you can read, explore MRI-scans and also detect some injures of knee (anterior cruciate ligament, meniscus, and cartilage) by means of artificial intelligence. This application was created for research goals and assisting clinicians in interpreting and making diagnoses of knee-injures, not for self-medication.

👈 **Select Main page from the sidebar** to explore your scans and use decision-making support system for diagnosing knee abnormalities.

How to use?

- Download .npy-files of each anatomical plane
- Explore them, using sliders to change slices. You can aslo download slice in jpeg
- If needed, generate gif-file to investigate slices sequentially
- Make sure that you uploaded appropriate files and get AI`s diagnosis
- Have questions? Watch demo 👇

Fig. B.1: Initial state of introduction page



Select a page above

- Have questions? Watch demo 👇

Welcome

Here you can explore your MRI-scans and detect injures. Please, load coronal, sagittal and axial plane, respectively

Choose the file of **Coronal View**

Drag and drop file here
Limit 200MB per file • NPY

coronal007.npy 1.5MB

Choose the file of **Axial View**

Drag and drop file here
Limit 200MB per file • NPY

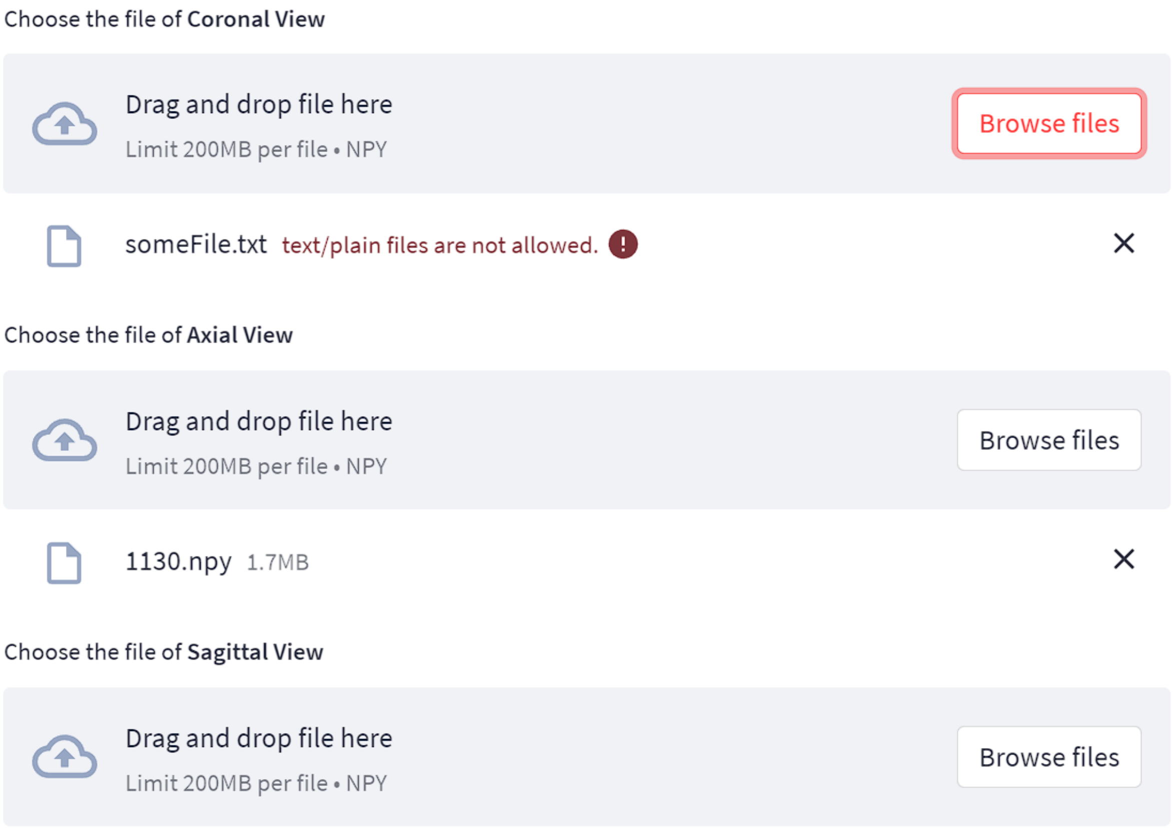
axial007.npy 2.0MB

Choose the file of Sagittal View

How it was done?

- Ansemble of 9 convolution neural networks was built be means of Python programming language and Tensorflow Keras library
- Dataset for training consists of 1,370 knee MRI exams and was taken from Stanford University Medical Center
- Interface was created with help of Streamlit

Fig. B.2: Introduction page. Instructive video



Choose the file of **Coronal View**

Drag and drop file here

Limit 200MB per file • NPY

someFile.txt text/plain files are not allowed.

Choose the file of **Axial View**

Drag and drop file here

Limit 200MB per file • NPY

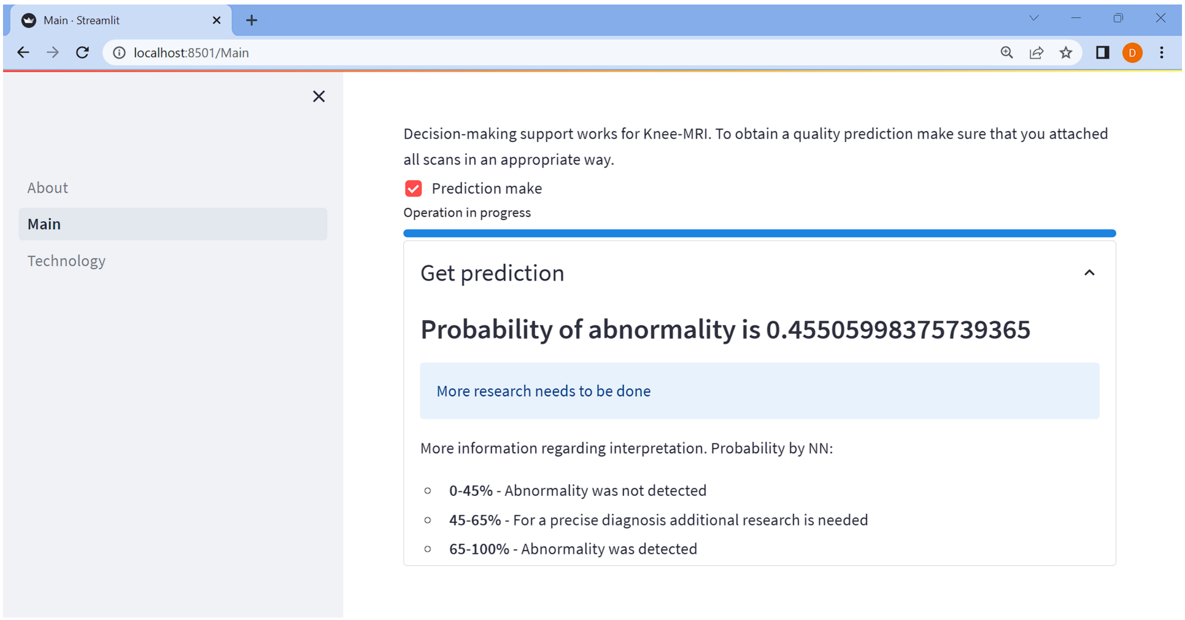
1130.npy 1.7MB

Choose the file of **Sagittal View**

Drag and drop file here

Limit 200MB per file • NPY

Fig. B.3: States of file uploader on main the page. Examples of alert while uploading unsuitable file-format, correct uploading and not yet uploaded



Decision-making support works for Knee-MRI. To obtain a quality prediction make sure that you attached all scans in an appropriate way.

☑ Prediction make

Operation in progress

Get prediction

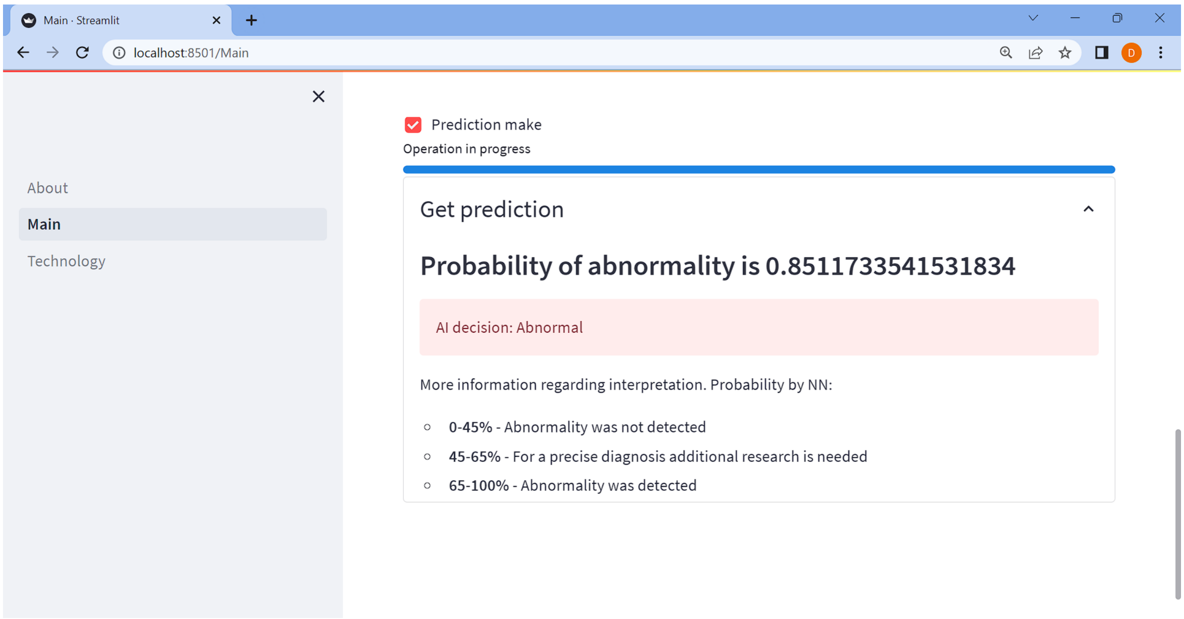
Probability of abnormality is 0.45505998375739365

More research needs to be done

More information regarding interpretation. Probability by NN:

- 0-45% - Abnormality was not detected
- 45-65% - For a precise diagnosis additional research is needed
- 65-100% - Abnormality was detected

Fig. B.4: Blue indicator while interpreting probability done by NN, when addition research is needed for precise AI solution



☑ Prediction make

Operation in progress

Get prediction

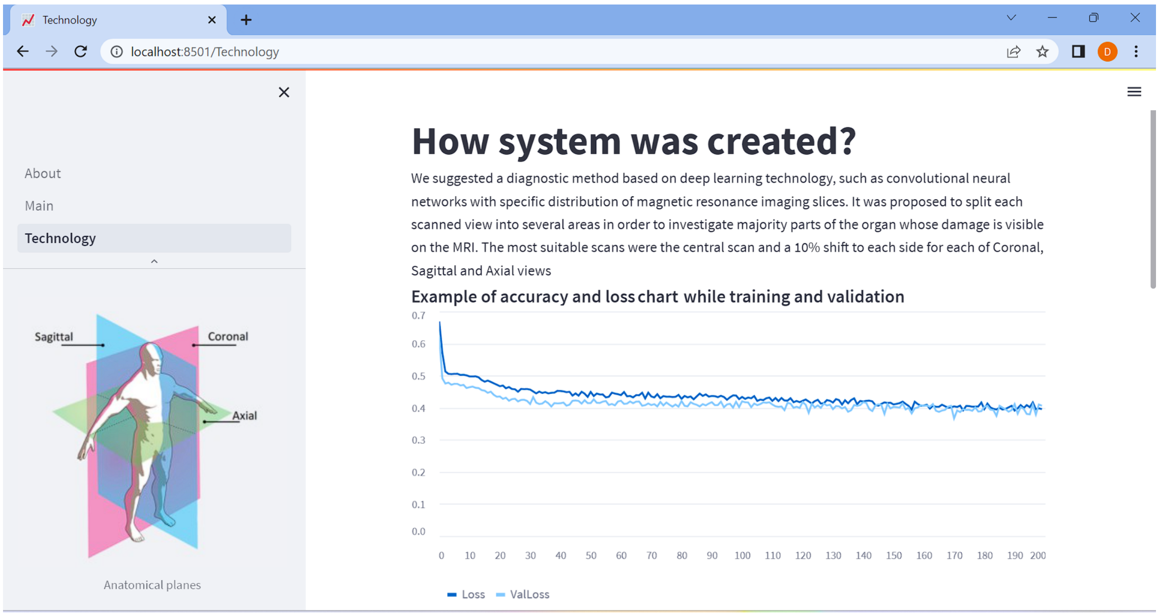
Probability of abnormality is 0.8511733541531834

AI decision: Abnormal

More information regarding interpretation. Probability by NN:

- **0-45%** - Abnormality was not detected
- **45-65%** - For a precise diagnosis additional research is needed
- **65-100%** - Abnormality was detected

Fig. B.5: Red indicator, while interpreting probability done by NN, when some abnormalities were detected

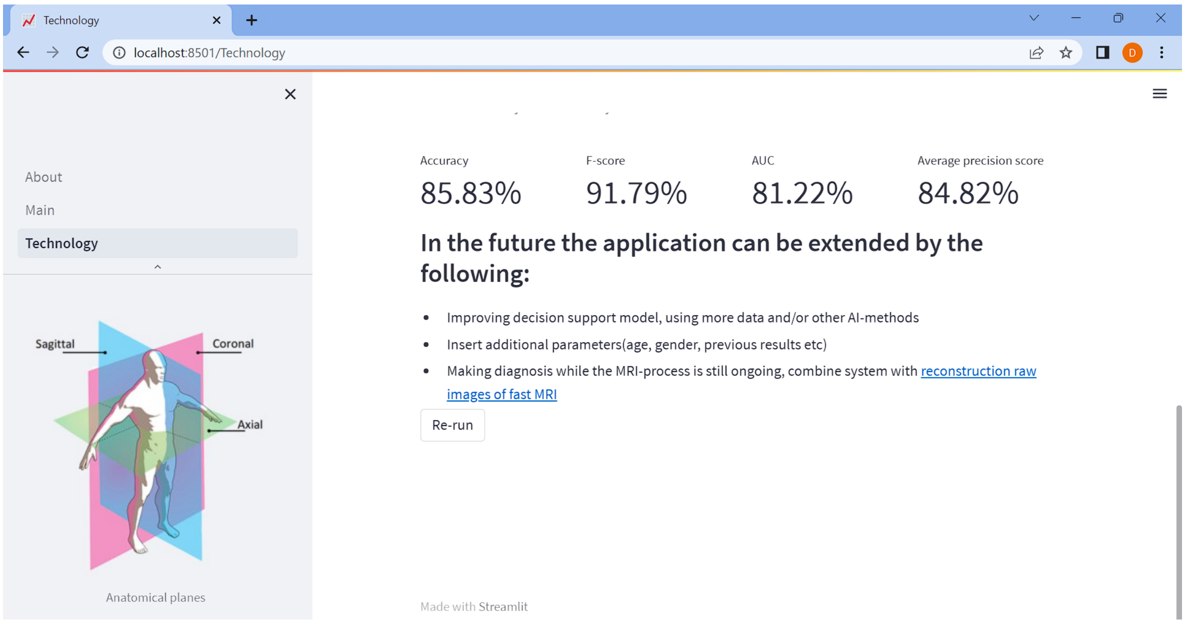


How system was created?

We suggested a diagnostic method based on deep learning technology, such as convolutional neural networks with specific distribution of magnetic resonance imaging slices. It was proposed to split each scanned view into several areas in order to investigate majority parts of the organ whose damage is visible on the MRI. The most suitable scans were the central scan and a 10% shift to each side for each of Coronal, Sagittal and Axial views

Example of accuracy and loss chart while training and validation

Fig. B.6: Initial state of technology page



Accuracy
85.83%

F-score
91.79%

AUC
81.22%

Average precision score
84.82%

In the future the application can be extended by the following:

- Improving decision support model, using more data and/or other AI-methods
- Insert additional parameters(age, gender, previous results etc)
- Making diagnosis while the MRI-process is still ongoing, combine system with reconstruction raw images of fast MRI

Fig. B.7: Technology page. Metrics and future work