

Slovenská technická univerzita v Bratislave
Fakulta informatiky a informačných technológií

FIIT-182905-86077

Bc. Timotej Zaťko

**Uplatnenie interpretovateľnosti a
vysvetliteľnosti neurónových sietí pri
vyhodnocovaní medicínskych obrazových
dát**

Diplomová práca

Študijný program: Inteligentné softvérové systémy

Študijný odbor: Informatika

Miesto vypracovania: Ústav počítačového inžinierstva a aplikovanej informatiky
(FIIT)

Vedúci práce: Ing. Martin Tamajka

Bratislava 2021

Zadanie diplomovej práce

Meno študenta: **Bc. Timotej Zat'ko**

Študijný program: Inteligentné softvérové systémy

Študijný odbor: Softvérové inžinierstvo – hlavný studijný odbor
Umelá inteligencia – vedľajší studijný odbor

Názov práce: **Uplatnenie interpretovateľnosti a vysvetliteľnosti neurónových sietí pri vyhodnocovaní medicínskych obrazových dát**

Samostatnou výskumnou a vývojovou činnosťou v rámci predmetov Diplomový projekt I, II, III vypracujte diplomovú prácu na tému, vyjadrenú vyššie uvedeným názvom tak, aby ste dosiahli tieto ciele:

Všeobecný cieľ:

Vypracovaním diplomovej práce preukážte, ako ste si osvojili metódy a postupy riešenia relatívne rozsiahlych projektov, schopnosť samostatne a tvorivo riešiť zložité úlohy aj výskumného charakteru v súlade so súčasnými metódami a postupmi študovaného odboru využívanými v príslušnej oblasti a schopnosť samostatne, tvorivo a kriticky pristupovať k analýze možných riešení a k tvorbe modelov.

Špecifický cieľ:

Vytvorte riešenie zodpovedajúce návrhu textu zadania, ktorý je prílohou tohto zadania. Návrh bližšie opisuje tému vyjadrenú názvom. Tento opis je záväzný, má však rámcový charakter, aby vznikol dostatočný priestor pre Vašu tvorivosť.

Riadťe sa pokynmi Vášho vedúceho.

Pokiaľ v priebehu riešenia, opierajúc sa o hlbšie poznanie súčasného stavu v príslušnej oblasti, alebo o priebežné výsledky Vášho riešenia, alebo o iné závažné skutočnosti, dospejete spoločne s Vašim vedúcim k presvedčeniu, že niečo v texte zadania a/alebo v názve by sa malo zmeniť, navrhnite zmenu. Zmena je spravidla možná len pri dosiahnutí kontrolného bodu.

Miesto vypracovania: Ústav počítačového inžinierstva a aplikovanej informatiky, FIIT STU Bratislava

Vedúci práce: **Ing. Martin Tamajka**

Termíny odovzdania:

Podľa harmonogramu štúdia platného pre semester, v ktorom máte príslušný predmet (Diplomový projekt I, II, III) absolvovať podľa Vášho študijného plánu

Predmety odovzdania:

V každom predmete dokument podľa pokynov na www.fit.stuba.sk v časti:
home > Informácie o > štúdiu > harmonogram štúdia > diplomový projekt.

V Bratislave dňa 17. 2. 2020

SLOVENSKÁ TECHNICKÁ UNIVERZITA
V BRATISLAVE

Fakulta Informatiky a informačných technológií

Ilkovičova 2, 842 16 Bratislava 4

1

doc. Ing. Peter Lacko, PhD.

riaditeľ Ústavu informatiky, informačných systémov
a softvérového inžinierstva

Čestne vyhlasujem, že som túto prácu vypracoval samostatne, na základe konzultácií a s použitím uvedenej literatúry.

V Bratislave, 17. máj 2021

Timotej Zaťko

Anotácia

Slovenská technická univerzita v Bratislave
Fakulta informatiky a informačných technológií
Študijný program: Inteligentné softvérové systémy

Autor: Bc. Timotej Zaťko

Diplomová práca: Uplatnenie interpretatívnosti a vysvetliteľnosti neurónových sietí pri vyhodnocovaní medicínskych obrazových dát

Vedúci diplomového projektu: Ing. Martin Tamajka

máj 2021

Súčasný vplyv umelej inteligencie na spoločnosť je nespochybniteľný. Využitie si už našla v rôznych oblastiach našich životov či už je to v smartfónoch pri odomykaní tvárou alebo najnovšie pri kontrole používania ochranného rúška pri vstupe do obchodov. Umelá inteligencia sa postupne dostáva do oblasti medicíny, kde má potenciál zachraňovať životy. Aby, teda mohla byť spoľahlivým pomocníkom doktorov pri diagnóze ochorení je nevyhnutné, aby jej rozhodnutie bolo možné vysvetliť.

V oblasti medicíny je možné použiť neurónových sietí, pretože dokážu veľmi dobre pracovať s obrazovými dátami, a tak sa dajú využiť napríklad pri diagnóze Alzheimerovej choroby z rádiologických snímkov. Ich problémom však je, že sa správajú ako "čierna skrinka" čo bráni v tom, aby sa dostali do bežnej praxe.

V tejto práci sme navrhli nový spôsob vysvetlovania rozhodnutí neurónových sietí, navrhli sme spôsob porovnania s existujúcimi prístupmi a overenia pri vysvetľovaní rozhodnutí neurónovej siete detekujúcich Alzheimerovu chorobu z MRI snímkov.

Annotation

Slovak University of Technology Bratislava
Faculty of Informatics and Information Technologies
Degree Course: Intelligent Software Systems

Author: Bc. Timotej Zatko

Diploma's Thesis: Application of interpretability and explainability of neural networks in the evaluation of medical images

Supervisor: Ing. Martin Tamajka

2021, May

The current impact of artificial intelligence on society is undeniable. It has already been used in various areas of our lives, whether it is in smartphones for unlocking via face recognition or, most recently, for controlling the use of protective masks when entering shops or groceries. Artificial intelligence is entering the field of medicine, where it has the potential to save lives. Thus, in order to be a reliable assistant to doctors for example in the diagnosis of the disease, it is necessary that its decisions can be explained.

In the field of medicine, the usage of neural networks is possible, because they can work very well with image data, and so they can be used, for example, in the diagnosis of Alzheimer's disease from radiological images. However, their problem is that they behave like a "black box" which prevents them from getting into common practice.

In this work, we proposed a novel method of interpreting neural networks, we proposed a process of comparison with existing approaches and verification in explaining the neural network decisions detecting Alzheimer disease from MRI images.

Pod'akovanie

Ďakujem môjmu školiteľovi Ing. Martinovi Tamajkovi za odbornú pomoc a vedenie pri tvorbe tejto práce.

Obsah

1	Úvod	7
2	Analýza	9
2.1	Alzheimerova choroba	9
2.1.1	Diagnostika Alzheimerovej choroby	10
2.1.2	Biologické ukazovatele	10
2.1.3	Obrazové a rádiologické ukazovatele	11
2.2	Neurónové siete	12
2.2.1	Neurón	14
2.2.2	Dopredné neurónové siete	15
2.2.3	Konvolučné neurónové siete	15
2.2.4	Architektúry konvolučných neurónových sietí	18
2.2.5	Interpretovanie neurónovej siete	19
2.2.6	Vysvetľovanie predikcie neurónovej siete	20
2.2.6.1	Analýza senzitivity	22
2.2.6.2	LRP (angl. layer-wiser relevance propagation) . . .	23
2.2.6.3	Riadená spätná propagácia (angl. Guided Backprop)	24
2.2.6.4	GradCAM	25
2.2.6.5	Riadený GradCAM (angl. Guided GradCAM) . . .	25
2.2.6.6	RISE - Randomized Input Sampling for Explanation	26
2.3	Využitie neurónových sietí pri odhaľovaní Alzheimerovej choroby .	28

Obsah

2.3.1	Vysvetľovanie rozhodnutí neurónových sietí detegujúcich Alzheimerovu chorobu	31
2.4	Spracovanie obrazu	32
2.4.1	Rekonštrukcia obrazu	33
2.5	Zhrnutie	34
3	Ciele práce	37
3.1	Vytvorenie novej, alebo vylepšenie existujúcej metódy pre vysvetľovanie rozhodnutí neurónových sietí	37
3.2	Využitie vytvorennej metódy na určenie miery správnosti modelu neurónovej siete detegujúcej Alzheimerovu chorobu	38
4	Návrh riešenia	39
4.1	RISEI - Randomized Input Sampling for Explanation with Inpainting	40
4.2	Overenie riešenia	43
4.2.1	Dátová sada	43
4.2.2	Experimenty	43
4.2.2.1	Určenie kvality metódy vysvetľovania rozhodnutí modelu	44
4.2.2.2	Určenie správnosti modelu	44
4.3	Zhrnutie	45
5	Implementácia	47
5.1	Metóda RISEI	47
5.1.1	Generovanie masiek	47
5.1.2	Vytvorenie tepelných máp	55
5.1.3	Vyhodnotenie tepelných máp	57
5.1.3.1	Metriky insertion & deletion	57
5.2	Model na detekciu Alzheimerovej choroby na základe MRI snímok .	58
5.2.1	Dátová sada	59
5.2.1.1	Predspracovanie	59

Obsah

5.2.1.2	Augmentácie	60
5.2.2	Model	61
5.2.3	Trénovanie	63
5.3	Zhrnutie	66
6	Overenie riešenia	69
6.1	Experimenty	69
6.1.1	Porovnanie metódy RISE na natrénovaných modeloch . . .	69
6.1.2	Stabilita tepelných máp	70
6.1.2.1	Experiment 1 (jeden snímok)	71
6.1.2.2	Experiment 2 (viacero snímkov)	71
6.1.3	RISE vs RISEI	73
6.1.4	Hľadanie optimálnych parametrov	76
6.1.5	Porovnanie s existujúcimi metódami	76
6.2	Zhrnutie	76
7	Zhodnotenie	77
Literatúra		79
Dodatok A	Plán práce	
A.1	Letný semester - DP1	
A.2	Zimný semester - DP2	
A.2.1	Vyjadrenie k plneniu plánu	
A.3	Letný semester - DP3	
A.3.1	Vyjadrenie k plneniu plánu	
Dodatok B	Technická dokumentácia	
B.1	Príprava vývojového prostredia	
B.2	Závislosti (použité knižnice)	
B.3	Technické riešenie	
B.4	Moduly	
B.4.1	Modul: src.risei	

Obsah

B.4.1.1 Trieda: RISEI

Dodatok C Opis digitálnej časti práce

Zoznam použitých skratiek

AD angl. Alzheimier disease (Alzheimerova choroba) – používa sa na označenie pacientov trpiacich Alzheimerovou chorobou

AUC angl. area under curve (plocha pod krivkou)

CN angl. cognitive normal (kognitívne zdravý) – používa sa na označenie pacientov bez kognitívneho poškodenia (tj. zdravých jedincov)

MCI angl. mild cognitive impairment (mierne kognitívne poškodenie) – používa sa na označenie pacientov s miernym kognitívnym poškodením

MRI angl. magnetic resonance imaging (magnetická rezonancia) – je spôsob tvorby rádiologických snímkov ľudského tela, používa sa pri diagnostike Alzheimerovej choroby

PET angl. positron emission tomography (pozitrónová emisná tomografia) – je spôsob tvorby rádiologických snímkov ľudského tela, používa sa pri diagnostike Alzheimerovej choroby

TP angl. true positive (správne pozitívny) – výraz pre pozorovanie, ktoré je pozitívne a bolo označené správne

Obsah

TN angl. true negative (správne negatívny) – výraz pre pozorovanie, ktoré je negatívne a bolo označené správne

FP angl. false positive (nesprávne pozitívny) – výraz pre pozorovanie, ktoré je negatívne, ale bolo neprávne označené ako pozitívne

FN angl. false negative (nesprávne negatívny) – výraz pre pozorovanie, ktoré je pozitívne, ale bolo neprávne označené ako negatívne

1. Úvod

Umelá inteligencia sa už dávno stala súčasťou nášho každodenného života. Prichádzame s ňou do kontaktu neustále, keď odomykáme telefón vlastnou tvárou alebo keď pomocou prekladača prekladáme text to iného jazyka. Jej využitie je tiež rozšírené v oblasti medicíny, kde má potenciál zachraňovať životy. Využíva sa pri výrobe liekov, monitorovaní zdravia, analýze zdravotných plánov, chirurgických zákrokov a aj pri odhaľovaní chorôb [1]. Práve pri odhaľovaní chorôb sa častokrát využívajú hlboké neurónové siete, a to napríklad pri detekcii rakoviny kože, rakoviny pľúc alebo Alzheimerovej choroby z obrazových dát.

Neurónovým sieťam sa už podarilo dosiahnuť také dobré výsledky, že sú porovnatelné s expertmi v medicínskej oblasti. Ich problémom však je, že sa správajú ako "čierna skrinka", čo v oblasti medicíny nie je žiadúce. Preto je nevyhnutné, aby boli rozhodnutia neurónovej siete interpretovateľné a pacient s lekárom vedeli, na základe čoho sa neúronová sieť rozhodla. Lekári by si mali svoje rozhodnutia viedieť obhájiť. Aby sa teda neurónové siete mohli stať bežným pomocníkom lekárov, je vysvetliteľnosť ich rozhodnutí dôležitá. Avšak toto nie je jedinou motiváciou pre vysvetliteľnosť rozhodnutí neurónových sietí. Novovznikajúce regulácie, ako napríklad pripravovaná regulácia s názvom "Right to Explanation" od Európskej Únie [2] vyžadujú vysvetliteľnosť systémov umelej inteligencie. Motivácia je teda aj legislatívna.

2. Analýza

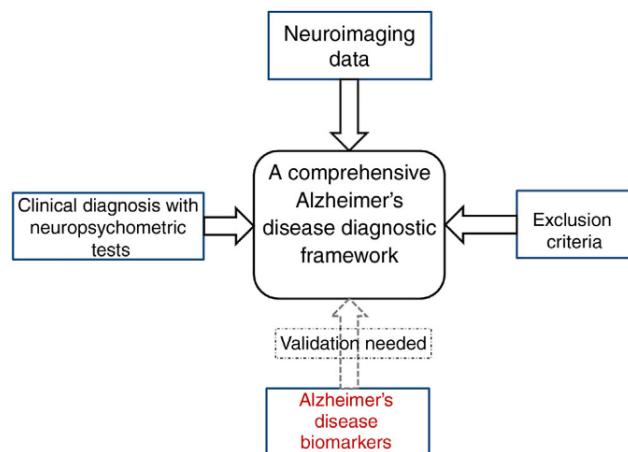
2.1 Alzheimerova choroba

Alzheimerova choroba je najčastejšou príčinou demencie. Prvotné príznaky tejto choroby sú zhoršenie pamäti, zabúdanie nedávnych udalostí, mien, neschopnosť rozoznávať známe miesta či orientovať sa v čase [3]. Jej priebeh sa vyznačuje postupným poklesom kognitívnych funkcií, postupným zhoršením pamäte, myslenia, rozprávania a schopnosti učenia sa [4]. Najčastejšie sa vyskytuje u ľudí starších ako 65 rokov, s pravdepodobnosťou výskytu až 50% po dovršení 85 rokov života [4]. S narastajúcim vekom človeka sa zvyšuje pravdepodobnosť ochorenia. Pravdepodobnosť ochorenia zvyšujú taktiež úrazy hlavy, poruchy prekrvenia mozgu, pozitívna rodinná anamnéza či vzdelanie (protože ľudia s nižším vzdelaním majú väčšie riziko rozvoja tohto ochorenia) [3]. Toto ochorenie sa vyskytuje častejšie u žien ako u mužov, v pomere 2:1 [5].

Alzheimerova choroba nie je “iba” o strate pamäti, ale aj šiestou najčastejšou príčinou smrti v USA [6]. Medzi rokmi 2000 až 2017 sa počet úmrtí v USA viac ako zdvojnásobil [6]. Ľudia starší ako 65 rokov ktorým bola diagnostikovaná táto choroba sa v priemere dožívajú 4 až 8 rokov po jej diagnóze [6].

2.1.1 Diagnostika Alzheimerovej choroby

Alzheimerova býva diagnostikovaná kombináciou viacerých ukazovateľov. Pri určovaní diagnózy sa používajú neuropsychometrické (kognitívne) testy, rádiologické snímky (angl. neuroimaging data), biologické ukazovatele a špecifické kritériá, na základe ktorých je možné vylúčenie iných chorôb u pacienta z jeho história vývoja ochorenia [5]. T. Khan zadefinoval tieto ukazovatele do tzv. komplexného rámca pre diagnózu Alzheimerovej choroby (Obr. 2.1). V súčasnosti sa v tejto oblasti skúmajú biologické ukazovateľe (ich identifikácia a použitie), keďže používanie (a teda aj vytvorenie) rádiologických ukazovateľov je drahé [5] (vyžaduje si to zaškolený personál a vybavenie). Biologické ukazovateľe zatiaľ nie sú dostatočne spoľahlivé [5].



Obr. 2.1: **Komplexný rámec pre diagnózu alzheimerovej choroby.** Pozostáva z neuropsychometrických testov, rádiologických snímok (z PET, MRI...), biologických ukazovateľov (napr. úrovne hladín určitých proteínov v krvnej plazme) Alzheimerovej choroby a kritérií vylúčenia iných neurologických chorôb.[5]

2.1.2 Biologické ukazovatele

Biologické ukazovatele (angl. biomarkers) sú merateľné biologické ukazovatele slúžiace na detekciu prítomnosti choroby. National Institute of Health definguje bio-

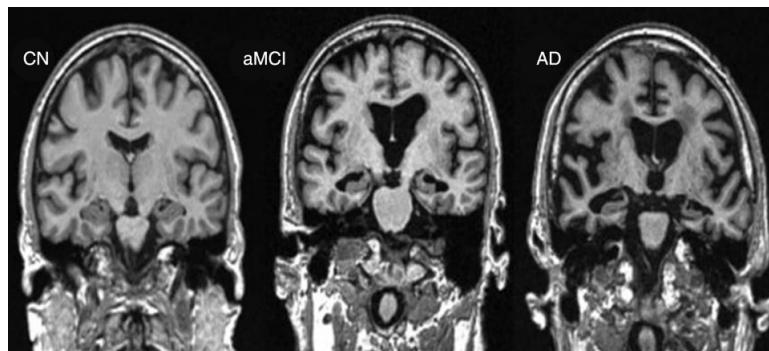
logický ukazovateľ ako indikátor určitého objektívneho merania a hodnotenia biologického procesu, patogénneho procesu alebo farmakologického hodnotenia terapeutickej účinnosti [7]. Alzheimerova choroba môže byť identifikovaná sledovaním týchto biologických ukazovateľov napríklad v krvnej plazme [5] alebo v mozgovo-miechovej tekutine (angl. cerebrospinal fluid) (ako úrovne hladín proteínov P-tau and A β 42) [5] (angl. cerebrospinal fluid).

2.1.3 Obrazové a rádiologické ukazovatele

Identifikovanie Alzheimerovej choroby je v súčasnosti možné aj z rádiologických snímok. Tvorba rádiologických snímok je v súčasnosti možná pomocou techník akými sú počítačová tomografia s jednou fotónovou emisiou (angl. single-photon emission computed tomography - SPECT), pozitrónová emisná tomografia (angl. positron emission tomography PET), počítačová tomografia (angl. computed tomography - CT), magnetická rezonancia (magnetic resonance imaging - MRI) a magnetická rezonančná spektroskopia (angl. magnetic resonance spectroscopy - MRS) [5].

Snímky z magnetickej rezonancie (MRI) dokážu zachytiť odumieranie tkaniva (na základe biologických procesov), ktoré sa odohráva v rôznych častiach mozgu [5]. Príklad takého snímku sa nachádza na obrázku 2.2.

Snímky z pozitrónovej emisnej tomografie (PET) dokážu zachytiť pokles mozgovej aktivity, ktorá je u pacientov s Alzheimerovou chorobou nižšia. Mozgová aktivita odráža úroveň metabolizmu glukózy v mozgu. Na miestach v mozgu, ktoré sú touto chorobou postihnuté, je úroveň metabolizmu glukózy nižšia. Tento jav je znázornený na obrázku 2.3.



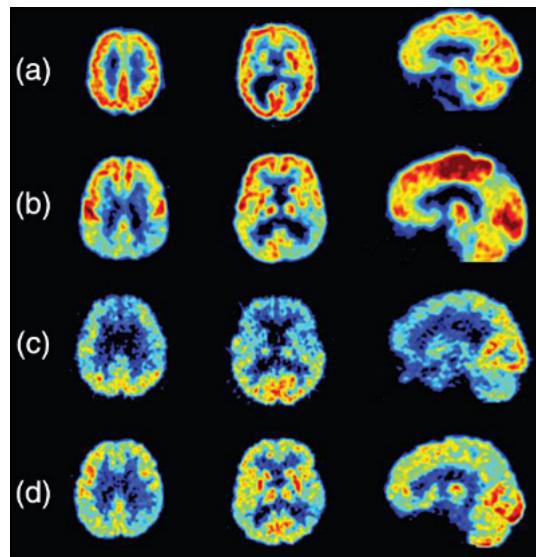
Obr. 2.2: **Typické odumieranie mozgového tkaniva zachytené magnetickou rezonanciou.** Obrázok zľava, označený ako CN (angl. cognitive normal), reprezentuje kognitívne normálneho jedinca. Obrázok v strede, označený ako aMCI (angl. amnestic mild cognitive impairment) reprezentuje jedinca s miernym kognitívnym poškodením - na obrázku je zreteľný úbytok mozgového tkaniva (šedá farba) najmä v strede mozgu (ale aj na jeho okrajoch) oproti kognitívne normálnemu jedincovi. Posledný obrázok označený ako AD (angl. Alzheimer's disease) reprezentuje jedinca s Alzheimerovou chorobou - na obrázku je zreteľný značný úbudok mozgového tkaniva. [5]

2.2 Neurónové siete

Neurónové siete patria medzi obľúbené techniky strojového učenia. Špeciálnou kategóriou sú hlboké neurónové siete (často označované skratkou DNN od angl. deep neural network), ktoré sa oproti obyčajným neurónovým sieťam odlišujú počtom vrstiev. Hlbokým neurónovým sieťam sa doteraz podarilo dosiahnuť v mnohých úlohách výnimočné výsledky, v ktorých častokrát už dokázali prekonať človeka. V našej oblasti obrazových rádiologických dát sa používajú najmä konvolučné neurónové siete.

Haykin et al. [8] definujú neurónovú sieť nasledovne:

Neurónová sieť je veľký paralelný distribuovaný procesor tvorený jednoduchými procesorovými jednotkami, ktorý má prirodzený sklon ukladať poznatky a sprístupňovať ich na použitie. Ľudskému mozgu sa podobá v dvoch aspektoch:



Obr. 2.3: **Snímky normálneho mozgu a mozgu postihnutého Alzheimerovou chorobou z pozitrónovej emisnej tomografie (PET).** [5] Na obrázkoch je viditeľná úroveň metabolizmu glukózy, u pacientov s Alzheimerovou chorobou je táto úroveň nižšia (žltá a modrá farba na obrázkoch). (a) Mozog kognitívne zdravého jedinca - vyznačuje sa vyššou mozgovou aktivitou. (b) Mozog vyznačujúci symptómy Alzheimerovej choroby - je vidieť nižšiu aktivitu v niektorých častiach mozgu oproti kognitívne zdravému jedincovi. (c) Mozog postihnutý frontotemporálnou demenciou (angl. frontotemporal dementia), tiež sa vyznačuje nižšou mozgovou aktivitou. (d) Mozog postihnutý Alzheimerovou chorobou.

1. Neurónová sieť získava vedomosti zo svojho prostredia prostredníctvom procesu učenia.
2. Na uchovanie získaných poznatkov sa používajú prepojenia medzi jednotlivými neurónami.

Neurónové siete sú teda inšpirované fungovaním mozgu človeka, keďže napodobňujú jeho fungovanie.

2.2.1 Neurón

Neurón (Obr. 2.4) je základnou stavebnou jednotkou neurónových sietí. Matematicky sa dá zapísat ako [8]:

$$y_k = \varphi(b_k + \sum_{j=1}^m w_{kj} \cdot x_j) \quad (2.1)$$

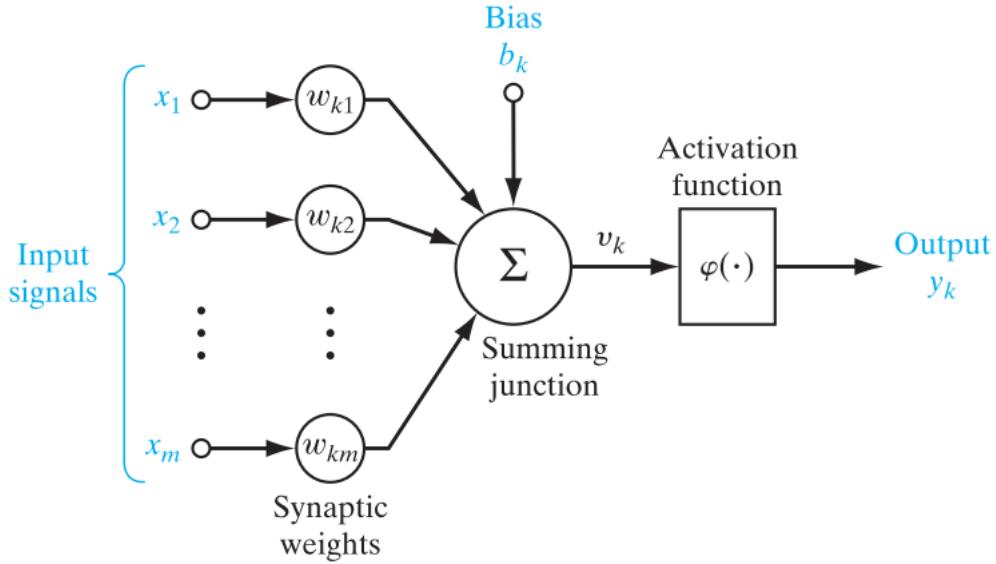
Kde:

- x_1, x_2, \dots, x_m sú vstupné signály
- $w_{k1}, w_{k2}, \dots, w_{km}$ sú váhy neurónu k
- b_k je sklon neurónu k
- $\varphi(\dots)$ je aktivačná funkcia
- y_k je výsupný signál neurónu k

Parametrami, ktoré sa počas trénoania neurónovej siete menia sú váhy w_{kj} a sklon b_k , tieto parametre sú takzvané trénovateľné parametre. Tieto parametre sa upravujú pri spätej propagácii (angl. backpropagation), kedy sa minimalizuje chybová funkcia (angl. loss function).

V neurónových sietiach s viac vrstvami sa stávajú výstupné signály y neurónov jednej vrstvy vstupom x do ďalšej.

Aktivačná funkcia zabezpečuje nelinearitu neurónu, medzi najpoužívanejšie aktivačné funkcie patria Sigmoid ($S(x) = \frac{1}{1+e^{-x}}$), Tanh alebo ReLU ($ReLU(x) = \max(0, x)$). Jednotlivé neuróny si môžeme predstaviť ako nelineárne funkcie, ktorých spojením do viac vrstiev dokážu skladať ešte zložitejšie a komplexnejšie funkcie.



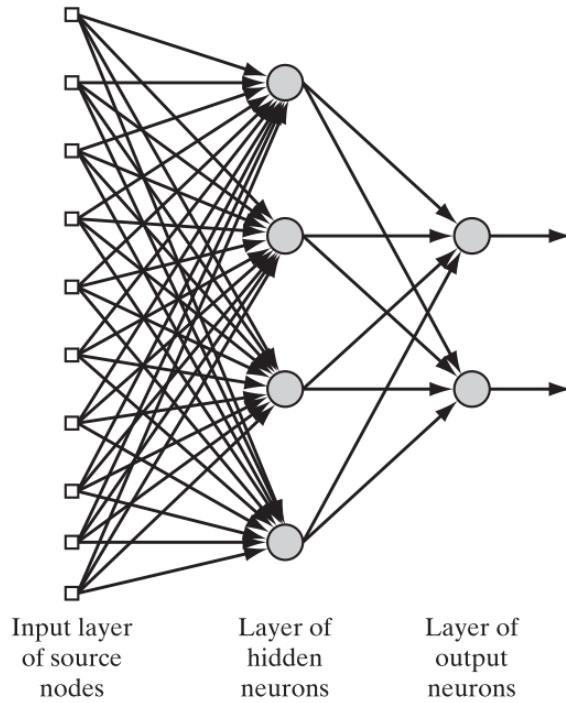
Obr. 2.4: **Model neurónu.** [8] Neurón sa skladá zo vstupných signálov a váh, ktoré sú na tieto signály aplikované, sklon (b_k - angl. bias) a aktivačnej funkcie, ktorá zabezpečuje nelinearitu. Vzorec 2.1 matematicky popisuje správanie neurónu.

2.2.2 Dopredné neurónové siete

Dopredné neurónové siete (Obr. 2.5) sú jednou z mnoha architektúr neurónových sietí. V dopredných neurónových sieťach výstupný signál z jednej vrstvy nemôže byť vstupným signálom do jej predošej vrstvy. Signál je prenášaný iba v jednom smere – dopredu. Dopredné neurónové siete sa môžu skladať z viacerých vrstiev. Základom je vstupná a výstupná vrstva a ľubovoľný počet skrytých vrstiev. Ich počet nie je limitovaný, avšak v hlbokých neurónových sieťach (tj. sieťach s veľkým početom skrytých vrstiev) môže nastáť problém miznúceho gradientu.

2.2.3 Konvolučné neurónové siete

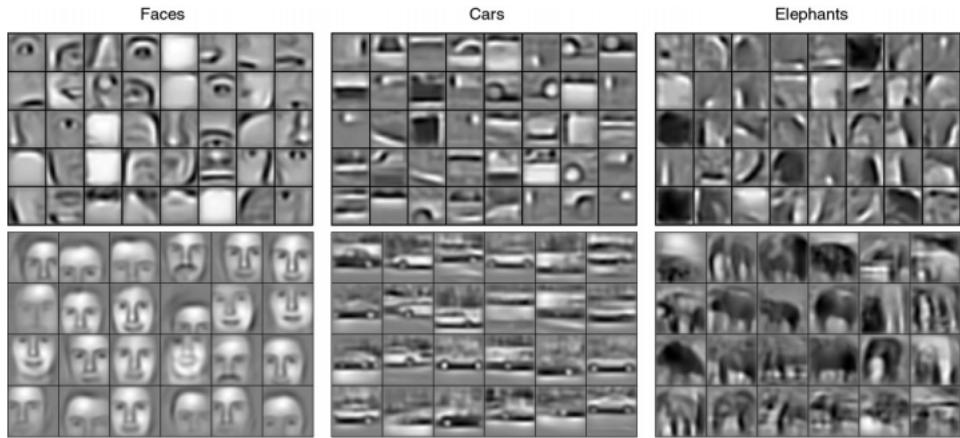
Konvolučné neurónové siete sa používajú prevažne v doméne obrazových dát. Tieto siete majú schopnosť naučiť sa rozpoznávať špecifické štruktúry/tvary z obrázka. Toto dokážu pomocou takzvaných konvolučných filtrov, ktoré sa v nižších vrstvách



Obr. 2.5: **Model doprednej neurónovej siete.** [8] Dopredné neurónové siete sa skladajú zo vstupnej vrstvy, skrytých vrstiev a výstupnej vrstvy. Keď hovoríme o počte vrstiev vstupnú vrstvu nepočítame. Neurónová sieť na obrázku má teda dve vrstvy.

naučia rozoznávať jednoduchšie tvary, akými sú napríklad obrysy alebo hrany (Obr. 2.6). V tých vyšších vrstvách sú to zložitejšie štruktúry akými môžu byť celé objekty v závislosti od typu úlohy na ktorú boli trénované. Ak bola neurónová sieť trénovaná napríklad na klasifikáciu zvierat, môže tým objektom byť pes alebo morča, v prípade ak je úlohou neurónovej siete detekcia Alzheimerovej choroby možu týmito objektami byť niektoré väčšie časti mozgu (napr. hippocampus).

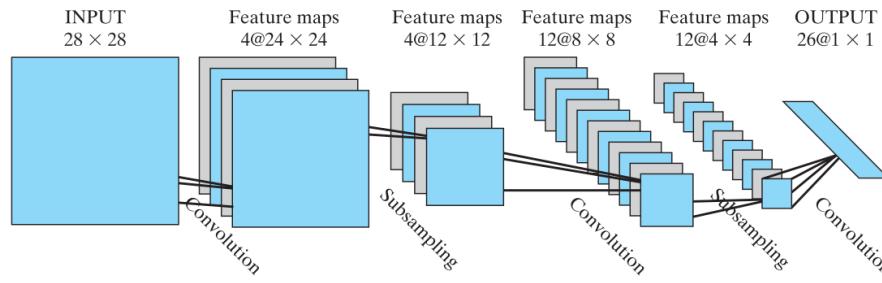
Základnými stavebnými blokmi konvolučných neurónových sietí sú konvolučné vrstvy (angl. convolutional layers) a združovacie vrstvy (angl. pooling layers).



Obr. 2.6: Vizualizácia druhej (hore) a tretej vrstvy (dole) konvolučných neurónových sietí naučených na špecifické kategórie objektov (tváre, autá a slony). [9] Nižšie vrstvy rozoznávajú jednoduchšie štruktúry zatiaľ čo vyššie už dokážu rozoznať aj tie zložitejšie.

Konvolučné vrstvy Pomocou konvolučných vrstiev sa neurónová sieť učí extrahovať črty z obrázka [8]. Konvolúcia prebieha tak, že tzv. jadro (angl. kernel) sa posúva po tzv. mape vlastností (angl. feature map) a matematickými operáciami z pôvodnej mapy vlastností a svojich parametrov vytvára novú mapu vlastností. Tieto parametre sú trénovateľné, čo umožňuje sa každému jadru naučiť určitú črtu - napr. hranu. Konvolučná vrstva tiež dokáže znižovať komplexitu modelu (a teda aj počet jeho parametrov) jej hyper parametrami (angl: stride, padding, depth).

Združovacie vrstvy Cieľom združovacích vrstiev je postupne znižovať dimenzionalitu dát, tým znižovať počet parametrov modelu, a teda aj jeho komplexitu [10]. Najčastejšie sa používajú vrstvy združujúce maximom (angl. max-pooling), ale existujú aj vrstvy združujúce priemerom či súčtom.



Obr. 2.7: Príklad architektúry konvolučnej neurónovej siete.

[8] V tejto architektúre neurónovej siete sa používajú tri konvolučné vrstvy (označené ako *convolution*) a dve združovacie vrstvy (označené ako *subsampling*). Môžeme si všimnúť, že konvolučné vrstvy postupne pridávajú mapy vlastností (tiež označované ako: angl. "volumes") a tiež mierne znižujú ich veľkosť. Združovacie vrstvy zasa výrazne znižujú ich veľkosť (až o polovicu) a tým aj počet parametrov v neurónovej sieti.

2.2.4 Architektúry konvolučných neurónových sietí

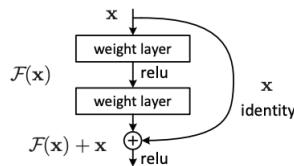
Architektúra neurónovej siete hovorí o tom, ako neurónová sieť vyzerá - koľko má vrstiev, z akých vrstiev sa skladá (konvolučné, združovacie, husté), koľko filtrov je v jednotlivých vrstvách a pod. Nie každá architektúra je vhodná na každý problém. Ak je problém jednoduchý, môže byť použitie veľmi hlbokej neurónovej siete zbytočné. Taktiež, jednoduchšia architektúra potrebuje menej výpočtových zdrojov na natrénovanie a je odolnejšia voči pretrénovaniu. Spomeniem niekoľko najznámejších architektúr, ktoré sú používané najmä pri klasifikácii obrazových dát.

- VGG [11] - hlboká neurónová sieť, so 16 alebo s 19 vrstvami. Skladá sa s konvolučných a združovacích vrstiev.
- ResNET [12] - hlboká neurónová sieť skladajúca sa z reziduálnych blokov. Reziduálne bloky obsahujú skracovacie spojenia, "skratky" (angl. shortcut connections) ako nástroj na zabránenie miznúcemu a explodujúcemu gradientu. Táto architektúra bola navrhnutá s 20, 32, 44, 56, 110 a 1202 vrstvami. Na klasifikáčných úlohách na dátovej sade ImageNet táto architektúra prekonala

architektúru VGG.

- Inception (GoogLeNet) [13] - hlboká neurónová sieť skladajúca sa s inception blokov. Každý blok robí niekoľko rôznych konvolúcii zo vstupu daného bloku, ktoré sú následne spojené v združovacom bloku.

Taktiež existuje niekoľko ďalších vylepšení Inception architektúry (Inception v1 až v4), dokonca aj kombinácia s architektúrou ResNet.



Obr. 2.8: Reziduálny blok v architektúre ResNET. Informácia z predhádzajúceho bloku je súčasťou výstupu aktuálneho bloku pomocou skracovacieho spojenia. [12]

2.2.5 Interpretovanie neurónovej siete

Montavon; Samek; Müller (2018) definujú interpretovanie ako mapovanie abstraktného konceptu (napríklad predikovanej triedy) do domény, ktorej človek dokáže porozumieť. Ako príklad domény, ktorá je interpretovateľná uvádzajú obrázky (pole pixelov) alebo text (sekvencia slov) [14]. Medzi domény, ktoré nie sú interpretovateľné zaradujú napríklad latentné vektorové reprezentácie slov (angl. word embeddings) alebo iné abstraktné vektorové reprezentácie [14]. Na rozdiel od vstupných dát do neurónovej siete, ktoré sú zvyčajne interpretovateľné, neuróny na výstupnej vrstve a v skrytých vrstvách sú abstraktné a vyžadujú dodatočné úsilie na ich interpretovanie. Jedným zo spôsobov interpretovania týchto neurónov je maximalizácia aktivácie (angl. activation maximization).

Maximalizácia aktivácie (angl. Activation maximization) Maximalizácia aktivácie je metóda na nájdenie takého vstupného prototypu, ktorý vyprodukuje najväčšiu mieru aktivácie pre zvolený neurón (zvyčajne je to neurón hľadaný

trydy na najvyššej vrstvy). Takýto vstupný prototyp je nájdený tak, že neurónovej sieti je daný na vstup neutrálny obrázok, ktorý v danej doméne nereprezentuje žiadnu triedu (zvyčajne sa jedná o šedý obrázok) a je optimalizovaná funkcia maximalizácie aktivácie pomocou poklesu gradientu [14] (angl. gradient descent). Pri aplikovaní tejto metódy na obrazové dátá výsledné prototypy vyzerajú tak ako na obrázku 2.9.

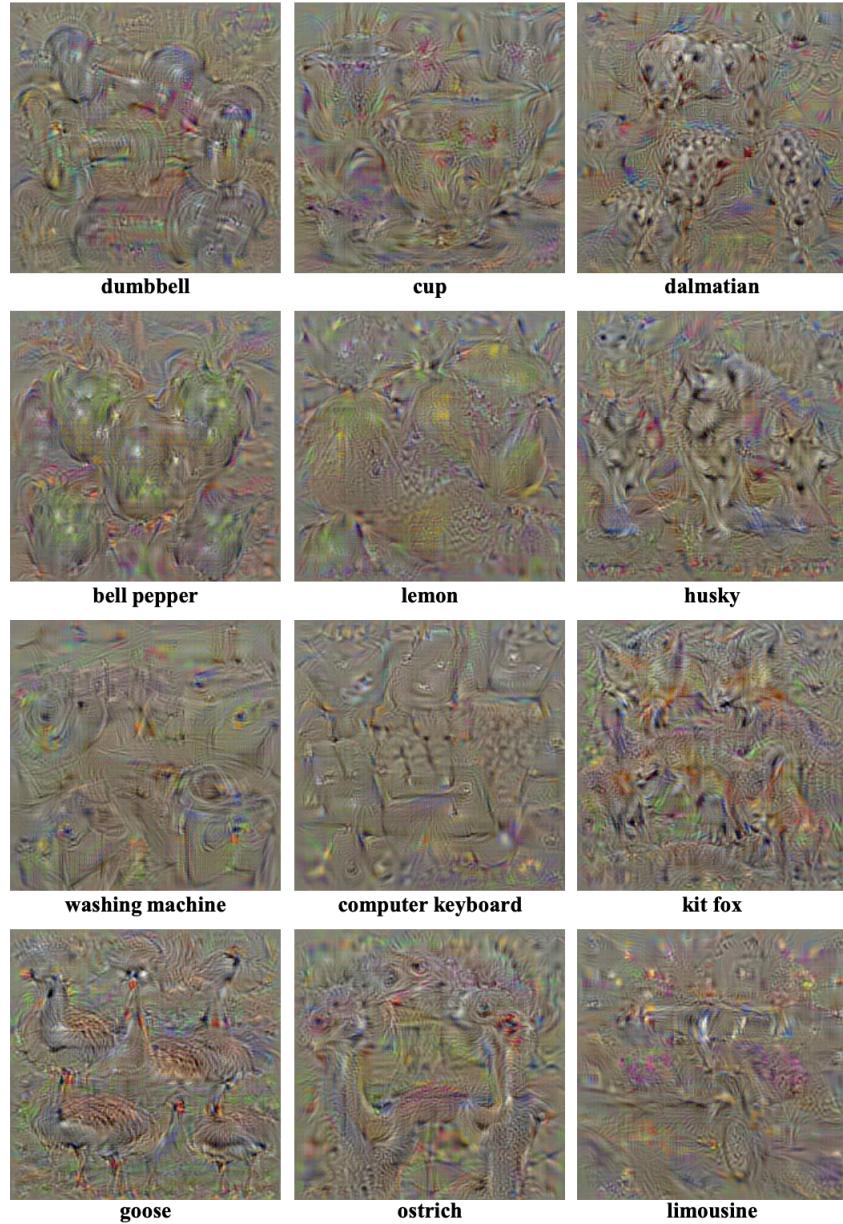
Maximalizácia aktivácie s expertom Na získanie realistickejších prototypov (prototypov, ktoré sa viac podobajú vstupným dátam) l_2 -regularizácia (používaná v maximalizácii aktivácie) je nahradená takzvaným “expertom”, ktorý sa snaží učiť distribúciu hľadanej triedy [14]. Oproti l_2 -regularizácii, ktorá hľadá vstup maximalizujúci pravdepodobnosť triedy, expert hľadá taký vstup, ktorý je najpravdepodobnejší pre zvolenú triedu. Ako “expert” môže byť použitý napríklad Gaussian RBM (angl. Restricted Boltzmann machine) [14].

2.2.6 Vysvetľovanie predikcie neurónovej siete

Montavon; Samek; Müller (2018) definujú vysvetľovanie ako kolekciu vlastností dát, ktoré sú z interpretovateľnej domény, ktoré prispeli k výslednému rozhodnutiu (napr. zaradenie do určitej triedy - klasifikácia) pre určité pozorovanie [14]. Rozdiel oproti interpretovaniu teda je, že pri interpretovaní hľadáme vzorový prototyp (vzorové pozorovanie) pre zvolenú triedu, zatiaľ čo pri vysvetľovaní sa snažíme zistiť prečo, a teda ktoré z vlastností vstupu najviac prispeli (tj. sú najviac relevantné) k výslednej predikcii neurónovej siete (napr. zaradenie pozorovania do určitej triedy).

Niekteré metódy vysvetľovania fungujú na základe zakrývania častí obrázka a sledovaním zmeny predikcie predikovanej triedy – perturbačné metódy, iné zasa na základe spätného šírenia (angl. backpropagation) – napr. LRP, analýza senzitivity.

Každá z metód má svoje výhody a nevýhody, napríklad výhodou perturbačných



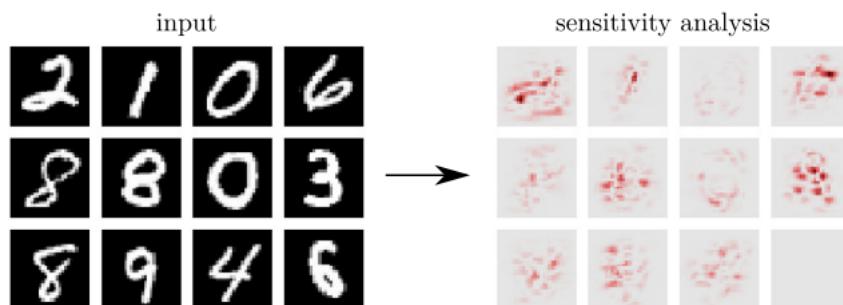
Obr. 2.9: Maximalizácia aktivácie aplikovaná na obrazové dátá. [15] Výsledné vzorové prototypy pre jednotlivé triedy nevyzerajú prirodzene, sú prevažne šedé s farebnými črtami objektov. Tieto vzorové prototypy nereprezentujú príklady vstupov "z reálneho sveta" ale ideálne vstupy pre jednotlivé triedy. Takéto vstupy nerónová sieť bežne nedostane.

metód je, že môžu byť použité na akýkoľvek model, keďže jediné čo potrebujú je výstup (predikciu) z modelu. Ich nevýhodou však je, že sú pomalé. Niektoré z metód vysvetľovania bližšie opíšeme v tejto sekcií.

2.2.6.1 Analýza senzitivity

Analýza senzitivity slúži na vysvetľovanie predikcie neurónovej siete. Táto metóda identifikuje, ktoré z vlastností vstupného pozorovania najviac prispievajú výslednej predikcii. Najviac dôležité sú také vlastnosti, ktorých zmenou sa najvýraznejšie zmení výsledná predikcia. Na takéto vlastnosti je výsledná predikcia najviac citlivá [14].

Výsledok analýzy senzitivity znázornený v tepelnej mape (angl. heatmap) je zobrazený na obrázku 2.10. Analýza senzitivity zachytáva teda vlastnosti vstupného pozorovania, ktoré k výslednej predikcii prispievajú pozitívne aj negatívne (napr. zmenením určitej vlastnosti vstupu sa výrazne zníži zaradenie do danej triedy). Na výslednej tepelnej mape vlastnosti, ktoré k výslednej predikcii prispievajú pozitívne, a vlastnosti, ktoré k výslednej predikcii prispievajú negatívne (proti), nevieme rozlísiť. Vieme len, že zmenením danej vlastnosti výrazne ovplyvníme predikciu.



Obr. 2.10: **Analýza senzitivity** aplikovaná na konvolučnú neurónovú sieť trénovanú na dátovej sade MNIST. [14]

Červenou farbou sú zobrazené miesta ktoré najviac prispievajú, či už pre alebo proti, výslednej predikcii. Čím je červená farba výraznejšia, tým viac je výsledok senzitívny na zmenu daného pixela.

2.2.6.2 LRP (angl. layer-wiser relevance propagation)

Metóda vrstvami propagovanej relevancie, ďalej len LRP (angl. layer-wise relevance propagation), sa od analýzy senzitivity odlišuje tým, že vo výslednej tepelnej mape dokáže odlišiť vlastnosti, ktoré prispeli pozitívne alebo negatívne k výslednej predikcii (v závislosti od použitých parametrov α a β).

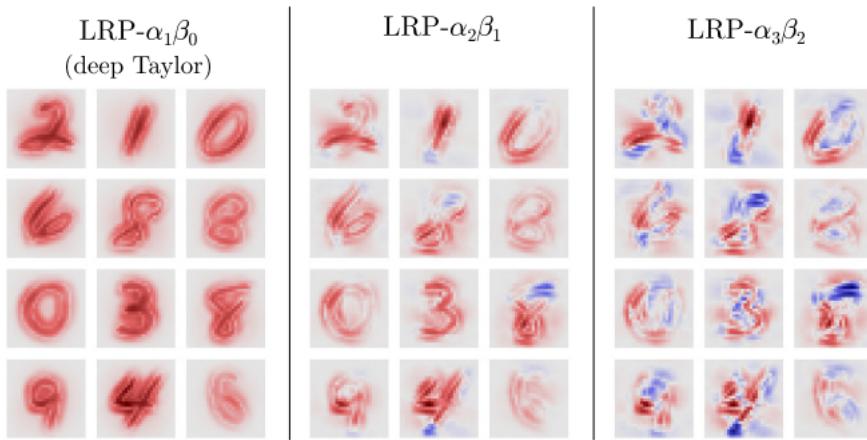
Táto technika funguje tak, že vstupný obrázok (metóda sa dá použiť aj na iné ako obrazové dátu, keďže pracujeme práva s obrazovými dátami metódy budeme vysvetľovať práve na nich) dopredným šírením "prejde" neurónovou sieťou, pričom sú zozbierané aktivácie neurónov v jednotlivých vrstvách. Následne je neurónovou sieťou spätným šírením propagované skóre z výstupu neurónovej siete v podobe relevancie až k vstupnému obrázku.

Nasledovné vzorce 2.2, 2.3, 2.4 [14] vyjadrujú spôsob výpočtu propagovanej relevancie medzi vrstvami. j a k sú jednotlivé vrstvy, pričom k je vrstva, z ktorej je relevancia R propagovaná. Parametre α a β upravujú, koľko pozitívnej (α) alebo negatívnej (β) relevancie je vytvorennej počas fázy spätného šírenia relevancie. Pri ich nastavovaní musí platiť, že $\alpha - \beta = 1$ a zároveň $\beta \geq 0$. Súčet pozitívnej a negatívnej relevancie je však medzi vrstvami vždy rovnaký [14], výsledok použitia rôznych hodnôt α a β je znázornený na obrázku 2.11. $R_{j \leftarrow k}^+$ (Obr. 2.2) a $R_{j \leftarrow k}^-$ (Obr. 2.4) vyjadrujú množstvo pozitívnej (+), resp. negatívnej (-) relevancie propagovanej z vrstvy k do vrstvy j . a_j je aktivácia neurónu, na ktorý je propagovaná relevancia.

$$R_{j \leftarrow k}^+ = \frac{a_j w_{jk}^+}{\sum_j a_j w_{jk}^+} \quad (2.2)$$

$$R_{j \leftarrow k}^- = \frac{a_j w_{jk}^-}{\sum_j a_j w_{jk}^-} \quad (2.3)$$

$$R_j = \sum_k (\alpha R_{j \leftarrow k}^+ - \beta R_{j \leftarrow k}^-) R_k \quad (2.4)$$



Obr. 2.11: Výsledné vysvetlenie (v podobe tepelnej mapy) vytvorené použitím LRP s rôznymi hodnotami α a β na dátovej sade MNIST. [14] Pozitívna relevancia je zobrazená červenou farbou [14]. Negatívna relevancia je zobrazená modrou farbou [14]. V prípade, že použijeme $\alpha = 1$ a $\beta = 0$ stráčame informáciu o tom, ktoré pixely negatívne (tj. sú proti výslednej predikcii) prispeli k výslednej predikcii (a opačne).

Výhodou LRP oproti iným metódam, ako napríklad dekonvolúcii je, že vysvetlenie (výsledná tepelná mapa) vytvorené technikou LRP je pre rôzne obrázky vždy rôzne [16]. Naopak, pri dekonvolúcii je vysvetlenie vždy rovnaké pokiaľ v architektúre neurónovej siete neboli použité združovacie vrstvy (angl. pooling layers) [16]. Ďaľším rozdielom je (aj oproti analýze senzitivity), že vo výslednom vysvetlení LRP rozlišuje, ktoré vlastnosti pozitívne alebo negatívne prispeli k negatívnej predikcii.

2.2.6.3 Riadená spätná propagácia (angl. Guided Backprop)

Metóda Guided Backprop je rozšírením metódy dekonvolúcie (angl. deconvolution) [14]. Metóda využíva aktivácie *ReLU* pre smerovanie signálu (tj. výsledného "tepla") na príslušné miesta vstupného obrazu [14]. Pri spätnom šírení sú negatívne gradienty nahradené hodnotou nula. Rovnako aj gradienty z neurónov, ktoré mali pri doprednom šírení po aplikácii aktivačnej funkcie (*ReLU*) hodnotu 0. Obrázok

2.12 zobrazuje výslednú tepelnú mapu po použití riadenej späťnej propagácie, a porovnáva ju s inými metódami. Na rozdiel od metódy LRP, táto metóda nezobrazí na tepelnej mape oblasti, ktoré k výslednej predikcii prispievajú negatívne.

2.2.6.4 GradCAM

GradCAM [17], alebo Gradientom-vážené mapovanie aktivácií triedam (angl. Gradient-weighted Class Activation Mapping) je spôsob vysvetľovania rozhodnutí aplikovaný na neurónové siete, ktoré používajú konvolučné vrstvy. Cieľom tejto metódy je vysvetliť, ktoré časti vstuпу sú dôležité pre vybranú triedu (jednu z tried, ktoré neurónová sieť predikuje). Metóda funguje nasledovne:

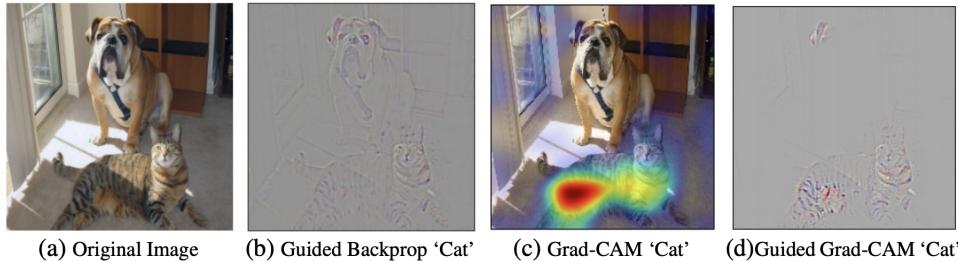
- Vstupný obraz ”prejde” neurónovou sieťou (tj. inferencia).
- Vypočítame gradient poslednej konvolučnej vrstvy voči výstupu na poslednej vrstve neurónovej siete pre vybranú triedu.
- Výsledok predchádzajúceho kroku má rovnaký rozmer ako veľkosť poslednej konvolučnej vrstvy, hodnoty v tejto matici sčítame cez dimenziu kanálov. Z konvolučnej vrstvy o rozmere (x, y, c) , kde c je dimenzia kanálov (angl. channels) vznikne matica o rozmere (x, y) .
- Na tento výsledok použijeme funkciu *ReLU* čím odstránime z tepelenej mapy časti obrázku, ktoré prispievajú proti predikovanej triede negatívne.
- Túto maticu pomocou bilineárnej interpolácie zväčšíme na veľkosť vstupu.

Výsledkom je tepelná mapa pre vstupný obrázok voči vybranej predikovanej triede. Vyššie hodnoty v tepelnej mape vyjadrujú doležitejšie časti obrazu pre vybranú triedu, nižšie hodnoty vyjadrujú tie menej doležité.

2.2.6.5 Riadený GradCAM (angl. Guided GradCAM)

Guided GradCAM kombinuje metódu riadenej späťnej propagácie s metódou Grad-CAM. Výsledné tepelné mapy z oboch metód, pre vstupný obraz a vybranú triedu

sú navzájom vynásobené (nejedná sa o maticové násobenie ale násobenie po prvkoch) čím vznikne nová tepelná mapa [17] (Obr. 2.12).



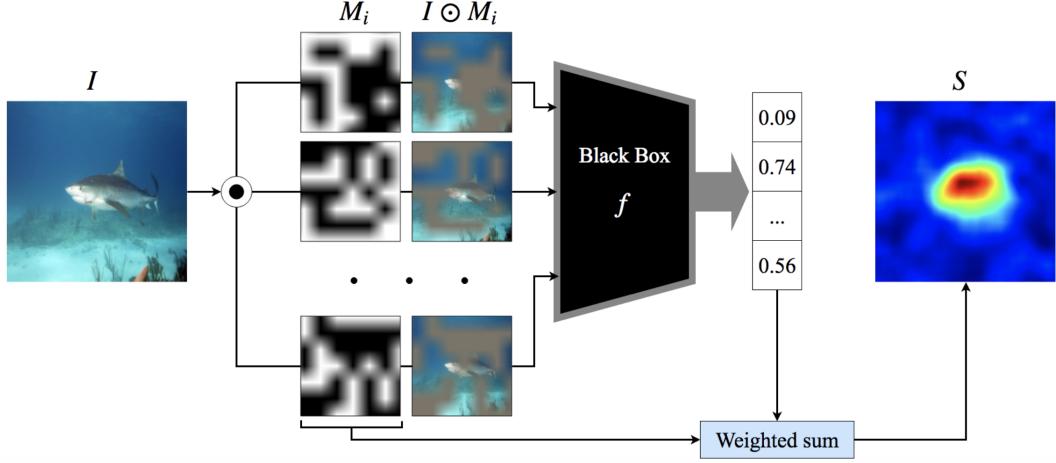
Obr. 2.12: Porovnanie metód Guided Backprop (b), GradCAM (c) a Guided GradCAM (d) [17]. Tepelná mapa (d) je výsledkom násobenia medzi tepelnými mapami (b) a (c), o tom svedčí väčšina "tepla" v spodnej časti obrázku. Teplo má polohu z (c) a tvar z (b).

2.2.6.6 RISE - Randomized Input Sampling for Explanation

Túto metódu môžeme zaradiť medzi perturbačné metódy, keďže je tiež založená na zakrývaní jednotlivých častí obrazu a sledovaním zmeny výslednej predikcie modelu. Už z názvu modelu (*Randomized Input Sampling for Explanation*) je zrejmé, že táto metóda využíva náhodu na zakrývanie jednotlivých častí vstupného obrazu. Vstupný obraz je prekrytý náhodou maskou, ktorá je vytvorená nasledovne [18]:

- Je vytvorená náhodná binárna (tj. iba z bielej a čiernej farby) maska o malej veľkosti (napríklad 8px x 8px).
- Táto maska je zväčšená (angl. upsampled) pomocou bilineárnej interpolácie [18] (angl. bilinear interpolation) na veľkosť ktorá je mierne väčšia ako veľkosť obrázka s ktorým bude prekrytá (kvôli oreznávaniu). Tým sa zníži jej kvalita a ostré hrany medzi bielymi a čiernymi časťami sa zjemnia. Masky už teda nie sú binárne.
- Z masky je náhodne vyrezaná náhodná časť o veľkosť prekrývaného obrázka.

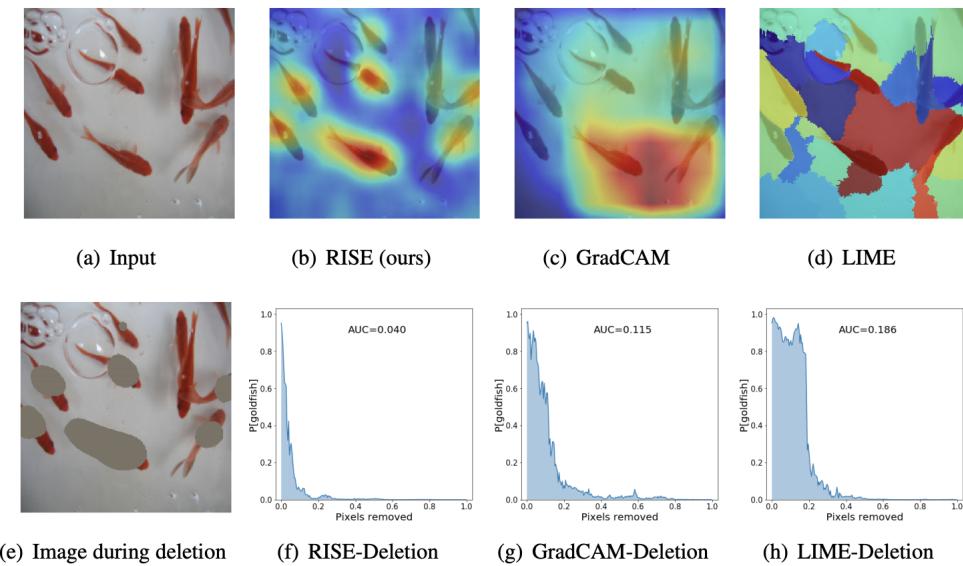
Toto sa opakuje N krát. Výsledná tepelná mapa je vypočítaná ako vážený priemer všetkých vygenerovaných masiek, kde váhy sú skóre (pravdepodobnosť predikovanej triedy) z modelu. Tento proces je zobrazený na obrázku 2.13.



Obr. 2.13: Metóda *Rise*. [18] Vygenerované masky nahradzajú vstupný obrázok na, ktorý sú aplikované. Z výstupných predikcií jednotlivých masiek je nakoniec vypočítaná tepelná mapa.

Autori porovnali túto metódu s metódami *GradCAM* (Selvaraju et al. 2017) [17] a *LIME* (Ribeiro et al. 2016) [19]. Metóda *Rise* si oproti týmto dvom metódam počínala lepšie (Obr. 2.14). Vykonali niekoľko experimentov, v ktorých porovnali architektúry neurónových sietí *ResNet50* (He et al. 2016) [12] a *VGG16* (Simonyan; Zisserman 2014) [11] natrénované na dátových sadách PASCAL VOC07 (Everingham et al. 2010) [20] a MSCOCO2014 (Lin et al. 2014) [21]. Sledovali metriky *insertion* a *deletion* (Obr. 2.14). Metrika *insertion* je vyjadrená ako plocha pod krivkou (AUC) funkcie $y = f(x)$, kde y je istota predikcie a x je počet pridaných najdôležitejších pixelov, dôležitosť pixelov je určená metódou vysvetľovania predikcie neurónovej siete a môže byť zobrazené pomocou tepelnej mapy. Metrika *deletion* naopak odoberá najdôležitejšie pixely z obrázka.

Výhodou tejto metódy je, že oproti bežným perturbačným metódam je výrazne rýchlejšia.



Obr. 2.14: Porovnanie metódy *Rise* s *GradCAM* alebo *LIME*. [18] V prvom riadku sú tepelné mapy jednotlivých metód pre vstup. V druhom riadku je znázornená porovávaná metrika *deletion*. Táto metrika sleduje vzťah medzi odobratím najdôležitejších pixelov a výslednou predikciou modelu. Je vyčíslená pomocou výpočtu plochy pod krivkou (AUC). Na grafoch si môžeme všimnúť, že metóda *Rise* potrebuje odobrať menej pixelov na to aby klesla pravdepodobnosť predikovanej triedy. To znamená, že tepelná mapa (metódy *Rise* oproti ostatným metódam) lepšie zaznamenáva dôležité pixely pre predikovanú triedu.

2.3 Využitie neurónových sietí pri odhalovaní Alzheimerovej choroby

Neurónovým sieťam sa doposiaľ podarilo dosiahnuť veľmi dobré výsledky pri odhalovaní Alzheimerovej choroby. Ako vstup používajú rádiologické snímky ako sú z MRI či PET. Tieto rádiologické ukazovateľe sme bližšie popísali v sekciu 2.1.3. Okrem rádiolgických snímok môžu byť vstupom do neurónovej siete demografické údaje o pacientovi, či výstupy z rôznych klinických alebo kognitívnych testov. Ta-keťo údaje o pacientoch obsahuje populárna dátová množina *ADNI-1* [22].

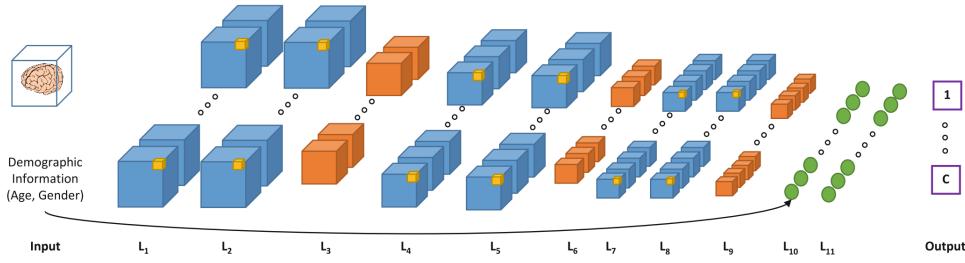
Neurónové siete natrénované na predikciu Alzheimerovej choroby sa líšia najmä

v:

- **predspracovaní** - vstupné dáta sú zmenšené/zväčšené rôznymi algoritmami na rôzne veľkosti, častokrát sa z rádiologických snímok odstraňuje lebka
- **type vstupných dát** - môžu to byť rádiologické snímky (MRI, PET), vlastné črty extrahované z rádiologických snímok (MRI, PET), alebo kombinácia takýchto snímok/črt, s inými, napríklad demografickými údajmi
- **architektúre** - môžu to byť konvolučné siete s 2D konvolúciami (v prípade, že sa používa iba časť rádiologickej snímky, alebo vlastné črty) alebo 3D konvolúciami (ak je vstup celý rádiologický snímok, angl. "full volume"), alebo iné architektúry ako ResNET (reziduálne neurónové suete) alebo VGG
- **ako boli natrénované** - pri niektorých neurónových sieťach autori využili učenie prenosom (angl. transfer learning) a rôzne spôsoby augmentácie vstupov

Ako príklad 3D konvolučnej neurónovej siete uvediem neurónovú sieť od Esmaeilzadeh et al. s presnosťou **94.1%** (a s F_2 skóre 0.93) na populárnej dátovej množine s názvom *ADNI-1* (Obr. 2.15). Tento výsledok dosiahli v úlohe klasifikácie iba do tried CN a AD (bez MCI). Vstupom do tejto neurónovej siete boli snímky z magnetickej rezonancie (MRI) ale aj demografické informácie, akými sú napríklad vek alebo pohlavie. Autori článku neuvádzajú úspešnosť modelu, ktorý bol natrénovaný iba z obrazových dát, táto úspešnosť by bola pravdepodobne nižšia, nakoľko vek aj pohlavie sú významnými faktormi ovplyvňujúcimi rozvoj Alzheimerovej choroby.

V prípade klasifikácie do všetkých troch tried - CN, MCI a AD autori tejto práce dosiahli horšie výsledky oproti binárnej klasifikácii. Ich model dokázal správne zaradiť pacienta s presnosťou **61.1%** (a s F_2 skóre 0.62) [23]. Pri dosiahnutí tohto výsledku použili tzv. učenie s prenosom (angl. transfer learning), ktoré im zlepšilo úspešnosť modelu až o 7.1% z pôvodných 54%. Model, z ktorého učili prenosom je už skôršie spomínaný model na binárnu klasifikáciu pacientov s Alzheimerovou chorobou.



Obr. 2.15: Architektúra konvolučnej neurónovej siete použitej pri detekcii Alzheimerovej choroby. [23] Modré kocky sú konvolučné vrstvy, oranžové kocky sú *max-pooling* vrstvy, posledné dve (zelené) vrstvy sú plne prepojené vrstvy. Môžeme si všimnúť, že do posledných dvoch plne prepojených vrstiev okrem obrazových dát vstupujú aj informácie o veku a pohlaví.

Autori experimentovali trénovaním dvoch rôznych modelov, jedného jednoduchšieho a druhého zložitejšieho. Lepší bol jednoduchší model, pretože neboli tak náchylní na pretrénovanie. V týchto modeloch použili dropout, l_2 regularizáciu a augmentované dátá (obrázky otočili po osi x). Tieto "vylepšenia" pridávali postupne a sledovali rozdiel v úspešnosti modelu, každé jedno z týchto vylepšení výrazne zlepšilo úspešnosť modelu. V kroku predspracovania dát odstránili z obrázkov také časti, ktoré nepredstavovali tkivo mozgu (napr. lebka) technikou s názvom BET (Smith 2002) [24], pretože z nich sa Alzheimerova choroba nedá diagnostikovať.

Niekteré práce (Suk et al. 2016) sa zaobrali dokonca klasifikáciou do štyroch tried: AD, CN, pMCI (angl. progressive MCI - pacienti ktorí pokročili k AD do 18 mesiacov), sMCI (angl. stable MC - pacienti ktorí nepokročili k AD do 18 mesiacov). Táto úloha je samozrejme náročnejšia, najlepší model v tomto prípade dosahoval presnosť 53.72% [25]. V prípade binárnej klasifikácie (AD vs CN) sa autorom podarilo dosiahnuť presnosť až **95.09%**, oproti Esmaeilzadeh et al. však použili aj rádiologické snímky z PET. Táto práca sa ďalej vyznačuje adaptívou selekciou črt, vďaka ktorej sa autorom podarilo dosiahnuť tak dobré výsledky. V tejto práci autori taktiež vykonali odstránenie lebky zo vstupných snímok počas fázy predspracovania.

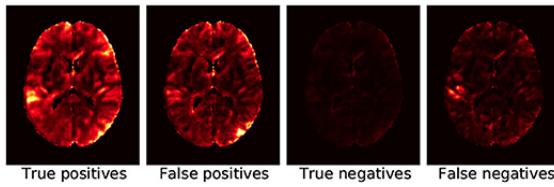
Učenia prenosom (angl. transfer learning) je veľmi dobrým spôsobom na zrýchlenie trénovania a zlepšenie úspešnosti modelu. Hosseini-Asl et al. využili učenie prenosom a to tak, že najskôr netrénovali 3D konvolučný autoenkovodér, ktorý mal za úlohy rekonštruovať vstup - tj. vstupný radiologický snímok. Z tohto autoenkovodéra zobraťi jednu jeho časť - enkodér za ktorý dali konvolučné vrstvy, ktoré dotrénovali na detekciu Alzheimerovej choroby. Enkodér teda slúžil na ekstrakciu črt.

Neurónové siete sa v niektorých prácach používajú v kombinácii s inými algoritmi strojového učenia. Suk et al. použili kombináciu riedkych regresných modelov (angl. sparse regression models) a 2D konvolučnej neurónovej siete, kde výstupy z týchto regresiných modelov slúžili ako vstup do neurónovej siete.

2.3.1 Vysvetľovanie rozhodnutí neurónových sietí detegujúcich Alzheimerovu chorobu

Existujúce práce sa už zaobrali metódami vysvetľovania rozhodnutí neurónových sietí detekujúcich Alzheimerovu chorobu. Böhle; Eitel; Weygandt; Ritter 2019 uviedli možnosti analýzy rozhodnutí za účelom ich vysvetľovania. Konkrétnie sa zaobrali metódami vrstvami propagovanej relevancie (LRP) a vedenou spätnou propagáciou (angl. guided backpropagation). Uvádzajú LRP ako metódu na vysvetľovanie inividuálnych rozhodnutí neurónovej siete kde naopak vedenú spätnú propagáciu ako metódu na zistenie oblastí, na ktoré je neurónová sieť senzitívna. Tieto metódy skúmali porovnávaním priemerov tepelných máp (angl. heatmaps) všetkých pozorovaní v predikovaných triedach (2 - AD, HC). Taktiež porovnávali priemerné tepelné mapy pozorovaní podľa spôsobu zaradenia výslednej predikcie (4 - true positive, true negative, false positive, false negative) (Obr. 2.16). Okrem iného porovnávali mieru relevancie pri metóde LRP v jednotlivých častiach mozgu u pozorovaní s Alzheimerovou chorobou a u pozorovaní bez nej. Možným vylepšením tejto práce je vyskúšanie metódy LRP aj na pacientoch s miernym kognitívnym poškodením (angl. mild-cognitive impairment), nie len na pacientoch s

Alzheimerovoch chorobou a zdravých jedincoch.



Obr. 2.16: Priemerná relevancia (z metódy LRP - $\beta = 0$)
pozorovaní podľa spôsobu zaradenia výslednej predikcie

Najviac relevancie je na miestach so žltou farbou. [28]

2.4 Spracovanie obrazu

Kedže pri diagnostike Alzheimerovej choroby sa pracuje s rádiologickými snímkami, čo sú trojrozmerné obrazové dátá, pri jej detekcii neurónovými sieťami je potrebné tieto dátá spracovať technikami spracovania obrazu.

Metódy spracovania obrazu podľa Chen [29] rozdeľujeme do nasledovných kategórií:

- vylepšovanie obrazu (angl. image enhancement)
- rekonštrukcia obrazu (angl. image restoration)
- analýza obrazu (angl. image analysis)
- kompresia obrazu (angl. image compression)

Pri **vylepšovaní obrazu** je obraz upravovaný predovšetkým heuristickými technikami [29], môže sa napríkald jednať o upravenie jasu, kontrastu alebo farieb. Cieľom **rekonštrukcie obrazu** je zrekonštruovať poškodené časti obrazu, napr. pri fotografiách to môžu byť ich vyblednuté časti. Metódy **analýzy obrazu** umožňujú obraz spracovať tak, že je možné z neho automaticky získať (extrahovať) informácie [29]. Príkladmi analýzy obrazu je segmentácia obrazu, extrakcia hrán alebo analýza textúry. **Kompresia obrazu** umožňuje zmenšenie veľkosti obrazu znižovaním

počtom potrebných bitov na jeho reprezentáciu [29]. Môže sa jednať o zmenšenie rozmerov obrazu, alebo počtu farieb potrebných na jeho reprezentáciu.

V našej doméne budeme pracovať so všetkými týmito technikami. Ako príklad môžem uviesť odstránenie takých častí obrazu, ktoré nepredstavujú mozgové tkanivo (BET - Smith 2002). Táto technika je kombináciou analýzy obrazu - identifikácia častí na odstránenie a vylepšenia obrazu - samotné odstránenie tých častí. Kompresia obrazu sa používa, v časti predspracovania pred tým ako je samotný snímok použitý ako vstup do neurónovej siete. Metódy rekonštrukcie obrazu sa bežne v tejto oblasti nepoužívajú, avšak my by sme ich chceli v našej práci použiť pri vytváraní novej metódy, preto sa im budeme bližšie venovať.

2.4.1 Rekonštrukcia obrazu

Metódy rekonštrukcie obrazu, alebo inak nazývané aj dokreslenia obrazu (angl. inpainting), podľa Ravi; Pasupathi; Muthukumar.; Krishnan [30] môžeme zaraďiť do nasledovných kategórií:

- dokresľovanie založené na syntéze textúr
- poloautomatické a rýchle digitálne dokresľovanie
- dokresľovanie založené na parciálnej diferenciálnej rovnici
- dokresľovanie na základe predlohy a vyhľadávania
- hybridné dokresľovanie

Tieto metódy sa líšia rýchlosťou dokresľovania, schopnosti dokreslovať veľké/malé plochy a predovšetkým kvalitou dokreslenia. Metódy dokresľovania založené na syntéze textúr fungujú dobre pre väčšie chýbajúce oblasti, avšak v ich výsledku môžu vzniknúť nežiadúce hrany [30]. Dokresľovanie na základe predlohy má zas problémy so zakrivenými štruktúrami [30]. Obr. 2.17 zobrazuje príklady použitia niektorých techník dokreslenia obrazu.



Obr. 2.17: Príklady dokreslenia obrázkov rôznymi metódami [30].

2.5 Zhrnutie

Alzhemierova choroba je bez pochyby veľmi nebezpečnou chorobou, keďže nie je "iba" o strate pamäti ale patrí k častým príčinám smrti (Sek. 2.1). Diagnostika

tejto choroby pozostáva najmä z neuropsychometrických testov a analýzy rádiologických snímok (napr. z PET, MRI). V súčasnosti tieto rádiologické snímky posudzujú doktori samotný. Práve tu je priestor pre umelú inteligenciu, aby im pri posudzovaní týchto snímok pomohla.

V doméne obrazových dát sa používajú najmä konvolučné neurónové siete, pretože majú veľmi dobrú schopnosť naučiť sa rozoznávať špecifické objekty z obrázka. Konvolučné neurónové siete sa v nižších vrstvách naučia rozoznávať jednoduchošie tvary/hrany a vo vyšších zložitejšie šruktúry až celé objekty. Keďže jednou z možností diagnostiky Alzheimerovej choroby je diagnostika pomocou rádiologických snímok, je možné použiť neurónové siete práve pri detekcii tohto ochorenia.

Neurónovým sietiam sa doteraz podarilo dosiahnuť veľmi dobré výsledky pri detekcii Alzheimerovej choroby, niektoré state-of-the-art riešenia dosahujú presnosť až **95.09%** (Suk et al. 2016). S takto vysokou úspešnosťou môžu byť veľmi dobrým pomocníkom doktorov. Do úvahy však musíme zobrať, že tieto výsledky boli dosiahnuté bez klasifikácie MCI pacientov. V reálnom svete doktora navštívia všetky typy pacientov - CN, MCI a AD. V tomto prípade neurónové siete dosahujú rádovo nižšiu presnosť (**61.1%**, Böhle et al. 2019). Niektoré práce dosiahli tieto výsledky použitím informácií o veku a pohlaví pacienta. Keďže pravdepodobnosťou výskytu Alzheimerovej choroby po dovršení 85 rokov života je až 50% (Sek. 2.1), je možné, že sa pri vyššom veku pacienta model začne rozhodovať najmä na základe tejto informácie a nie na základe obrazových dát. Zároveň to však môže neurónovej sieti pomôcť, ak nebude brať tento atribút ako hlavný indikátor Alzheimerovej choroby, ale skôr ako pomocný atribút, ktorý bude meniť jej správanie u rôznych typov pacientov. Tu je však dôležité, takúto neurónovú sieť podrobiť dôkladnej analýze jej rozhodnutí. Osobne si ale myslím, že v produkčnom modeli by sa tento atribút mal vynechať.

Ďalším problémom neurónových sietí je, že sa správajú ako čierne skrinky. Preto je potrebné ich rozhodnutia interpretovať, aby bolo pre doktora zrejmé na základe čoho neurónová sieť urobila svoju predikciu. V tomto práve môžu pomôcť metódy na vysvetľovanie rozhodnutí neurónovej siete (tzv. white-box metódy), alebo iné

Kapitola 2. Analýza

black-box metódy vysvetľovania rozhodnutí modelov (napr. RISE, LIME...).

Bežnému používaniu neurónových sietí ako pomocníka pre doktorov, nebráni len ich vysvetliteľnosť, ale aj ich schopnosť detektie ochorenia, keďže aj tu je priestor na zlepšenie - napr. úspešnosti klasifikácie do tried CN, MCI a AD.

Pre pochopenie správania sa neurónových sietí poznáme metódy jej interpretovania a vysvetľovania jej rozhodnutí. Interpretovaním neurónovej siete zistujeme, ako vyzerá vzorové pozorovanie pre jednu z tried, ktorú klasifikuje. Vysvetľovaním jej rozhodnutí zas zistujeme na základe čoho neurónová sieť spravila svoje rozhodnutie, a teda ktoré zo vstupných vlastností pozorovania ju navideli k zaradeniu do určitej triedy. Niektoré z týchto metód (LRP a vedená spätná propagácia) už boli použité pri vysvetľovaní rozhodnutí neurónových sietí detekujúcich Alzheimerovu chorobu, avšak zatiaľ len pri binárnej klasifikácii pacientov.

3. Ciele práce

Vychádzajúc zo zadania projektu a na základe poznatkov nadobudnutých z analýzy domény a problému, sme si stanovili nasledovné ciele.

3.1 Vytvorenie novej, alebo vylepšenie existujúcej metódy pre vysvetľovanie rozhodnutí neurónových sietí

Existujú rôzne metódy pre vysvetľovanie rozhodnutí neurónových sietí. Či už sú to tzv. white-box metódy (ako napríklad LRP) alebo tzv. black-box metódy, ktoré je možné použiť na ľubovoľný typ modelu. Žiadna z týchto metód nie je dokonalá (každá má svoje plusy a minusy v rôznych aspektoch) a je tu teda priestor na vytvorenie novej (lepšej) alebo vylepšenie existujúcej metódy. V prípade vylepšenia existujúcej metódy je nutné túto metódu porovnať najmä s vylepšovanou metódou a následne s inými metódami. Cieľom je teda vytvoriť novú metódu, ktorá vytvára presnejšie vysvetlenia ako iné metódy, alebo vylepsiť existujúcu metódu, ktorá vytvára presnejšie vysvetlenia ako metóda, z ktorej vychádza.

3.2 Využitie vytvorenej metódy na určenie miery správnosti modelu neurónovej siete detegujúcej Alzheimerovu chorobu

Pri neurónových sieťach detekujúcich Alzheimerovu chorobu je dôležité, aby sa naučili klasifikovať pacientov na základe relevantných črt z rádiologických snímkov. Práve preto je potrebné určiť mieru správnosti modelu podľa toho či sa model rozhoduje práve na základe týchto črt a nie iných. Na to sa využívajú metódy na vysvetľovanie rozhodnutí neurónových sietí, v tomto prípade sa použije novovytvorená metóda. Cieľom je teda určiť správnosť modelu detegujúceho Alzheimerovu chorobu pomocou vytvorenej metódy pre vysvetľovanie rozhodnutí neurónovej siete.

4. Návrh riešenia

Pre použitie neurónových sietí v bežnej praxi doktorov pri diagnostike Alzheimerovej choroby je nevyhnutné, aby sa rozhodnutia neurónových sietí dali vysvetliť. Preto navrhujeme metódu na vyvsetľovanie rozhodnutí neurónových sietí, ktorú overíme na MRI snímkoch u pacientov (CN, MCI a AD).

Vychádzajúc cieľa práce *3.1 Vytvorenie novej alebo vylepšenie existujúcej metódy pre vysvetľovanie rozhodnutí neurónových sietí* navrhujeme metódu, ktorá vychádza z už existujúcej metódy *RISE* (Sek. 2.2.6.6). Táto metóda dosiahla veľmi dobré výsledky oproti metódam GradCAM a LIME a považujem ju teda vhodný základ pre ďalšie vylepšenia. Metóda RISE funguje na princípe zakrývania častí obrázka (tak ako iné perturbačné/oklúzne metódy) jednou hodnotou (tj. farbou). Po takomto prekrytí avšak nevznikajú žiadné ostré hrany, ktoré by mohli neurónovú sieť myliť ako u iných metódach, ktoré fungujú na princípe zakrývania častí obrazu. Keďže metóda RISE bola pôvodne použitá na obrázky vo farebnom priesotre RGB, tento prekryv sa zvyčajne robí v čiernej farbe - tj. v ($r = 0, g = 0, b = 0$). MRI snímky nepoužívajú žiadnu farebnú schému, ale zachytávajú intenzitu (hodnoty sú zväčša reálne čísla). V tomto prípade môžeme zakrývať maximálnou alebo minimálnou hodnotou (minimálna hodnota je ekvivalentná RGB v prípade šedej). Toto zakrytie môže byť práve ďalším zdrojom zmätenia pre neurónovú sieť, keďže úbytky tkaniva sú vyjadrené nízkymi hodnotami na snímkoch. Preto navrhujeme zakrývané miesta dokresliť určitou metódou spracovania obrazu (Sek. 2.4) alebo na zakrytie použiť inú hodnotu. Pôvodná metóda bola ale narvhnutá pre obrázky (tj. 2D) a nie 3D volumetrické dátá, preto budeme musieť metódu

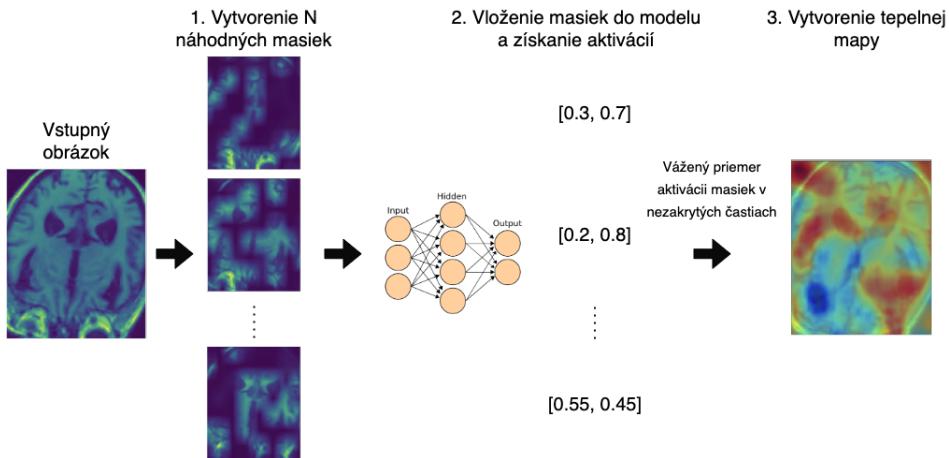
RISEI upraviť aby vedela pracovať s 3D dátami - tj. budeme generovať 3D masky a podobne.

4.1 RISEI - Randomized Input Sampling for Explanation with Inpainting

Metódu sme pomenovali *Randomized Input Sampling for Explanation with Inpainting* (tj. náhodné vzorkovanie vstupu pre vysvetlovanie s dokreslovaním) so skratkou RISEI.

Kedžže metóda vychádza už z existujúcej metódy, časť našej metódy je samozrejme rovnaká. Proces vytvorenia vysvetlenia klasifikácie (Obr. 4.1) do triedy T pre obrázok O modelom je teda nasledovný:

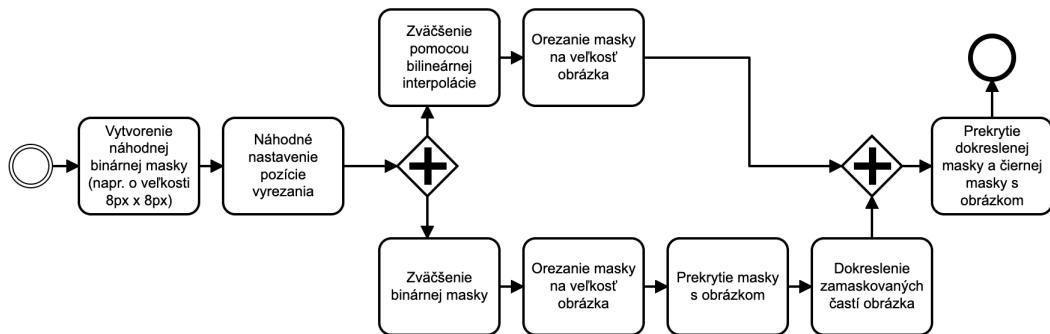
1. Vytvorenie N náhodne zamaskovaných obrázkov z obrázka O .
2. Vloženie zamaskovaných obrázkov do modelu a následné získanie pravdepodobnosti pre triedu T .
3. Vytvorenie a vizualizácia vysvetlenia pomocou tepelnej mapy.



Obr. 4.1: Proces vysvetlenia klasifikácie - vytvorenia tepelnej mapy.

Toto sú 3 hlavné kroky z ktorých pozostáva táto metóda, ďalej bližšie popíšeme jednotlivé z nich.

Vytvorenie náhodne zamaskovaných obrázkov. Vytvorenie náhodne zamaskovaných obrázkov tiež pozostáva z niekoľkých krokov, pričom niektoré z nich môžu byť vykonávané paralelne. Tento krok sme znázornili diagramom (Obr. 4.2). Masky sa vytvárajú paralelne, pretože ”čierna” maska ma jemné hrany a na dokreslenie potrebujeme naopak masku s ostrými hranami.



Obr. 4.2: BPMN diagram generovania jedného obrázka prekrytého maskou

Oproti metóde *Rise* vytvárame o jednu masku naviac, a teda je originálny obrázok prekrytý s viacerými maskami. Jednotlivé masky cez seba prekryjeme, pričom každej z nich nastavíme určité množstvo priehľadnosti. S týmto pomerom môžeme ďalej experimentovať a výsledky porovnávať. Môžeme porovnať použitie iba dokreslenej masky s iba ”čierной” maskou a tiež s použitím oboch v rôznych pomeroch.

Vytvorenie ”čiernej” masky je rovnaké, ako pri metóde *Rise*. Dokreslená maska vznikne dokreslením zakrytych (zamaskovaných) častí obrázka pomocou jedného z algoritmov na dokreslovanie (angl. inpainting). Tieto algoritmy sme popísali v sekcií 2.4 Spracovanie obrazu. Obrázok 4.3 je príkladom dokreslenia častí vzorového obrázka na základe masky náhodne vygenerovanej masky (tentotýk je v 2D, naša metóda bude pracovať s 3D). V našej metóde budeme experimentovať s

rôznymi hodnotami prekrytie (priemer, maximum, minimum, medián).



Obr. 4.3: Niektoré časti vzorového obrázka (vľavo) boli dokreslené podľa náhodne vygenerovanej binárnej masky (v strede). Výsledný obrázok (vpravo) môže byť ešte prekrytý ”čiernou” maskou s určitou priehľadnosťou.

Vytvorenie a vizualizácia vysvetlenia pomocou tepelnej mapy. Tento krok je identický s originálnou metódou *RISE*. Nasledovný vzorec 4.1 vyjadruje výpočet dôležitosti I pre každý voxel $[x, y, z]$ snímky, kde n je počet všetkých zamaskovaných snímok. Funkcia $p(k, x, y, z)$ vracia vracia predikciu (tj. aktiváciu v kontexte neurónových sietí) pre predikovanú triedu (v prípade binárnej klasiifikácie) z modelu pre zamaskovaný snímok k . Funkcia $c(k, x, y, z)$ vracia mieru zakrytie/dokreslenia maskou, pričom $H(c) = <0, 1>$, kde 1 znamená úplné prekrytie/dokreslenie a 0 žiadne prekrytie/dokreslenie. Rovnako, ako metóde *Rise*, počítame vážený priemer.

$$I_{x,y,z} = \frac{\sum_k^n p(k, x, y, z) * (1 - c(k, x, y, z))}{\sum_k^n p(k, x, y, z)} \quad (4.1)$$

Navrhovaná metóda do originálnej metódy pridáva niekoľko parametrov a najmä výpočtovo náročné dokreslovanie, preto bude nutné nájsť vhodné nastavenie parametrov, aby výpočet vysvetlenia neboli príliš časovo náročný. Práve výpočtová náročnosť môže byť jednou zo slabín tejto metódy. Takisto aj samotná dokreslená časť obrázka môže byť príčinou zmätenia neurónovej siete.

4.2 Overenie riešenia

Našu metódu budeme najskôr porovnávať s originálnou metódou RISE (tj. či sa nám podarilo vytvoriť lepsiu metódu) a následne s inou existujúcou metódou (LRP, GradCAM, Guided Backprop alebo Guided GradCAM). Z týchto metód je najviac vhodná metóda LRP, keďže už bolo jej použitie pri vysvetľovaní rozhodnutí neurónových sietí detekujúcich Alzheimerovu chorobu (Sekcia 2.3.1) skúmané. Tieto experimenty môžeme vykonávať na CN a AD vzorkách; a aj na CN, MCI a AD vzorkách. Budeme sledovať kvalitu navrhnutej metódy (oproti ostatným metódam) a na základe týchto tepelných máp budeme vyhodnocovať mieru správnosti modelu.

4.2.1 Dátová sada

Experimenty budeme vykonávať na dátovej sade ADNI, ktorá obsahuje MRI snímky AD pacientov. Táto dátová sada bola použitá aj na trénovanie state-of-the-art modelu na diagnóstiku Alzheimerovej choroby [23], ale aj pri vysvetľovaní rozhodnutí neurónovej siete pomocou LRP [28]. Na tejto dátovej sade budeme musieť vykonať rovnaké predspracovanie ako Böhle et al., aby sme sa s ich výsledkami mohli porovnať. Prípadne môžeme vykonať vlastné predspracovanie, ale budeme musieť vykonať aj experimenty s metódou LRP.

4.2.2 Experimenty

Najskôr budeme vyhodnocovať nami navrhnutú metódou pomocou sledovania kvality tepelných máp. Následne budeme overovať správnosť modelu pomocou nami navrhnutej metódy, avšak je nutné aby metóda generovala kavlitné tepelné mapy.

4.2.2.1 Určenie kvality metódy vysvetľovania rozhodnutí modelu

Kvalitu metódy vysvetľovania rozhodnutí modelu budeme sledovať určovaním kvality tepelnej mapy. Tá v kontexte našej práce hovorí o tom, do akej miery táto mapa odzrkadľuje to, na základe čoho sa model rozhoduje. Toto budeme merať metrikami *insertion (AUC)* a *deletion (AUC)*, ktoré sme bližšie popísali v sekciu 2.2.6.6. Táto metrika nám povie, aká dobrá je naša metóda na vysvetľovanie.

Keďže naša metóda generuje tepelné mapy pomocou vygenerovania veľkého množstva náhodných masiek, je vhodné skúmať, ako sú tieto tepelné mapy konzistentné pri niekoľkých požitiach metódy na tom istom MRI snímku. Konzistentnosť máp môžeme merať pomocou podobnosti medzi jednotlivými tepelnými mapami vygenerovanými pre tú istú snímku (napr. ako súčet absolútnej hodnôt rozdielov medzi voxelmi v oboch tepelných mapách). Čím je táto podobnosť väčšia, tým je metóda pri generovaní máp viac konzistentná.

4.2.2.2 Určenie správnosti modelu

Správnosť modelu budeme určovať na základe tepelných máp vytvorených pomocou metódy na vysvetľovanie predikcií modelu. Budeme overovať do akej miery dávajú tepelné mapy zmysel v kontexte skutočnej anatómie mozgu, tj. či tepelná mapa pre správnu predikciu ukazuje na klinicky relevantné oblasti mozgu. Sledujeme, že či tepelná mapa nehovorí o tom, že sa model rozhodol na základe takej oblasti mozgu, z ktorej sa Alzheimerova choroba nedá zistiť. Veľkú úlohu pri určovaní správnosti modelu zohráva aj kvalita natrénovaného modelu, tú môžeme merať pomocou metrik z práce od Böhle et al. v ktorej sa autori zaoberali vyhodnocovaním tepelných máp vypočítaných pomocou metódy LRP. Tieto metriky sú nasledovné (relevancia je v našom prípade teplota na tepelnej mape):

- súčet relevancie v jednotlivých častiach mozgu (podľa segmentačných masiek) pre AD a CN
- hustota relevancie v jednotlivých častiach mozgu (podľa segmentačných ma-

siek) pre AD a CN, berie ohľad na veľkosť danej časti mozgu

- prírastok relevancie v jednotlivých častiach mozgu (podľa segmentačných mäsiek) vypočítaný ako pomer priemernej relevancie každej triedy v danej časti mozgu

4.3 Zhrnutie

V tejto kapitole sme navrhli metódu na vysvetľovanie rozhodnutí modelov strojového učenia a spôsob jej implementácie. Navrhnutú metódu budeme overovať na neurónových sieťach detegujúcich Alzheimerovu chorobu s cieľom odhaľovania nesprávnych rozhodnutí.

5. Implementácia

5.1 Metóda RISEI

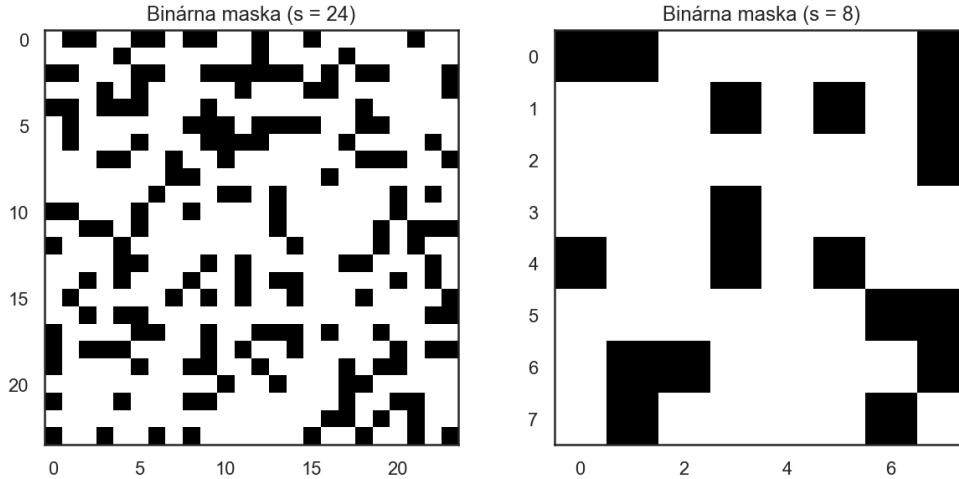
Metódu RISEI sme sa rozhodli implementovať v jazyku Python, keďže plánujeme používať knižnice pre strojové učenie akými sú *tensorflow* či *scikit-learn*.

5.1.1 Generovanie masiek

Na základe BPMN diagramu (Obr. 4.2) sme implementovali proces generovania masiek. Generovanie masiek prebieha paralelne vo viacerých procesoch použitím knižnice Python *multiprocessing*. Metóda RISE pracuje s trojrozmernými dátami, avšak diagramy v tejto sekcii zobrazujú snímky a masky v 2D (konkrétnie určitú vrstvu z 3D snímku) kvôli jednoduchšej vizualizácii. V tejto sekcii popíšeme jednotlivé kroky generovania masiek.

Vytvorenie náhodnej binárnej masky Náhodné binárne masky generujeme pomocou knižnice *numpy*. Pomocou nasledovného kódu vygenerujeme N náhodných masiek 3D binárnych matice. Obr. zobrazuje takúto binárnu maticu, ale v 2D. *size* (veľkosť) a *probability* (pravdepodobnosť) sú hyper-parametrami RISEI metódy. *size* hovorí o veľkosti generovanej masky, čím je toto číslo väčšie tým bude výsledná maska viac fragmentovaná na malé plochy. *probability* hovorí o tom, s akou pravdepodobnosťou daná plocha neprekrytá maskou. RISE používa predvolenú hodnotu *size* = 8.

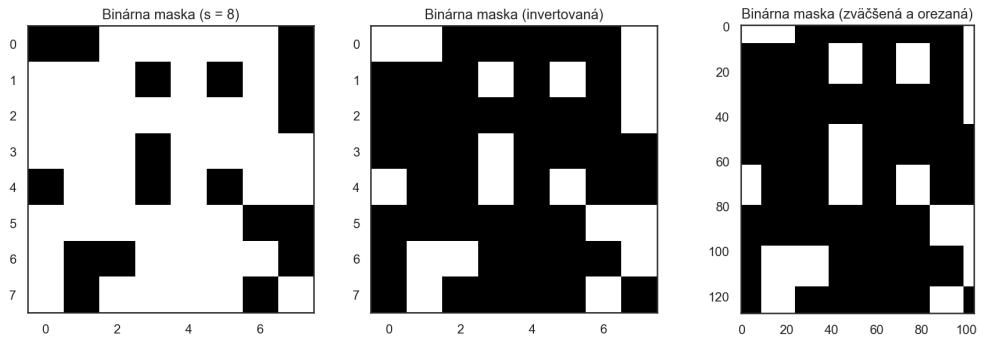
```
binary_masks = np.random.rand(N, size, size, size) < probability
```



Obr. 5.1: Porovnanie dvoch binárnych masiek s rôznou veľkosťou (*size*), čím väčšia veľkosť, tým je obrázok viac fragmentovaný. Keď je fragmentácia vyššia, zakrývame menšie časti mozgu, predpokladáme, že takto sa nám nepodarí zakryť relevantné časti z čo zapríčiní nižšiu kvalitu tepelnej mapy (predpokladáme, že v takomto prípade bude "teplo" rovnomerne rozmiestnené po celej snínke).

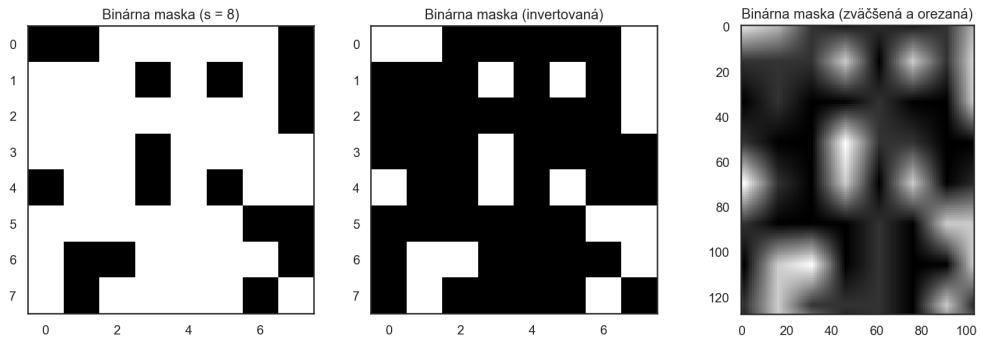
Náhodné nastavenie pozície vyrezania, zväčšenie binárnej masky a orezenie na veľkosť obrázka Binárnu masku zväčšíme na veľkosť vstupnej snímky plus menší offset (o veľkosti *size*). Následne zo zväčšenej masky na náhodnej pozícii vyrežeme masku o veľkosťi vstupnej snímky (Obr. 5.2). Táto maska určuje, ktoré miesta na snímke bude treba dokresliť - biele miesta, čiže jednotky. Tento krok v pôvodnej implementácii RISE nie je.

Zväčšenie pomocou bilineárnej interpolácie a orezanie masky na veľkosť obrázka Tak ako v poôvodnej implementácii RISE, vytvoríme "čiernu" masku na zakrytie častí obrázku. Pôvodnú binárnu masku pomocou bilineárnej interpolácie (funkcia *resize* z knižnice *scikit-learn*) zväčšíme na veľkosť o niečo väčšiu ako je vstupná snímka (aby sme mohli vykonať náhodný posun), následne vyrežeme na



Obr. 5.2: Vygenerovaná maska je zväčšená a orezaná na veľkosť vstupnej snímky (o veľkosti [104, 128, 104] pričom na obrázkoch je vizualizovaná druhá a tretia dimenzia). Úplne vľavo je binárna maska o veľkosti 8. V strede je invertovaná binárna maska (kvôli ďalšiemu pracovanou s ňou) a vpravo je orezaná binárna maska o veľkosti vstupnej snímky.

náhodnej pozícii masku o veľkosti vstupnej snímky (táto náhodná pozícia je rovnaká ako pri orezávaní binárnej masky bez interpolácie, preto je v BPMN diagrame v samostatnom kroku).



Obr. 5.3: Vygenerovaná maska je zväčšená pomocou bilineárnej interpolácie a orezaná na veľkosť vstupnej snímky (ten je o veľkosti [104, 128, 104] pričim na obrázkoch je vizualizovaná druhá a tretia dimenzia). Úplne vľavo je binárna maska o veľkosti 8. V strede je invertovaná binárna maska (kvôli ďalšiemu pracovanou s ňou) a vpravo je orezaná interpolovaná "čierna" maska o veľkosti vstupnej snímky.

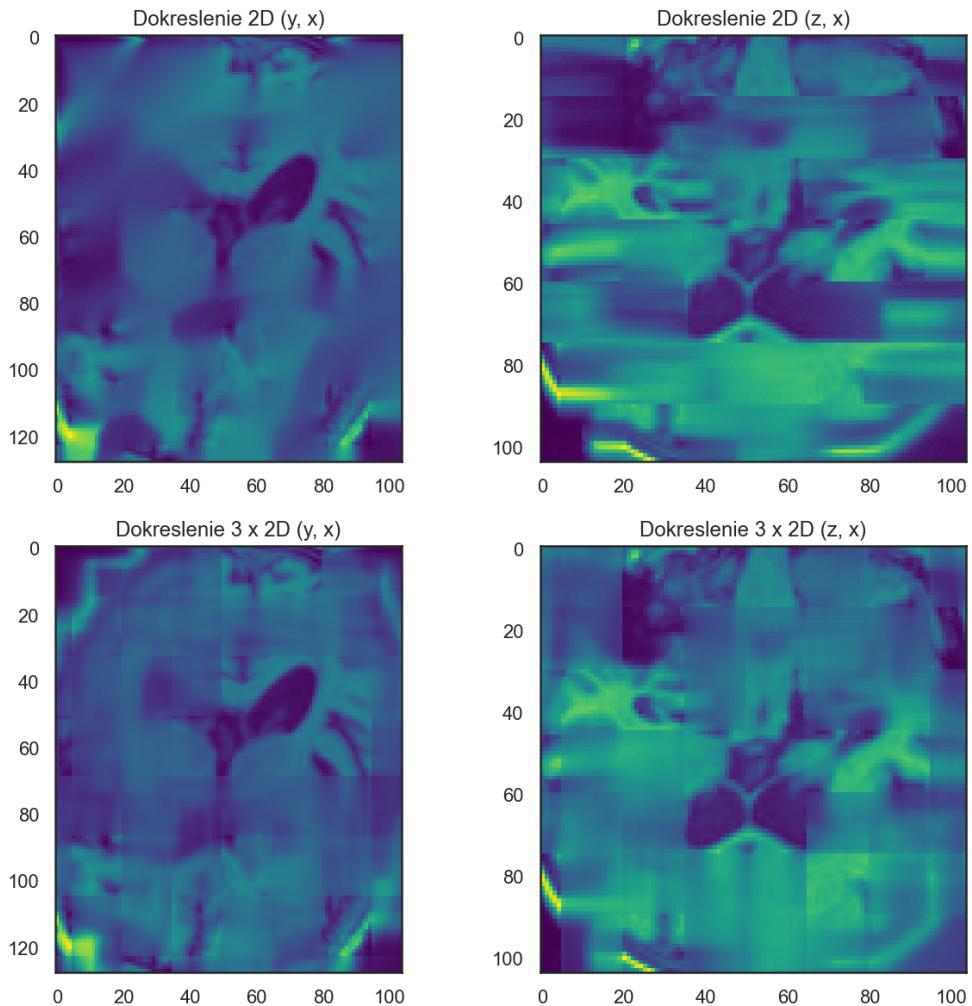
Prekrytie masky s obrázkom a dokreslenie zamaskovaných častí obrázka

Keďže pracujeme nad trojrozmernými dátami, pokúsili sme sa použiť dokreslovanie obrázka v 3D. Na to sme sa pokúsili použiť funkciu *inpaint* s knižnicou *scikit-image*, avšak dokreslenie jednej masky bolo veľmi časovo náročné (trvanie bolo až v minútach kde dokreslenie v 2D je v sekundách) a my ich potrebujeme generovať tisíce, preto sme od trojrozmerného dokreslovania upustili.

Dokreslovanie dvojrozmerných snímok z 3D snímku má avšak svoje nevýhody. Nech máme snímky o veľkosti $[z, y, x]$, pri 2D dokreslení musíme dokreslovať z snímok o veľkosti $[y, x]$ (alebo y snímok o veľkosti $[y, x]$, alebo x snímok o veľkosti $[y, z]$). Pri takomto dokreslovaní, dokreslenie z pohľadu $[y, x]$ vyzerajú byť správne, avšak z iného pohľadu, napr. $[z, x]$ sa javí byť dokreslenie nesprávne, najmä kvôli vzniknutým ostrým hranám (Obr. 5.4). Tento problém sme adresovali tak, že dokreslovanie vykonávame vo všetkých troch rovinách a následne počítame priemer pre každý voxel zo všetkých troch dokreslení. Takto je výsledok o niečo lepší, tj. z každej strany je dokreslenie lepšie ako nesprávne dokreslenie z 2D ale o niečo horšie ako správne dokreslenie z 2D. Na označenie miest, ktoré treba dokresliť sme použili zväčšenú binárnu masku (Obr. 5.2). Dokreslenie vykonávame funkciou *inpaint* z knižnice *cv2 (Open CV)*. Používame dokreslovací algoritmus *cv2.INPAINT_TELEA*, keďže pomocou neho sme dosahovali vizuálne najlepšie výsledky. Funkcia *cv2.inpaint* vyžaduje ako parameter *inpaint_radius* (Obr. 5.6), čo je jedným z hyper parametrov našej metódy.

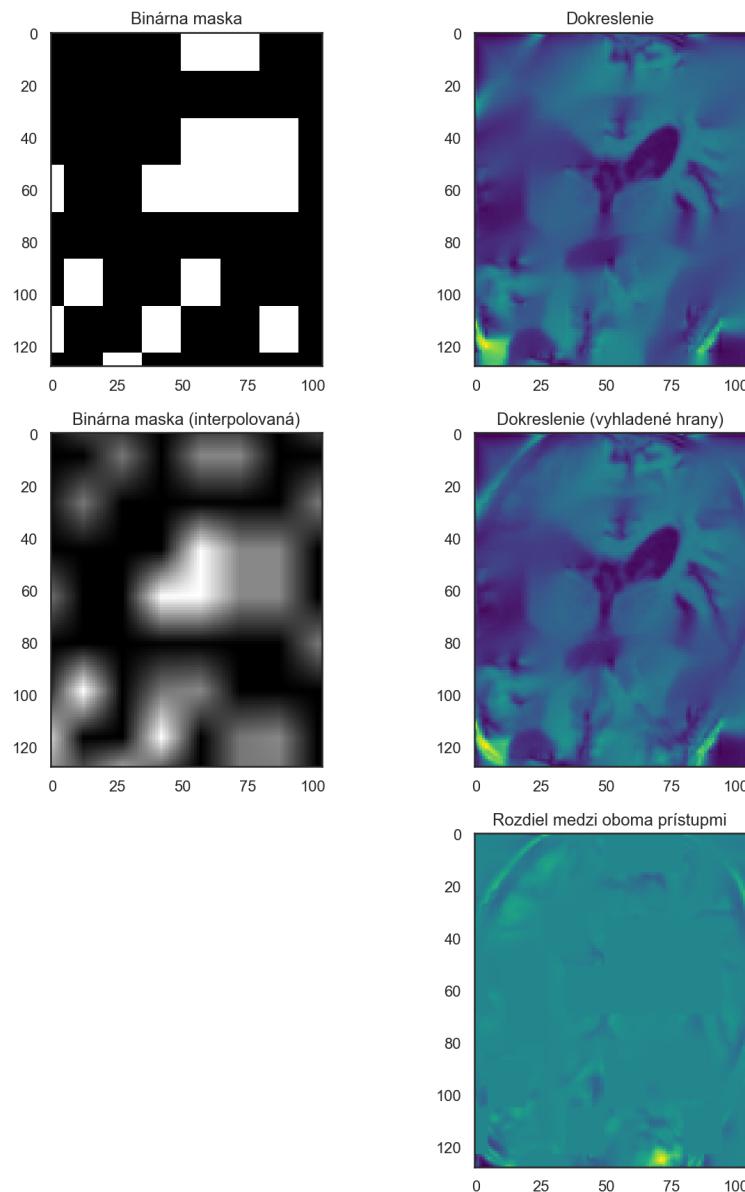
Keďže sa pôvodná implementácia RISE prekrýva miesta tak, aby nevznikali ostré hrany medzi zakrytím miestom a pôvodným obrázkom, a teda vznikol plynulý prechod, aj pri dokreslení vytvárame plynulý prechod medzi dokreslením a pôvodným obrázkom (Obr. 5.5). Tento prechod je implementovaný nasledovne.

```
# binary_mask int[z, x, y] - upsized binary mask
# image float[z, x, y] - original image
# mask float[z, x, i] - upsized and interpolated binary mask
# inpaint_radius int
inpainted = cv.inpaint(image, binary_mask, inpaint_radius,
```

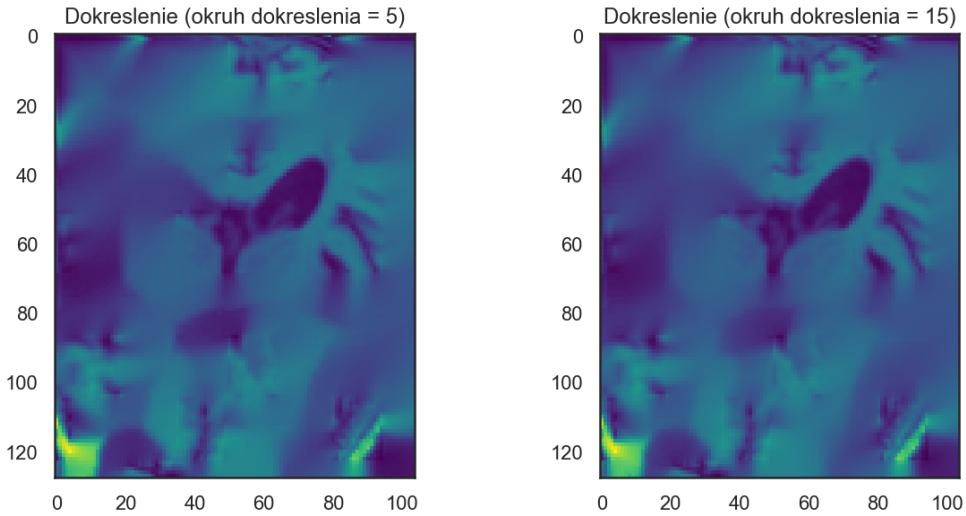


Obr. 5.4: Porovnanie 2D dokreslenia (iba v jednej dimenzií) a spriemerovaného 3x 2D dokreslenia (v každej dimenzií). Použitie iba 2D dokreslenia je kvalitné iba v jednej dimenzií a v ostatných je deštruktívne - vytvára ostré hrany. Použitie 3x 2D dokreslenia a spriemerovanie pre každý voxel produkuje celkom dobré dokreslenia po všetkých dimenziách.

```
cv2.INPAINT_TELEA)  
inpaintedImage = image * mask + inpainted * (1 - mask)
```



Obr. 5.5: Príklad vyhladzovania hrán dokreslenia - splynutie dokreslenia s pôvodným snímkom (štvrty snímok). Druhý snímok zobrazuje ostré hrany po dokreslení - bez splývania s obrázkom. Piaty snímok zobrazuje rozdiel medzi oboma prístupmi. Môžeme si všimnúť, že na obrázku sú viditeľné miesta, kde sa nachádza prechod na interpolovanej binárnej maske. O tieto miesta (informácie) je dokreslenie s vyhladenými hranami "bohatšie".

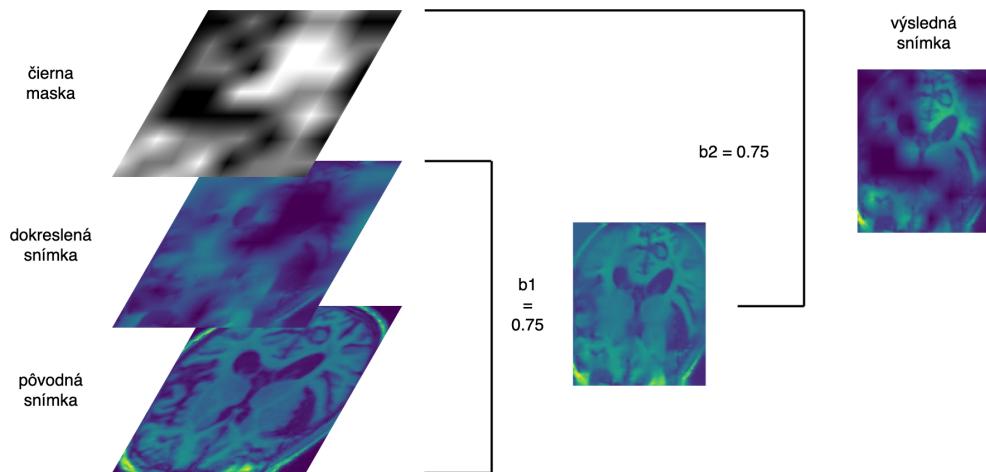


Obr. 5.6: Porovnanie okruhov dokreslenia (parameter *inpaint_radius*), rozdiel vo výsledku nie je veľmi viditeľný, avšak s väčším oruhom dokreslenia je generovanie rádovo pomalšie. (pri generovaní bolo vypnuté splynutie dokreslenia so snímkom aby bol rozdiel aspoň trochu viditeľný)

Prekrytie dokreslenej masky a čiernej masky s obrázkom Kedže v rámci metódy sa prekrývajú tri rôzne vrstvy - originálna snímka, čierna maska a dokreslená snímka môžeme tieto vrstvy skombinovať v rôznom pomere a tým vytvoriť novú snímku.

Toto sme implementovali zavedením parametrov $b1$ a $b2$ (skratka od slova prechod, angl. blend), ktoré hovoria o pomere medzi originálnym snímkom a dokresleným snímkom, a originálnym snímkom spojeným s dokreslením a čierou maskou (Obr. 5.7). Pri týchto parametroch platí, že $0 \leq b1, b2 \leq 1$. Takto zadefinované parametre mi umožňujú vytvoriť zakaskovaný snímok iba s čierou maskou ($b1 = 0, b2 = 1$) či iba s dokreslením ($b1 = 1, b2 = 0$).

Názov "čierna" maska pochádza z pôvodnej implementácie RISE, kde sa obrázok prekrýval čierou maskou. V našej implementácii neprekrývame farbou, ale hodnotou, tj. "čierna" je hodnota 0 (minimum). Okrem použitia hodnoty 0, môžeme použiť aj 1, *priemer* či *medián* (toto je ďaľším hyper-parametrom našej metódy).



Obr. 5.7: Príklad, ako vyzerá spojenie originálnej snímky, dokreslenej snímky a čiernej masky. V diagrame je zobrazený aj výsledok medzikroku spojenia dokreslenej snímky a pôvodnej snímky. Parametre boli nastavené na $b1 = 0.75$ a $b2 = 0.75$.

Zjednodušená (a menej efektívna, v produkčnej implementácii sa niektoré inštrukcie nevykonávajú keď $b1$ je 0 alebo $b2$ je 0) implementácia spojenia jednotlivých vrstiev vyzerá nasledovne.

```
# image float[z, x, y] - original image
# inpainted_blend float[z, x, y] - inpainted image
# mask float[z, x, i] - upsized and interpolated binary mask
# b1 float <0, 1>
# b2 float <0, 1>
# b2_value string - what value use in "black" mask
(min/max/mean/median)

# merge with inpainted image
new_image = (1 - b1) * original_image + b1 * inpainted_blend

value = 0 # black
if b2_value == 'max':
    value = 1 # white
elif b2_value == 'mean':
```

Názov	Dátový typ	Popis
s	int	Veľkosť strany binárnej 3D matice.
p	float	Pravdepodobnosť, že plocha nebude prekrytá maskou.
b1	float	Miera prekrytia medzi originálnym snímkom a dokresleným snímkom.
b2	float	Miera prekrytia s "čierrou" maskou.
b2_value	string	Hodnota "čiernej" masky, môže to byť minimum, maximum, medián, priemer.
in_paint_radius	float	Polomer dokreslenia algoritmom z knižnice OpenCV.

Tabuľka 5.1: Zoznam parametrov metódy RISEI.

```
value = np.mean(original_image)
elif b2_value == 'median':
    value = np.median(original_image)
# merge with "black" mask
new_image = b2 * mask * new_image + (b2 * (1 - mask) * value)
```

Kompletný zoznam parametrov metódy RISEI sa nachádza v tabuľke 5.1.

5.1.2 Vytvorenie tepelných máp

Na základe návrhu (Sekcia 4.1) sme implementovali vytváranie tepelných máp. Keďže generovanie tepelnej mapy si vyžaduje vygenerovať veľký počet zamaskovaných snímok, ktoré v istom momente musia byť všetky uložené v pamäti, generujeme a vyhodnocujeme zamaskované snímky v dávkach (angl. batch). Zdrojový kód nižšie, implementuje vytvorenie jednej tepelnej mapy. Príklad vytvorennej tepelnej mapy uvádzame na obrázku 5.8.

```
# image_x float[z, x, y, 1] - original image
# masks_count int - how many masks are generated to create a heatmap
# batch_size - how many masks to evaluate on model
# risei_batch_size int - how many masks to generate in one batch
```

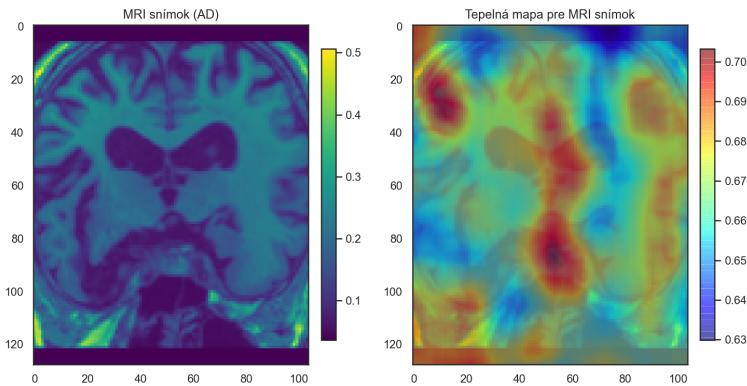
```
# seed int int - seed for mask generation
# cls_idx int - index of target class in model output vector
# model tf.keras.Model - instance of tensorflow model

risei = RISEI(s=8, p=0.5, b1=0.5, b2=0.5, b2_value='median',
    in_paint_radius=5)
heatmap = np.zeros(shape=image_x.shape[:3])
batch_count = math.ceil(masks_count / risei_batch_size)
weights = 0

for batch_idx in range(batch_count):
    batch_masks_count = min(risei_batch_size, masks_count - batch_idx *
        risei_batch_size)
    # reshape input for RISEI since it works with [z, y, x] shape
    # batch_x float[z, x, y] - images to evaluate with masks already
        applied
    # masks float[z, x, y] - interpolated binary masks (so we know which
        places we inpainted or masked)
    batch_x, masks = risei.generate_masks(batch_masks_count,
        image_x.reshape(image_x.shape[:3]), seed=seed)
    y_pred_batch_x = model.predict(batch_x.reshape((-1, *image_x.shape)),
        batch_size=batch_size)

    for mask, y_pred in zip(masks, y_pred_batch_x):
        # invert the mask, since 1 is for no masking
        # y_pred is the activation for the input masked image on last
            layer (softmax)
        heatmap = heatmap + y_pred[cls_idx] * (1 - mask)
        weights += y_pred[cls_idx]

heatmap = heatmap / weights
```



Obr. 5.8: Príklad vytvorennej tepelnej mapy (vpravo) k MRI snímku (vľavo). Mierka vujadruje priemernú mieru aktivácie pre daný voxel.

5.1.3 Vyhodnotenie tepelných máp

Zatiaľ sme implementovali, podľa návrhu riešenia (Sekcia 4.2.2.1), iba metriky *insertion* a *deletion*.

5.1.3.1 Metriky insertion & deletion

Tieto metriky fungujú tak, že postupne odstraňujeme/pridávame pixely z obrázku a tieto obrázky vkladáme do modelu a zaznamenávame si aktiváciu na poslednej vrstve pre predikovanú triedu. V prípade obrázkov, a teda dvojrozmerných dát je to ešte výpočtovo zvládnuteľné, avšak v prípade trojdimenzionálnych rádiologických simkov to už môže byť problém. Naše vstupné snímky majú po zmenení rozmer [104, 128, 104], čiže ak aby sme odstraňovali zo snímku po jednom voxelu, museli by sme vykonať 1 384 448 evaluácií pomocou nášho modelu (čo trvá niekoľko hodín, aj pri evaluovaní v maximálnych možných dávkach vzhľadom na pamäť grafickej karty). Preto sme sa rozhodli, že budeme pridávať po n (100) voxeloch v každom kroku. V prípade metódy *insertion* vkladáme do snímku plného núl (môžeme prípadne aj jednotiek). Keďže kód je rozsiahlejší, uvedieme len pseudokód.

```
method = 'insertion'
```

```
step_size = 150 # how many voxels to insert/delete in one evaluation
image_x, image_y = get_image()
image_y_pred = model.predict(image_x)
heatmap = get_heatmap()
voxels = get_ordered_voxels_by_heat(heatmaps)
sequence = get_images_sequence(voxels, step_size) # create a sequence
    from images where each next image has n inserted/deleted voxels
y_pred = []

for batch_x, batch_y in sequence:
    batch_y_pred = model.predict(batch_x)
    for y in batch_y_pred:
        y_pred.append(y)

auc = metrics.auc([i * step_size for i in range(len(y_pred))], y_pred) /
    get_voxels_count(image_x)
```

5.2 Model na detekciu Alzheimerovej choroby na základe MRI snímok

V tejto sekcií popíšem implementáciu, z ktorého predikcií budeme vytvárať tepelné mapy. Náš model - neuónovú sieť sme sa rozdihodli implementovať v knižnici Tensorflow (v2.3.0). Naším cieľom nie je natrénovať najlepší model na detektciu Alzheimerovej choroby, ale model ktorý je použiteľný na overenie nami narvhnutej metódy. Preto nevykonáme komplexnejšie prístupy k detektii Alzheimerovej choroby, ktoré sme popísali v analýze (Sekcia 2.3), ako je napríklad učenie prenosom pomocou autoenkodéra.

5.2.1 Dátová sada

Použili sme dátovú sadu ADNI. Ako vstup modelu je celý MRI snímok (tj. všetky tri dimenzie), nepoužívame žiadné ine údaje z dátovej sady ADNI, ako napríklad demografické údaje a pod. keďže model plánujeme používať iba na vytváranie tepelných máp pre vstupné snímky.

V dátovej sade sa nachádza celkom 502 MRI snímok, z toho 311 pacientov s Alzheimerovou chorobou (AD) a 191 bez (CN). Dátovú sadu máme teda nevyváženú a model môže začať preferovať jednu triedu. Na predĺženie tomuto javu existuje niekoľko techník, napríklad nadzorkovanie (angl. oversampling) alebo podzorkovanie (angl. undersampling) kedy sa doplní synetickými minoritná trieda, alebo sa odstránia nejaké pozorovania z majoritnej triedy. My sme sa však rozhodli nastaviť predikovaným triedam váhy, ktoré sú zohľadnené v chybovej funkcií, taktiež sme nainicializovali chybu¹ pre neuróny na poslednej vrstve aby reflektovala to, že triedy sú nevyvážené.

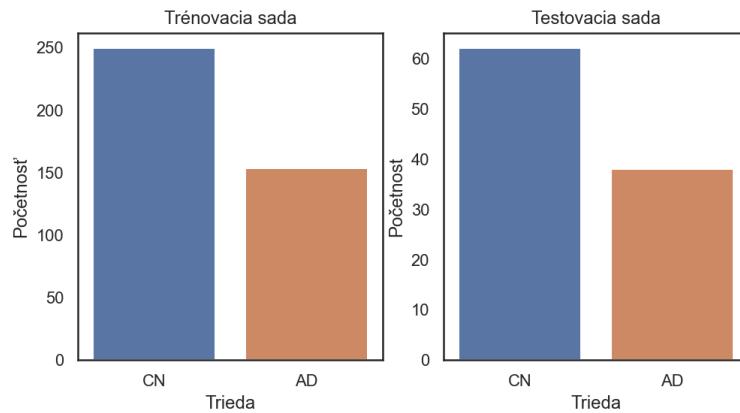
Dátovú sadu MRI snímkov pacientov sme náhodným výberom rozdelili na trénovaciu a testovaciu v pomere 80/20. Validačnú sadu sme nevytvárali, z dôvodu malého množstva dát, ktoré máme k dispozícii a taktiež neplánujeme prehľadávať priestor hyper parametrov za účelom nájsť ich najoptimálnejšiu kombináciu. V budúcnosti možno zvážime vykonanie krížovej validácie (angl. cross validation) pri trénovaní. Aj po rozdelení sa nám podarilo zachovať pôvodný pomer medzi triedami – 62/38 (Obr. 5.9).

5.2.1.1 Predspracovanie

MRI snímky boli predspracované štandardnou postupnosťou nástroja freesurfer², avšak nevykonali sme odstránenie lebky z MRI snímkov. Okrem iného sme vykonali:

¹https://www.tensorflow.org/tutorials/structured_data/imbalance_data#optional_set_the_correct_initial_bias

²<https://surfer.nmr.mgh.harvard.edu/>



Obr. 5.9: Početnosť tried medzi trénovacou a testovacou sadou - je zrejmá prevaha triedy AD.

- Upravenie vstupných snímok na rovnakú veľkosť $104 \times 128 \times 104$ voxelov. Esmaeilzadeh et al. upravili vstupné snímky na veľkosť $116 \times 130 \times 83$, k týmto číslam sme sa pokúsili priblížiť. Pomer veľkostí dimenzií ale nemáme rovnaký, aj z dôvodu, že sme nevykonali odstránenie lebky zo vstupných snímok.
- Štandardizáciu vstupných dát (preškálovanie na rozsah $< 0,1 >$) nasledovným vzorcom: $\frac{(image_x - images_min)}{(images_max - images_min)}$.

5.2.1.2 Augmentácie

Kedže použitá dátová sada obsahuje malé množstvo dát, rozhodli sme sa dát augmentovať (Obr. 5.10). Dáta náhodne augmentujeme v každej dávke (angl. batch). Implementovali sme nasledovné augmentácie:

- Vymenenie hemisfér mozgu (Esmaeilzadeh et al. [23]) s pravdepodobnosťou 50%
- Náhodná rotácia o 0 až 5 stupňov s pravdepodobnosťou 20%
- Náhodné priblíženie do 80% veľkosti snímku s pravdepodobnosťou 20%

- Náhodné gaussovské rozmazanie ($\max \sigma = <0.85, 1>$) s pravdepodobnosťou 20%
- Náhodný gaussovský šum pravdepodobnosťou 20%

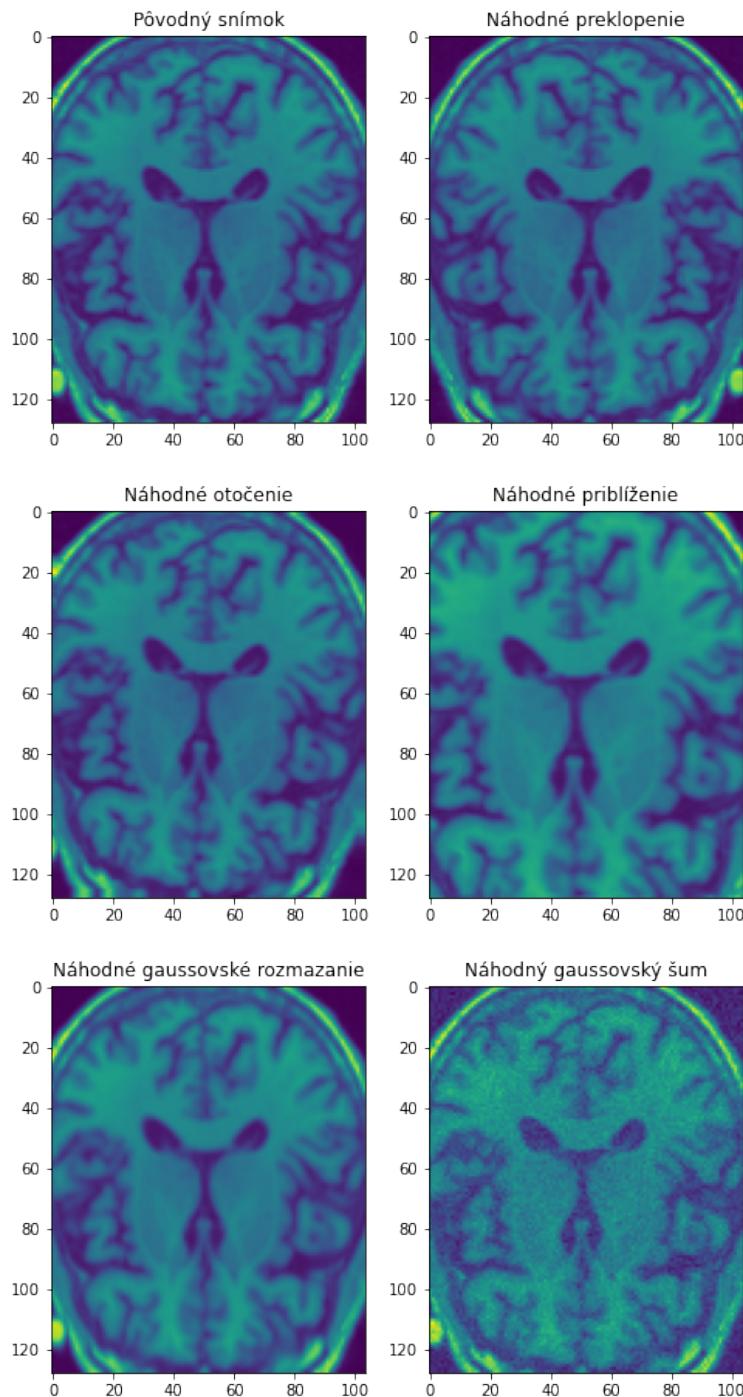
5.2.2 Model

Rozhodli sme sa implementovať a porovnať niekoľko architektúr neurónových sietí.

3D konvolučná neurónová sieť od Esmaeilzadeh et al. Túto neurónovú sieť sme sa rozhodli implementovať, pretože jej autori pomocou nej dosiahli veľmi dobré výsledky (94.1% presnosť). Implementovali sme jednoduchšiu verziu, ktorá dosahovala lepšie výsledky, opísali sme ju v sekcii 2.3. Táto neurónová sieť ma celkovo $2\ 899\ 778$ parametrov.

2D ResNet a 3D ResNet Keďže reziduálne neúronové siete dosahujú pri klasifikačných úlohách nad obrazovými dátami veľmi dobre výsledky vyskúšame aj tieto architektúry. V prípade 2D ResNet-u používame 2D konvolúcie, tie nám budú fungovať aj napriek tomu, že máme 3D dátu. Vstup do 2D ResNet-u je tiež 3D matica, ktorej tretia dimenzia býva o obvykle o dĺžky 1 alebo 3 (RGB), v našom prípade bude o veľkosti poslednej dimenzie snímku. Rozmery vstupných dát pre 2D ResNet budú [104, 128, 104] a pre 3D ResNet [104, 128, 104, 1]. Za konvolučné vrstvy a globálnu združovaciu vrstvu sme pripojili dve plne prepojené vrstvy s 512, 256 a 128 neurónami a s aktivačnou funkciou *ReLU*, následne už nasleduje iba posledná vrstva s aktiváciou *softmax*. Tieto neurónové siete majú celkovo $12\ 689\ 602$, resp. $34\ 356\ 354$ parametrov.

Do neurónových sietí sme ešte pridali dropout a dávkovú normalizáciu (angl. batch normalization). Dropout sme pridali pred plne prepojené vrstvy. Dávkovú normalizáciu sme pirdali v konvolučných vrstvách pred aplikovaním nonlinearity, tak ako je to odporučené od Ioffe et al. v *Batch Normalization: Accelerating Deep Network*



Obr. 5.10: Príklady aplikácie implementovaných augmentácií.

Training by Reducing Internal Covariate Shift. Na poslednej vrstve sa nachádzajú dva neuróny s aktivačnou funkciou *softmax*.

5.2.3 Trénovanie

Pri trénovaní sme použili:

- kategorickú entropiu (angl. categorical crossentropy) ako chybovú funkciu s podporou pre nevyvážené triedy (tj. táto funkcia brala ohľad na váhy tried, ktoré sme nastavili nepriamo úmerne ich veľkosti),
- optimalizačný algoritmus Adam s prevolenými nastaveniami,
- exponenciálne tlmenie rýchlosťi učenia (angl. learning rate decay), s hodnotou 0,96 každých 25 epoch,
- skoré zastavenie trénovania ak sa metrika AUC (plocha pod krivkou) nezlepšila za posledných 50 epoch,
- veľkosť dávky (angl. batch size) – 10 (vždy tak, aby sme naplno využili pamäť grafickej karty),
- l_2 regularizáciu (rovnako ako Esmaeilzadeh et al.).

Trénovali sme tak, že sme, začali s obyčajným modelom a postupne sme pridávali augmentácie, dávkovú normalizáciu, dropout a regularizáciu pričim sme postupne dodačovali parametre. Najlepšie výsledky sme dosiahli s architektúrou 3D ResNet s presnosťou 80% (Tabuľka 5.2). Nepodarilo sa nám nám teda priblížiť k výsledkom analyzovaných prác, čo však v konečnom dôsledku ani nie je cieľom tejto práce.

	3D CNN			3D ResNet			2D ResNet		
	Acc.	Sens.	Spec.	Acc.	Sens.	Spec.	Acc.	Sens.	Spec.
Baseline	0.71	0.76	0.63	0.71	0.79	0.57	0.67	0.77	0.50
+ Augmentácie	0.67	0.68	0.66	0.69	0.84	0.45	0.76	0.90	0.52
+ Batch Norm	0.75	0.77	0.71	0.80	0.85	0.71	0.77	0.89	0.58
+ Dropout	0.75	0.76	0.74	0.74	0.94	0.42	0.77	0.85	0.63
+ Regularizácia (12)	0.71	0.70	0.71	0.79	0.87	0.66	0.78	0.89	0.61

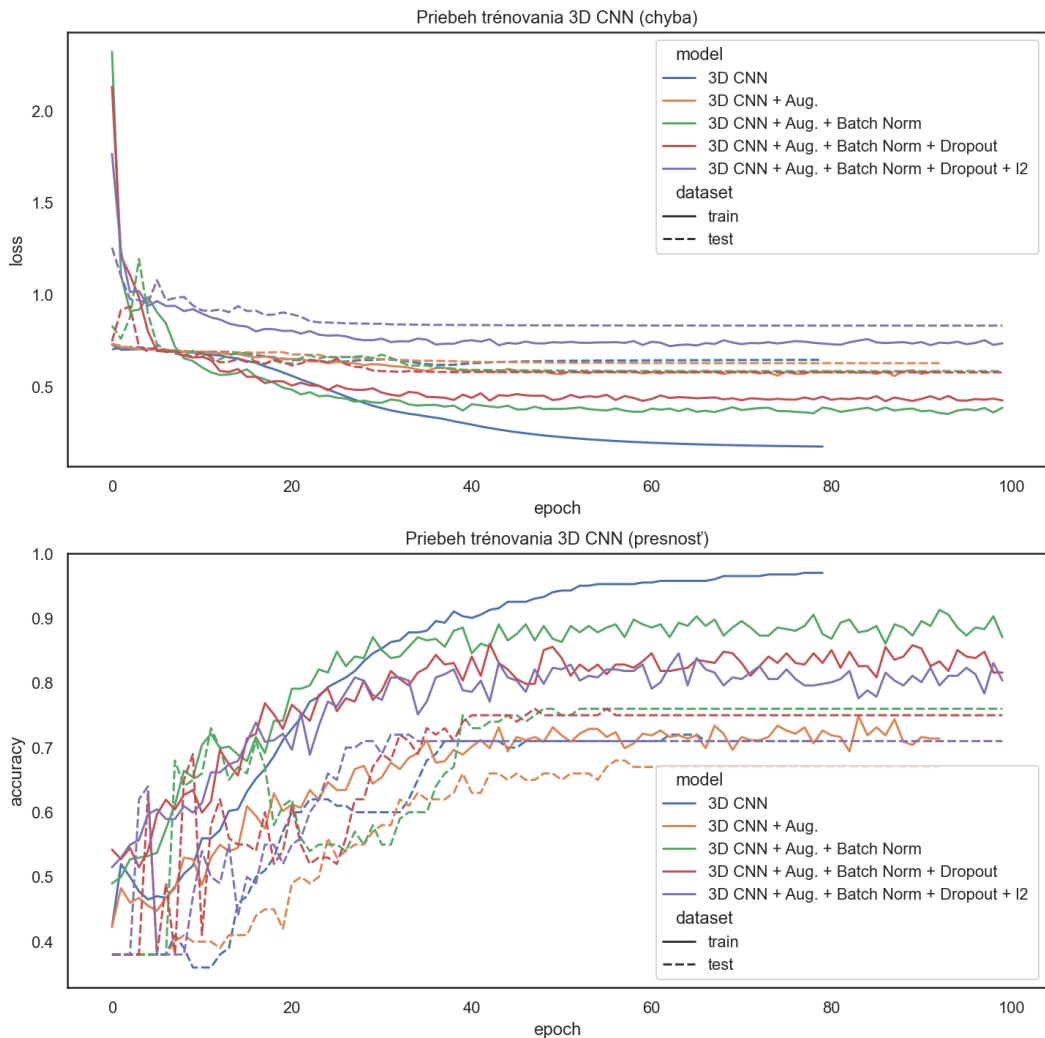
Tabuľka 5.2: Výsledky trénovania. Acc. = presnosť (angl. Accuracy), Sens. = senzitivita (angl. Sensitivity), Spec. = Specificita (angl. Specificity)

Oproti Esmaeilzadeh et al., ktorí dosiahli presnosť až 94%, sme dosiahli presnosť len 72% avšak sme mali menej dát (o 339 pozorovaní menej), neodstraňovali sme zo snímok lebku a nepoužili sme pri klasifikácii vek pacienta. Avšak robili sme viac augmentácií, no po ich pridaní sa úspešnosť modelu zhoršila (Obr. 5.11) (ale následne sa už iba zlepšovala), je teda možné, že niektoré augmentácie nie sú nekorektné a deštruktívne voči vstupným snímkam a vedú k zhoršeniu výkonnosti modelu. Aj po pridaní veľmi slabej regularizácie, sa úspešnosť modelu zhoršila. V prípade 2D a 3D ResNet architektúr sa nám podarilo dosiahnuť lepšie výsledky, avšak v ich prípade sa sieť neskôr začala pretrénovať (Obr. 5.12) aj napriek použítej regularizácii, čo môže naznačovať, že sú tieto architektúry na náš problém príliš komplexné.

Identifikovali sme nasledovné možné vylepšenia (zoradené podľa subjektívneho pomera úsilie/vplyv):

- Porovnať jednotlivé augmentácie a vyhodiť tie deštruktívne.
- Odstránenie lebky zo snímok.
- Nájsť a použiť viac dát.
- Krížová validácia v prípade väčšieho nastavovania hyperparametrov.
- Učenie prenosom pomocou autoenkodéra [26].

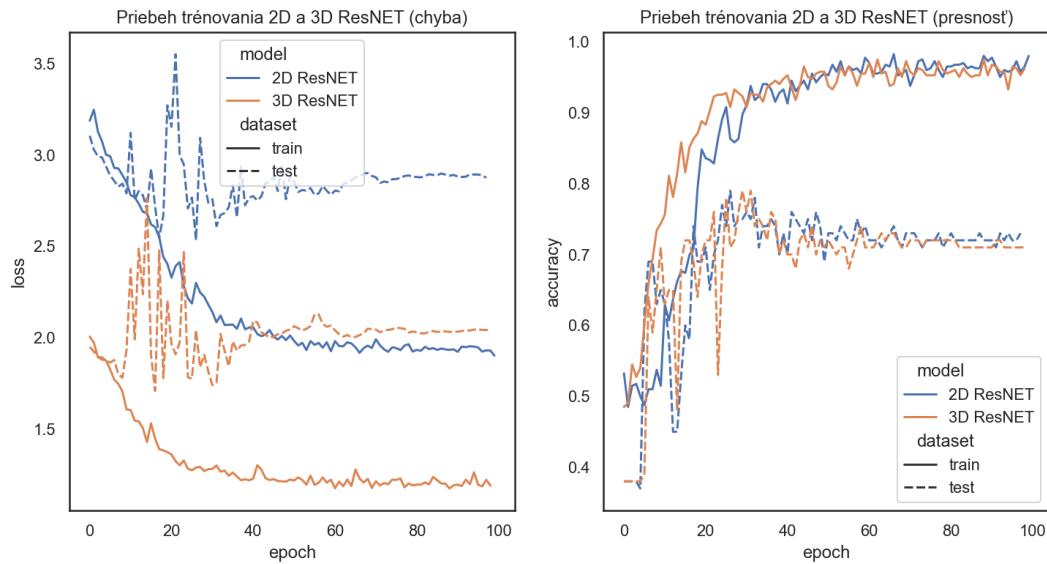
Vzhľadom na to, že naším cieľom nie je natrénovať najlepšiu neurónovú sieť, nevykonali sme všetky identifikované možné vylepšenia. Vykonali sme porovnanie jednotlivých augmenácií na architektúre 3D CNN a to tak, že sme pre každú augmentáciu natrénovali samostatný model (bez dropout-u a regularizácie) a augmentovali sme vstupné snímky s pravdepodobnosťou 50%. Sledovali sme, či sa aj s augemntovanými snímkami netrénované modely bez akejkoľvek regularizácie dokážu preučiť. Model sa nepreucil iba pri augmentácii *náhodné priblíženie*. Následne sme natrénovali modely pre jednotlivé architektúry znova, lepší model sa nám podarilo natrénovať iba pre architektúru 3D CNN, presnosť - **0.78**, senzitivita - **0.81**, špecifickita - **0.73**.



Obr. 5.11: Priebeh trénovania 3D konvolučnej neurónovej siete, čím viac sme pridali regularizácie (l2 alebo dropout) tým sme dosiahli horšie výsledky. Po pridaní augmentácií sa úspešnosť modelu zhoršila, avšak následne sa už iba zlepšovala.

5.3 Zhrnutie

V tejto kapitole sme opísali, ako sme implementovali nami navrhovanú metódu a spôsob jej overenie. Opísali sme implemetáciu kľúčových prvkov (generovanie masiek a vytvorenie tepelnej mapy) navrhovanej metódy RISEI a jej hyper parametre. Rovnako sme opísali aj implementáciu neurónovej siete, architektúru, spôsob tré-



Obr. 5.12: Priebeh trénovania 2D a 3D ResNet neurónovej siete. V oboch prípadoch sme použili dropout aj regularizáciu. Od približne 30-tej epochy sa neurónová sieť začala pretrénovať, aj napriek regularizácii. Ako vylepšenie je možné skúsiť silnejšiu regularizáciu (väčšiu hodnotu l2 a dropout), ak ani to nepomôže, je možné, že architektúra je príliš komplexná náš problém (neurónová sieť je príliš hlboká). Môžeme ďalej skúsiť odstrániť jednu plne prepojenú vrstvu alebo znížiť počet neurónov v týchto vrstvách.

novania a použitú dátovú sadu. Metódu RISEI sa nám podarilo implementovať a je ju možné použiť v experimentoch. Podarilo sa nám natrénovať niekoľko architektúr neurónových sietí, pričom sme identifikovali možné príčiny výsledkov, ktoré dosahujú (a navrhli spôsob ako ich riešiť). Nie všetky identifikované príčiny sa nám podarilo vyriešiť, keďže cieľom tejto práce nie je natrénovať najlepší model, ale natrénovať taký model, ktorý je postačujúci na overenie navrhnutej metódy.

6. Overenie riešenia

Kedže sme natrénovali viacero modelov, ako prvé sme vykonali ich porovnanie pomocou implementáciou metódy RISE pre 3D snímky. Následne sme vybrali z nich najlepší model, ktorý sme používali v ďalších experimentoch.

Adresovanie náhodnosti metódy RISE pri porovnávaní dvoch roznych modelov Kedže vygenerované masky sú náhodné, pri porovnávaní dvoch metód (kombinácií rôznych parametrov metódy RISEI) generujeme rovnaké binárne masky pre každý i-ty snímok, tj. zakryté pozície sú rovnaké, rôzna je len hodnota zakrycia. Vygenerované binárne masky pre jednotlivé snímky v testovacej sade sú stále náhodné (a teda aj medzi sebou rôzne). Rovnaké sú len binárne masky pre dve rôzne generovania masiek pre ten istý snímok v testovacej sade. Takto dosiahneme kvalitnejšie výsledky, pretože žiadna metóda nebude mať výhodu, že náhodou zakryla rôzne časti snímky lepšie a teda bude dôležité ako ich zakryla.

6.1 Experimenty

6.1.1 Porovnanie metódy RISE na natrénovaných modeloch

V tomto experimente sme porovnali metódu RISE na nami natrénovaných modeloch. Vybrali sme najlepšie natrénované modely pre každú architektúru (Tabuľka

		3D CNN	3D ResNET	2D ResNET
Insertion (AUC)	priemer	0.50	0.46	0.38
	medián	0.53	0.13	0.30
Deletion (AUC)	priemer	0.53	0.81	0.62
	medián	0.60	0.83	0.43

Tabuľka 6.1: **Porovnanie metódy RISE na rôznych architektúrach.** Vybrali sme najlepšie natrénované modely pre každú architektúru (Tabuľka 5.2). Pre insertion sú lepšie vyššie hodnoty (očakávame, že keď vložíme najpodstatnejšie voxely aktivácia bude stúpať), pre deletion sú lepšie nižšie hodnoty (očakávame, že keď ostránieme najdôležitejšie voxely, aktivácia bude klesať).

5.2). Použili sme nami implementovaný metódu RISEI pričom sme nastavili jej parametre tak, aby fungovala ako metóda RISE. Parametre sme nastavili nasledovne: $s = 8$, $p = 1/3$, $b1 = 0$, $b2 = 1$ (opis parametrov sa nachádza v tabuľke 5.1). Na vytvorenie tepelnej mapy sme vygenerovali 1024 masiek. Pri vyhodnocovaní metrík insertion a deletion sme nastavili veľkosť kroku na 2500 voxelov (takto trvalo vyhodnotenie tepelnej mapy 3 minúty). Vybrali sme 25 náhodných snímkov z testovacej sady, generovanie a vyhodnotenie tepelných máp k nim trvalo približne 1 hodinu. Najlepšie výsledky sme dosiahli na architektúre 3D CNN (Tabuľka 6.1).

6.1.2 Stabilita tepelných máp

Keďže metóda RISEI (aj metóda RISE z ktorej vychádzame) používa na generovanie tepelným máp náhodné masky, výsledná tepelná mapa je touto náhodnosťou ovplyvnená a je teda do určitej miery náhodná. Očakávame, že čím viac masiek vygenerujeme, tým bude vplyv náhodny na výslednú tepelnú mapu nižší. Keď teda pre tú istú snímkovu vygenerujeme niekoľko tepelných máp, tieto tepelné mapy sa budú lísiť čo najmenej = budú stabilné.

Metóde RISEI sme nastavili nasledovné parametre $s = 8$, $p = 1/3$, $b1 = 0$, $b2 = 1$ a $b2_value = 1$ - nepoužívame dokreslenie, keďže je časovo náročné a vykonávame veľké množstvo experimentov. Uvažujeme tak, že nezáleží na tom, čo je hodnota

zakrytie vo vygenerovaných maskách, pokiaľ tá hodnota nie je náhodná.

Porovnanie vytvorených tepelných máp N vytvorených tepelných máp porovnávame tak, že počítame štandardnú odchílku pre každý voxel medzi vytvorenými tepelnými mapami. Tak z tepelných máp o rozmere $[N, z, y, x]$ vznikne 3D matica štandardných odchílok $[z, y, x]$. Ak má voxel rovnakú/blízku hodnotu tepla medzi tepelnými mapami, štandardná odchýlka nula/blízka nule. Z 3D matice štandardných odchílok vypočítame priemernú/strednú hodnotu - táto hodnota reprezentuje vzniknutú chybu medzi tepelnými mapami plynúcu z náhodnosti tepelných masiek.

6.1.2.1 Experiment 1 (jeden snímok)

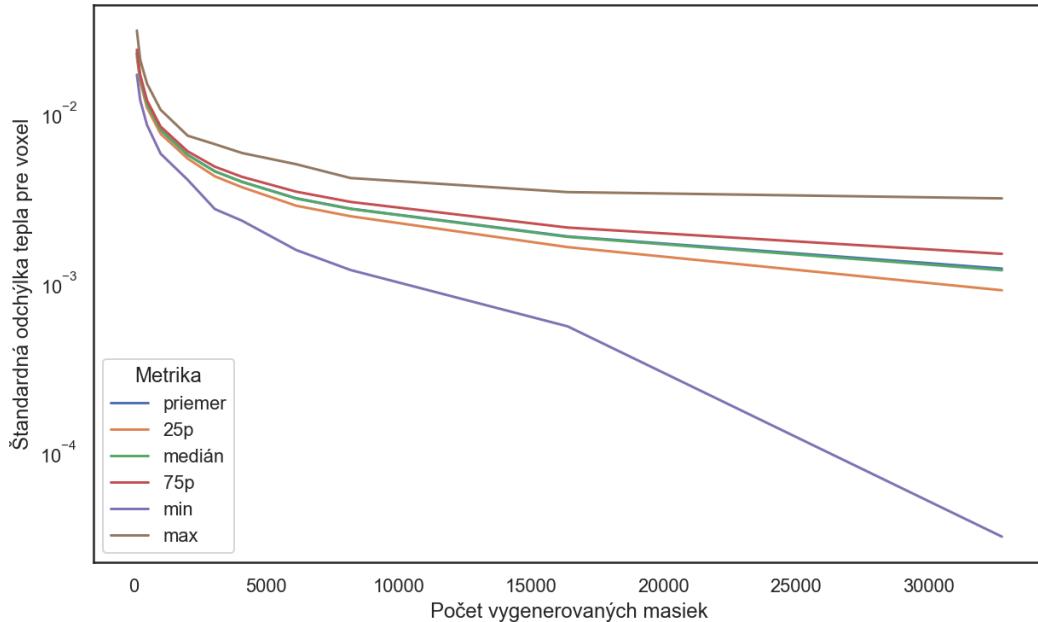
Pre náhodný snímok vytvoríme K tepelných máp, pričom tepelné mapy vytvárame z 16, 128, 256, 512, 1024, 2048, 3072, 4096, 6144, 8192, 16384 alebo 32768 masiek. Kvôli vysokej pamäťovej náročnosti, v experimentoch, v ktorých vytvárame tepelné mapy z vysokého počtu masiek vytvoríme menej tepelných máp (Tabuľka 6.2). Zistili sme, že s vyšším počtom vygenerovaných masiek chyba klesá logaritmicky (Obr. 6.1).

6.1.2.2 Experiment 2 (viacero snímkov)

Kedže sme v predchádzajúcom experimente overovali stabilitu iba na jednom snímku, v tomto experimente overíme stabilitu na viacero snímkoch. Z testovacej sady sme vybrali 5 TP (viď. Zoznam použitých skratiek), 5 TN, 5 FP a 5 FN pozorovaní (tj. celkovo 20 pozorovaní), podľa toho ako ich neurónová sieť označila. Takto zabezpečíme vyváženosť tried pozorovaní v experimente. Kvôli časovej aj pamäťovej náročnosti tohto experimentu sme vytvárali 10 tepelných máp pre každé jedno pozorovanie. Výsledky boli takmer identické s predchádzajúcim experimentom (Sekcia 6.1.2.1) a trend poklesu chyby pri zvyšujúcom počte masiek bol zachovaný (6.3). Z oboch experimentov vyplýva, že je vhodné použiť vyšší počet

Počet vygenerovaných masiek	Počet vytvorených tepelných máp	Medián štandardnej odchýlky pre voxel (chyba)
16	100	0.0640
128	100	0.0225
256	100	0.0160
512	100	0.0113
1024	100	0.0080
2048	100	0.0057
3072	50	0.0045
4096	50	0.0039
6144	25	0.0031
8192	25	0.0027
16384	15	0.0019
32768	5	0.0011

Tabuľka 6.2: Stabilita vytvorených tepelných máp podľa počtu vygenerovaných masiek pre jeden snímok. S vyšším počtom vygenerovaných máp chyba výrazne klesá. Už pri 2048 maskách je chyba zanedbateľná, keďže hodnoty voxelov v tepelných mapách sú z intervalu $< 0, 1 >$.



Obr. 6.1: Stabilita vytvorených tepelných máp podľa počtu vygenerovaných masiek. Os y je v logaritmickej škále a reprezentuje chybu. Táto chyba klesá logaritmicky s vyšším počtom vygenerovaných masiek. Priemer sa veľmi blíži mediánu, preto ho na diagrame nie je takmer vôbec vidieť.

masiek pri vytváraní tepelných máp, aby sa odstránil vplyv náhody. Ako vhodným počet považujem 2048 masiek, pri tomto počte je chyba vzhľadom na hodnoty v tejepnej mape minimálna (Tavuľka 6.2, 6.3).

6.1.3 RISE vs RISEI

V tomto experimente sme porovnali RISE a rôzne nastavenia metódy RISEI. Použili sme 3D CNN model s úspešnosťou 72%, senzitivitou 73% a špecifitou 71%, toto nie je rovnaký model ako ten, ktorý sme použili v predchádzajúcom experimente, ale jeden z prvých modelov, ktorý sme natrénovali. pretože sme sa snažili overiť metódu RISEI čo najskôr. Najnovší 3D CNN model sme ešte nestihli použiť v tomto experimente.

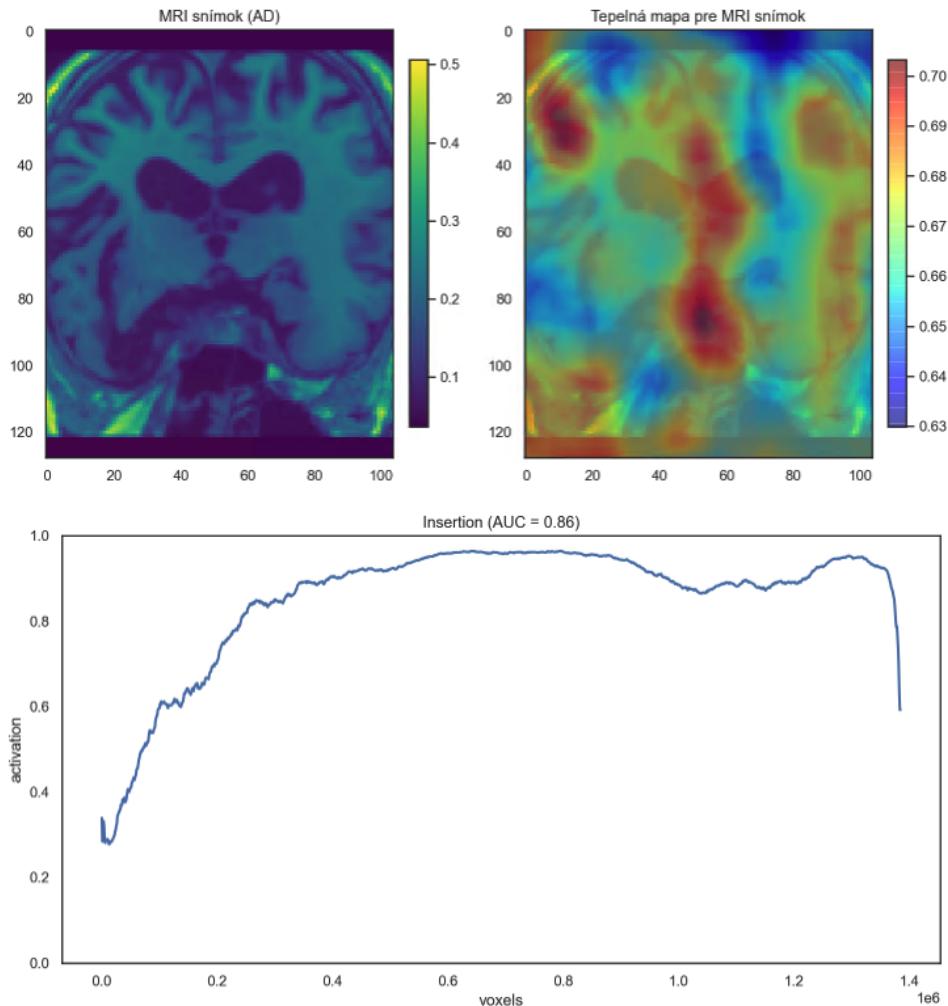
Počet vygenerovaných masiek	Medián štandardnej odchýlky pre voxel
16	0.0594
128	0.0207
256	0.0160
512	0.0105
1024	0.0074
2048	0.0052
3072	0.0043
4096	0.0037

Tabuľka 6.3: Stabilita vytvorených tepelných máp podľa počtu vygenerovaných masiek pre 20 snímkov. Trend poklesu chyby oproti prvému experimentu (Tabuľka 6.2) ostal zachovaný.

	Insertion	
	Priemer	Medián
b1 = 0, b2 = 1, b2_value = min (RISE)	0.43	0.37
b1 = 0, b2 = 1, b2_value = max	0.65	0.67
b1 = 0, b2 = 1, b2_value = median	0.52	0.48
b1 = 1, b2 = 0	0.53	0.47
b1 = 1, b2 = 0.25, b2_value = min	0.49	0.42
b1 = 1, b2 = 0.5, b2_value = min	0.44	0.37
b1 = 1, b2 = 0.75, b2_value = min	0.39	0.30

Tabuľka 6.4: **Porovnanie rôznych nastavení metódy RISEI.**
Najlepšie výsledky dosiahla metóda bez použitia dokreslenia ale s prekrytím hodnotou jedna.

Použili sme rovnaké nastavenie parametrov ako v predchádzajúcim experimente (Sekcia 6.1.1) a menili sme len parametre $b1$, $b2$ a $b2_value$. Parametre in_paint_radius sme nastavili na hodnotu 5. Vyhodnocovali sme zatiaľ len metriku *insertion*, najlepšie výsledky sme dosiahli bez použitia dokreslenia ale s prekrytím hodnotou jedna (Tabuľka 6.4). Metóda s dokreslením si počína horšie, ale stále lepšie ako prekrytie s hodnotou nula pôvodná metóda RISE). Obrázok 6.2 zobrazuje príklad vygenerovanej tepelnej mapy pre MRI snímkov a výsledný graf zmeny aktivácie pre metriky *insertion*.



Obr. 6.2: Vygenerovaná tepelná mapa a graf zmeny aktivácie po pridávaní voxelov pre vybraný MRI snímok. Metrika AUC je pomerne vysoká, avšak je potrebné tepelnú mapu ešte vyhodnotiť z pohľadu segmentačných masiek. Tepelná mapa bola vygenerovaná s parametrami $b1 = 1$, $b2 = 0$.

Ako možná zmena do ďalších experimentov je používať hodnotu *max* ako hodnotu pre parameter *b2_value* v kombinácii s $b1 = 1$ (tj. metóda s dokreslením) a $b2 < 1$, keďže dosiahla zatiaľ najlepšie výsledky.

6.1.4 Hľadanie optimálnych parametrov

6.1.5 Porovnanie s existujúcimi metódami

6.2 Zhrnutie

Zistili sme, že metóda RISE s dokreslovaním dosahuje lepšie výsledky ako pôvodná metóda ktorá zakrývala minimálnou hodnotou. Pokial ale minimálnu hodnotu nahradíme maximálnou, dosiahneme lepšie výsledky ako s dokreslovaním, ktoré dosahuje podobné výsledky ako zakrývaním mediánom. Avšak, vyhodnocovali sme zatial len pomerne malej vzorku a nebrali sme do úvahy početnosť tried (AD a CN) v tejto vzorke čo je nedostatkom našich experimentov (v ďalších experimentoch by mali byť tieto triedy vyvážené). Zároveň na dátach z tejto vzorky nemali testované modely 100%-nú úspešnosť.

Na generovanie tepelných máp sme použili 1000 masiek (kedže autori metódy RISE v experimentoch použili podobný počet), avšak my máme iný typ dát (3D volumetrické dátá), preto je vhodné vyskúšať rôzne počty a nájsť vhodný počet pre nás problém.

Kedže sú masky pri generovaní tepelných máp náhodné, je možné, že pre jeden MRI snímok metóda vygeneruje rôzne tepelné masky. V ďalších experimentoch by sme mali vyhodnocovať aj konzistenciu tepelných máp - tj. ako veľmi sa líšia medzi rôznymi použitiami metódy na tom istom snímku. Predpokladáme, že väčší počet použitých másk bude viest ku konzistentnejším tepelným mapám. Tákyto meraním môžeme dospieť k optimálnemu počtu masiek, ktorý je potrebný na generovanie tepelnej mapy.

Taktiež je potrebné vyhodnotiť správnosť tepelných máp vzhľadom na segmentačné masky, tak ako sme uviedli v návrhu riešenia (Sekcia 4.2.2.2). Ďalej je potrebné porovnať navrhovanú metódu s inými existujúcimi metódami, napr. LRP alebo analýza senzitivity.

7. Zhodnotenie

Aktívne sme pracovali a pracujeme na splnení oboch zadefinovaných cieľov. Podarilo sa nám implementovať navrhnutú metódu a vykonať prvé experimenty, z ktorých výsledkov sme navrhli úpravy týkajúce sa natrénovaného modelu, ale aj nastavenia parametrov navrhнутej metódy. Ďalej sa osobitne vyjadrimo k jednotlívým cieľom.

Vytvorenie novej, alebo vylepšenie existujúcej metódy pre vysvetľovanie rozhodnutí neurónových sietí (Sekcia 3.1) Navrhli a implementovali sme vylepšenie existujúcich metódy RISE, ktorá sa dá použiť nie len na vysvetľovanie rozhodnutí neurónových sietí. Metóda dokáže pracovať s ktorýmkolvek modelom za predpokladu, že daný model na výstupe uvádzza pravdepodobnosť pre predikovanú triedu/triedy. Z dosiahnutých výsledkov ešte nemôžeme tvrdiť, či sme navrhli lepšiu alebo horšiu metódu. Predpokladáme, že to môže byť závislé od typu dát a teda domény.

Využitie vytvorenej metódy na určenie miery správnosti modelu neurónovej siete detegujúcej Alzheimerovu chorobu (Sekcia 3.2) Metódu sme použili na vytvorenie tepelných máp a vykonali sme prvé experimenty s rôznymi nastaveniami tejto metódy. Pre splnenie tohto cieľa je nutné vykonať ďalšie experimenty (napr. porovnanie s inými metódami), tak ako sme uviedli v návrhu riešenia (Sekcia 4.2).

Literatúra

1. AMISHA, Paras Malik; PATHANIA, Monika; RATHAUR, Vyas Kumar. Overview of artificial intelligence in medicine. *Journal of family medicine and primary care*. 2019, roč. 8, č. 7, s. 2328.
2. GILPIN, Leilani H; BAU, David; YUAN, Ben Z; BAJWA, Ayesha; SPECTER, Michael; KAGAL, Lalana. Explaining explanations: An overview of interpretability of machine learning. In: *2018 IEEE 5th International Conference on data science and advanced analytics (DSAA)*. 2018, s. 80–89.
3. 2019. Dostupné tiež z: <http://www.alzheimer.sk/informacie/alzheimerovachoroba.aspx>.
4. DUTHEY, Béatrice. Background paper 6.11: Alzheimer disease and other dementias. *A Public Health Approach to Innovation*. 2013, s. 1–74.
5. KHAN, Tapan. *Biomarkers in Alzheimer's Disease*. Academic Press, 2016.
6. 2017. Dostupné tiež z: <https://www.alz.org/alzheimers-dementia/facts-figures>.
7. WORKING, G Biomarkers Definitions. Biomarkers and surrogate endpoints: preferred definitions and conceptual framework. *Clin Pharmacol Ther*. 2001, roč. 69, č. 3, s. 89–95.
8. HAYKIN, Simon S et al. *Neural networks and learning machines/Simon Haykin*. New York: Prentice Hall, 2009.
9. LEE, Honglak; GROSSE, Roger; RANGANATH, Rajesh; NG, Andrew. Unsupervised Learning of Hierarchical Representations with Convolutional Deep Belief Networks. Dostupné z DOI: [10.1145/2001269](https://doi.org/10.1145/2001269).

10. O'SHEA, Keiron; NASH, Ryan. An introduction to convolutional neural networks. *arXiv preprint arXiv:1511.08458*. 2015.
11. SIMONYAN, Karen; ZISSERMAN, Andrew. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*. 2014.
12. HE, Kaiming; ZHANG, Xiangyu; REN, Shaoqing; SUN, Jian. Deep residual learning for image recognition. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, s. 770–778.
13. SZEGEDY, Christian; LIU, Wei; JIA, Yangqing; SERMANET, Pierre; REED, Scott; ANGUELOV, Dragomir; ERHAN, Dumitru; VANHOUCKE, Vincent; RABINOVICH, Andrew. Going deeper with convolutions. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, s. 1–9.
14. MONTAVON, Grégoire; SAMEK, Wojciech; MÜLLER, Klaus-Robert. Methods for interpreting and understanding deep neural networks. *Digital Signal Processing*. 2018, roč. 73, s. 1–15.
15. SIMONYAN, Karen; VEDALDI, Andrea; ZISSERMAN, Andrew. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*. 2013.
16. MÜLLER, Klaus-Robert; SAMEK, Wojciech; MONTAVON, Gregoire; LAPUSCHKIN, Sebastian; ARRAS, Leila. *Explaining and Interpreting Deep Neural Networks*. Dostupné tiež z: http://iphome.hhi.de/samek/pdf/DTUSummerSchool2017_1.pdf.
17. SELVARAJU, Ramprasaath R; COGSWELL, Michael; DAS, Abhishek; VEDANTAM, Ramakrishna; PARIKH, Devi; BATRA, Dhruv. Grad-cam: Visual explanations from deep networks via gradient-based localization. In: *Proceedings of the IEEE international conference on computer vision*. 2017, s. 618–626.
18. PETSIUK, Vitali; DAS, Abir; SAENKO, Kate. Rise: Randomized input sampling for explanation of black-box models. *arXiv preprint arXiv:1806.07421*. 2018.

19. RIBEIRO, Marco Tulio; SINGH, Sameer; GUESTRIN, Carlos. "Why should i trust you?Explaining the predictions of any classifier. In: *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*. 2016, s. 1135–1144.
20. EVERINGHAM, Mark; VAN GOOL, Luc; WILLIAMS, Christopher KI; WINN, John; ZISSERMAN, Andrew. The pascal visual object classes (voc) challenge. *International journal of computer vision*. 2010, roč. 88, č. 2, s. 303–338.
21. LIN, Tsung-Yi; MAIRE, Michael; BELONGIE, Serge; HAYS, James; PERONA, Pietro; RAMANAN, Deva; DOLLÁR, Piotr; ZITNICK, C Lawrence. Microsoft coco: Common objects in context. In: *European conference on computer vision*. 2014, s. 740–755.
22. 2017. Dostupné tiež z: <http://adni.loni.usc.edu/>.
23. ESMAEILZADEH, Soheil; BELIVANIS, Dimitrios Ioannis; POHL, Kilian M; ADELI, Ehsan. End-to-end Alzheimer's disease diagnosis and biomarker identification. In: *International Workshop on Machine Learning in Medical Imaging*. 2018, s. 337–345.
24. SMITH, Stephen M. Fast robust automated brain extraction. *Human brain mapping*. 2002, roč. 17, č. 3, s. 143–155.
25. SUK, Heung-Il; LEE, Seong-Whan; SHEN, Dinggang; INITIATIVE, Alzheimer's Disease Neuroimaging et al. Deep sparse multi-task learning for feature selection in Alzheimer's disease diagnosis. *Brain Structure and Function*. 2016, roč. 221, č. 5, s. 2569–2587.
26. HOSSEINI-ASL, Ehsan; KEYNTON, Robert; EL-BAZ, Ayman. Alzheimer's disease diagnostics by adaptation of 3D convolutional network. In: *2016 IEEE International Conference on Image Processing (ICIP)*. 2016, s. 126–130.
27. SUK, Heung-Il; LEE, Seong-Whan; SHEN, Dinggang; INITIATIVE, Alzheimer's Disease Neuroimaging et al. Deep ensemble learning of sparse regression models for brain disease diagnosis. *Medical image analysis*. 2017, roč. 37, s. 101–113.
28. BÖHLE, Moritz; EITEL, Fabian; WEYGANDT, Martin; RITTER, Kerstin. Layer-wise relevance propagation for explaining deep neural network decisions.

- ons in MRI-based Alzheimer's disease classification. *Frontiers in aging neuroscience*. 2019, roč. 11, s. 194.
- 29. CHEN, Wai Kai. *The electrical engineering handbook*. Elsevier, 2004.
 - 30. RAVI, S.; PASUPATHI, P.; MUTHUKUMAR., S.; KRISHNAN, N. Image in-painting techniques - A survey and analysis. In: *2013 9th International Conference on Innovations in Information Technology (IIT)*. 2013, s. 36–41.
 - 31. IOFFE, Sergey; SZEGEDY, Christian. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*. 2015.

A. Plán práce

A.1 Letný semester - DP1

V tomto semestri plánujem pracovať na analýze domény, návrhu metódy a jej implementácií.

A.2 Zimný semester - DP2

V tomto semestri plánujem pracovať na implementácii navrhnutej metódy, ktorú budem overovať v experimentoch a postupne vylepšovať. V tomto semestri plánujem:

- natrénovať model na detekciu Alzheimerovej choroby z MRI snímkov,
- implementovať navrhnutú metódu,
- experimentovať s hyper-parametrami navrhnutej metódy,
- skúmať dosiahnuté výsledky, hľadať príčiny a možné vylepšenia,
- priebežne písat' prácu – implementáciu a dosiahnuté výsledky.

A.2.1 Vyjadrenie k plneniu plánu

V tomto semestri sa nám podarilo splniť všetky stanovené ciele. Natrénovali sme niekoľko modelov detekujúcich Alzheimerovu chorobu z MRI snímkov. Čo sa týka úspešnosti týchto modelov, bohužiaľ sa nám nepodarilo dosiahnuť tak dobré výsledky ako u iných prác. Avšak, naším cieľom nie je natrénovať najlepší model, takže táto úspešnosť vyzerá byť zatial pre nás postačujúca.

Metódu sme implementovali, tak, ako sme ju navrhli, pričom sme pridali vylepšenia ako multiprocessing - paralelné generovanie masiek.

S hyper-parametrami navrhnutej metódy sme experimentovali (ale nie so všetkými, pretože ich je veľa), avšak sme nerobili žiadne prehľadávanie optimálnych parametrov.

Dosiahnuté výsledky sme skúmali a diskutovali ich v závere overenia riešenia pričom sme navrhli ďalšie kroky.

A.3 Letný semester - DP3

V tomto semestri budem pracovať na finalizácii tejto práce, navrhnutú metódu plánujem už iba vylepšovať a pracovať na záverečnom dokumente. V tomto semestri plánujem:

- písat prácu a jej jednotlivé časti - implementácia, technická dokumentácia, dosiahnuté výsledky, záver,
- vykonať úpravy v navrhnutej metóde na základe doterajších výsledkov experimentov,
- vyhodnotiť stabilitu tepelných máp,
- optimalizovať vstupné parametre do RISEI metódy,
- porovnať navrhnutú metódu s existujúcimi metódami,

- vyhodnotiť a porovnať vykonané experimenty,
- odovzdať prácu.

A.3.1 Vyjadrenie k plneniu plánu

Plán práce sa nám v tomto semestri podarilo dodržať. Z experimentov nevzišli žiadne potrebné úpravy metódy, preto žiadne úpravy metódy neboli v tomto semestri vykonané. Taktiež sme vyhodnotili stabilitu tepelných máp. Rovnako sme hľadali aj optimálnu kombináciu vstupných parametrov metódy RISEI tak, že sme si vytvorili možné kombinácie parametrov, ktoré sme vyhodnotili. Neoptimalizovali sme ale všetky vstupné parametre, pri niektorých sme uznali, že to u nich nedáva zmysel. Vytvorenú metódu sme taktiež porovnali s existujúcimi metódami GradCAM, Guided Backprop a Guided GradCAM. Vykonali sme veľké množstvo experimentov (rôzne parametre RISEI, kombinácia RISEI a guided backprop atď.), ktoré sme vyhodnotili (napr. vyhodnotenie na segmentačných maskách) a porovnali.

Dodatok A. Plán práce

B. Technická dokumentácia

Metóda RISEI je implementovaná v jazyku Python, rovnako ako aj jej vyhodnotenie a porovnanie s ostatnými metódami.

B.1 Príprava vývojového prostredia

Na správu python-ovských balíkov a vývojového prostredia je použitá conda, ktorú je nutné nainštalovať. Condu je možné nainštalovať cez distribúciu Anaconda (Anaconda obsahuje grafické rozhranie a množstvo nástrojov/programov) alebo menšiu distribúciu Miniconda. V prípade, že potrebujete šetriť miesto na disku, odporúčam menšiu distribúciu Miniconda.

Po inštalácii condy zadajte nasledovný príkaz v koreňovom adresári repozitára. Tento príkaz vytvorí nové conda prostredie v ktorom nainštaluje potrebné python balíky.

```
1 $ conda env create -f environment.yml
```

Následne pre aktiváciu conda prostredia zadajte nasledovný príkaz.

```
1 $ conda activate dp-timzatko
```

Teraz je možné používať shell, v ktorom bolo aktivované conda prostredie *dp-timzatko*, na spúšťanie Python skriptov a Jupyter notebookov.

Následne spustite Jupyter notebook klienta.

```
1 $ jupyter-notebook
```

Teraz je možné, prezerať, spúštať a upravovať jupyter notebooky v repozitári.

B.2 Závislosti (použité knižnice)

Na implementáciu riešenia sme použili nasledovné Python knižnice (uvádzame len tie najvýznamnejšie). Kompletný zoznam sa nachádza v súbore */REPOZITÁRJ/environment.yml*.

- numpy - na prácu s vektormi, maticami a matematickými operáciami nad nimi.
- pandas - na vytváranie, a ukladanie tabuľkami.
- seaborn - vykreslovanie grafov.
- matplotlib - vykreslovanie rádiologických snímkov, tepelných mám a segmentačných masiek.
- opencv - na dokreslenie v RISEI.
- tensorflow (v2.3.1), tensorboard - implementácia, trénovanie, evaluácia modelu na predikciu alzheimerovej choroby.
- torch, torchvision - evaluácia modelu na predikciu alzheimerovej choroby. Pytorch je potrebný, pretože je závislosťou knižnice *captum*, ktorú používame na vytváranie tepelných masiek existujúcimi metódami (GradCAM a pod.).
- SimpleITK - načítanie volumetrických dát z disku.
- scikit-image - práca s vizuálnymi dátami (augmentácie, zmena veľkosti).

B.3 Technické riešenie

Implementácia riešenia (RISEI, model, evaluácia atď.) sa nachádza v adresári */REPOZITÁR/src*. Funkcionalita z tohto adresára je následne importovaná jupyter notebookmi v adresári */REPOZITÁR/conda_notebooks*. Každý z notebookov má iný účel - trénovanie modelu, vyhodnotenie metódy RISEI, porovnanie metód a pod (detailný opis k týmto notebookom sa nachádza v prílohe C Opis digitálnej časti práce).

B.4 Moduly

Adresár */REPOZITÁR/src* obashuje nasledovné Python moduly, ktoré majú uvedené zodpovednosti.

- **src.risei** - implementácia metódy RISEI (exportuje triedu RISEI) - generovanie masiek.
- **src.model** - obsahuje pomocné funkcie na prácu s tensorflow modelom (načítanie checkpointu atď.).
- **src.model.cnn_3D** - implementácia 3D konvolučnej neurónovej siete v tensorflow-e.
- **src.model.res_net** - implementácia 3D a 2D siete ResNet v tensorflow-e.
- **src.model.compile_model** - komplilácia tensorflow modelu a nastavenie metrík, optimizéru, chybovej funkcie a predvolených nastavení.
- **src.model.create_model** - vytvorenie modelu.
- **src.model.training** - spustenie trénovania modelu, vrátane vytvorenie tensorflow datasetu, nastavenia augmentácií, pripojenia k tensorboardu a pod. na základe vstupných parametrov.

- **src.model.mri_tensorboard_callback** - výpis rádiologických snímkov po epochách/iteráciách do tensorboard-u pri trénovaní.
- **src.model.evaluation** - vyhodnotenie modelu (matica zmätenia, klasifikačné metriky) a priebehu jeho trénovalia.
- **src.model.torch.cnn_3D** - implementácia 3D konvolučnej neurónovej siete v pytorch-y.
- **src.data** - práca s volumetrickými dátami, konverzia tensorflow sequence do numpy a opačne.
- **src.data.description** - popisné štatistiky o dátovej sade.
- **src.data.augmentations** - augmentácie.
- **src.data.mri_sequence** - načítanie MRI snímkov, štítkov a segmentačných masiek z disku. Zmena veľkosti snímkov, orezanie snímkov, šandardizácia dát, rozdelenie dát do dávok.
- **src.data.train_test_split** - rozdelenie dátovej sady na trénovaciu, testovaciu a validačnú.
- **src.data.selector** - výber záznamov z dátovej sadny na základe triedy AD/CN a správnosti klasifikácie modelom.
- **src.data.evaluation.segmentation_masks** - evaluácia tepelných máp podľa segmentačných masiek.
- **src.data.heatmaps** - generovanie tepelných máp.
- **src.data.heatmaps.evaluation** - evaluácia kvality tepelnej mapy podľa metrík *insertion* a *deletion*, perzistencia histórie - tepelných máp a ich evaluácie, načítanie histórie evaluácie, vizualizácie v diagramoch.

Funkcie a triedy v moduloch sú v primeranom rozsahu dokumentované pomocou komentárov. Ďalej bližšie opíšeme najviac dôležitý modul implementujúci navrhnutnú metódu *RISEI*.

B.4.1 Modul: src.risei

Tento modul poskytuje triedu RISEI ktorá slúži na generovanie masiek, z ktorých sa vytvárajú tepelné mapy.

B.4.1.1 Trieda: RISEI

Trieda RISEI slúži na generovanie masiek.

```
class src.risei.RISEI(input_size, s=8, p1=0.5, b1=0.8, b2=0.5,
                      b2_value=0,
                      in_paint='2d',
                      in_paint_radius=20,
                      in_paint_algorithm=cv2.INPAINT_NS,
                      in_paint_blending=True,
                      in_paint_2d_to_3d=False,
                      processes=4,
                      debug=False,
)
```

Parametre

- **s** - veľkosť mriežky, z ktorej sa vytvára binárna maska.
- **p1** - pravdepodobnosť, že pixel v mriežke bude biely.
- **b1** - miera prekryvu medzi pôvodným obrázkom a dokreslením. Ak je θ tak sa dokreslenie vôbec nevykoná.
- **b2** - miera prekryvu medzi pôvodným obrázkom s dokreslením a "čierrou" maskou.
- **b2_value** - hodnoty v "čiernej" maske.

- **in_paint** - typ dokreslenia. Môže byť *2d* alebo *3d*. V prípade *2d* je dokreslenie realizované iba v prvej dimenzií.
- **in_paint_radius** - rádius dokreslenia (posúva sa ďalej do knižnice *opencv*).
- **in_paint_algorithm** - algoritmus dokreslenia, môže byť *cv2.INPAINT_NS* alebo *cv2.INPAINT_TELEA*.
- **in_paint_blending** - ak *True* dokreslenie bude prekryté s pôvodným snímkom podľa interpolovanej čiernej masky (tak nevzniknú žiadne ostré hrany).
- **processes** - počet procesov v ktorých sa budú generovať masky.
- **debug** - ak *True* budú do pamäte ukladané medzivýsledky z generovania masiek (binárna maska, interpolovaná maska atď.).

Metódy Metódy triedy RISEI.

```
generate_masks(n, image, log=True, seed=None)
```

Parametre:

- **n** - počet masiek na vygenerovanie,
- **image** - 3D snímka (*x*, *y*, *z*),
- **log** - ak *True* na štandardnom výstupe bude zobrazený aktuálny stav generovania masiek,
- **seed** - seed pre náhodu.

```
show_from_last_run(i, z, figsize=(12, 8), dim=0):
```

Parametre:

- **i** - i-ta maska na zobrazenie,

- **z**,
- **figsize** - veľkosť vykresleného diagarmu,
- **dim** - reprezentuje rozmer - 0, 1, 2.

Dodatok B. Technická dokumentácia

C. Opis digitálnej časti práce

Evidenčné číslo práce v informačnom systéme FIIT-182905-86077.

Obsah digitálnej časti práce (archív ZIP):

- `/DP_TimotejZatko.pdf` — Práca vo formáte PDF.
- `/DP_prilohy_TimotejZatko.pdf` — Prílohy vo formáte PDF.
- `/repo/thesis/` — Zdrojové súbory práce vo formáte L^AT_EX.
- `/repo/tmp/` — Dátová sada, zoserializované natrénované modely, záznamy z trénovania modelov, zoserializované výsledky experimentov.
- `/repo/src/` — Všetky zdrojové súbory, vrátane metódy RISEI, modelov, generovania tepelných máp. Obsahuje Python balíky.
- `/repo/scripts/` — Pomocný skript na stiahnutie dát z Google Drive.
- `/repo/assets/` — Zdrojové súbory diagramov použitých v práci.
- `/repo/colab_notebooks/` — Obsahuje Jupyter notebooky z trénovania modelov na platforme Google Colab s využitím GPU aj TPU (obsahuje prvé natrénované modely)
- `/repo/conda_notebooks/` — Obsahuje Jupyter notebooky.
- `/repo/conda_notebooks/dataset.ipynb` — Rozdelenie dátovej sady.
- `/repo/conda_notebooks/tensorboard.ipynb` — Tensorboard.

- `/repo/conda_notebooks/augmentations.ipynb` — Vizualizácia implementovaných augmentácií.
- `/repo/conda_notebooks/training/training_history.ipynb` — Vizualizácia priebehu trénovania.
- `/repo/conda_notebooks/training/augmentations/` — Porovnanie vplyvu augmentácií na výsledný model.
- `/repo/conda_notebooks/training/2d_ResNet18/` — Trénovanie modelu 2D ResNet18 (viacero jupyter notebookov).
- `/repo/conda_notebooks/training/3d_ResNet18/` — Trénovanie modelu 2D ResNet18 (viacero jupyter notebookov).
- `/repo/conda_notebooks/training/3d_cnn/` — Trénovanie modelu 3D CNN (viacero jupyter notebookov).
- `/repo/conda_notebooks/risei/model_comparison_v1/` — Porovnanie metódy RISEI a jej parametrov na natrénovaných modeloch (prvá iterácia, staré modely, viacero jupyter notebookov).
- `/repo/conda_notebooks/risei/model_comparison_v2/` — Porovnanie metódy RISEI a jej parametrov na natrénovaných modeloch (druhá iterácia, nové modely, viacero jupyter notebookov).
- `/repo/conda_notebooks/risei/evaluation_history_ins_del.ipynb` — Vyplňanie štatistik pre vybranú evaluáciu metódy.
- `/repo/conda_notebooks/risei/evaluation_history_ins_del_comparison.ipynb` — Porovnanie vybraných dvoch evaluácií metód.
- `/repo/conda_notebooks/risei/evaluation_history_segmentation_masks.ipynb` — Vyhodnotenie metódy voči segmentačným maskám.
- `/repo/conda_notebooks/risei.evaluation_all.ipynb` — Porovnanie všetkých experimentov.

Dodatok C. Opis digitálnej časti práce

- `/repo/conda_notebooks/risei/risei.ipynb` — RISEI - zobrazenie generovaných masiek podľa nastavených parametrov.
- `/repo/conda_notebooks/risei/experiments/methods/` — Vyhodnotenie tepelných máp z iných, existujúcich metód (viacero jupyter notebookov).
- `/repo/conda_notebooks/risei/experiments/parameters/` — Hľadanie optimálneho počtu parametrov (viacero jupyter notebookov).
- `/repo/conda_notebooks/risei/experiments/stability/` — Vyhodnotenie kvality stability tepelných máp podľa počtu vygenerovaných masiek (viacero jupyter notebookov).