ČESKÉ VYSOKÉ
UČENÍ TECHNICKÉ
V PRAZE

FAKULTA
BIOMEDICÍNSKÉHO
INŽENÝRSTVÍ

BAKALÁŘSKÁ PRÁCE

2022

DALIBOR
JELÍNEK

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE

FAKULTA BIOMEDICÍNSKÉHO INŽENÝRSTVÍ

Katedra biomedicínské informatiky

Využití hlubokých neuronových sítí pro prioritizaci RTG snímků plic

Use of deep neural networks for chest x-ray prioritization

Bakalářská práce

Studijní program:	Biomedicínská a klinická technika
Studijní obor:	Biomedicínská informatika

Autor bakalářské práce:	Dalibor Jelínek
Vedoucí bakalářské práce:	Mgr. Radim Krupička, Ph.D.

Kladno 2022

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení:	**Jelínek**	Jméno:	**Dalibor**	Osobní číslo:	**491785**
Fakulta:	**Fakulta biomedicínského inženýrství**				
Garantující katedra:	**Katedra biomedicínské informatiky**				
Studijní program:	**Biomedicínská a klinická technika**				
Studijní obor:	**Biomedicínská informatika**				

II. ÚDAJE K BAKALÁŘSKÉ PRÁCI

Název bakalářské práce:

Využití hlubokých neuronových sítí pro prioritizaci RTG snímků plic

Název bakalářské práce anglicky:

Use of deep neural networks for chest x-ray prioritization

Pokyny pro vypracování:

Cílem práce je vytvořit systém využívají hlubokých neuronových sítí pro prioritizaci RTG snímků plic. V rámci práce se seznamte s možnostmi hlubokých neuronových a konvolučních sítí pro automatickou klasifikaci RTG snímků. Pro potřebu strojového učení a prioritizaci vyberte ve spolupráci s radiology z FNKV vhodnou datovou sadu a definujte prioritu řazení. Navhrněte způsob, jak bude automaticky priorita určována a vyberte vhodnou architekturu a implementaci neuronové sítě pro její určení. Systém upravte, natrénujte a validujte na vybrané datové sadě. Výsledky poronejte s expertním hodnocením lékařů.

Seznam doporučené literatury:

[1] Pytorch, Pytorch tutorials, 1.11.2021, https://pytorch.org/tutorials/
[2] Yashvi Chandola, Jitendra Virmani, H.S Bhadauria, Papendra Kumar, Deep Learning for Chest Radiographs, ed. 1, Elsevier, 2021, ISBN 9780323906869

Jméno a příjmení vedoucí(ho) bakalářské práce:

Mgr. Radim Krupička, Ph.D.

Jméno a příjmení konzultanta(ky) bakalářské práce:

MUDr. David Girsa

Datum zadání bakalářské práce: **14.02.2022**
Platnost zadání bakalářské práce: **18.09.2023**

doc. Ing. Zoltán Szabó Ph.D.
vedoucí katedry

prof. MUDr. Jozef Rosina, Ph.D., MBA
děkan

PROHLÁŠENÍ

Prohlašuji, že jsem bakalářskou práci s názvem „Využití hlubokých neuronových sítí pro prioritizaci RTG snímků plic“ vypracoval samostatně a použil k tomu úplný výčet citací použitých pramenů, které uvádím v seznamu přiloženém k bakalářské práci.

Nemám závažný důvod proti užití tohoto školního díla ve smyslu § 60 Zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů.

V Kladně 11. května 2022 …..……..……...

Dalibor Jelínek

PODĚKOVÁNÍ

Rád bych poděkoval vedoucímu projektu Mgr. Radimu Krupičkovi, Ph.D. za jeho rady a připomínky, které práci výrazně zlepšily. A za odborné konzultace děkuji MUDr. Davidovi Girsovi a MUDr. Kataríně Naďové.

ABSTRAKT

Využití hlubokých neuronových sítí pro prioritizaci RTG snímků plic

Práce zkoumá možnost vytvoření systému pro automatickou prioritizaci rentgenových snímků plic dle závažnosti nálezu. Byla zvolena datová sada rentgenových snímků s popisem – CheXpert a upraven existující software konvoluční neuronové sítě od JF HEALTHCARE, který je schopen se na této datové sadě učit. Neuronová síť pak byla rozšířena na detekci 13 nálezů a naučena na této datové sadě. Při testu na testovací sadě, která byla vyčleněna sady CheXpert, bylo pro různé nálezy dosaženo výsledků Youdenovy J statistiky mezi 0,20 a 0,59. Na jiné testovací sadě byla vypočtena priorita dle závažnosti detekovaných nálezů a ta pak byla porovnána s prioritou určenou konzultujícím lékařem – neuronová síť určila stejnou prioritu jako lékař jen ve 20 % případů a v 80 % případů prioritu nadhodnotila. Lze říci, že neuronovou síť by pro požadovaný účel šlo použít, a práce nastiňuje možnosti, jak by šlo její výsledky zlepšit.

Klíčová slova

Automatická prioritizace rentgenových snímků plic dle závažnosti (triáž), strojové učení, konvoluční neuronová síť, datová sada CheXpert.

ABSTRACT

Use of deep neural networks for chest x-ray prioritization

The paper examines the possibility of creation of a system for automatic prioritization (triage) of chest X-ray images based on severity of medical findings. CheXpert data set and JF HEALTHCARE´s neural network software were chosen as a core of this system. The software was then extended to detect 13 medical findings and trained using the data set. After the final trial of a testing data set the values of Youden´s J-statistics were between 0.20 and 0.59. The priority of detected findings was calculated on another test data set and then compared to the priority determined by consulting the MD beforehand – the neural network assessed the same priority as the MD in 20 % of cases and in 80 % of cases the priority was assessed higher. The paper suggests that this neural network could be used for the requested purpose and proposes some ways how to improve its performance.

Keywords

Machine learning, automatic prioritization of chest x-ray according to seriousness (triage), deep learning, convolutional neural network, CheXpert data set.

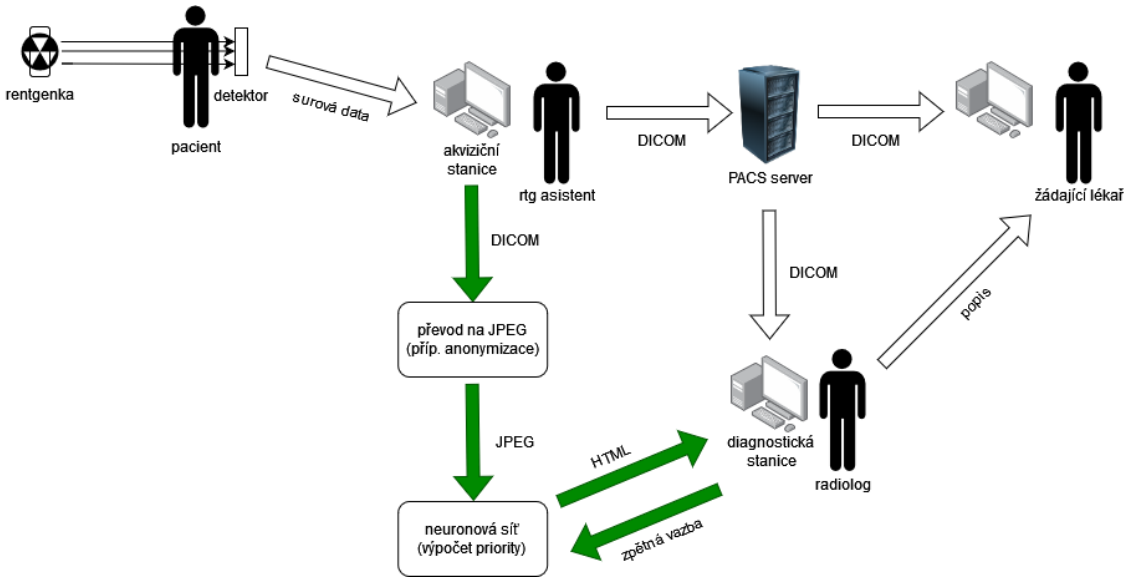
Obsah

1 Úvod

1.1 Motivace

Rentgenový snímek hrudníku (*Chest X-ray*) je neinvazivní vyšetření, které pomocí velmi malé dávky ionizujícího záření vytvoří snímek vnitřku hrudníku (srdce, plic, dýchacích cest, cév a kostí páteře a hrudníku). Používá se k hodnocení stavu plic, srdce, hrudní stěny a může pomoci diagnostikovat příčinu problémů s dýcháním, horečky, bolesti na hrudi nebo zranění. Dále může pomoci s diagnózou a sledováním léčby různých plicních onemocnění jako je zápal plic, rozedma plic nebo rakovina. Je to rychlé a jednoduché vyšetření, takže je hodně důležité i v urgentní medicíně. Zároveň se jedná o nejčastěji prováděné rentgenové vyšetření [1].

Každý pořízený rentgenový snímek je prohlédnut a popsán odborníkem – radiologem. K tomuto popisu musí dojít v relativně krátké době, ale je běžné, že ve frontě na popis čeká více snímků a na některé se dostane až druhý den. Mezi čekajícími snímky jsou samozřejmě snímky zdravých lidí, lidí s chronickými chorobami, ale i snímky urgentní, které je potřeba popsat co nejdříve. Bylo by vhodné mít k dispozici systém, který by snímky čekající ve frontě automaticky zhodnotil, odhadl jejich závažnost a upozornil radiologa na ty, které nesnesou odkladu.



Obrázek 1.1: Workflow rentgenového snímku

Takový systém by se skládal z několika komponent, které by zajišťovaly:

- připojení systému na frontu pořizovaných rentgenových snímků z akvizičních stanic na vyšetřovnách
- převod snímku z formátu DICOM do formátu JPEG nebo PNG

- anonymizaci dat (aby při případném externím vyhodnocování citlivá data neopouštěla nemocnici)
- zaslání snímku na automatické vyhodnocení
- určení priority
- a konečně prezentaci výsledků na monitoru popisujícího lékaře.

V neposlední řadě systém musí obsahovat nějakou správu fronty, aby nedocházelo k neustálému odsouvání neprioritních snímků. Ve FNKV bývá maximální délka fronty čekajících na popis asi 15 snímků, což znamená, že lékař má celkem dobrý přehled o tom, jak dlouho snímky ve frontě jsou a může zajistit, aby se na každý snímek dostalo v rozumném čase. Nicméně by bylo vhodné, aby byl v praxi doplněn nějaký mechanismus řízení fronty snímků na popis, aby nemohlo dojít k situaci, že by neustále přicházely snímky s vysokou prioritou a na některý snímek s prioritou nízkou by se dostalo až po velmi dlouhém čase. Nabízí se třeba umělé povyšování priority snímků po určité době strávené nečinně ve frontě, nebo jejich zvýraznění.

Velmi vítaným dalším rozšířením systému by pak bylo zpětné hodnocení prioritizovaných snímků po jejich popisu, a to jak jejich „skutečnou“ prioritou, tak i nálezy, které na nich byly popsány. Tímto způsobem by šlo vytvořit anonymní datovou sadu, podle které by bylo možno algoritmus v budoucnosti dále učit a zpřesňovat jeho predikce, a to přesně podle zvyklostí a postupů daného pracoviště.

1.2 Cíle práce

Cílem této práce je navrhnout samotné jádro systému (na obrázku 1.1 označenou „neuronová síť (výpočet priority)“), které snímky již převedené do formátu JPEG automaticky vyhodnocuje a stanovuje jejich prioritu, aby popisující lékař mohl nejdříve věnovat svou pozornost snímkům pacientů se závažnějšími stavy.

Ke splnění cílů bude potřeba vytvořit software, který bude schopen stanovit prioritu jednotlivých snímků bez znalosti dalších informací o pacientech. K tomu bude zapotřebí najít použitelnou sadu snímků hrudníku, pokud možno s již stanovenými prioritami, případně priority stanovit alternativním způsobem, a na té software otestovat. Na závěr je nutné provést srovnání výsledků systému s prioritami, které stanoví lékaři, kteří se popisu rentgenových snímků věnují.

1.3 Struktura práce

Kapitola „Přehled současného stavu“ stručně shrnuje způsoby, kterými lze problém prioritizace řešit a které byly skutečně ve světě použity.

V kapitole „Metody“ popisuji datovou sadu, kterou jsem vybral pro učení neuronové sítě, způsob volby software, který bude popisovat snímky hrudníku, a metodu stanovení priority snímku.

Kapitola „Implementace“ dokumentuje instalaci, nastavení a používání zvolené neuronové sítě, tak jak byla původně napsána. Dále je provedena zkouška a porovnání s publikovanými výsledky.

Kapitola „Experimenty“ sestává z jednotlivých experimentů, kterými jsem postupně upravoval neuronovou síť, aby dokázala popsat více nálezů a s přesnějšími výsledky.

„Výsledky“ provedených testů modifikované neuronové sítě jsou uvedeny ve zvláštní kapitole.

Následuje „Diskuse“, ve které jsou dosažené výsledky kriticky zhodnoceny a je navrženo několik postupů, které by měly vést k jejich zlepšení v další práci.

2 Přehled současného stavu

Systém počítačového hodnocení snímků plic je vysoce aktuální téma. Při hledání zdrojů na počátku tohoto projektu (duben 2021) jich bylo relativně málo, ale během roku, který uplynul, na toto a podobná témata vznikla řada nových prací a také již komerčních produktů. Všechny, které v práci diskutuji, používají strojové učení – neuronové sítě. Protože alfou i omegou neuronových sítí jsou data, která lze použít pro jejich učení, je možné takové produkty vyvíjet, jen pokud takové datové sady máme.

Trend započal v roce 2017 uvolněním 112 000 snímků z klinického centra NIH. Pak jen v roce 2019 bylo uvolněno více než 755 000 snímků ve třech popsaných sadách (CheXpert, MIMIC-CXR a PadChest). Publikace těchto dat měla zásadní dopad na počet vydaných prací zabývajících se hlubokým strojovým učením v této oblasti. Mezi lety 2015 a 2021 (březnem) tak vzniklo 296 prací. [2]

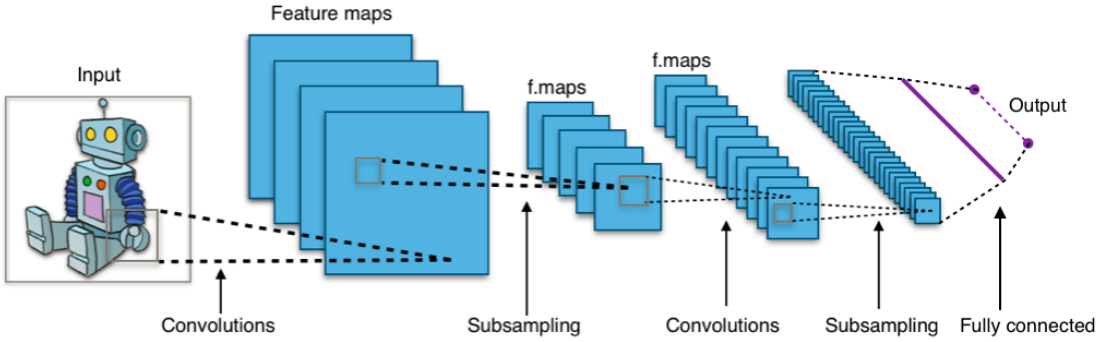
Dnes lze najít již hotové komerční produkty, které na snímku plic hledají abnormality, vizualizují nálezy, automaticky rozdělují normální a abnormální studie a také prioritizují worklisty. Jako příklady lze uvést:

- Chest Solution od Nanox.AI
- Critical Care Suite od GE Healthcare
- Annalise CXR od annalise.ai
- ChestLink od OXIPIT
- Chest X-Ray Classifier od Quibim
- AI-Rad Companion Chest X-ray od Siemens Healthineers
- a mnoho dalších lze nalézt na stránkách [3].

Také lze nalézt práce, které mají stejný cíl jako tato, a které uvádí, že se povedlo naučit hluboké konvoluční neuronové sítě tak, že dosahovaly senzitivity 71 % a specificity 95 % při detekci normálních snímků plic [4]. Nebo práci, která měřila dopad použití umělé inteligence na prioritizaci worklistu radiologa, a zjistila, že průměrná doba čekání na popis pneumotoraxu se zkrátila z 80 na 36 minut [5]. Lze tedy říci, že problém je pomocí dobře naučené neuronové sítě řešitelný.

Konvoluční neuronová síť (*CNN*) je neuronová síť, která je navržena právě pro zpracování rastrových obrázků. Při zpracování rastrového obrázku klasickou plně propojenou sítí bychom brzy narazili na limity strojového času, protože každý jednotlivý pixel by se stal vstupní hodnotou plně propojené sítě, což by vedlo k příliš vysokému počtu parametrů sítě. Navíc by každý drobný posun a otočení objektu v obrázku hodnotila síť jako něco nového, neznámého. CNN tento problém řeší tak, že na svém začátku sestává z posloupnosti konvolučních vrstev a subsamplingových vrstev. Konvoluční vrstvy postupně na celý obrázek aplikují malou (třeba 3 x 3 pixely velkou) masku. Protože maska má třeba jen 9 vstupů (pro černobílý obrázek) a aplikuje se na celý obrázek, jedná se o významné snížení počtu parametrů oproti klasické plně propojené síti. Následná

subsamplingová vrstva sníží rozlišení výsledku a její výstup je vstupem další konvoluční vrstvy. Po několika vrstvách je vytvořen mnohem menší obrázek, který je vstupem do klasické plně propojené sítě, která data finálně klasifikuje.



Obrázek 2.1: Schéma konvoluční neuronové sítě
(zdroj: WikiMedia, autor: Aphex34)

Pro použití neuronové sítě je ovšem zásadní mít kvalitní data, ze kterých by se síť učila. Na úspěšné trénování sítě jsou potřeba desítky tisíc snímků s patřičnými popisy. Vytvoření takové sady dat z jedné nemocnice by trvalo léta, protože větší česká nemocnice dělá zhruba 30 až 40 snímků hrudníku denně, a proto je potřeba najít nějakou již existující datovou sadu rentgenových snímků hrudníku s popisy.

V posledních letech byly na internetu publikovány některé datové sady, které lze využít pro učení neuronové sítě. Jedná se třeba o:

- MIMIC-CXR-JPG – chest radiographs with structured labels [6]
  sada 377 110 snímků hrudníku s popiskami, které byly odvozeny z textových zpráv lékařů
- VinDr-CXR: An open dataset of chest X-rays with radiologist annotations [7]
  sada 18 000 snímků ručně popsaných radiology
- NIH Dataset [8]
  sada více než 100 000 snímků více než 30 000 pacientů
- CheXpert [9]
  sada 224 000 snímků od 65 000 pacientů s popiskami, které bylo strojově přečteny ze zpráv lékařů ze Stanfordské nemocnice
- PadChest [10]
  sada 160 000 snímků od 67 000 pacientů ve vysokém rozlišení a 16bitové hloubce šedé s popiskami z nemocnice San Juan ve Španělsku

Publikované datové sady obsahují popisky nálezů, které na snímcích lékaři našli (pneumotorax, kardiomegálie, otok, ...), neobsahují ale zadáním požadovanou prioritu.

3 Metody

Protože není k dispozici žádná datová sada, která by přímo hodnotila snímky z pohledu jejich závažnosti, je potřeba prioritu odvodit. Problémem jejího stanovení je, že závisí jednak na nálezech, které na snímku objektivně jsou, ale také na jejich míře a její určení je i do jisté míry subjektivní. Vyjdu tedy z toho, že se pokusím naučit neuronovou síť určovat ze snímků co nejvíc nálezů a prioritu snímku budu stanovovat na základě jejich detekce, což ovšem bude zcela pomíjet míru jejich „velikosti“. Priority jednotlivých nálezů, kterém budu umět popsat, budou určeny konzultujícím lékařem.

Na začátku práce vyčlením z použité datové sady dvě množiny snímků. Jedna bude použita pro kontrolu přesnosti naučení neuronové sítě při rozpoznávání nálezů. Druhou pak posoudí konzultující lékař a stanoví priority jednotlivým snímkům dle svých zkušeností a praxe. Tu pak porovnám s prioritou stanovenou podle nálezů určených neuronovou sítí.

Rozhodl jsem se použít sadu CheXpert [9]. Jednak patří k těm rozsáhlejším a pak je s ní spojena soutěž, ve které se různé algoritmy snaží co nejlépe popsat snímky hrudníku v sadě obsažené. Některé z těchto algoritmů jsou publikovány i se zdrojovými kódy a dá se na nich dále stavět.

3.1 Datová sada

Datová sada CheXpert obsahuje cca 224 tisíc snímků od 65 tisíc pacientů pořízených v Stanford Hospital v letech 2002-2017. Jednotlivé snímky byly v nemocnici normálně v průběhu let radiology popsány a tyto jejich popisy pak byly najednou strojově přečteny, zpracovány a byly vytvořeny značky pro 14 běžných nálezů s tím, že každý nález je označen jako buď pozitivní, negativní nebo nejistý (což znamená, že buď radiolog vyjádřil nejistotu ohledně nálezu nebo samotná interpretace popisu snímku je nejistá). Těchto 14 plicních nálezů je vybraných podle publikace „*Fleischner Society: Glossary of Terms for Thoracic imaging (2008)*“ – viz tabulka 3.1.

Sadu snímků si lze po krátké registraci stáhnout ze stránek [9], kde se nachází ve dvou verzích *CheXpert-v1.0 Original* (což jsou snímky ve vysokém rozlišení o celkové velikosti asi 439 GB) a *CheXpert-v1.0 Downsampled* (celkem o velikosti 11 GB, která obsahuje stejné snímky ve formátu JPEG zmenšené na kratší stranu velkou 320 pixelů). Dále jsou přítomny dva soubory ve formátu CSV. Jeden pro učící data (*train.csv* - 223 415 řádků) a druhý pro validační data (*valid.csv* - 235 řádků). Soubory CSV obsahují tyto sloupce:

- Path – cesta k JPEG souboru
- Sex – pohlaví
- Age – věk

- Frontal/Lateral – čelní/boční projekce
- AP/PA – předozadní/zadopřední projekce
- a pak sloupce jednotlivých 14 nálezů, u kterých jsou hodnoty:

  prázdné = nezmíněno

  0 = negativní

  -1 = nejistý

  1 = pozitivní

No Finding	bez nálezu
Enlarged Cardiomediastinum	zvětšené mediastinum
Cardiomegaly	kardiomegálie
Lung Opacity	opacity mléčného skla
Lung Lesion	léze na plicích/nález na plicích
Edema	edém/otok
Consolidation	konsolidace
Pneumonia	pneumonie / zápal plic
Atelectasis	atelektáza
Pneumothorax	pneumotorax
Pleural Effusion	plicní výpotek
Pleural Other	jiný problém pleury
Fracture	zlomenina
Support Devices	pomocná zařízení

Tabulka 3.1: Nálezy značené v datech CheXpert

Značky ve validačním souboru *valid.csv* byly vytvořeny ručně na základě shody názoru tří radiologů, kteří validační snímky popisovali.

Poznámka: V souborech *train.csv* a *valid.csv* je potřeba před použitím opravit úvodní řádek hlavičky na nahradit v něm mezery podtržítky (např. *Pleural_Effusion*), jinak jsou později použitým softwarem neustále vypisována varování:

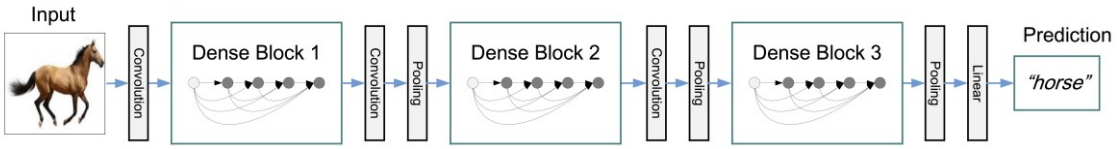
```
INFO:root:Summary name train/loss_Pleural Effusion is illegal;
using train/loss_Pleural_Effusion instead.
```

Existuje ještě třetí sada 500 snímků, která byla rovněž ručně popsána osmi radiology. Tato sada se nazývá *Test* a není volně dostupná. Slouží pro vyhodnocení přesnosti neuronových sítí, které jsou na sadě CheXpert naučeny rozeznávat pět nálezů (atelektáza, kardiomegálie, konsolidace, edém a plicní výpotek). Do soutěže zaslané algoritmy jsou vyhodnoceny na této sadě a pak jsou zařazeny do tabulky pořadí podle dosažené AUC a toho, kolik radiologů překonaly v hodnocení. V tabulce je aktuálně 186 zápisů.

3.2 Neuronová síť

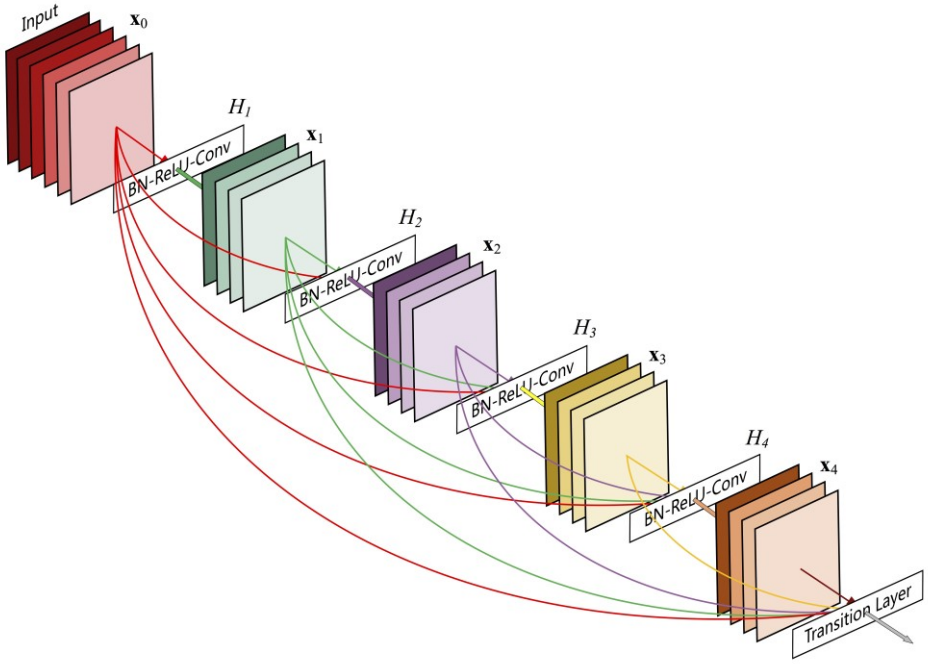
Pro další práci jsem si zvolil software neuronové sítě vytvořený firmou JF HEALTHCARE. Důvodem bylo jednak jeho vysoké umístění v žebříčku soutěže (páté místo) a také protože měl publikované zdrojové kódy. Software je napsaný v jazyce Python a je k dispozici na GitHubu [11], odkud ho lze volně stáhnout a používat dle licence *Apache License, Version 2.0, January 2004*.

Základem software je neuronová síť typu *DenseNet*, která je evolucí klasické konvoluční neuronové sítě (*CNN*).



Obrázek 3.1: Schéma sítě *DenseNet* se třemi bloky [12]

Klasické sítě při zvětšování své hloubky začnou trpět tím, že informace o vstupu nebo gradientu prochází příliš mnoha vrstvami a než dojde do konce sítě, může se tato informace vytratit nebo „vyblednout“. Bylo navrženo několik architektur neuronových sítí, které tento problém řeší. Dělají to většinou tak, že přidávají zkratky vedoucí z dřívějších do následných vrstev. *DenseNet* také používá tento nápad, ale aplikuje ho důsledně na každou vrstvu – vytváří tedy zkrácené propojení mezi všemi vrstvami (o stejné velikosti *feature-map*). [12]



Obrázek 3.2: Schéma propojení vrstev neuronové sítě *DenseNet* [12]

Výhodou tohoto návrhu sítě je, že dovoluje objevené *features* použít libovolnou následující vrstvou bez dalšího zkreslení. Každá vrstva totiž získává přímý vstup ze všech předchozích vrstev a její výstup je stejně tak k dispozici všem vrstvám následujícím. Má-

li tedy síť L vrstev, pak u klasické CNN existuje L propojení, kdežto u *DenseNet* jich je L*(L+1)/2. Při použití této architektury má síť méně parametrů a také se zlepší tok informací a gradientů sítí. To má za důsledek snazší učení a tím pádem možnost použití hlubších sítí, než bylo možné dříve s klasickou CNN. [12]

3.3 Prioritizace

Původně jsem předpokládal, že neuronová síť určí nálezy, které na snímku najde a ty pak seřadím podle jejich závažnosti. Ovšem takové řazení je relativně subjektivní a nelze jednoznačně říct, že jeden druh nálezu je vždy závažnější než jiný. Navíc je v tomto kontextu jemná priorita nepotřebná a spíše věci komplikuje. Vznikaly by pak problémy, jak například správně prioritizovat snímek, na kterém je vysoká pravděpodobnost méně závažného nálezu, ale zároveň nižší pravděpodobnost velmi závažného nálezu. Při konzultaci s lékaři jsme došli k závěru, že v praxi budou stačit tři úrovně priority:

- urgentní (3)
- závažné (2)
- může počkat (1)

Přiřazení priorit bylo stanoveno takto:

Nález anglicky	Nález česky	Priorita hrubá
Lung Opacity	opacity mléčného skla	urgentní
Edema	edém / otok	urgentní
Pneumonia	pneumonie	urgentní
Pneumothorax	pneumotorax	urgentní
Cardiomegaly	kardiomegálie	závažné
Lung Lesion	léze na plicích	závažné
Atelectasis	atelektáza	závažné
Pleural Effusion	plicní výpotek	závažné
Pleural Other	jiný problém pleury	závažné
Support Devices	pomocná zařízení	závažné
Enlarged Cardiomediastinum	mediastinum zvětšené	může počkat
Consolidation	konsolidace	může počkat
Fracture	zlomenina	může počkat
No Finding	bez nálezu	může počkat

Tabulka 3.2: Priorita jednotlivých nálezů

Pokud tedy hodnocení neuronovou sítí přesáhne zvolený práh, bude se předpokládat, že daný nález je na hodnoceném snímku přítomen. Pak se použije nejvyšší dosažená priorita ze všech nálezů a ta bude prezentována lékaři.

4 Implementace

4.1 Instalace

Součástí balíčku software je soubor *requirements.txt*, který popisuje další balíčky, které je třeba doinstalovat pro provozování software a které jsou někdy požadovány v konkrétní verzi:

```
torch
torchvision
numpy==1.16.2
matplotlib==3.0.3
scikit-learn==0.20.3
tensorflow==1.15.4
tensorboardX==1.6
easydict==1.9
opencv-python==4.0.0.21
```

Tyto balíčky lze stáhnout a doinstalovat příkazem:

```
pip install -r requirements.txt
```

Ovšem vzhledem k tomu, že tento software je tři roky starý a vývoj mezitím pokročil, tak na testovaných systémech tento postup k úspěchu nevedl. Instalace skončila podivnou chybou:

```
The C/C++ header for freetype2 (ft2build.h) could not be found.
```

Řešením je vymazat ze souboru *requirements.txt* požadované verze:

```
torch
torchvision
numpy
matplotlib
scikit-learn
tensorflow
tensorboardX
easydict
opencv-python
```

a nechat stáhnout a instalovat poslední verze balíčků. Program pak sice při běhu vypisuje varovná hlášení o zastaralosti některých funkcí, ale jinak se zdá být funkční.

4.2 Trénink

Neuronová síť se učí na snímcích a popiskách popsaných CSV souborem, jehož jméno a cesta k němu jsou uloženy v proměnné `train_csv`. Ve výchozím nastavení se

učí jen pět tříd: *Cardiomegaly, Edema, Consolidation, Atelectasis* a *Pleural Effusion*, což odpovídá soutěžní úloze.

Učení se spustí příkazem:

```
~/projekt/pyTest/bin/python Chexpert/bin/train.py
Chexpert/config/example.json logdir --num_workers 1 --device_ids "0" -
-logtofile True &
```

Což znamená: spusť učení podle konfigurace v souboru *example.json*, zkopíruj celý adresář s programem do podadresáře *logdir/classification* a do samotného podadresáře *logdir* ulož i výsledky a log učení *log.txt*.

Učení se spustí s jedním procesem zavádění dat pro neuronovou síť a na jednom GPU grafické karty. Autoři používali čtyři karty *GeForce GTX 1080 Ti*, takže příkazový řádek měl parametry `--num_workers 8 --device_ids "0,1,2,3"` Moje konfigurace při startu učení doporučovala použít pro *DataLoader* jen jeden *worker process*, takže jsem to tak udělal. Počet CUDA zařízení v počítači lze zjistit tímto skriptem:

```
#!pyTest/bin/python
import torch
print(torch.cuda.is_available())
print(torch.cuda.device_count())
```

Na mém stroji s kartou *GeForce RTX 3080 Ti* skript vypíše:

```
True
2
```

Nicméně pokus o spouštění na dvou GPU (`--device_ids "0,1"`) vedl k varování:

```
UserWarning:
There is an imbalance between your GPUs. You may want to exclude GPU 1 which has less
than 75% of the memory or cores of GPU 0. You can do so by setting the device_ids
argument to DataParallel, or by setting the CUDA_VISIBLE_DEVICES environment variable.
```

Učení jsem tedy spouštěl jen na jednom GPU.

Během učení jsou na obrazovku, či do logovacího soubor, vypisovány následující informace:

```
…
INFO:root:2021-11-25 02:05:35, Train, Epoch : 3, Step : 590, Loss : 0.65808 0.68677
0.89595 0.94767 0.72263, Acc : 0.730 0.730 0.590 0.580 0.720, Run Time : 2.19 sec

INFO:root:2021-11-25 02:05:37, Train, Epoch : 3, Step : 600, Loss : 0.71389 0.64765
0.64188 1.06405 0.62426, Acc : 0.660 0.700 0.610 0.610 0.730, Run Time : 2.19 sec
INFO:root:2021-11-25 02:05:39, Dev, Step : 600, Loss : 0.89576 0.96725 0.54344
0.82828 0.76913, Acc : 0.706 0.806 0.783 0.448 0.285, Auc : 0.696 0.843 0.768 0.766
0.796,Mean auc: 0.774 Run Time : 2.08 sec
INFO:root:2021-11-25 02:05:41, Train, Epoch : 3, Step : 610, Loss : 0.78658 0.64102
0.55593 0.95179 0.55576, Acc : 0.630 0.700 0.620 0.570 0.730, Run Time : 4.29 sec
```

```
INFO:root:2021-11-25 02:05:43, Train, Epoch : 3, Step : 620, Loss : 0.81860 0.63575
0.56316 0.91405 0.70703, Acc : 0.740 0.720 0.620 0.610 0.630, Run Time : 2.34 sec
INFO:root:2021-11-25 02:05:46, Train, Epoch : 3, Step : 630, Loss : 0.82537 0.68206
0.52319 0.92670 0.73163, Acc : 0.700 0.710 0.630 0.690 0.690, Run Time : 2.19 sec
INFO:root:2021-11-25 02:05:48, Train, Epoch : 3, Step : 640, Loss : 0.60103 0.51252
0.61537 0.89644 0.56146, Acc : 0.740 0.710 0.650 0.640 0.680, Run Time : 2.21 sec
INFO:root:2021-11-25 02:05:50, Train, Epoch : 3, Step : 650, Loss : 0.90733 0.72631
0.97592 0.91865 0.69649, Acc : 0.620 0.720 0.630 0.590 0.690, Run Time : 2.20 sec
INFO:root:2021-11-25 02:05:52, Train, Epoch : 3, Step : 660, Loss : 0.86860 0.97632
0.82099 1.12747 0.66276, Acc : 0.680 0.690 0.690 0.560 0.700, Run Time : 2.20 sec
INFO:root:2021-11-25 02:05:55, Train, Epoch : 3, Step : 670, Loss : 0.61839 0.74932
0.70619 1.08156 0.78062, Acc : 0.660 0.680 0.590 0.590 0.650, Run Time : 2.25 sec
INFO:root:2021-11-25 02:05:57, Train, Epoch : 3, Step : 680, Loss : 0.51798 0.79012
0.72206 0.85982 0.61352, Acc : 0.750 0.720 0.660 0.610 0.660, Run Time : 2.27 sec
INFO:root:2021-11-25 02:05:59, Train, Epoch : 3, Step : 690, Loss : 0.79218 0.72450
0.60305 0.89375 0.65692, Acc : 0.700 0.620 0.590 0.540 0.630, Run Time : 2.26 sec
INFO:root:2021-11-25 02:06:02, Dev, Step : 693, Loss : 0.92476 0.95939 0.54502
0.83435 0.74905, Acc : 0.710 0.806 0.813 0.427 0.298, Auc : 0.710 0.843 0.771 0.775
0.799,Mean auc: 0.779 Run Time : 2.14 sec
INFO:root:2021-11-25 02:06:02, Best, Step : 693, Loss : 0.92476 0.95939 0.54502
0.83435 0.74905, Acc : 0.710 0.806 0.813 0.427 0.298,Auc :0.710 0.843 0.771 0.775
0.799,Best Auc : 0.779
```

Nejčastějším záznamem v tomto výpisu je zpracování snímků z učící sady (*Train*), kde je vypsána aktuální epocha učení (*Epoch*) a aktuální krok (*Step*). Následně jsou zobrazeny hodnoty ztrátové funkce a přesnosti odhadů (poměr správných odhadů) pro jednotlivé nálezy. Na konci je doba trvání této dávky. Záznam je vypisován dle nastavení parametru `"log_every": 10`, tedy pro každý desátý snímek.

Záznamy jsou pravidelně prokládány testem pomocí snímků ze sady *Dev,* a to podle parametru `"test_every": 100`, tedy po každých 100 naučených snímcích. Zde se navíc vypisují AUC (*Area Under Curve*) pro jednotlivé nálezy a jejich průměr.

Pokud je hodnota AUC změřená v tomto kroku lepší než nějaká nalezená dříve, jsou aktuální parametry uloženy (ukládají se celkem tři modely dle nastavení parametru `"save_top_k": 3` do souborů *best*N.*ckpt* Jednotlivé modely se zapisují kruhově, takže ten s nejvyšším číslem není nutně ten nejlepší.) a do logu je vypsán záznam *Best*.

Učení se dá spustit i s doplňkovým parametrem `--verbose True`, který vypisuje další informace o neuronové síti (její architekturu) před jejím vlastním učením. K této funkci je ovšem nejdříve potřeba doinstalovat balíček *torchsummary*, který chybí v dodaném souboru *requirements.txt*, pomocí příkazu `pip install torchsummary`.

Jedna epocha učení na celých vstupních datech trvá na mém stroji 1 hodinu 52 minut. Provede se celkem 26 376 kroků. Což neodpovídalo počtu snímků, protože soubor *train.csv* má 223 414 řádků se snímky pro učení. Vysvětlení spočívá v parametrech

`enhance_index` a `enhance_times`, které do učící sady snímků zahrnují vybrané snímky opakovaně, viz kapitola 4.5 Konfigurace.

Po naučení jedné celé epochy lze v konfiguračním souboru zvednout počet epoch, po které se má učit, třeba na `"epoch": 6` a spustit učení znovu s parametrem `--resume 1`. Učení bude pokračovat a zpřesňovat se od místa, kde dříve skončilo.

Poznámku si zaslouží způsob nakládání s hodnotou -1, což znamená *nejisté*. Program zde, obsahuje napevno vložené řádky kódu (viz soubor *data/dataset.py)*, které dělí pět zpracovávaných nálezů do dvou skupin:

řádek 34: `if index == 5 or index == 8:`

`index` zde odkazuje na nálezy edém a atelektáza, u kterých *nejistý* se přepisuje na *pozitivní*.

řádek 40: `elif index == 2 or index == 6 or index == 10:`

zde podmínka vybírá zbývající nálezy kardiomegálie, konsolidace a plicní výpotek a přepisuje *nejistý* na *negativní*.

Nepovedlo se mi najít zdůvodnění, proč je to nastaveno zrovna takto. Dva oslovení lékaři neviděli žádný medicínský důvod, proč by se s prvními dvěma nálezy mělo zacházet jinak než se zbylými třemi. Je pravděpodobné, že toto nastavení bylo prostě určeno náhodně a pak otestováno na validačních datech a zvoleno to, které dávalo lepší výsledek.

Zkoušel jsem trénovat neuronovou síť dalšími a dalšími epochami a zjistil jsem, že při výchozích pěti nálezech a daných datech jsem dosáhl nejlepší hodnoty AUC po čtyřech epochách a pak až do deseti testovaných epoch celkem se již výsledek nezlepšil. Toto samozřejmě nelze brát za dané, protože soubory jsou při učení voleny náhodně `(shuffle=True)`, takže při jiném učení to může dopadnout trochu jinak.

4.3 Klasifikace

Testování neuronové sítě se spouští příkazy

```
cd ~/projekt/logdir
cp best1.ckpt best.ckpt
~/projekt/pyTest/bin/python classification/bin/test.py
    --num_workers 1 --device_ids "0"
```

Program `test.py` provede klasifikaci pro všech 234 testovacích snímků, které jsou vyjmenovány v souboru *dev.csv*. Pro každý snímek vypíše pravděpodobnost, se kterou se na něm vyskytuje jeden z nálezů, který neuronová síť rozpoznává. Ve výchozím nastavení vypisuje 5 hodnot.

Snímky jsou programem ve stažené verzi klasifikovány jen podle modelu *best1.ckpt*, což je ovšem v rozporu s dokumentací, kde se uvádí, že před klasifikací je třeba zvolit, který uloženy model chceme testovat a provést příkaz např. `cp best1.ckpt best.ckpt`

	A	B	C	D	E	F
1	Path	Cardiomegaly	Edema	Consolidation	Atelectasis	Pleural Effusion
2	/storage/ssd2/CheXpert-v1.0-small/valid/patient64541/study1/view1_frontal.jpg	0.464575529	0.434478372	0.470965564	0.519793212	0.479236543
3	/storage/ssd2/CheXpert-v1.0-small/valid/patient64542/study1/view1_frontal.jpg	0.366135031	0.389618188	0.445144534	0.502534568	0.450414807
4	/storage/ssd2/CheXpert-v1.0-small/valid/patient64542/study1/view2_lateral.jpg	0.409645975	0.375840843	0.463714898	0.506009161	0.456597984
5	/storage/ssd2/CheXpert-v1.0-small/valid/patient64543/study1/view1_frontal.jpg	0.440722406	0.444408	0.466417223	0.508457541	0.468244195
6	/storage/ssd2/CheXpert-v1.0-small/valid/patient64544/study1/view1_frontal.jpg	0.382389814	0.403907478	0.459039241	0.502116859	0.454221636
7	/storage/ssd2/CheXpert-v1.0-small/valid/patient64545/study1/view1_frontal.jpg	0.423046529	0.43170312	0.484745204	0.518974602	0.482895315
8	/storage/ssd2/CheXpert-v1.0-small/valid/patient64546/study1/view1_frontal.jpg	0.45238325	0.430958837	0.471476287	0.518969834	0.476953089
9	/storage/ssd2/CheXpert-v1.0-small/valid/patient64547/study1/view1_frontal.jpg	0.415414006	0.412391692	0.460407913	0.50012809	0.450654149
10	/storage/ssd2/CheXpert-v1.0-small/valid/patient64547/study1/view2_frontal.jpg	0.450314701	0.42701444	0.46805203	0.512666881	0.472048104
11	/storage/ssd2/CheXpert-v1.0-small/valid/patient64547/study1/view3_lateral.jpg	0.445048124	0.418020159	0.474557787	0.505829394	0.46590665
12	/storage/ssd2/CheXpert-v1.0-small/valid/patient64548/study1/view1_frontal.jpg	0.455922633	0.447018564	0.485234499	0.515789926	0.477207601
13	/storage/ssd2/CheXpert-v1.0-small/valid/patient64549/study1/view1_frontal.jpg	0.441122383	0.438047677	0.474975079	0.514878392	0.469056696

Obrázek 4.1: Výstupní soubor

Výsledky klasifikaci jsou vypsány na obrazovku a také do výstupního CSV souboru *logdir/test/test.csv* (název výstupního souboru lze změnit pomocí parametru `--out_csv_path [test/test.csv]`). Na obrazovku je navíc vypsán i nejlepší krok a jeho AUC (*Area Under Curve*), např.:

```
Save best is step: 1000 AUC: 0.6576341906158493
```

4.4 ROC

Nejprve je třeba doinstalovat balíček *pandas* příkazem `pip install pandas` protože balíček chybí v *requirements.txt*.

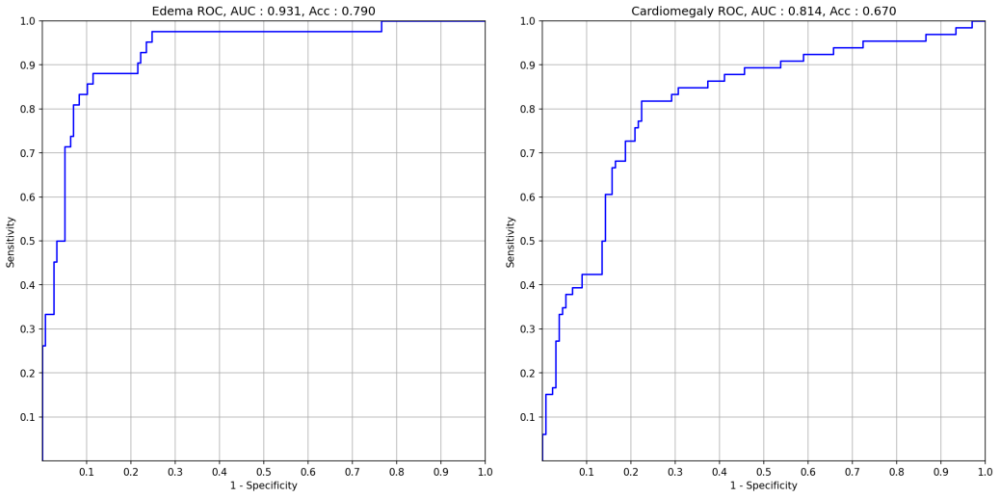
Před vykreslením ROC je potřeba nepřeskočit krok 4.3 Klasifikace, kterým se vytvoří CSV soubory s kontrolou výsledků ohodnocení testovacích souborů.

Pak příkazy

```
cd ~/projekt/logdir
~/projekt/pyTest/bin/python classification/bin/roc.py DJplotname
```

vykreslíme graf ROC pěti hodnocených nálezů do souborů s názvy:

DJplotname_Atelectasis_roc.png
DJplotname_Cardiomegaly_roc.png
DJplotname_Consolidation_roc.png
DJplotname_Edema_roc.png
DJplotname_Pleural_Effusion_roc.png



Obrázek 4.2: Příklady ROC křivek

Zároveň jsou vypsány hodnoty AUC pro jednotlivé nálezy.

Během výpočtu jsou vytvořeny pomocné soubory *pred_csv_done.csv* a *true_csv_done.csv*.

4.5 Konfigurace

Autoři z *JF HEALTHCARE* vytvořili dokumentaci zhruba v rozsahu čtyř stran, která je pro porozumění činnosti programu nedostačující. Navíc je místy nepřesná. Zároveň se nedostává i vysvětlujících komentářů v kódu programu. Na mnoho věcí bylo potřeba přijít pečlivým čtením kódu a metodou pokus-omyl. Zde shrnuji doposud zjištěné poznatky:

Vzorové nastavení programu neuronové sítě je uloženo v souboru *config/example.json*:

```
{
    "train_csv": "/home/user/projekt/Chexpert/config/train.csv",
    "dev_csv": "/home/user/projekt/Chexpert/config/dev.csv",
    "backbone": "densenet121",
    "width": 512,
    "height": 512,
    "long_side": 512,
    "fix_ratio": true,
    "pixel_mean": 128.0,
    "pixel_std": 64.0,
    "use_pixel_std": true,
    "use_equalizeHist": true,
    "use_transforms_type": "Aug",
    "gaussian_blur": 3,
    "border_pad": "pixel_mean",
    "num_classes": [1,1,1,1,1],
```

```
    "batch_weight": true,
    "enhance_index": [2,6],
    "enhance_times": 1,
    "pos_weight": [1,1,1,1,1],
    "train_batch_size": 10,
    "dev_batch_size": 10,
    "pretrained": true,
    "log_every": 10,
    "test_every": 100,
    "epoch": 3,
    "norm_type": "BatchNorm",
    "global_pool": "AVG_MAX",
    "fc_bn": true,
    "attention_map": "FPA",
    "lse_gamma": 0.5,
    "fc_drop": 0,
    "optimizer": "Adam",
    "criterion": "BCE",
    "lr": 0.0001,
    "lr_factor": 0.1,
    "lr_epochs": [2],
    "momentum": 0.9,
    "weight_decay": 0.0,
    "best_target": "auc",
    "save_top_k": 3,
    "save_index": [0,1,2,3,4]
}
```

Bohužel v dokumentaci nejsou jednotlivé parametry vůbec popsány a jejich význam je třeba odvozovat přímo ze zdrojového kódu. Význam jednotlivých parametrů:

- `"train_csv": "/home/user/projekt/Chexpert/config/train.csv`
  cesta k CSV souboru, který popisuje, kde najít jednotlivé snímky a jejich popisky. Tyto soubory jsou používány v rámci učení sítě pro standardní učení podle obsažených popisek nálezů.

- `"dev_csv": "/home/user/projekt/Chexpert/config/dev.csv"`
  cesta k CSV souboru, ve kterém jsou cesty ke snímkům určených pro testování výsledku učení a jejich popisky. Soubory jsou používány i pro testování pokroku během učení. V datové sadě CheXpert je označován jako *valid.csv*, ale jedná se o stejný soubor.

- `"train_batch_size": 10` a `"dev_batch_size": 10`
  velikost batche – počet položek učících/testovacích dat, které budou zpracovány v jedné iteraci algoritmu. Autoři používali na čtyřech GPU hodnotu 56, ale ta je na mém systému příliš velká a model se nevejde do paměti grafické karty:
  `RuntimeError: CUDA out of memory. Tried to allocate 20.00 MiB (GPU 0; 5.94 GiB total capacity; 5.19 GiB already allocated;`

`16.50 MiB free; 5.22 GiB reserved in total by PyTorch)`
Nejvyšší hodnota *batch_size*, která fungovala, byla pouze 2. Po instalaci druhé karty se ji podařilo zvednout na 5. Později byla instalována grafická karta *GeForce RTX 3080 Ti*, která dovolila použít hodnotu 10, ale až po tom, co se nainstalovaly nové balíčky *torch* příkazem:
`pip3 install torch==1.10.0+cu113 torchvision==0.11.1+cu113 torchaudio==0.10.0+cu113 -f https://download.pytorch.org/whl/cu113/torch_stable.html`

- `"backbone": "densenet121"`
  model neuronové sítě je hustě propojená konvoluční síť (více viz https://arxiv.org/abs/1608.06993)

- `"pretrained": true`
  pokud je nastaveno na *true*, použije se předtrénovaný model z ImageNet.

- `"optimizer": "Adam"`
  algoritmus stochastické optimalizace. Více viz https://pytorch.org/docs/stable/generated/torch.optim.Adam.html

- `"criterion": "BCE"`
  ztrátová funkce měřící *Binary Cross Entropy*. Jediná volba – žádná jiná není podporována.

- `"num_classes": [1,1,1,1,1]`
  počet výstupních tříd (jako pole jedniček), musí velikostí odpovídat i parametrům `pos_weight` a také `save_index`

- `"log_every": 10`
  po kolika souborech z *train.csv* se má vypsat logovací záznam na obrazovku/do logovacího souboru.

- `"test_every": 100`
  po kolika souborech z *train.csv* se má provést test kvality naučení sítě pomocí testovacích souborů z *dev.csv*

- `"epoch": 6`
  do jaké epochy učení budeme počítat. Číslo epochy je na konci učení uloženo a pokud ho v konfiguračním souboru zvýšíme a spustíme učení znovu s parametrem `--resume 1`, bude učení inkrementálně pokračovat dalšími epochami.

- `"best_target": "auc"/"acc"/"loss"`
  podle jakého paramteru učení určujeme, zda jsme dosáhli pokroku. Autoři používají *AUC - Area Under Curve*.

- `"save_top_k": 3`
  pokud je výsledek procesu učení na *Dev* datech lepší (podle kritéria `best_target`) než ten doposud dosažený, uloží se natrénovaný model do souboru *bestN.ckpt* Jednotlivé modely se zapisují kruhově, takže ten s nejvyšším číslem není nutně ten nejlepší. Tento parametr určuje, kolik modelů se bude celkem ukládat.

- `"enhance_index": [2,6] a "enhance_times": 1`
  datové sloupečky nálezů 2 a 6 (počítáno po oříznutí prvních pěti sloupečků CSV souboru) – tedy v tomto případě kardiomegálie a konsolidace – budou v případě, že jsou *pozitivní* (tedy 1) zahrnuty do souboru učících dat ještě `enhance_times` krát (zde tedy budou *pozitivní* nálezy zahrnuty pro učení celkem dvakrát).

4.6 Pilotní test

Nejdříve jsem zkoušel napodobit výsledky, ke kterým došli autoři. Postupně jsem s parametrem `--resume 1` učil další a další epochy. Jedna epocha učení na mém stroji trvala cca 2 hodiny. Výsledky jsou vidět v tabulce:

Epocha	1	2	3	4	5	6	7	8	9	10	Autoři
Save best is step	23 600	28 800	52 800	79 700	79 700	79 700	79 700	79 700	79 700	79 700	
AUC	0.8858	0.8912	0.8937	0.8948	0.8948	0.8948	0.8948	0.8948	0.8948	0.8948	
Cardiomegaly AUC	0.8344	0.8752	0.8573	0.8143	0.8143	0.8143	0.8143	0.8143	0.8143	0.8143	0.8703
Edema AUC	0.9176	0.9173	0.9228	0.9305	0.9305	0.9305	0.9305	0.9305	0.9305	0.9305	0.9436
Consolidation AUC	0.9102	0.8876	0.9208	0.9258	0.9258	0.9258	0.9258	0.9258	0.9258	0.9258	0.9334
Atelectasis AUC	0.8797	0.8757	0.8805	0.8849	0.8849	0.8849	0.8849	0.8849	0.8849	0.8849	0.9029
Pleural_Effusion AUC	0.8940	0.9149	0.9150	0.9366	0.9366	0.9366	0.9366	0.9366	0.9366	0.9366	0.9166

Tabulka 4.3: Výsledky učení pěti nálezů

Jak je vidět, výsledky učení se zlepšovaly během prvních čtyř epoch a pak už byly jen horší. Výsledkům publikovaným autory software jsem se celkem přiblížil, ale zcela jich nedosáhl. Rozdíl u kardiomegálie AUC 0,814 vs. 0,870 je relativně velký. Příčinou může být náhoda obsažená v procesu učení. Možná kdybych učení provedl vícekrát, dostal bych se k lepším výsledkům. Druhý faktor je, že používám zmenšenou verzi snímků (*CheXpert-v1.0-small*), které mají kratší stranu velkou 320 pixelů. Není úplně jasné, jak velké snímky použili autoři, někde říkají, že si myslí, že 1024 pixelů a jinde jsem našel 512 pixelů.

5 Experimenty

V této části jsou popsány jednotlivé experimenty, které byly se softwarem neuronové provedeny. Nejprve byla síť rozšířena na rozpoznávání sedmi nálezů a pak na požadovaných třináct nálezů. Dále byl otestován dopad použití snímků o vyšším rozlišení. Následně je popsán proces vytvoření nových větších učících sad. Potom bylo vyzkoušeno, zda není vhodnější na každý nález vytrénovat zvláštní neuronovou síť. Závěrem byly provedeny zkoušky na nových testovacích sadách a popsán výpočet rozhodovacích prahů a priorit snímků.

5.1 Rozšíření počtu nálezů

Software, tak jak je nabízen ke stažení, pracuje jen s těmi pěti nálezy, které se používají při soutěži. Takže je potřeba software modifikovat tak, aby pracoval s více nálezy.

Nejdříve jsem zkoušel změnit počet nálezů na sedm – přidáním pneumonie a pneumotoraxu, což byly dva nejvíce prioritní nálezy, které původní kód nesledoval. K tomu jsou potřeba následující úpravy kódu programu:

5.1.1 config/example.json

Rozšíření proměnných, které se týkají počtu tříd na sedm:

```
"num_classes": [1,1,1,1,1,1,1],
"pos_weight": [1,1,1,1,1,1,1],
"save_index": [0,1,2,3,4,5,6],
```

Protože snímků s těmito diagnózami je relativně málo (viz tabulka 5.1), je třeba jejich pozitivní výskyt zopakovat při učení vícekrát.

```
"enhance_index": [2,6,7,9],
"enhance_times": 1
```

Sloupec v CSV	Sloupec v CSV po oříznutí	Nález anglicky	Nález česky	Jemná priorita	Počet 0 negativní	Počet 1 pozitivní	Počet -1 nejisté
5	0	No Finding	bez nálezu	14	0	22 381	0
6	1	Enlarged Cardiomediastinum	mediastinum zvětšené	8	21 638	10 798	12 403
7	2	Cardiomegaly	kardiomegálie	9	11 116	27 000	8 087
8	3	Lung Opacity	opacity mléčného skla	4	6 599	105 581	5 598
9	4	Lung Lesion	léze na plicích	6	1 270	9 186	1 488
10	5	Edema	edém / otok	2	20 726	52 246	12 984
11	6	Consolidation	konsolidace	5	28 097	14 783	27 742
12	7	Pneumonia	pneumonie	3	2 799	6 039	18 770
13	8	Atelectasis	atelektáza	7	1 328	33 376	33 739
14	9	Pneumothorax	pneumotorax	1	56 341	19 448	3 145
15	10	Pleural Effusion	plicní výpotek	10	35 396	86 187	11 628
16	11	Pleural Other	jiný problém pleury	11	316	3 523	2 653
17	12	Fracture	zlomenina	13	2 512	9 040	642
18	13	Support Devices	pomocná zařízení	12	6 137	116 001	1 079

Tabulka 5.1: Nálezy, čísla jejich sloupců a počty případů v *Train* datech

5.1.2 data/dataset.py

Zde je potřeba doplnit označené řádky. Pozor na změněné číslování datových sloupců v CSV souboru – viz přechozí tabulka 5.1.

```
self._label_header = [
      header[7],
      header[10],
      header[11],
      header[13],
      header[15],
      header[12],
      header[14]
      ]
```

```
elif index == 2 or index == 6 or index == 7 or index == 9 or index ==
10:
```

Tímto řádkem zároveň určujeme, že *nejisté* u těchto dvou přidávaných nálezů budeme brát jako *negativní*. Pokud bychom to chtěli naopak, doplnili bychom tyto nálezy na dřívější řádek

```
if index == 5 or index == 8:
```

V praxi by asi bylo vhodnější řešit toto nastavení nějakou proměnnou než takovouto editací kódu.

5.1.3 bin/train.py

Zde není při rozšíření potřeba nic měnit.

5.1.4 bin/test.py

Zde se doplní názvy nově přidaných nálezů.

```
test_header = [
    'Path',
    'Cardiomegaly',
    'Edema',
    'Consolidation',
    'Atelectasis',
    'Pleural Effusion',
    'Pneumonia',
    'Pneumothorax']
```

A opraví se načítání testované sady parametrů z `best1.ckpt` na `best.ckpt`

```
ckpt_path = os.path.join(args.model_path, 'best1.ckpt')
```

5.1.5 bin/roc.py

Podobně jako v souboru *data/dataset.py* rozšíříme řádek:

```
elif index == 2 or index == 6 or index == 7 or index == 9 or index ==
10 :
```

nebo při opačném požadavku na práci s *nejisté* podobně rozšíříme řádek:

```
if index == 5 or index == 8:
```

Dále je potřeba doplnit řádky:

```
outfile['Pneumonia'] =
    groups['Pneumonia'].mean().reset_index()['Pneumonia']
outfile['Pneumothorax'] =
    groups['Pneumothorax'].mean().reset_index()['Pneumothorax']
```

a opravit řádek

```
num_labels = len(header_true) - 5
```

na

```
num_labels = 7
```

a konečně doplnit

```
header = [ header_true[7], header_true[10], header_true[11],
  header_true[13], header_true[15], header_true[12], header_true[14] ]
```

Pak by už model měl jít natrénovat i otestovat. Po třech epochách učení jsem se dostal k těmto výsledkům AUC (adresář *logdir.7kat3*):

Nález	AUC
Cardiomegaly	0.802
Edema	0.919
Consolidation	0.904
Atelectasis	0.832
Pleural_Effusion	0.833
Pneumonia	0.841
Pneumothorax	0.894

Tabulka 5.2: Výsledky učení – 7 nálezů

Autoři software ovšem uvádí, že se dostali k těmto hodnotám AUC, které jsou výrazně lepší:

Nález	AUC
Cardiomegaly	0.870
Edema	0.944
Consolidation	0.933
Atelectasis	0.903
Pleural_Effusion	0.917

Tabulka 5.3: Výsledky učení autorů – 5 nálezů

Výsledky by byly patrně o trochu lepší, kdybych prošel učením šest epoch. Ovšem tak dobrých výsledků, jaké prezentují autoři, nejsem schopen dosáhnout ani při vyšším počtu opakování. S programem je dodávána i sada před-trénovaných koeficientů neuronové sítě, ale z nějakého důvodu je program umožňuje použít pouze pro další učení a nikoliv pro samotné spuštění neuronové sítě. Jejich využití pro mé účely je ovšem stejně malé, protože jsou natrénovány jen na pět soutěžních nálezů.

5.2 Další rozšíření

Dalším krokem je rozšíření na všech 13 nálezů stejnou metodou, což se ovšem na první pokus nedaří. Učení končí bezvýsledně s varováním:

```
WARNING:root:NaN or Inf found in input tensor.
```

Hodnota *Loss* ve výpise pro nález *Fracture* je nula a odpovídající AUC je *NaN*. Zkouším tedy vynechat *Fracture* a spustit učení znovu. Učení dvakrát padne na stejnou chybu výše, ale pak napotřetí už projde. Jedna epocha s 12 nálezy trvá 3 hodiny a 25 minut. Po šesti epochách a 21 hodinách učení se dostávám k tomuto výsledku (adresář *logdir.12kat6*):

Nález	AUC	Acc
Enlarged_Cardiomediastinum	0.725	0.640
Cardiomegaly	0.748	0.725
Lung_Opacity	0.853	0.420
Lung_Lesion	0.256	0.180
Edema	0.882	0.800
Consolidation	0.879	0.840
Pneumonia	0.839	0.955
Atelectasis	0.812	0.625
Pneumothorax	0.655	0.530
Pleural_Effusion	0.836	0.735
Pleural_Other	0.945	0.975
Support_Devices	0.742	0.635

Tabulka 5.4: Výsledky učení – 12 nálezů

Znovu se pokouším rozšířit učení na všech 13 nálezů. Učení neustále padá na chybu `ValueError: Input contains NaN, infinity or a value too large for dtype('float32').`

Opakovaně ho zkouším spouštět a při devátém pokusu se podaří jednu epochu dokončit. Epocha trvá 3 hodiny 35 minut. Program ovšem při učení vypisuje varování: `/home/user/projekt/pyTest/lib/python3.8/site-packages/sklearn/metrics/_ranking.py:949: UndefinedMetricWarning: No positive samples in y_true, true positive value should be meaningless` Zjišťuji, že ve validačním souboru *Dev.csv* není obsažen žádný případ *Fracture,* a ani některé jiné kategorie nejsou zrovna hojně zastoupené (ideálně by měly být zastoupeny asi ve stejném poměru jako v učících datech):

No Finding	38
Enlarged Cardiomediastinum	109
Cardiomegaly	68
Lung Opacity	126
Lung Lesion	1
Edema	45
Consolidation	33
Pneumonia	8
Atelectasis	80
Pneumothorax	8
Pleural Effusion	67
Pleural Other	1
Fracture	0
Support Devices	107

Tabulka 5.5: Četnost nálezů v *Dev.csv* datech

Myslím, že tohle možná bude dělat problémy při učení neronové sítě, konkrétně při výpočtu AUC. Asi by to bylo třeba nějak zohledit nebo upravit. V tuto chvíli přesouvám dva případy *Fracture* ze souboru *Train.csv* do souboru *Dev.csv*, takže vyhodnocovací funkce má alespoň nějaký případ a konečně začne ukládat nejlepší modely do souborů *BestN.ckpt*.

Zkouším tedy znovu spustit učení pro všech třináct nálezů. Bohužel zde se nedaří ani po mnohém opakování, učení stále padá na chybu `ValueError: Input contains NaN, infinity or a value too large for dtype('float32')`. Na 26. pokus nakonec se podaří epochu dokončit. Problém nastane při spuštění klasifikace, která hlásí chybný počet sloupců na jednom řádku v *Dev.csv*. Při přepisu oněch dvou případů jsem udělal chybu a výsledky jsou nepoužitelné, protože jeden *Dev* soubor byl označen chybně.

Chybu opravuji a znovu pouštím učení. Tentokrát prochází na první pokus s průměrným AUC 0,686 (adresář *logdir.13kat1 – best2.ckpt*):

Nález	AUC	Acc
Enlarged_Cardiomediastinum	0.753	0.522
Cardiomegaly	0.705	0.662
Lung_Opacity	0.762	0.637
Lung_Lesion	0.495	0.965
Edema	0.770	0.791
Consolidation	0.769	0.159
Pneumonia	0.422	0.960
Atelectasis	0.712	0.627
Pneumothorax	0.544	0.164
Pleural_Effusion	0.748	0.383
Pleural_Other	0.765	0.995
Fracture	0.965	0.005
Support_Devices	0.658	0.493

Tabulka 5.6: Výsledky učení – 13 nálezů, 1. epocha

Později ještě zkouším dalších pět epoch učení. Ve druhé epoše dojde k menšímu zlepšení průměrného AUC na 0,690:

Nález	AUC	Acc
Enlarged_Cardiomediastinum	0.804	0.642
Cardiomegaly	0.772	0.527
Lung_Opacity	0.788	0.577
Lung_Lesion	0.155	0.020
Edema	0.797	0.791
Consolidation	0.859	0.841
Pneumonia	0.679	0.960
Atelectasis	0.745	0.373
Pneumothorax	0.485	0.030
Pleural_Effusion	0.758	0.318
Pleural_Other	0.635	0.995
Fracture	0.910	0.005
Support_Devices	0.634	0.507

Tabulka 5.7: Výsledky učení – 13. nálezů, 2. epocha

Až do šesté epochy se pak již průměrné AUC nezlepší (adresář *logdir.13kat6 – best1.ckpt*).

Z učení neuronové sítě jsem zcela vynechal nález *No Finding* (tedy *Bez nálezu*), protože se chci vyhnout celkem dobře představitelné situaci, kdy by neuronová síť dospěla současně k závěru, že snímek obsahuje na 66 % pneumotorax a na 80 % neobsahuje žádný nález. Tento rozpor nevím, jak bych vyřešil. Asi bych z hlediska péče o pacienta stejně musel upřednostnit závěr, že pacient může mít pneumotorax.

5.3 Použití větších snímků

Protože výsledné AUC z posledního učení není příliš dobré, zvažuji přechod na vyšší velikost rentgenových snímků. Doposud jsem používal snímky velikosti delší strany 320 pixelů, což může být pro spolehlivou detekci některých nálezů málo. Ze stránek soutěže CheXpert jsem stáhnul úplnou datovou sadu *CheXpert-v1.0 Original.* Sada má velikost 439 GB, což znamená, že se stahuje asi 25 hodin a je potřeba před jejím stažením myslet na to, že bude potřeba v cílovém umístění ještě zhruba stejně tolik volného místa na rozbalení ZIP souboru.

Ovšem důležitou otázkou je, s jak velkými snímky pracovat v neuronové síti. Autoři na webovém fóru tvrdí, že pokud si pamatují, zmenšili originální snímky na 1 024 x 1 024 pixelů. Ovšem v konfiguračním souboru ve zdrojovém kódu jsou hodnoty:

```
"width": 512,
"height": 512,
"long_side": 512,
```

Rozhodl jsem se tedy pracovat s plnými snímky, které se při výpočtu zmenšují na snímky o straně 512 pixelů, i když je samozřejmě možné v budoucnu vyzkoušet i 1 024 pixelů.

První epochu jsem pouštěl čtyřikrát neúspěšně, kdy učení padalo se známou chybou `ValueError: Input contains NaN, infinity or a value too large for dtype('float32')`, ale na šestý pokus se zadařilo. Běh jedné epochy se prodloužil ze 3 hodin 35 minut na 8 hodin 32 minut. Další epochy běžely již bez havárie a během pěti dní se mi povedlo provést šest epoch učení, abych měl výsledky srovnatelné se stejným nastavením ale menšími snímky. Výsledky učení se tentokrát zlepšovaly až do čtvrté epochy (s menšími snímky se zlepšovaly jen do druhé) a došel jsem nakonec k průměrnému AUC 0,796 (adresář *logdir.13katL4*):

Nález	AUC	Acc
Enlarged_Cardiomediastinum	0.682	0.478
Cardiomegaly	0.800	0.682
Lung_Opacity	0.863	0.448
Lung_Lesion	0.755	0.537
Edema	0.769	0.289
Consolidation	0.905	0.841
Pneumonia	0.883	0.955
Atelectasis	0.816	0.373
Pneumothorax	0.843	0.935
Pleural_Effusion	0.842	0.682
Pleural_Other	0.885	0.995
Fracture	0.925	0.498
Support_Devices	0.609	0.502

Tabulka 5.8: Výsledky učení – větší snímky

Je vidět, že použití větších snímků se odrazilo na zlepšení průměrného AUC z 0,690 na 0,796, což je velký skok. Zejména se zlepšila detekce *Lung Lesion* z 0,155 na 0,755 a *Pneumothorax* z 0,485 na 0,843. Nicméně se za toto zlepšení platí značným prodloužením doby učení, což ale vadí samozřejmě jen v době trénování neuronové sítě.

5.4 Vytvoření nových učících sad

Validační sada (nazývaná v programu *dev.csv* nebo *valid.csv*) se používá tak, že po každých `test_every` krocích (výchozí nastavení je 100) se z ní spočítá AUC, Acc a Loss a podle toho, který z těchto parametrů je zvolen jako optimalizační kritérium (parametrem `best_target`) se uloží daný stav neuronové sítě do souboru *bestN.ckpt*

Problém je, že tvůrci neuronové sítě používali síť jen pro pět soutěžních nálezů, pro které jim validační sada *dev.csv* obsažená v datové sadě CheXpert fungovala dobře. Jenže při rozšíření na 13 nálezů je dle mého názoru původní validační sada nevhodná, protože neobsahuje všechny nálezy v dostatečném počtu (viz tabulka 5.1).

Přistupuji tedy k vytvoření nové sady validačních dat *valid.csv* a testovacích dat *test.csv*. Jednotlivé snímky vybírám postupně náhodně ze sady *train.csv*. Cílem je, aby obě výsledné sady obsahovaly alespoň 200 výskytů od každého nálezu. K tomuhle účelu jsem napsal skript, který výběr provádí (*CheXpert/bin/DJdata.py*). Jelikož většina snímků je popsána více nálezy, jsou některé nálezy v celkovém součtu mnohem početnější:

Nález	Valid	Test
Enlarged_Cardiomediastinum	210	210
Cardiomegaly	269	274
Lung_Opacity	876	884
Lung_Lesion	212	213
Edema	397	375
Consolidation	237	237
Pneumonia	204	201
Atelectasis	301	305
Pneumothorax	241	251
Pleural_Effusion	656	688
Pleural_Other	200	200
Fracture	211	210
Support_Devices	835	885
Počet snímků	1 794	1760

Tabulka 5.9: Počty nálezů v nových Valid.csv/Test.csv

Po přesunu vybraných snímků do sad *valid.csv* a *test.csv* zbývá v sadě *train.csv* ještě 219 858 snímků.

Zde je třeba si uvědomit, že s vytvořením nových validačních a testovacích sad přicházím při učení o výhodu, že původní soubor *dev.csv* byl ručně zkontrolován třemi radiology. Nově vytvořené soubory jsou založeny už jen na strojovém čtení lékařských zpráv, které pak nikdo neověřoval.

S novými sadami spouštím učení, které havaruje na chybě `int() argument must be a string, a bytes-like object or a number, not 'NoneType'`. Tato chyba je způsobena tím, že v původním *dev.csv* jsou na místech nálezů jen nuly a jedničky, kdežto v *train.csv* se používají i prázdné hodnoty a -1. Nahrazuji tedy prázdné hodnoty nulou, protože, když je nález nezmíněn, tak tam pravděpodobně není. Pak se rozhoduji, že i hodnoty -1 nahradím nulou, protože pokud nevím, v jakém kontextu byl nález zmíněn, tak budu předpokládat, že nebyl pozitivní. Jedná se o 962 výskytů hodnoty -1 v souboru *valid.csv* z celkových 23 322 hodnot. (Při zpětném pohledu by možná bylo lepší se pokusit

nálezům s nejistým hodnocením vyhnout úplně a do testovací a validační sady je nezahrnovat vůbec).

Po úpravě CSV souborů spouštím učení znovu, a to opakovaně padá při výpočtu ROC křivky na známou chybu `Input contains NaN, infinity or a value too large for dtype('float32').` Protože doposud se mi podařilo se této chybě vyhnout několikerým opakováním první epochy učení, zkouším učení spustit celkem dvacetkrát, ale bezvýsledně. Mám podezření, že má testovací data jsou stále moc „řídká“, tedy že pozitivních případů je někdy v učení málo. Upravuji tedy parametr `"enhance_times"` na 5 a tím nálezy `"enhance_index": [1,2,4,6,7,9,11,13]` pouštím do učení šestkrát častěji. Výsledkem je, že jedna epocha učení prochází už značný počet – 973 248 snímků, ale na šestý pokus probíhá již úspěšně. Jen běží dlouhých 41 hodin, což je způsobeno zejména tím, že validační soubor obsahuje 1 794 snímků, a tak každý krok validace trvá 83 sekund. Další věcí, která přispívá k velkému prodloužení učení, je, že se každý snímek při učení zmenšuje na požadovanou velikost. Pokud bych se již v této fázi dokázal rozhodnout, jaká velikost snímku je optimální, pak by se samozřejmě vyplatilo celou sadu zmenšit jednorázově. Tato fáze učení neuronové sítě mi nakonec trvala 14 dní téměř nepřetržitých výpočtů.

Výsledně dostávám AUC 0,738 a spouštím další dvě epochy učení, které dle očekávání již prochází bez pádu. Druhá epocha zlepší AUC na 0,749 a třetí již zlepšení nepřinese (adresář *logdir.41.2*):

Nález	AUC	Acc
Enlarged_Cardiomediastinum	0.680	0.883
Cardiomegaly	0.850	0.849
Lung_Opacity	0.649	0.514
Lung_Lesion	0.760	0.882
Edema	0.849	0.222
Consolidation	0.711	0.863
Pneumonia	0.741	0.886
Atelectasis	0.629	0.168
Pneumothorax	0.810	0.866
Pleural_Effusion	0.768	0.367
Pleural_Other	0.758	0.890
Fracture	0.698	0.883
Support_Devices	0.832	0.467

Tabulka 5.10: Výsledky učení – nové učící sady

Výsledky u některých nálezů nejsou špatné, ovšem výsledků deklarovaných autory původního software zdaleka nedosahují – hodnoty AUC vyšší než 0,9 se mi zdaleka nepodařilo dosáhnout.

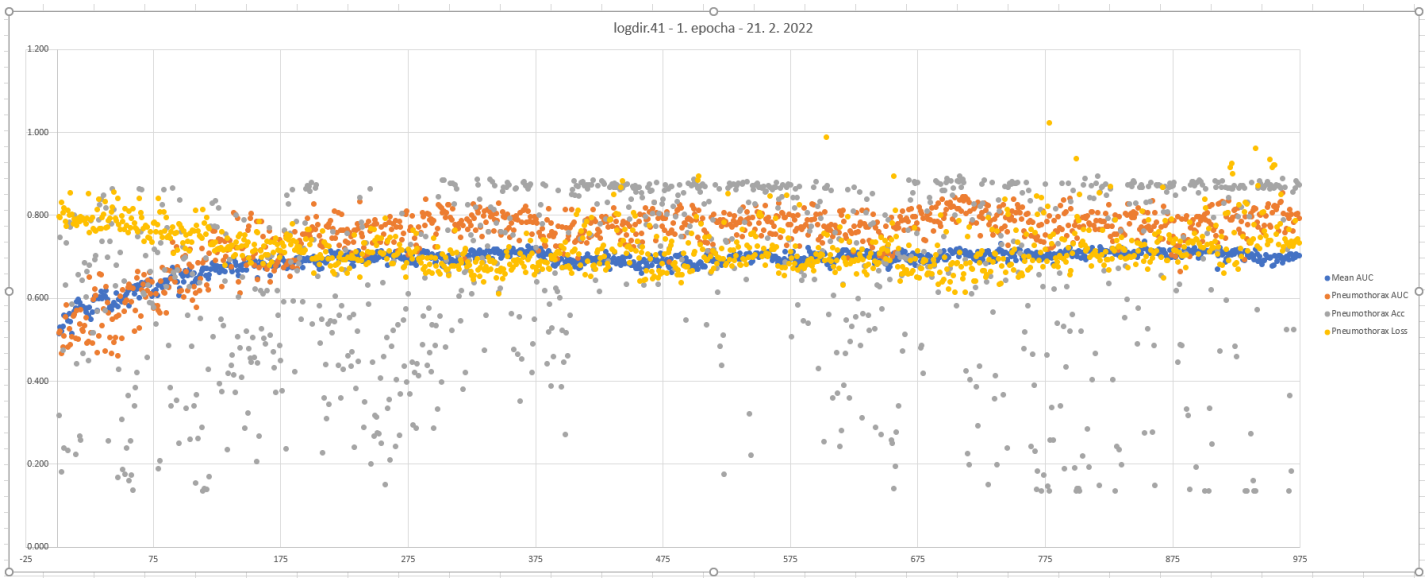
5.5 Snížení počtu kategorií

Během učení neuronové sítě jsem si postupně ukládal jednotlivé výsledky, které síť označila jako nejlepší. Z každého běhu učení (jedné epochy či více epoch po sobě jdoucích) ukládá software tři nejlepší sady parametrů do souborů *bestN.ckpt*. Pro vyhodnocení, která sada je nejlepší, se pak používá průměrné AUC ze všech sledovaných nálezů. Tento přístup ovšem nevede k optimálním výsledkům pro jednotlivé nálezy. Dá se to dobře pozorovat na hodnotách AUC postupně dosahovaných během učení:

krok	36 900	74 000	85 300	164 400	182 600
průměrné AUC	0.725	0.732	0.738	0.740	0.749
Enlarged_Cardiomediastinum	0.608	0.668	0.685	0.681	0.680
Cardiomegaly	0.830	0.829	0.842	0.848	0.850
Lung_Opacity	0.635	0.639	0.650	0.646	0.649
Lung_Lesion	0.715	0.732	0.754	0.747	0.760
Edema	0.848	0.846	0.837	0.843	0.849
Consolidation	0.680	0.687	0.682	0.688	0.711
Pneumonia	0.677	0.703	0.724	0.713	0.741
Atelectasis	0.637	0.650	0.639	0.624	0.629
Pneumothorax	0.812	0.807	0.786	0.818	0.810
Pleural_Effusion	0.788	0.814	0.825	0.811	0.768
Pleural_Other	0.715	0.735	0.727	0.743	0.758
Fracture	0.663	0.582	0.645	0.716	0.698
Support_Devices	0.823	0.827	0.798	0.738	0.832

Tabulka 5.11: Postupné výsledky AUC

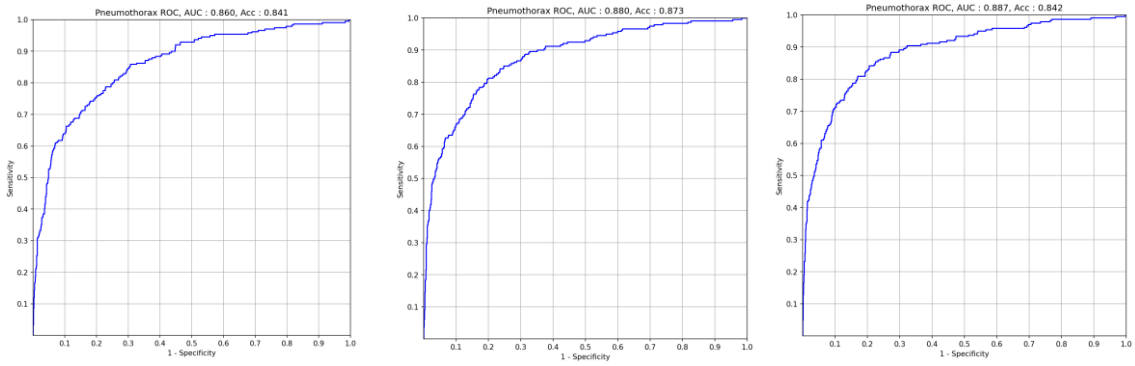
V tabulce 5.11 jsou vyznačeny nejlepší AUC, kterých bylo pro daný nález dosaženo. Jak je vidět například plicní výpotek (*Pleural Effusion*) měl AUC 0,825 v kroku 85 300, ale výpočet skončil v kroku 182 600 s AUC jen 0,768.



Obrázek 5.1: Graf postupných výsledků učení

Na obrázku 5.1 je dobře vidět postupné zlepšování průměrného AUC (modré body) během učení první epochy (adresář *logdir.41*). Ovšem také je vidět, že hodnoty AUC pro pneumotorax (oranžové body) se pohybují v poměrně výrazném rozptylu o asi 0,15. Když tedy ukončíme učení v nějakém místě podle průměrného AUC, ani zdaleka nemusíme mít výborné AUC pro každý nález.

Rozhodl jsem se tedy experimentovat a zkusit učení provádět jen s jednou třídou/nálezem. Začal jsem s pneumotoraxem. Jedna epocha učení pro samotný pneumotorax trvala 19 hodin. Zkusil jsem učení po dobu tří epoch a dostal se postupně k výsledkům AUC 0,860 → 0,880 → 0,887.



Obrázek 5.2: Zlepšující se ROC pneumotoraxu

Opakoval jsem stejný postup pouze s otokem (*Edema*). Zde jedna epocha trvala jen 13,5 hodiny, protože jsem nevkládal do učení opakovaně pozitivní výsledky parametrem `"enhance_times"`. Prošel jsem opět tři epochy a dostal AUC 0,832 → 0,847 → 0,859.

Další jsem zkusil samotnou pneumonii. Tady to bylo s opakováním a epocha trvala 15,25 hodiny. Postupné výsledky jednotlivých epoch byly AUC 0,642 → 0,656 → 0,684 → 0,691. Zde jsem zkusil čtyři epochy, protože jsem k mému překvapení dostal horší výsledek, než kterého jsem dosáhl při učení se všemi nálezy najednou. Výsledek se sice s každou epochou zlepšoval, ale jen o tisíciny.

Nakonec jsem otestoval ještě opacity mléčného skla (*Lung Opacity*). Zde jsem prošel dvě epochy s výsledným AUC 0,620 → 0,641. Tyto dvě epochy jsem spustil s nastavením, které nejistou hodnotu -1 v datech interpretuje jako 0. Potom jsem přepsal kód, aby -1 použil jako 1 a zkusil, jaký to bude mít dopad na výsledné AUC. Došel jsem k horší hodnotě AUC 0,616. Zdá se tedy, že alespoň pro tento nález je lepší interpretovat -1 jako 0. To samozřejmě nemusí platit pro ostatní nálezy a kdyby jich bylo méně, bylo by hezké postupně otestovat efekt tohoto nastavení na výsledky u každého nálezu. I tak si tento experiment vyžádal 11 dní učení.

Zde je souhrn výsledků dosažených při učení po jednotlivých nálezech versus učení všech 13 nálezů najednou:

Nález	Epocha 1	Epocha 2	Epocha 3	Epocha 4	Vše najednou
Pneumothorax	0.860	0.880	0.887	-	0.810
Edema	0.832	0.847	0.859	-	0.849
Pneumonia	0.642	0.656	0.684	0.691	0.741
Lung_Opacity -1=>0	0.620	0.641	-	-	0.649
Lung_Opacity -1=>1	0.616	-	-	-	0.649

Tabulka 5.12: Výsledky učení jednotlivě vs. najednou

Jak je vidět při učení nálezů jednotlivě lze dosáhnout i výrazně lepších výsledků (0,887 vs. 0,810), což jsem očekával, protože zde nedochází k negativnímu vlivu průměrování s ostatními výsledky. Ovšem velkým překvapením pro mě je, že u pneumonie jsem nebyl schopen se k výsledku učení všech nálezů najednou ani přiblížit, natož ho zlepšit (0,691 vs. 0,741). Nastavení neuronové sítě jsem několikrát kontroloval a nenašel jsem žádnou chybu. Samozřejmě platí, že při učení neuronové sítě se data zpracovávají v náhodném pořadí, a tedy každý pokus povede k lehce odlišnému výsledku, ale takto velký rozdíl je překvapivý (pokud by byl prostor pro další experimenty, zkusil bych se stejným nastavením spustit opakovaně první epochu učení, abych viděl, jaký je rozptyl výsledků). Možným vysvětlením tohoto jevu také je, že v případě, kdy je neuronová síť trénována na více nálezů, pak mohou být interně neuronovou sítí výsledky hledání jednoho nálezu použity i pro detekci nálezu jiného.

Druhým závěrem je, že se zde nabízí jiný přístup k řešení problému, který je sice pracnější a časově náročnější, ale mohl by dávat o něco přesnější výsledky – a to neučit jednu neuronovou síť na všechny nálezy, ale postupně naučit několik neuronových sítí pro každý nález zvlášť. Hodnocení výsledků v praxi by sice trvalo déle, ale v rozsahu sekund – což není podstatné, ale výhodou by bylo, že pokud by se některý nález ukázal problematicky hodnotitelný, pak by se dalo soustředit jen na něj a třeba zvýšit rozlišení snímků, které neuronová síť zpracovává.

5.6 Více projekcí

Při vyhodnocování výsledků čtvrté epochy učení samotné pneumonie jsem si všiml, že AUC, které hlásí program *test.py* je 0,691, kdežto výsledek při vykreslování ROC křivky programem *roc.py* je odlišný – 0,694. Přestože je to rozdíl malý, pátral jsem po jeho příčině. Po nějaké době jsem si všiml části kódu v *roc.py*, který jsem do té doby pouze mechanicky měnil, aniž bych se nad ním zamyslel. Jsou to tyto řádky:

```
outfile['Cardiomegaly'] =
   groups['Cardiomegaly'].min().reset_index()['Cardiomegaly']
outfile['Edema'] =
   groups['Edema'].max().reset_index()['Edema']
outfile['Consolidation'] =
   groups['Consolidation'].mean().reset_index()['Consolidation']
outfile['Atelectasis'] =
   groups['Atelectasis'].mean().reset_index()['Atelectasis']
```

```
outfile['Pleural Effusion'] =
   groups['Pleural Effusion'].mean().reset_index()['Pleural Effusion']
```

Při výpočtu AUC a vykreslení ROC křivky pracuje program *roc.py* s možnými více projekcemi u jednoho pacienta. V originálních datech *dev.csv* je 200 pacientů, z nichž 30 má dvě různé projekce snímků plic – PA projekci a LAT projekci – (3 pacienti mají dokonce projekce tři). V kódu učení *train.py* a vyhodnocení *test.py* se těmito projekcemi nijak zvlášť nepracuje, v *roc.py* se ale berou v úvahu. Různé projekce stejného pacienta se slučují při výpočtu dohromady, a to ještě každý nález jiným způsobem. Jak je vidět, u kardiomegálie se uvažuje minimum z dosažených hodnot, u otoku maximum a u zbývajících tří nálezů pak průměr hodnot. Podobně jako u výše zmíněné různé interpretace nejistých hodnot -1 jako 0 nebo 1 pro jednotlivé nálezy, ani zde nevím, proč se autoři k tomuto způsobu ovlivnění výpočtu rozhodli. Předpokládám, že tak prostě dosáhli lepšího výsledku. Samozřejmě z medicínského hlediska má velký smysl u jednotlivých pacientů zohledňovat všechny dostupné projekce. Při příštím vytváření souborů *valid.csv* a *test.csv* by bylo vhodnější vkládat do nich vždy celé soubory projekcí pacienta, jsou-li dostupné.

V praxi se ovšem přínos zkoumání více projekcí najednou nemusí projevit, protože záleží na zvyklostech daného pracoviště. Například v Nemocnici Valašské Meziříčí snímky hrudníku dělají, pokud to je to možné, vždy dvěma projekcemi – PA a LAT. Na druhou stranu ve FNKV se více projekcí dělá zřídka – pouze několik za měsíc.

5.7 Zkouška testovací sady dat

Vrátil jsem se k neuronové síti ve verzi se všemi třinácti nálezy a většími snímky (adresář *logdir.41.2*) a jejímu nejlepšímu výsledku *best2.ckpt* s AUC 0,749 a zkusil jsem ověřit její výsledky tak, že jsem nechal příkazy

```
cd ~/projekt/logdir.41.2
cp best2.ckpt best.ckpt
~/projekt/pyTest/bin/python classification/bin/test.py --num_workers 1
--in_csv_path classification/config/test.csv
~/projekt/pyTest/bin/python classification/bin/roc.py DJTest --
true_csv_path classification/config/test.csv
```

vyhodnotit testovací soubor *test.csv* neuronovou sítí a pak vypočítat AUC pro jednotlivé nálezy. Data obsažená v souboru *test.csv* neuronová síť dosud neviděla.

Nález	AUC valid.csv	AUC test.csv
průměr	0.749	0.741
Enlarged_Cardiomediastinum	0.680	0.682
Cardiomegaly	0.850	0.864
Lung_Opacity	0.649	0.661
Lung_Lesion	0.760	0.725
Edema	0.849	0.827
Consolidation	0.711	0.685
Pneumonia	0.741	0.706
Atelectasis	0.629	0.635
Pneumothorax	0.810	0.783
Pleural_Effusion	0.768	0.770
Pleural_Other	0.758	0.756
Fracture	0.698	0.717
Support_Devices	0.832	0.827

Tabulka 5.13: Porovnání AUC *valid.csv* s *test.csv*

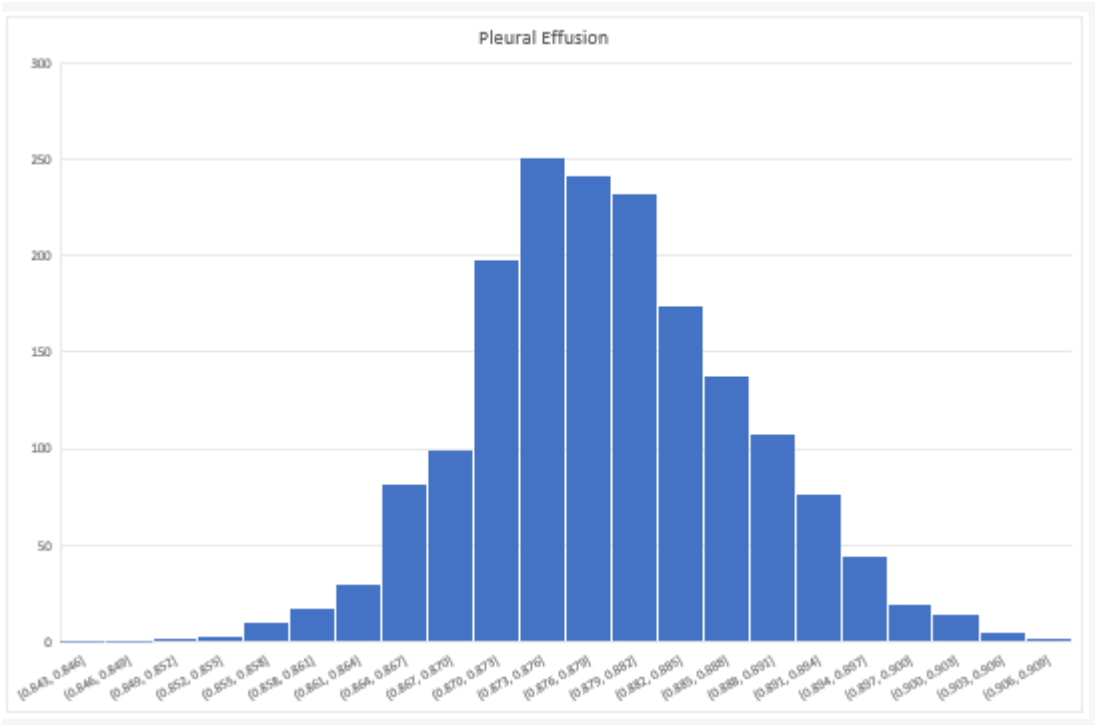
Výsledky AUC jednotlivých nálezů jsou podobné ve validační i testovací sadě, takže neuronová síť zřejmě netrpí přetrénováním (*overfitting*).

Nicméně číselné výsledky výpočtu neuronové sítě zapsané do souboru *pred_csv_done.csv* se vymykají očekávání. Mělo by se mělo jednat o čísla mezi 0 a 1, která lze interpretovat jako sílu přesvědčení neuronové sítě, že snímek obsahuje daný nález. Hodnota okolo 0,5 by pak měla znamenat, že neuronová síť si není jistá. Nicméně v mém případě jsou hodnoty vychýlené a posunuté mimo 0,5:

Nález	Min	Max
Enlarged Cardiomediastinum	0.189	0.440
Cardiomegaly	0.132	0.488
Lung Opacity	0.407	0.521
Lung Lesion	0.254	0.573
Edema	0.560	0.742
Consolidation	0.334	0.530
Pneumonia	0.197	0.528
Atelectasis	0.521	0.670
Pneumothorax	0.130	0.482
Pleural Effusion	0.843	0.908
Pleural Other	0.017	0.128
Fracture	0.267	0.418
Support Devices	0.725	0.852

Tabulka 5.14: Minima a maxima výsledků hodnocení

Například plicní výpotek (*Pleural Effusion*) se u dat z *test.csv* pohybuje pouze mezi 0,843 a 0,908. Na opačném konci spektra pak je *Pleural Other*, kde dosahuji hodnot mezi 0,017 a 0,128. Neznám vysvětlení tohoto jevu. Přitom ale vím, že se podle dříve spočítané plochy pod ROC křivkou (AUC 0,770 a 0,756) dají tato čísla použít pro rozhodování o nálezech celkem dobře.



Obrázek 5.3: Histogram výsledků plicního výpotku

Předpokládám tedy, že je potřeba najít vhodný práh, kterým se data rozdělí na pravda/nepravda.

5.8 Výpočet prahů

Výpočet prahu lze velmi lehce provést při výpočtu AUC v programu *roc.py* na místě, kde se počítá ROC křivka. Je třeba jen se rozhodnout, jakým způsobem práh určit. Samozřejmě zněna hodnoty prahu mění poměr mezi falešně pozitivními a falešně negativními nálezy. Je těžké teoreticky stanovit, co je pro lékaře v případě daného nálezu cennější, zda chce radši přednostně vidět i snímky, které nález obsahují s menší jistotou, nebo zda je lepší zabývat se prioritně těmi jistějšími s rizikem přehlédnutí nálezu jiného. Představuji si, že v praxi by aplikace pro prioritizaci byla vybavena posuvníky, kterými by se daly měnit prahy v závislosti na tom, jak dobře by byla síť naučená na ten či onen nález a také jak by ho dokázala detekovat v praxi na reálných snímcích.

Na webové stránce [13] lze najít několik způsobů výpočtu rozhodovacího prahu. Do kódu *roc.py* jsem přidal tři způsoby výpočtu prahů:

```
threshold1 = thresholds[np.argmax(tpr - fpr)]
threshold2 = thresholds[np.argmin(abs(tpr-(1-fpr)))]
threshold3 = thresholds[np.argmin((1 - tpr) ** 2 + fpr ** 2)]
```

Výsledky výpočtu jsou v tabulce níže a jsou si numericky poměrně podobné. Zvolil jsem první způsob výpočtu, což je Youdenova J statistika a hodnoty pro jednotlivé nálezy ukládám do pomocného souboru *test/cut_off.json*. Pokud tento soubor existuje, pak upravený program *test.py* nevypisuje výsledky číselnými hodnotami mezi 0 a 1, ale načte si uložené prahy a vypisuje rovnou hodnoty 0 a 1.

Nález	threshold1	threshold2	threshold3
Enlarged_Cardiomediastinum	0.312	0.295	0.295
Cardiomegaly	0.293	0.283	0.280
Lung_Opacity	0.449	0.450	0.449
Lung_Lesion	0.321	0.323	0.321
Edema	0.667	0.673	0.669
Consolidation	0.439	0.442	0.439
Pneumonia	0.361	0.371	0.369
Atelectasis	0.625	0.627	0.625
Pneumothorax	0.191	0.191	0.191
Pleural_Effusion	0.880	0.879	0.880
Pleural_Other	0.031	0.033	0.032
Fracture	0.336	0.333	0.336
Support_Devices	0.790	0.790	0.790

Tabulka 5.15: Různě vypočtené prahy pro *test.csv*

5.9 Výpočet priority

Ještě je potřeba stanovit prioritu snímku podle zjištěných nálezů. Postupuji dle priorit, které byly pro jednotlivé stanoveny lékaři FNKV (viz tabulka 3.2).

Toho dosahuji dopsáním kódu do *test.py*, který doplní výstupní CSV soubor nejvyšší prioritou podle vypočtených nálezů:

```
if predBool[0]=="1" or predBool[5]=="1" or predBool[11]=="1":
      priority = "1"

if predBool[1]=="1" or predBool[3]=="1" or predBool[7]=="1" or
 predBool[9]=="1" or predBool[10]=="1" or predBool[12]=="1":
      priority = "2"

if predBool[2]=="1" or predBool[4]=="1" or predBool[6]=="1" or
 predBool[8]=="1":
      priority = "3"

batch = batch + ',' + priority
```

Je třeba upozornit, že aktuální kód může stanovit i prioritu 0, pokud na snímku nenajde vůbec žádný nález, ale k tomu dochází velmi zřídka. Pokud by to někde v dalším zpracování vadilo, stačí nastavit v *test.py* inicializaci `priority = "0"` na `"1"`.

6 Výsledky

6.1 Test klasifikace

Testovací data v souboru *test.csv* jsem nechal otestovat neuronovou síti, tentokrát za použití vypočtených prahů. Výsledky z neuronové sítě i původní popisky z CheXpert jsem zkombinoval (soubor *Test 220415.xlsx*) a vypočetl TN, TP, FN, FP a některé další související ukazatele.

Nález	TP	TN	FP	FN	Sens.	Spec.	F-score	Youden's J statistic	AUC
Enlarged Cardiomediastinum	6 %	70 %	18 %	6 %	0.47	0.79	0.31	0.26	0.682
Cardiomegaly	12 %	71 %	14 %	4 %	0.75	0.84	0.57	0.59	0.864
Lung Opacity	32 %	30 %	20 %	18 %	0.64	0.61	0.63	0.25	0.661
Lung Lesion	9 %	55 %	33 %	3 %	0.72	0.63	0.33	0.35	0.725
Edema	18 %	51 %	27 %	3 %	0.85	0.65	0.54	0.50	0.827
Consolidation	9 %	50 %	36 %	4 %	0.70	0.58	0.32	0.28	0.685
Pneumonia	9 %	49 %	40 %	3 %	0.77	0.55	0.29	0.32	0.706
Atelectasis	11 %	47 %	35 %	6 %	0.63	0.57	0.34	0.20	0.635
Pneumothorax	10 %	63 %	23 %	4 %	0.72	0.73	0.43	0.45	0.783
Pleural Effusion	27 %	44 %	17 %	12 %	0.70	0.72	0.65	0.42	0.770
Pleural Other	9 %	52 %	36 %	2 %	0.81	0.59	0.32	0.40	0.756
Fracture	8 %	65 %	23 %	4 %	0.63	0.73	0.35	0.36	0.717
Support Devices	38 %	38 %	11 %	12 %	0.76	0.77	0.76	0.53	0.827

Tabulka 6.1: Výsledky zpracování dat z *test.csv*

6.2 Test prioritizace dle CheXpert

Na testovacím souboru *test.csv*, ve kterém je 1 760 snímků, jsem vyzkoušel určení priorit jednak podle popisek původní datové sady CheXpert a pak podle výsledků výpočtu neuronové sítě (viz soubor *Test 220415.xlsx*).

V tabulce jsou na svislé ose priority určené podle nálezu zapsaného v datové sadě a na vodorovné ose priority vypočtené neuronovou sítí. Na diagonále jsou tedy procenta snímků, kterým neuronová síť určila prioritu stejnou jako je vypočtena podle zdrojových dat (což samozřejmě neznamená, že je přesná shoda v diagnóze). Mimo diagonálu jsou případy, kde se priority liší. Priority jsou:

3 – urgentní
2 – závažné
1 – může počkat

Priorita		Neuronová síť		
Data		1	2	3
	1	0.1 %	1.6 %	2.7 %
	2	0.1 %	5.5 %	18.8 %
	3	0.0 %	6.5 %	64.7 %

Tabulka 6.2: Priorita nálezů dle CheXpert

V celkovém součtu byla priorita nastavena stejně jako podle zdrojových dat v 70,3 % případů. V 6,6 % případů určila neuronová síť prioritu nižší a ve 23,1 % případů stanovila prioritu vyšší.

6.3 Test prioritizace dle lékaře

Hned zkraje své práce jsem vybral 70 snímků z datové sady CheXpert (viz soubor *TestPriority 220424.xls*) a to tak, aby na nich podle popisů byl právě jeden nález. Z každé kategorie 13 nálezů + kategorie *Bez nálezu* jsem náhodně vybral pět snímků. Snímky jsem náhodně přejmenoval a zaslal lékaři, aby stanovil jejich prioritu podle svých zkušeností.

Výsledky (viz soubor *Kategorizace snímků lékař1 a.xlsx*) byly následující:

Priorita		Neuronová síť		
Lékař		1	2	3
	1	0.0 %	30.0 %	22.9 %
	2	0.0 %	7.1 %	27.1 %
	3	0.0 %	0.0 %	12.9 %

Tabulka 6.3: Priorita nálezů dle lékaře

Neuronová síť určila stejnou prioritu jako lékař jen ve 20 % případů a v 80 % případů prioritu nadhodnotila.

Lékař, kromě stanovení priority, popsal i nálezy, které na snímcích viděl. Podle datové sady CheXpert na nich měl být vždy jen jeden nález, ale lékař nálezů většinou popsal více – někdy až šest. Požádal jsem tedy o nové hodnocení s tím, že by měl být popsán jen jeden nález.

Výsledky (viz soubor *Kategorizace snímků lékař1 b.xlsx*) shody priority lékaře s neuronovou sítí na druhý pokus byly:

Priorita		Neuronová síť		
Lékař		1	2	3
	1	0.0 %	32.9 %	22.9 %
	2	0.0 %	4.3 %	27.1 %
	3	0.0 %	0.0 %	12.9 %

Tabulka 6.4: Priorita nálezů dle lékaře, druhý pokus

Neuronová síť tentokrát určila stejnou prioritu jako lékař jen ve 17 % případů a v 83 % případů prioritu nadhodnotila.

7 Diskuse

7.1 Určování nálezů

Při zkoušce na snímcích vyčleněných před učením ze sady CheXpert (*test.csv*) se ukázalo, že neuronovou síť lze naučit rozpoznávat nálezy na snímcích hrudníku.

Z tabulky 6.1 je vidět, že některé nálezy se nepovedlo naučit příliš dobře (např. *Atelectasis*, *Enlarged Cardiomediastinum, Lung Opacity* a *Consolidation*) s Youdenovou J-statistikou v rozmezí od 0,20 do 0,28. Jiné nálezy jsou detekovány celkem dobře (např. *Edema* - 0,50 a *Cardiomegaly* - 0,59). Hlavním hodnotícím kritériem je podle mě AUC a Youdenova J-statistika, která vyjadřuje pravděpodobnost správného výsledku testu. Nicméně by bylo vhodné dokázat dalšími úpravami neuronovou síť vycvičit na AUC vyšší než 0,9, aby dávala spolehlivější výsledky.

7.2 Návrhy pro další vývoj

Při dalším vývoji, a hlavně zlepšování přesnosti neuronové sítě, by bylo dobré věnovat více času a úsilí následujícím oblastem:

- zvýšit rozlišení snímků na 1 000 pixelů delší strany. Tím se sice značně prodlouží doba učení, ale pokud se zároveň zvýší i AUC, pak je možné všechny snímky v datové sadě jednorázově zmenšit na tuto velikost, čímž se učení zase zrychlí. Nebo by bylo možno jako kompromis zmenšit pouze snímky ve validační sadě.
- experiment s velikostí snímků bych prováděl pouze na pěti nálezech, které jsou součástí původní soutěže a původní sadě *valid.csv*. Cílem by mělo být dosáhnout srovnatelných AUC, které uvádí autoři původní práce.
- také by bylo vhodné s původním kódem a daty prozkoumat rozsahy čísel, která neuronová síť produkuje, zda jsou také, jako v mých výsledcích, posunuty mimo středovou hodnotu 0,5 a případně zjistit proč.
- dále by bylo vhodné se blíže zaměřit na to, proč se po rozšíření z pěti nálezů (viz tabulka 4.3) na sedm nálezů (viz tabulka 5.2) významně propadlo AUC i u nálezů, které se nerozšiřovaly.
- vytvořit nové validační a testovací sady, které by vůbec neobsahovaly nálezy s nejistým hodnocením (-1).
- zároveň při vytváření nových sad vždy zahrnout všechny dostupné projekce vyšetření pacienta. S tím pak souvisí určení, jak výsledky nálezů z jednotlivých projekcí vyhodnocovat – zda minimem, maximem či průměrem hodnot jednotlivých snímků.
- při vytváření nových sad zahrnout původní soubor *valid.csv*, protože byl ručně zkontrolován lékařem. Popis dalších snímků doplňovaných snímků, pokud

možno, zkontrolovat lékařem. Nebo, ideálně, vytvořit rozsáhlejší testovací a validační sadu jen podle popisů českého lékaře.
- otestovat, zda u jednotlivých nálezů vede k lepším výsledkům interpretace nejisté hodnoty -1 jako 0, nebo 1.
- vypustit nálezy s nízkou prioritou (*Enlarged Cardiomediastinum*, *Consolidation* a *Fracture*) a zkusit, zda jejich vynechání nepovede k vyšším AUC zbývajících nálezů.
- vypustit z výpočtu priority nálezy s nízkým AUC (<0,7), pokud se jejich výsledky nepovede dalším učením zlepšit.
- ve spolupráci s lékařem zhodnotit význam nálezu *Pleural Other* a případně ho zcela vypustit.
- vyzkoušet snímky pořízené v praxi a popsané lékaři, kteří by nástroj na prioritizaci snímků používali.
- velmi žádoucí by bylo i zvýšení výkonu pracovního stroje, protože počítání jedné epochy trvalo až 41 hodin, což je na experimentování příliš dlouhé.

7.3 Určování priority dle CheXpert

V tabulce 6.2 jsou výsledky porovnání priority, kde jsem podle popisek testovacích dat CheXpert spočetl prioritu a porovnal ji s prioritou spočtenou podle nálezů, které popsala neuronová síť. V celkovém součtu byla priorita nastavena stejně jako podle zdrojových dat v 70,3 % případů. V 6,6 % případů určila neuronová síť prioritu nižší a ve 23,1 % případů stanovila prioritu vyšší. To považuji za solidní výsledek, který by šlo v praxi použít. A tendence k přeceňování diagnóz je myslím vhodnější, než kdyby byly podceňovány.

7.4 Určování priority dle lékaře

V tabulkách 6.3 a 6.4 jsou výsledky prvního a druhého porovnání priorit, které stanovil lékař, s prioritami, které určila neuronová síť dle svých nálezů. Oba výsledky jsou si podobné: neuronová síť určila stejnou prioritu jako lékař jen ve 20 % případů a v 80 % případů prioritu nadhodnotila. Síť ani jednou nepoužila prioritu 1 – může počkat. Tento výsledek nelze považovat za dobrý. Jeho příčina je patrně vysvětlena v následujícím bodě.

7.5 Rozdíly popisů CheXpert a lékařů

Jak je zmíněno v kapitole 6.3 lékař snímky určené k otestování priority popsal jednotlivými nálezy. Tyto nálezy jsem porovnal s popisy v datech CheXpert. Výsledky se značně lišily. Připomínám, že testovací snímky byly popsány v sadě CheXpert každý právě jedním nálezem. Tento nález byl zmíněn lékařem jen ve 42 % případů. Pokud bych hodnotil celkovou shodu popisů snímků mezi lékařem a popisem v CheXpertu, došel bych k jen číslu 23 % (metrika je taková, že například mezi nálezy „1, 3, 9“ a „1, 9“ je shoda 2/3).

Když jsem pak na druhý pokus lékaře požádal o hodnocení snímků jen jedním „hlavním“ nálezem, tak výsledek byl ještě horší, protože nález z CheXpertu zmínil lékař jen v 29 % případů. Horší výsledek se se vlastně dal předpokládat, protože jsem lékaři omezil volnost hodnocení.

Kvůli značné neshodě mezi popisem lékaře a popisem v datové sadě CheXpert jsem požádal druhého lékaře, aby popsal stejných 70 snímků. Výsledek (viz soubor *Kategorizace snímků lékař2.xlsx*) byl, že nález popsaný v CheXpertu byl jmenován v 34 % případů a celková shoda byla 22 %. Tato čísla jsou zhruba podobná číslům, kterými jsem hodnotil popis prvního lékaře.

Provedl jsem ještě srovnání obou lékařů proti sobě (viz soubor *Kategorizace snímků srovnání lékařů.xlsx*) a zde celková shoda vychází na 71 %. Řekl bych tedy, že oba lékaři udělali při hodnocení dobrou práci a vzájemně si své výsledky víceméně potvrdili. Otázka je, proč se jejich hodnocení tak moc liší od hodnocení v sadě CheXpert.

Nabízí se několik vysvětlení:

- jelikož popisky v datech CheXpertu vytvářel algoritmus, určitě došlo k zanesení nějakých chyb a některé nálezy mohly být vynechány.
- můj výběr snímků, který se omezil na volbu jen z těch, které mají právě jeden nález, je možná z medicínského hlediska nevhodný, protože nálezů bývá většinou více. Zcela náhodný výběr by mohl dopadnout lépe.
- radiologové ve Stanfordu měli snímky pravděpodobně ve formátu DICOM zobrazené na diagnostických monitorech, kdežto zde jsme pracovali pouze se snímky v nižší kvalitě ve formátu JPEG a s nižší bitovou hloubkou, což může být problém při hledání např. pneumotoraxu nebo zlomeniny.
- některé popisy snímků v CheXpert mohly být cílené jen na nějakou konkrétní patologii a nepopisovaly se tedy všechny nálezy na snímku viditelné.
- u sady CheXpert je přímo zmíněno, že popisky jsou tvořeny podle: „*mentions from a list of observations from the Impression section of radiology reports, which summarizes the key findings in the radiographic study.*“ [14]. Tedy je možné, že původní zpráva lékaře mohla obsahovat více nálezů, ale CheXpert se zaměřil jen na ty klíčové.

V každém případě by si tento úkaz zasloužil podrobnější prozkoumání. Samozřejmě by bylo lepší, kdyby šlo neuronovou síť natrénovat na datech, která by byla popsána českými lékaři, ale to v tento moment není možné. Bylo by tedy vhodné vytvořit novou testovací sadu pro prioritu a tentokrát ji nelimitovat jen na právě jeden popsaný nález. Sadu bych vybral ze snímků, které jsou původně popsány v souboru *dev.csv*, což jsou snímky, které popisoval americký radiolog přímo. Nebo by, dle ochoty lékařů, bylo možno ji popsat i celou – tedy všech 235 snímků – a pak znovu provést srovnání českých a amerických popisů a přiřazených priorit.

8 Závěr

Pro prioritizaci rentgenových snímků hrudníku byl použit již existující software neuronové sítě vytvořený firmou *JF HEALTHCARE*. Tento software dosáhl v soutěži přesnosti určování nálezů vysokého umístění mezi obdobnými programy a jeho zdrojový kód je k dispozici pro další použití. Pro učení neuronové sítě byla použita datová sada CheXpert, která obsahuje cca 224 000 rentgenových snímků plic, a hlavně i popisky 13 nejčastějších nálezů. V práci jsou popsána některá úskalí při instalaci software a dále i jeho konfigurace, rozšíření a úprava pro požadované účely prioritizace snímků.

Po mnoha pokusech a času stráveném na učení neuronové sítě byla nalezena konfigurace, která slibuje použitelné výsledky.

Byly vytvořeny dvě testovací sady – jedna velká (1 760 snímků) pro testování učení neuronové sítě a druhá malá (70 snímků) pro porovnání priorit stanovených lékařem a sítí. Obě byly otestovány naučenou neuronovou sítí.

Na dosažené výsledky se lze dívat ze dvou hledisek. Při porovnávání priority vypočtené neuronovou sítí na velké testovací sadě snímků s prioritou stanovenou podle nálezů obsažených v datové sadě CheXpert bylo dosaženo správného přiřazení priority u 70 % snímků, což je dobrý výsledek.

Ovšem při porovnání priority vypočtené neuronovou sítí s prioritou určenou lékařem na malé testovací sadě, bylo dosaženo stejné priority jen u 20 % snímků a 80 % snímků síť přiřadila prioritu vyšší.

Problémem je, že při srovnání popisů snímků lékařem s popisy obsaženými v CheXpertu bylo zjištěno, že se tyto značně liší – shodují se jen na 29 %. Není jasné, co je příčinou tohoto rozdílu a je potřeba to dále zkoumat.

V tomto případě bych preferoval výsledek, který neuronová síť dává na datech a popisech, na kterých byla primárně naučena. Z tohoto pohledu by se dalo říct, že projekt prioritizace rentgenových snímků hrudníku je na dobré cestě, a i s takto naučenou neuronovou sítí by šlo systém v praxi spustit a vyzkoušet.

V předchozím textu jsem pak nastínil několik cest, které by měly vést ke zlepšení přesnosti výsledků neuronové sítě a které by stálo za to prozkoumat.

Jako velmi důležité pak vidím zapracování systému ukládání všech snímků, které jím projdou, a jejich popisu a priority přiřazené lékařem. Touto metodou by postupně mohlo být získáno dostatečné množství dat, kterými by šla neuronová síť natrénovat znovu a přesně podle popisů a požadavků českých lékařů.

Seznam použité literatury

1. X-ray (Radiography) - Chest. *RadiologyInfo.org For patients* [online]. [cit. 2022-04-22]. Dostupné z: https://www.radiologyinfo.org/en/info/chestrad

2. ÇALLI, Erdi, Ecem SOGANCIOGLU, Bram VAN GINNEKEN, Kicky G. VAN LEEUWEN a Keelin MURPHY. Deep learning for chest X-ray analysis: A survey. *Medical Image Analysis* [online]. 2021, **72** [cit. 2022-05-01]. ISSN 13618415. Dostupné z: doi:10.1016/j.media.2021.102125

3. *AI for Radiology: an implementation guide* [online]. [cit. 2022-05-01]. Dostupné z: https://grand-challenge.org/aiforradiology/

4. ANNARUMMA, Mauro, Samuel J. WITHEY, Robert J. BAKEWELL, Emanuele PESCE, Vicky GOH a Giovanni MONTANA. Automated Triaging of Adult Chest Radiographs with Deep Artificial Neural Networks. *Radiology* [online]. 2019, **291**(1), 196-202 [cit. 2022-05-01]. ISSN 0033-8419. Dostupné z: doi:10.1148/radiol.2018180921

5. BALTRUSCHAT, Ivo, Leonhard STEINMEISTER, Hannes NICKISCH, Axel SAALBACH, Michael GRASS, Gerhard ADAM, Tobias KNOPP a Harald ITTRICH. Smart chest X-ray worklist prioritization using artificial intelligence: a clinical workflow simulation. *European Radiology* [online]. 2021, **31**(6), 3837-3845 [cit. 2022-05-01]. ISSN 0938-7994. Dostupné z: doi:10.1007/s00330-020-07480-7

6. *Medical Information Mart for Intensive Care: CXR* [online]. [cit. 2022-04-26]. Dostupné z: https://mimic.mit.edu/docs/iv/modules/cxr/

7. VINBIGDATA: VinDr-CXR: An open dataset of chest X-rays with radiologist's annotations. *VINDr: Chest X-ray* [online]. [cit. 2022-04-26]. Dostupné z: https://vindr.ai/datasets/cxr

8. *NIH Clinical Center: CXR8* [online]. [cit. 2022-04-26]. Dostupné z: https://nihcc.app.box.com/v/ChestXray-NIHCC

9. *CheXpert: A Large Chest X-Ray Dataset And Competition* [online]. [cit. 2022-04-26]. Dostupné z: https://stanfordmlgroup.github.io/competitions/chexpert/

10. *PadChest: A large chest x-ray image dataset with multi-label annotated reports* [online]. [cit. 2022-05-01]. Dostupné z: https://bimcv.cipf.es/bimcv-projects/padchest/

11. JF&NNU - soutěžní kód v Pythonu - YWW (ensemble) [online]. [cit. 2022-04-16]. Dostupné z: https://github.com/jfhealthcare/Chexpert

12. Gao HUANG, Zhuang LIU, Laurens VAN DER MAATEN a Kilian Q. WEINBERGER. *Densely Connected Convolutional Networks*. arXiv, 2016. Dostupné z: doi:10.48550/ARXIV.1608.06993

13. *Roc curve and cut off point. Python* [online]. [cit. 2022-04-16]. Dostupné z: https://stackoverflow.com/questions/28719067/roc-curve-and-cut-off-point-python

14. IRVIN, Jeremy, Pranav RAJPURKAR, Michael KO, et al. *CheXpert: A Large Chest Radiograph Dataset with Uncertainty Labels and Expert Comparison*. 2019. Dostupné také z: http://arxiv.org/abs/1901.07031 https://stanfordmlgroup.github.io/competitions/chexpert/