



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV INTELIGENTNÍCH SYSTÉMŮ
DEPARTMENT OF INTELLIGENT SYSTEMS

GENEROVÁNÍ OBRÁZKŮ SÍTNICE S OMEZENÝM MNOŽSTVÍM TRÉNOVACÍCH DAT

GENERATING RETINAL IMAGES WITH A LIMITED AMOUNT OF TRAINING DATA

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

YAHOR SENICHAK

VEDOUCÍ PRÁCE
SUPERVISOR

MSc. ANDRII KAVETSKYI

BRNO 2023

Abstrakt

Účelem této studie je prozkoumat pokrok a aplikaci počítačového vidění a generativních adversariálních sítí (GAN 3.1) v diagnostice a studiu onemocnění očního pozadí. Zvláštní pozornost je věnována nejnovějším pokrokům v oblasti syntézy medicínských dat a vývoji vlastního algoritmu.

Byla provedena implementace nejnovějších pokroků v oblasti architektury hlubokého učení U-GAT-IT [22], která zahrnuje dva páry hlubokých neuronových sítí (dva generátory a dva diskriminátory). Tato implementace prošla tréninkem po dobu přibližně 300 000 iterací, během kterých byly dosaženy pozitivní výsledky. Byla zaznamenána dynamika procesu tréninku a provedeny testy, které prokázaly schopnost generovat kvalitní syntetické snímky očního pozadí nezávisle na vstupních datech.

Abstract

The purpose of this study is to explore the progress and application of computer vision and generative adversarial networks (GANs 3.1) in the diagnosis and study of fundus diseases. Particular attention is paid to the latest advances in the field of medical data synthesis and the development of our own algorithm.

Recent advances in the deep learning architecture U-GAT-IT [22], which includes two pairs of deep neural networks (two generators and two discriminators), have been implemented. This implementation was trained for approximately 300,000 iterations, during which positive results were obtained. The dynamics of the training process were recorded and tests were performed to demonstrate the ability to generate high-quality synthetic images of the ocular background independent of the input data.

Klíčová slova

Oční dno, strojové učení, automatizace diagnostiky, hluboké učení, konvoluční neuronové sítě, GANs, Lékařská segmentace, U-net, P-Unet, ResNet, FCN, lékařské zobrazování, U-GAT-IT, Adaptivní normalizace, Ztráta CAM, Ztráta cyklu, Generátor fundu

Keywords

Fundus, machine learning, diagnostic automation, deep learning, convolutional neural networks, GANs, Medical segmentation, U-net, P-Unet, ResNet, FCN, medical imaging, U-GAT-IT, Adaptive normalization, CAM loss, Cycle loss, Fundus generator

Citace

SENICHAK, Yahor. *Generování obrázků sítnice s omezeným množstvím trénovacích dat*. Brno, 2023. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce MSc. Andrii Kavetskyi

Generování obrázků sítnice s omezeným množstvím trénovacích dat

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana MSc. Andriiho Kavetskyho. Uvedl jsem všechny literární prameny, publikace a další zdroje, ze kterých jsem čerpal.

.....

Yahor Senichak
6. května 2024

Poděkování

Rád bych poděkoval vedoucímu MSc. Andriimu Kavetskyimu, MSc. Rostislavu Likhogruďovi a Bc. Vladislavu Varfolomejevovi za poskytnutí vědeckých zdrojů, potřebné literatury, cenných technických a organizačních rad a osobních konzultací.

Obsah

1	Úvod	4
2	Lidské oko	5
2.1	Stavba oční koule	5
2.2	Patologie oční koule	7
2.2.1	Diabetická retinopatie	7
2.2.2	Neproliferativní retinopatie	7
2.2.3	Proliferativní retinopatie	8
2.3	Oftalmoskopie	9
3	Technologie generativních sítí	10
3.1	Generativní kompetitivní síť (GAN)	11
3.2	Ztrátové funkce	11
3.2.1	Ztráta cyklu[21]	11
3.2.2	Ztráta identity	12
3.3	2D Konvoluce	12
3.3.1	Okraje	12
3.3.2	Krokování (max-pooling)	13
3.4	Aktivační funkce	13
3.4.1	ReLU (rektifikovaná lineární jednotka)	14
3.4.2	Funkce Softmax	14
3.5	Zadání klasifikace. Diskriminátor	14
3.5.1	VGG-16	14
3.6	Fully conected layer. Umělá inteligence	16
3.7	ResNet (34,50,101)	16
3.8	Generátor	17
3.9	Normalizace	18
4	Úspěchy v oblasti	20
4.1	U-net	21
4.1.1	P-Unet(33)	22
4.2	U-GAT-IT	22
4.2.1	Funkce ztrát	23
4.3	Výzvy Fundus GAN	24
5	Návrh vlastního algoritmu	25
5.0.1	Datové sady	25
5.0.2	Plán řešení	25

5.1	Implementace	26
5.2	Struktura programu	26
5.3	Trénování neuronové sítě	27
5.4	Testování a validace	28
6	Experimenty a analýza výsledků	30
6.1	Syntéza z původních dat	30
6.2	Syntéza z bílého šumu	31
6.3	Analýza výsledků	32
6.4	Zjištěné problémy	33
7	Závěr	34
	Literatura	35

Seznam obrázků

2.1	Stavba oční koule [34]	5
2.2	Diabetická retinopatie: A - Neproliferativní retinopatie, B - Proliferativní retinopatie [1]	8
2.3	Struktura oftalmoskopu [11]	9
3.1	Princip konvolučního kroku [30]	12
3.2	Striding [30]	13
3.3	Podrobná vizualizace max-pooling [30]	13
3.4	Architektura VGG-16 [6]	15
3.5	Neuronová síť s vrstvami úplného připojení [19]	16
3.6	Chyba školení (vlevo) a chyba testu (vpravo) s 20-vrstvými a 56-vrstvými "jednoduchými" sítěmi. Hlubší síť má větší trénovací chybu, a tedy chybu testování. [20]	17
3.7	Identity block [20]	17
3.8	Architektura generátoru [24]	18
4.1	Dvoustupňová generativně-adverzní síť (GAN), známá jako "Fundus GAN" [31], určená k syntéze obrazů sítnice, která pomáhá trénovat klasifikátory obrazů sítnice. [25]	20
4.2	Vstupní a výstupní políčka modelu P-Unet v průběhu 100 epoch.	22
4.3	Původní a syntetická data, a mapy pozornosti diskriminátora k nim.	24
4.4	A) Fundus GAN [31], B) Upravený U-GAT-IT	24
5.1	A) Fundus GAN [31], B) Upravený U-GAT-IT [22]	26
5.2	A) Generování syntetických dat ze skutečných lékařských dat, B) Generování syntetických dat z bílého šumu	27
5.3	A - Vstupní data. B - Syntetický strom cév. C - Syntetický obraz fundu.	28
5.4	Algoritmus pro zpracování vstupních a výstupních dat	29
6.1	Původní vstupní data a syntetický snímek	30
6.2	Generování z náhodných snímků	31
6.3	Výsledky generátoru A2B při 50,70,90,150,170 tisících iterací	32
6.4	A - Skutečný výsledek celé námi navržené architektury . B - Výsledky generátoru A2B se vstupní původní anotací cévního stromu	33

Kapitola 1

Úvod

V posledních letech jsou techniky hlubokého učení široce uznávány v medicíně, zejména v souvislosti s včasnou diagnostikou různých onemocnění. Na základě technik počítačového vidění byly klasifikátory natrénované na lékařských datech, úspěšně použity k analýze různých typů vzorků, jako jsou fluorografické snímky a snímky magnetické rezonance, jakož i snímky očního pozadí. Tyto algoritmy jsou schopny odhalit patologické změny a určit jejich povahu a závažnost.

Úspěšnost těchto algoritmů však do značné míry závisí na kvalitě a množství dostupných dat a anotací. Nedostatek kvalitních dat s podrobnými anotacemi zůstává významným problémem. Proces využití reálných lékařských dat často komplikují byrokratické postupy vyžadující osobní souhlas pacientů.

Kromě toho může kvalitu získaných dat významně ovlivnit různorodost zařízení používaných k vytváření lékařských snímků a rozdíly ve světelných podmírkách na různých klinikách. V důsledku toho nemusí algoritmy natrénované na některých typech dat fungovat efektivně na jiných.

Cílem tohoto článku je prozkoumat pokroky v oblasti fúze lékařských dat a navrhnut, implementovat a otestovat vlastní algoritmus pro vytváření různých stylizovaných očních dat. Součástí práce je také implementace veřejně dostupného řešení pro uživatele.

Následující kapitola 2 popisuje strukturu sítnice, proces získání skutečného obrazu fundu a patologie, které se v datech odrážejí. Kapitola "Technologie generativních sítí"³ popisuje kroky a techniky v úlohách hlubokého učení , hlavní komponenty **3.1** , metody hodnocení kvality mezivýsledků a principy trénování hlubokých sítí **3.2**. Kapitola 4 se zabývá nejnovějšími pokroky v oblasti fúze lékařských dat. Zvláštní pozornost je věnována architektonickým řešením používaným v této oblasti a jsou analyzovány nevýhody těchto architektur. Kapitola 5 popisuje technickou implementaci souborů architektury, trénovací data, na kterých byla síť trénována, provedené úpravy a průběžnou validaci celého řešení. Kapitola 6 popisuje výsledky získané během práce, matematickou přesnost výsledků a dynamiku kvality výsledků v procesu učení.

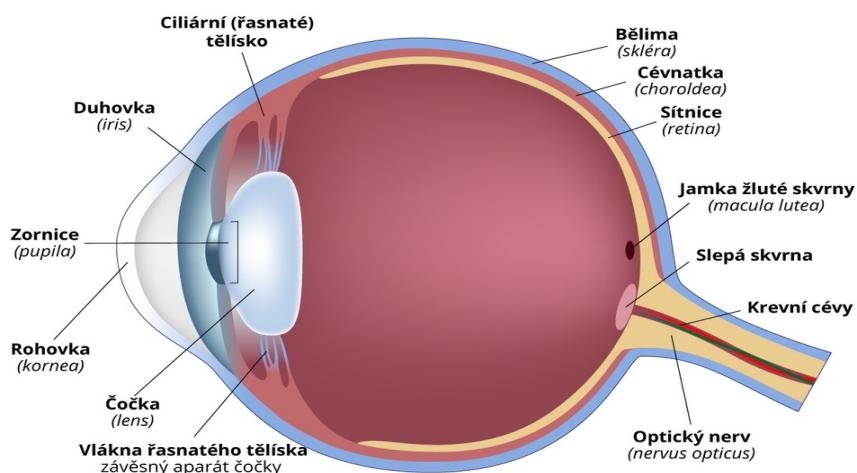
Kapitola 2

Lidské oko

Díky vysoké efektivitě a širokému využití multimodální vizualizace se tento obor rozvíjí a stává se standardní procedurou diagnostiky mnoha chorob v klinické praxi. Široká dostupnost multimodálních obrazů by mohla ve výzkumu značně zjednodušit vývoj a validaci počítacových metod zdravotnické diagnostiky. Nicméně v současné praxi je sériová produkce odpovídajících zobrazení velmi náročná a drahá, a to zejména z důvodu složitého procesu jejich získávání. Poptávka po vhodných datech velmi výrazně stoupla právě díky vývoji hlubokých neuronových sítí, které (což je ve pravdou v případě většiny úkolů, řešených pomocí strojového učení) zpravidla vyžadují velké objemy vstupních dat. Rozsáhlé výzkumy v poslední době prokázaly, že syntéza zobrazovacích metod může úspěšně zefektivnit klasifikaci, registraci nebo segmentaci při medicinské vizualizaci [27]. Proto zůstává možnost syntetizovat vysoce kvalitní a klinicky významné medicinské obrazy velmi palčivým, leč dosud zdaleka ne dořešeným problémem.

2.1 Stavba oční koule

Oko lze považovat za složitý optický přístroj. Jeho základním účelem je předávat správný obraz zrakovému nervu.



Obrázek 2.1: Stavba oční koule [34]

- **Rohovka** je průhledná vrstva buněk pokrývající přední část oka. Nenalezneme v ní krevní cévy, má velmi vysoký index lomu světla. Je součástí optické soustavy oka. Je ohrazena neprůhlednou vnější vrstvou oka – bělmem [2.1](#).
- **Přední komora oka** je prostor mezi rohovkou a duhovkou, je vyplňena tzv. komorou vodou. Duhovka má tvar kruhu s otvorem uprostřed (zornicí). Skládá se ze svalů, jejichž stažení nebo uvolnění způsobuje změnu velikosti zornice. Je součástí cévnatého obalu oka. Duhovka určuje barvu očí daného jedince (pokud obsahuje málo pigmentových buněk, jsou oči modré, pokud jich obsahuje více, pak hnědé). Plní stejnou funkci jako clona ve fotoaparátu – určuje množství procházejícího světla.
- **Zornice** je otvor v duhovce. Jeho rozměr závisí na množství světla v prostředí – čím více je světla - tím menší je zornice.
- **Čočka** v oku plní stejnou funkci jako čočka ve fotoaparátu. Jedná se o průhledný a elastický orgán, který může měnit tvar a prakticky okamžitě zaostřovat na objekty v dálce nebo nabízko. Nachází se v pouzdře zavřeném na řasnatém tělisku. Stejně jako rohovka je součástí optické soustavy oka.
- **Sklivec** je rosolovité průhledné těleso nacházející se v zadní části oka. Udržuje tvar oka a napomáhá látkové výměně mezi jeho jednotlivými částmi. Je také součástí optické soustavy oka.

- **Sítnice** je tenká vrstva pokrytá fotoreceptory, tedy buňkami citlivými na světlo, a nervy. Receptory se dělí na dva druhy: čípky a tyčinky. V těchto buňkách probíhá fotochemická reakce, umožňující přeměnu světelné energie (fotonů) na energii chemickou, která je dále předávána jako elektrochemický signál nervové soustavě.

Tyčinky, produkující světločivý pigment rodopsin, jsou vysoce citlivé a umožňují tak organismu vidět i za zhoršených světelných podmínek a dále zprostředkovávají takzvané periferní vidění. Čípky, produkující světločivé pigmenty jodopsiny, jsou naopak méně citlivé a vyžadují tak větší množství světla, nicméně díky tomu umožňují detailnější vidění (zprostředkovávají takzvané centrální vidění) a také rozlišování jednotlivých barev. Nejhustěji jsou čípky soustředeny v takzvané žluté skvrně (makule), která odpovídá za nejostřejší část zorného pole. Sítnice přiléhá k střední vrstvě oka ale na mnoha místech jen volně. Právě v těchto místech pak dochází při mnoha chorobách sítnice k jejímu odchlipování.

- **Bělma** je neprůhledná vnější vrstva oční koule, která přechází v přední části oka v průhlednou rohovku. K bělmu je upnuto šest okohybnných svalů. Obsahuje jen malé množství nervových zakončení a krevních cév.
- **Střední vrstva oka také cévnatka, vystýlá zadní část bělma a přiléhá k ní sítnice.** Cévnatka je zodpovědná za krevní oběh nitroočních těles. Při onemocněních sítnice je často zasažena v patologických procesech. V cévnatce nejsou nervová zakončení, a proto její onemocnění nevyvolávají bolest, která by v jiných orgánech mohla člověka upozornit na hrozící nebezpečí.
- **Zrakový nerv** je spojnicí, po níž se signály předávají z nervových zakončení do mozku.

2.2 Patologie oční koule

Doporučení k vyšetření očního dna většinou vydávají pacientovi specializovaní lékaři, jako například kardiologové, neurologové či endokrinologové. Důvodem je fakt, že obraz oblasti očního dna se může měnit při řadě jiných (neočních) patologických stavech, přičemž jejich počet je dokonce vyšší než v případě patologií oftalmologických. Při celé řadě onemocnění se do patologických procesů zapojují cévy v nejrůznějších orgánech: ledvinách, mozku, srdečním svalu atd. Analogické změny mohou proto být pozorovány i v očních cévách, které jsou nedílnou součástí krevního řečiště. Pozorováním očního dna lze učinit nepřímé závěry o stavu cév a mikrocirkulace krve v jiných orgánech. Například lze pozorovat počínající změny na cévách v případech tzv. angiopatie [23], ale také vážnější poruchy jako jsou kornatění cév (angioskleróza) nebo dokonce specifické změny na sítnici (retinopatie 2.2.1), která právě bude předmětem této práce. V závislosti na získaných výsledcích lze činit závěry ohledně závažnosti či stádia průběhu patologického procesu v tom či onom orgánu.

Nejčastěji se se změnami na očním dně setkáváme při hypertonii, diabetu, revmatismu, infekčních onemocněních (např. při toxoplazmóze, infekci HIV), onemocněních krve a při řadě neurologických patologií, revmatologických patologií, zánětech krevních cév, při rizikovém průběhu těhotenství s rozvojem toxikóz a v mnoha dalších případech [23].

2.2.1 Diabetická retinopatie

Projevy diabetické retinopatie zahrnují mikroaneurismata, krvácení do sítnice, exsudace, makulární edém a ischemii, krvácení do sklivce, neovaskularizaci a trakční odchlípení sítnice [28]. V různých stádiích onemocnění se tyto jednotlivé klinické symptomy nemusí projevit. Diagnóza se stanovuje pomocí oftalmoskopie, dodatečné informace lze získat pomocí angiografického vyšetření a optickou koherentní tomografií.

Diabetická retinopatie [17] je nejčastější příčinou slepoty, zvláště pak mezi dospělou populací. Projevení retinopatie závisí na následujících faktorech:

- Doba trvání cukrovky;
- Úroveň glukózy v krvi;
- Úroveň tepenného krevního tlaku.

2.2.2 Neproliferativní retinopatie

Neproliferativní retinopatie [17] se rozvíjí v začátku onemocnění a vede ke zvýšenému průtoku krve kapilárami, vzniku mikroaneurismat, krvácení exsudaci, ischemii a makulárním edémům (zesílení sítnice vlivem vytékání tekutin z kapilár). Nicméně i při probíhající retinopatií si může pacient udržet normální zrak. Častými prvními příznaky neproliferativní retinopatie jsou (viz. 2.2):

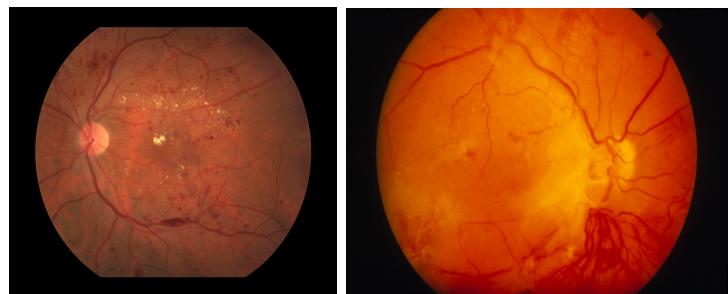
- Kapilární mikroaneurismata;
- Bodové či skvrnové krvácení do sítnice;
- Tvrdé exsudáty;
- Vatovité skvrny (měkké exsudáty).

Tvrdé exsudáty jsou diskrétní žlutá tělíska na sítnici, jsou příznakem chronického edému. Vatovité skvrny jsou oblasti mikroinfarktů ve vrstvě nervových vláken sítnice, které vedou k jejímu zamlžení. Měkké bílé exsudáty mají rozmyté okraje a překrývají hlouběji položené cévy. Během pozdních stádií onemocnění se mohou projevit:

- Makulární edém (diagnostikuje se pomocí biomikroskopie, jedná se o elevaci a zamlžení sítnice);
- Venózní dilatace a intraretinální mikrovaskulární poruchy.

2.2.3 Proliferativní retinopatie

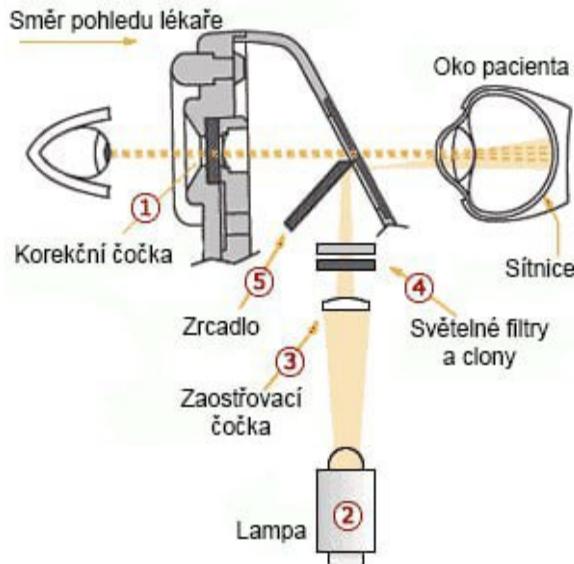
Symptomy mohou zahrnovat nejasné vidění, mušky (černé tečky) nebo mílhavá světýlka (fotopsie) v zorném poli, ale také nečekanou, náhlou a bezbolestnou plnou ztrátu zraku. Tyto symptomy jsou většinou vyvolány krvácením do sklivce nebo trakčním odchlípnutím sítnice. Proliferativní retinopatie se od neproliferativní odlišuje rozvojem preretinální neovaskularizace (viz. 2.2), která je viditelná na povrchu zrakového nervu a sítnice. Při oftalmoskopii je možné nalézt také makulární edém či krvácení do sítnice.



Obrázek 2.2: **Diabetická retinopatie:** A - Neproliferativní retinopatie, B - Proliferativní retinopatie [1]

2.3 Oftalmoskopie

Oftalmoskopie [11] je procedura, při níž lékař zkoumá oční dno pacienta pomocí speciálního přístroje – oftalmoskopu.



Obrázek 2.3: Struktura oftalmoskopu [11]

Vzhledem k tomu, že zakřivení čočky (a tedy i ostrost zraku) je u individuálních pacientů různá, oftalmoskop disponuje sadou korekčních čoček [1]. Čím větším počtem čoček je oftalmoskop vybaven, tím větší je přesnost, s níž může lékař přizpůsobit zařízení individuálním potřebám pacienta. Obraz je pak jasnější a proces diagnostiky je pro pacienta komfortnější. Počet čoček závisí na modelu přístroje a může se pohybovat od 16 do 89, což umožňuje korekci v rozsahu od +40 do -35 dioptrií. Podle číselného označení použitých korekčních čoček lze určit ostrost zraku pacienta. Rozptýlené světlo, které je vydáváno lampou [2] je pomocí spojné čočky soustředěno do paprsku a prochází soustavou apertur (jinak také zvaných clony) a filtrů [4]. V závislosti na konkrétním modelu může jít až o 7 různých apertur a až o 3 různé světelné filtry. V některých modelech přístrojů (Ri-scope L2 a L3) lze aperty a filtry kombinovat. Současné oftalmoskopy jsou schopny zaznamenat obraz očního dna v kvalitním rozlišení.

Kapitola 3

Technologie generativních sítí

V této kapitole se budeme zabývat současným pokrokem v oblasti syntézy lékařských dat, přezkoumáme stávající architektury řešení pro tento typ problémů a jejich nedostatky.

Dvě hlavní výzvy při použití hlubokého učení k vývoji počítačem podporované diagnostiky snímků fundu jsou nedostatek označených dat a právní problémy s ochranou soukromí pacientů. V současné době se vyvíjejí různé snahy o zvýšení množství dat bud rozšířením tréninkových snímků, zvýšením počtu dat, nebo syntézou realisticky vypadajících snímků fundu. Nicméně, rozšíření je omezeno množstvím dostupných dat a neřeší problém ochrany soukromí pacienta. [31]

- **Sběr dat:** Abychom neuronové sítě umožnili naučit se syntetizovat obrazy sítnice, musíme nejdříve nashromáždit dostatečně velký objem dat reálných obrazů sítnice. Je nutné se ujistit, že obrazy mají vysokou kvalitu a široké spektrum různých stavů sítnice.
- **Primární zpracování:** Primární zpracování dat zahrnuje přípravu obrazů ke strojovému učení pomocí oříznutí a změny rozlišení do standardního formátu a, pokud je to možné, také normalizaci pixelizace. Tato příprava zaručuje, že neuronová síť může zpracovávat jednotlivé obrazy efektivně a důsledně.
- **Výběr architektury neuronové sítě:** Existuje značné množství různých architektur neuronových sítí, které by bylo možné využít k syntéze obrazů sítnice. Mezi ty, které jsou v současné době pro podobné účely využívány, patří konvoluční neuronové sítě (CNN) [26] a generativní kompetitivní sítě (GAN) [18]. CNN[26] jsou sítě, které jsou vhodné spíše pro úkol následného zpracování obrazů, zatímco GAN jsou sítě, které vynikají ve vytváření syntetických obrazů, jen obtížně odlišitelných od reálných obrazů.
- **Učení:** Poté, co byla neuronová síť vyprojektována a spuštěna, lze přistoupit k jejímu učení na základě vstupních učebních dat. To zahrnuje vkládání obrazů z učebního okruhu do samotné sítě a následné nastavení jejích parametrů tak, aby případná chybovost mezi předpokládanými a skutečnými výsledky její práce byla co možná nejmenší. Je důležité práci sítě sledovat a srovnávat výsledné obrazy se sadou kontrolních obrazů, abychom se ujistili, že síť se učí efektivně a že se nesnaží přeučit tak, aby výsledky byly pouhými kopiami učebních dat.
- **Syntéza obrazů:** Jakmile se síť naučí požadovaný model, lze ji použít k syntéze nových obrazů sítnice tím, že jí coby vstupní data předložíme obraz zcela náhod-

ného šumu, který je následně pomocí daného modelu přetvořen do obrazu, který připomíná skutečnou sítnici. Konkrétní kroky tohoto procesu závisí na architektuře modelu a konkrétním programu nebo programové knihovně, které jsou k tomuto účelu používány.

3.1 Generativní kompetitivní síť (GAN)

GAN je architektura strojového učení sestávající ze dvou navzájem si konkurenčních neuronových sítí (generátoru a diskriminátora). Generátor („umělec“) se učí vytvářet obrazy, které vypadají jako skutečné, a diskriminátor („kritik“) se učí odlišovat od sebe skutečné obrazy a výtvory generátoru. Obě sítě se učí zároveň, přičemž generátor se snaží vytvořit taková data, která diskriminátor nebude schopen odlišit od skutečných obrazů, zatímco diskriminátor se snaží správně určit, zda je každý jemu předkládaný fragment dat skutečný či generovaný. Proto lze tento proces učení popsat pomocí hry typu MiniMax [8]. Učení je ukončeno ve chvíli, kdy je diskriminátor donucen své rozhodnutí o skutečnosti vstupních dat činit zcela náhodně. Od doby novátorské formulace GANs bylo provedeno značné množství výzkumu toho, jak správně formulovat optimalizační zadání vyváženosti mezi učením generativní sítě G, která vytváří realistické syntetické obrazy, na jedné straně a diskriminativní sítě D, která rozlišuje reálná a syntetická generovaná data, na straně druhé. Funkci ztrát pak lze popsat následujícím způsobem:

$$\min_G \max_D L_{GANS} = E_{x \sim p_{data}(x)}[\log D(x)] + E_{z \sim p_{prior}(z)}[\log(1 - D(z))] \quad (3.1)$$

V oblasti medicínských zobrazovacích metod prokázaly nejlepší výsledky modifikace architektury GAN, jako například Deep Convolutional Generative Adversarial Networks (Hluboké konvoluční generativní kompetitivní sítě, DCGANs) [29] a také Wasserstein GANs (WGANs) [10].

3.2 Ztrátové funkce

Ztrátové funkce v generativních kompetitivních sítích 3.1 hrají klíčovou roli při trénování modelů pro generování obrazu. Používají se k vyhodnocení kvality generovaných obrazů a k řízení trénování sítě.

Kromě dříve popsané ztrátové funkce diskriminátoru 3.1 existují další funkce pro kontrolu kvality výstupních dat generátoru.

3.2.1 Ztráta cyklu[21]

V kontextu architektur generativních adverzních sítí (GAN) [29] určených k provádění úkolů převodu obrazů mezi různými doménami je důležitou součástí hodnocení kvality modelu ztráta cyklu[21]. Účelem ztráty cyklu je udržet konzistence a zachovat informace během procesu překladu obrazu v obou směrech, a tím zmírnit problém ztráty diverzity (zhroucení režimu). Formální definice ztráty cyklu zahrnuje výpočet rozdílu pixelů mezi původními obrazy a jejich syntetickými verzemi po obou fázích generování a zpětného překladu. Matematický vzorec pro cyklickou ztrátu:

$$L_{B \rightarrow A \rightarrow B}^{cycle} = \mathbb{E}_B[\|B - G_{A \rightarrow B}(G_{B \rightarrow A}(B))\|] \quad (3.2)$$

kde:

$L_{B \rightarrow A \rightarrow B}^{cycle}$ - cyklická ztráta při přenosu z domény B do domény A a zpět.

\mathbb{E}_B - matematické očekávání na distribuci obrazů z domény B.

B - původní obrázek z domény B.

$G_{A \rightarrow B}$ $G_{B \rightarrow A}$ - generátory odpovědné za překlad z domény A do domény B a z domény B do domény A.

3.2.2 Ztráta identity

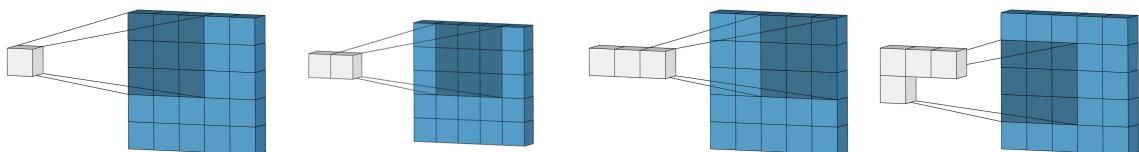
Ztrátová funkce identity se používá v generativních adverzních sítích (GAN)[3.1](#) k zajištění podobnosti rozložení barev vstupních a výstupních obrazů. Jejím účelem je zachovat původní vlastnosti vstupního obrazu poté, co byl transformován generátorem.

$$L_{s \rightarrow t}^{identity} = E_{x \sim X_t} [\|x - G_{s \rightarrow t}(x)\|_1]$$

Účelem ztrátové funkce identity je minimalizovat rozdíl mezi původním a syntetizovaným obrazem, aby se zajistilo, že při převodu obrazu budou zachovány barevné a texturní vlastnosti. To pomáhá zvýšit věrohodnost výsledků a zachovat ve výstupním výsledku důležité detaily a strukturu vstupního obrazu.

3.3 2D Konvoluce

Mechanismy hlubokého učení ve většině případů zahrnují jednu nebo několik konvolučních vrstev [\[30\]](#). Dvourozměrná konvoluce je operací, při níž jádro (čtvercová matice) klouže po vstupních datech a přitom provádí násobení po jednotlivých pixelech a sčítá jednotlivé výsledky do jediného výstupního pixelu.

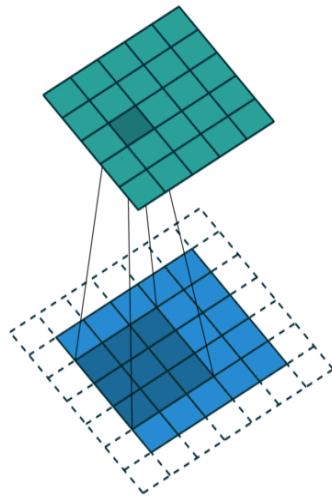


Obrázek 3.1: Princip konvolučního kroku [\[30\]](#)

Nicméně je třeba zmínit, že při provádění základní konvoluce se rozměry vstupních dat zmenšují, což není vždy žádoucí. Střed jádra také neprocesuje okrajové pixely, neboť v takovém případě by okrajová část jádra musela „vystoupit“ za hranici vstupních dat.

3.3.1 Okraje

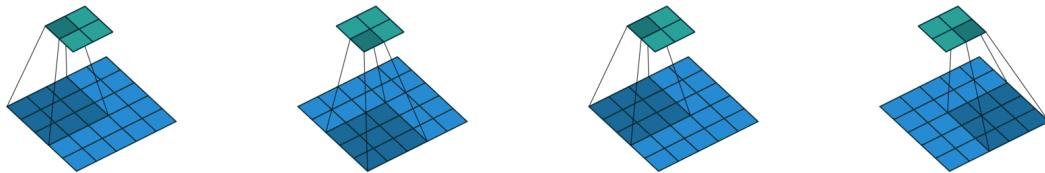
Abychom se vyhnuli nežádoucí kompresi vstupních dat, okraje vstupní matrice jsou vyplněny nulami, přičemž rozměr tohoto okraje (tzv. padding) musí být přímo závislý na rozměrech konvolučního jádra. Díky tomuto opatření pak střed konvolučního jádra vždy projde i nad okrajovými pixely vstupních dat.



Obrázek 3.2: Striding [30]

3.3.2 Krokování (max-pooling)

Často se stává, že při práci s konvoluční vrstvou je nutné získat výstup o menších rozměrech, než jaké mají vstupní data. To je obvykle potřeba v konvolučních neuronových sítích, kde prostorové rozměry klesají s rostoucím počtem kanálů. Při standardním postupu konvoluční jádro klouže po vstupních datech s posunem o jeden pixel. Pokud se krok posunu jádra (striding [30]) zvyšuje, velikost výstupních dat se odpovídajícím způsobem sníží.



Obrázek 3.3: Podrobná vizualizace max-pooling [30]

3.4 Aktivační funkce

Jednou z fází vývoje neuronové sítě je volba aktivační funkce neuronů. Aktivační funkce určuje hodnoty neuronu na základě signálů vysílaných z neuronů jemu předcházejících. Typ aktivační funkce do značné míry určuje funkční možnosti neuronové sítě a také metodu jejího učení. Aktivačních funkcí existuje celá řada, a protože v našem konkrétním případě předpokládáme, že roli generátoru i diskriminátoru budou plnit hluboké konvoluční sítě, svou pozornost zaměříme na funkce Relu 3.4.1 a Softmax 3.4.2.

3.4.1 ReLu (rektifikovaná lineární jednotka)

Výstupní hodnoty každé konvoluční vrstvy u většiny sítí rodiny CNN[26] procházejí přes aktivační funkci ReLU [16]. Funkce je vyjádřena vzorcem:

$$f(x) = \max(0, x) \quad (3.4)$$

Výhody používání ReLU:

- Její derivace je rovna buď jedné nebo nule, a proto zde nemůže docházet k růstu ani poklesu gradientů, protože vynásobením jedničky odchylka chybovosti dostaneme zpět odchylku chybovosti. Pokud bychom použili jinou funkci, například hyperbolickou tečnu, pak by se odchylka chybovosti mohla buď snížit, nebo zvýšit, nebo zůstat stejná, neboť derivace hyperbolické tečny vrací číslo s jiným znaménkem a hodnotou, která může výrazně ovlivnit pokles nebo růst gradientu. Navíc by použití této funkce by vedlo ke snižování hustoty použitých vah.
- Výpočet sigmoidy a hyperbolické tečny vyžaduje provádění operací náročných na zdroje, jako je umocňování, zatímco ReLU lze implementovat pomocí jednoduché prahové transformace aktivační matice na nulu.

3.4.2 Funkce Softmax

Konečnou funkcí architektur CNN [26] pro klasifikační úkoly je funkce Softmax, která převádí konečné hodnoty výstupních neuronů na pravděpodobnost shody s tou či jinou třídou. Součet hodnot všech výstupních neuronů po provedení Softmax je tedy 1 nebo 100%. Pro výpočet pravděpodobnosti každé jednotlivé třídy se číslo e umocňuje hodnotou výstupního neuronu a vydělí se součtem e umocněného na ai všech výstupních neuronů. Vzorec:

$$\sigma(\vec{z}_i) = \frac{e^{z_i}}{\sum_1^K e^{z_i}}$$

Funkce Softmax je užitečná v případech, kdy existují více než dvě možné teoretické třídy objektů. V našem případě je úkolem diskriminátoru rozlišit syntetický obraz od skutečného, takže existují dvě výstupní hodnoty (třídy).

3.5 Zadání klasifikace. Diskriminátor

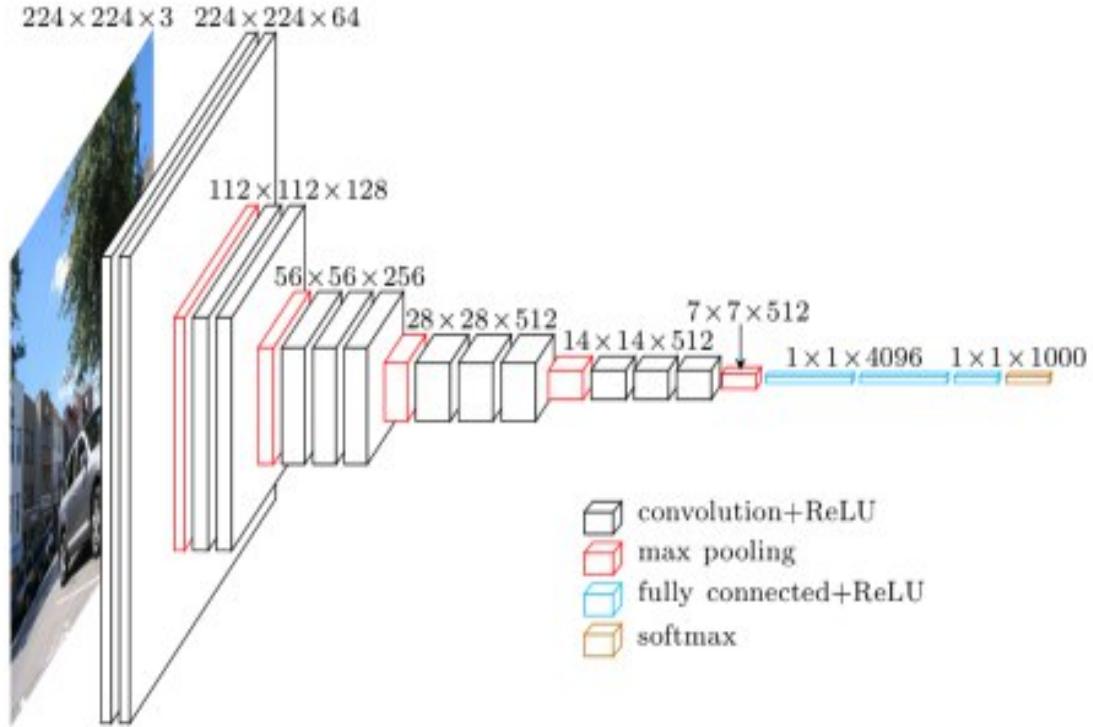
Diskriminátor je hluboká neuronová síť, která řeší úkol klasifikace vstupních dat. V našem případě se diskriminátor snaží odlišit od sebe výsledky práce generátoru od originálních výukových dat. Existuje několik modelů hlubokých neuronových sítí schopných řešit klasifikační úkoly. Většina z nich je modifikací struktury CNN[26] navržené Oxfordskou univerzitou v roce 2014 [32].

3.5.1 VGG-16

Architektura VGG-16 je popsána na obrázku níže 3.4.

Jako vstup pro vrstvu conv1 slouží RGB obrazy o rozměrech 224×224 , takže vstupní data představují tenzor s parametry $224 \times 224 \times 3$. V našem případě výsledné rozměry

obrazu činí 3500×2200 pixelů ve třech jednotlivých kanálech. Dále pak obrazy prochází přes řadu konvolučních vrstev, v nichž jsou použity filtry s velmi nízkým receptivním polem o rozměrech 3×3 (což je nejmenší velikost filtru pro získání představy o orientaci v prostoru – kde je nahore/dole, vlevo/vpravo a uprostřed). Konvoluční krok je fixován na hodnotě 1



Obrázek 3.4: Architektura VGG-16 [6]

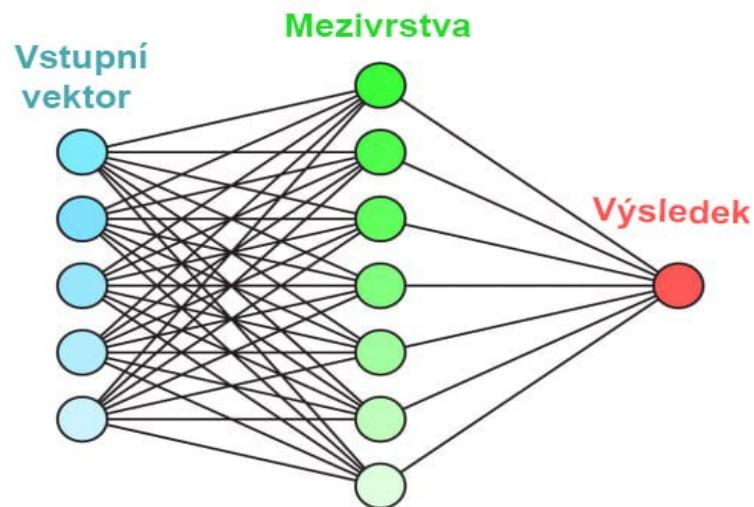
pixelu. Výplň prostoru (padding) vstupních dat konvoluční vrstvy je volen tak, aby bylo rozlišení výstupu zachováno i po konvoluci, tj. výplň při konvoluční vrstve 3×3 je rovna 1. Prostorové sdružování (pooling) se provádí pomocí pěti max-poolingových 3.3.2 vrstev, které následují po jedné z konvolučních vrstev (ne všechny konvoluční vrstvy musí nutně následovat max-poolingu). Operace max-poolingu se provádí v mřížce o rozměrech 2×2 pixely s krokem 2.

Po sérii konvolučních vrstev (která může mít v různých architekturách různou délku) následují tři plně propojené vrstvy: první dvě z nich mají 4096 kanálů, třetí pak 1000 kanálů. Poslední kanály jsou pak výstupním vektorem odpověďí při procesu klasifikace. Jelikož v našem modelu může architektura VGG-16 plnit předpokládanou funkci diskriminátoru, závěrečná vrstva síťe bude mít v našem konkrétním případě dva výstupní kanály.

Posledním krokem je vrstva soft-max. Konfigurace plně propojených vrstev je stejná ve všech neuronových sítích. Všechny skryté vrstvy jsou vybaveny ReLU 3.4.1. Jelikož vstupní obraz má vysoké rozlišení, počet konvolučních vrstev potřebných k získání požadované přesnosti musí být také vysoko nad běžným standardem. Dostatečně hluboké konvoluční sítě mají významný nedostatek – mají problém s nedoučováním, respektive s mizejícím gradiensem [33]. V souvislosti s tím byl navržen model „zbytkového učení“ (residual learning) 3.7.

3.6 Fully connected layer. Umělá inteligence

V modelech konvolučních neuronových sítí(CNN), které se zaměřují na úkoly klasifikace, je závěrečnou fází jedna (nebo více) vrstev úplného připojení, což je tradiční neuronová síť umělé inteligence .



Obrázek 3.5: Neuronová síť s vrstvami úplného připojení [19]

Poté, co vstupní obraz velikosti NxN Pixelů prochází řadou konvolučních vrstev jsou vstupy převedeny na vektor velikosti Mx1, což je ve své podstatě vstupní vektor hodnot pro neuronovou síť plného připojení . Výsledkem práce vrstev plného připojení bude neuron s náhodnou hodnotou na intervalu (0,1) a jednoduchá aktivační funkce ReLu s prahovou hodnotou 0.5. Pro realizaci architektury může být jedním z možných řešení metoda lineární regrese .

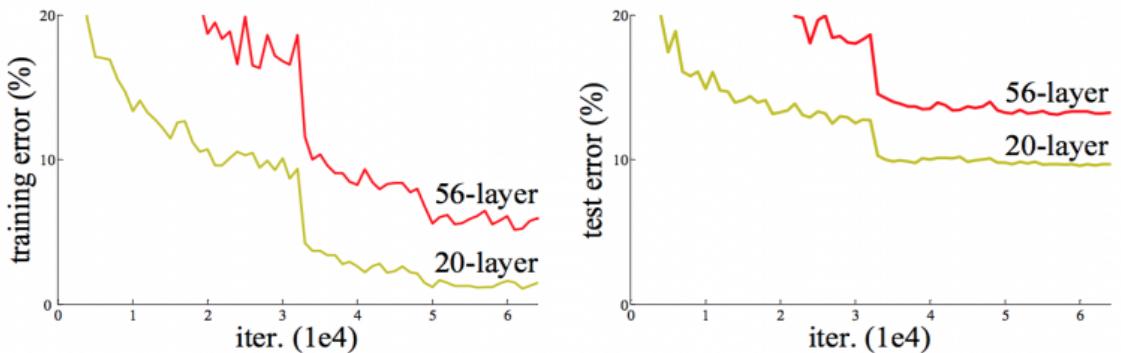
3.7 ResNet (34,50,101)

Pokud dostatečně hluboká síť začne konvolovat, vzniká problém. S rostoucí hloubkou sítě se přesnost zpočátku zvyšuje a následně prudce klesá. Propad přesnosti učení nám ukazuje, že ne všechny sítě lze snadno optimalizovat.

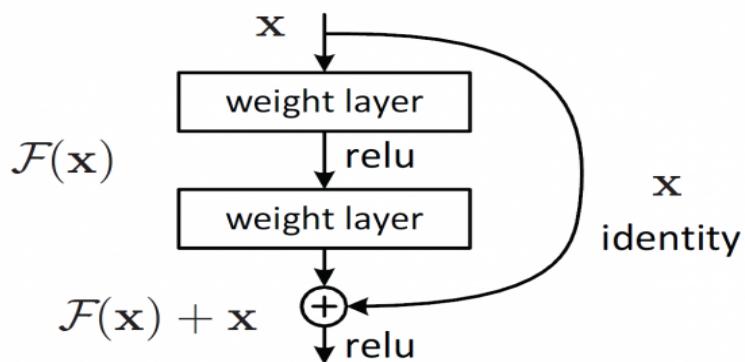
Aby byl tento problém překonán, společnost Microsoft zavedla hlubokou „zbytkovou“ strukturu učení. Místo toho, aby se síť spoléhala na to, že výsledek práce daného počtu skládaných vrstev vždy odpovídá základní představě, nechává tyto vrstvy přímo odpovídat „zbytku“. Vzorec $F(x) + x$ lze implementovat pomocí neuronových sítí se zkratkovými spojeními.

Použití ResNet může vyřešit celou řadu problémů, například:

- ResNet lze poměrně snadno optimalizovat. „Jednoduché“ sítě, které pouze skládají vrstvy, vykazují s rostoucí hloubkou vyšší chybovost učení [20].
- ResNet umožňuje poměrně snadno zvýšit přesnost pouhým navýšením hloubky, čehož jiné sítě dosahují jen s obtížemi.



Obrázek 3.6: Chyba školení (vlevo) a chyba testu (vpravo) s 20-vrstvými a 56-vrstvými "jednoduchými" sítěmi. Hlubší síť má větší trénovací chybu, a tedy chybu testování. [20]

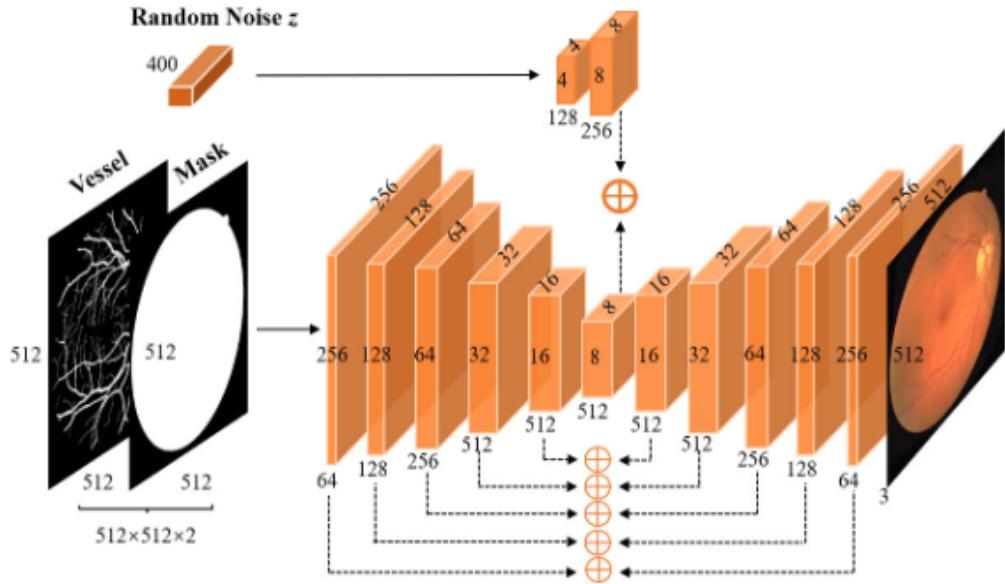


Obrázek 3.7: Identity block [20]

3.8 Generátor

Architektura generátoru se skládá ze dvou částí - kodéru a dekodéru, přičemž dekodér i kodér musí mít stejný počet vrstev, které si v párech navzájem odpovídají. Kodér je v podstatě síť rodiny CNN[26] a představuje několik konvolučních vrstev, jejichž vstupními daty jsou předem vygenerovaný cévní strom o rozměrech 512×512 pixelů a maska pozadí oční bulvy o stejných rozměrech. Tento vstup kodéru tak představuje trojrozměrnou matrice (tenzor) o rozměrech $512 \times 512 \times 2$. Výsledkem po zpracování pomocí konvolučních vrstev je pak tenzor o rozměrech $8 \times 8 \times 512$, u kterého práce kodéru končí. Třetím vstupním parametrem generátoru je bílý šum, což je vektor o velikosti 400. Vstupní vektor je nejprve převeden na trojrozměrný tenzor o rozměrech $4 \times 4 \times 128$, načež prochází plně propojenou vrstvou (fully connected layer) 3.6, čímž vzniká tenzor o rozměrech $8 \times 8 \times 256$. Poté se provede zřetězení mezi výstupním tenzorem operace kodéru a tenzorem bílého šumu. Výsledkem této operace je tenzor o rozměrech $8 \times 8 \times 768$, který bude vstupním parametrem dekodéru.

Dekodér funguje na principu dekonvoluce, přičemž po každé dekonvoluční vrstvě se výstupní hodnoty sečtou s výstupními hodnotami odpovídající vrstvy architektury kodéru. Je důležité zmínit, že hodnoty okrajů (padding) 3.3.1 a krokování (striding) 3.3.2 odpovídajících vrstev musí být stejné, jinak dekodér obrázek interpretuje nesprávně. Všechny vrstvy kodéru i dekodéru, s výjimkou finální, jsou vybaveny ReLu 3.4.1.



Obrázek 3.8: Architektura generátoru [24]

Pro generování vaskulárního stromu a masky oční bulvy se většinou používají mechanismy rodiny FCN 3.6.

3.9 Normalizace

V našem řešení se normalizační vrstvy stávají součástí dekodéru generátoru. Poté, co vstupní data (např. maska fundu a strom cév) projdou několika konvolučními vrstvami 3.3.2 a v kodéru projdou procesem komprese, se dekodér pokusí obnovit původní data pomocí dříve popsaného procesu dekonvoluce. Data je pak třeba normalizovat, aby se obnovila část ztracených informací, k nimž došlo v důsledku konvoluce.

Normalizaci mezidat lze provést několika způsoby: dávkou (batch), instancí (instance) a vrstvou (layer). Hlavní rozdíl mezi těmito normalizacemi spočívá ve způsobu, jakým se s daty zachází a jak je normalizují. Dávková normalizace posuzuje data v každé dárce společně, průměruje je a normalizuje. Normalizace podle instancí se zabývá daty v každém jednotlivém vzorku, což je užitečné v případech, kdy se statistiky jednotlivých vzorků mohou velmi lišit. A konečně normalizace podle vrstev se aplikuje na každý rys (např. kanál v konvoluční vrstvě) nezávisle na ostatních, takže je flexibilnější a přizpůsobivější různým rysům ve vrstvě.

Normalizace v hlubokých neuronových sítích má několik cílů, včetně:

1. Stabilizace učení

Normalizace pomáhá snížit vnitřní posun ((internal covariate shift)) během tréninku. Vnitřní kovariance vzniká, když se rozložení aktivací na vrstvách během tréninku mění, což může zpomalit konvergenci. Normalizace pomáhá udržovat stabilní rozložení aktivací, což urychluje učení.

2. Problém gradientního blednutí

Normalizace může v hlubokých sítích zmírnit problém blednutí nebo explodujících gradientů. To umožňuje efektivnější využití optimalizačních technik, jako je sestup po gradientu.

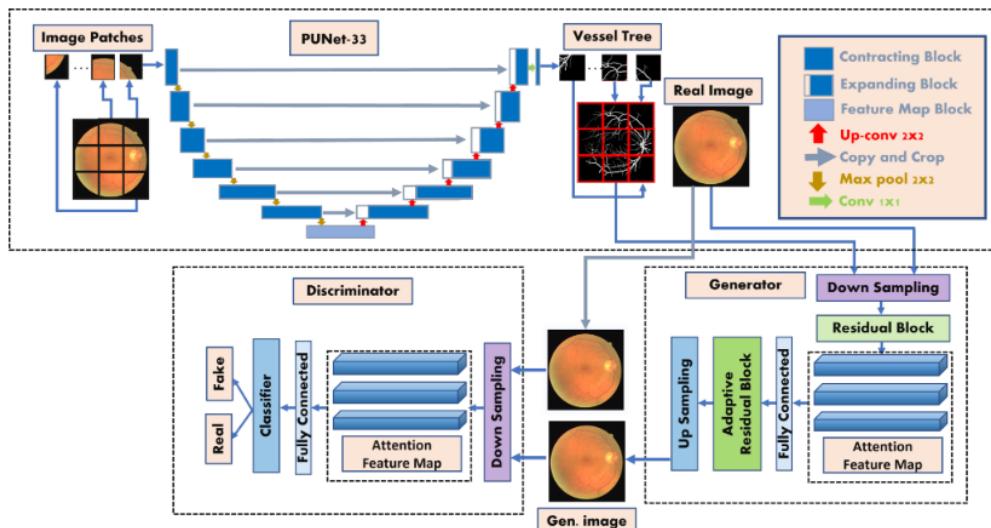
3. Lepší zobecnění

Normalizace může fungovat jako regulátor, který pomáhá zabránit nadměrnému přizpůsobení a zvýšit obecnost modelu. Pomocí normalizace lze použít vyšší míru učení bez rizika neudržitelného učení

Kapitola 4

Úspěchy v oblasti

V červenci 2022 na 44. konferenci IEEE Engineering in Medicine Biology Society (EMBC) představili Dereje Shenkut a Vijayakumar Bhagavatula dvoustupňový algoritmus pro generování realistických snímků očního pozadí [31].



Obrázek 4.1: Dvoustupňová generativně-adverzní síť (GAN), známá jako "Fundus GAN"[31], určená k syntéze obrazů sítnice, která pomáhá trénovat klasifikátory obrazů sítnice. [25]

Architektura "Fundus GAN"[31] se skládá ze dvou fází. V první fázi je původní obraz přiveden na vstup předtrénovaného modelu . Výsledkem modelu je binární obraz cévního stromu. Autoři článku zvolili jako model pro řešení segmentace v tomto problému P-U-net(33), což je modifikace původní architektury U-net[25] představené v roce 2015.

Po dokončení první fáze procesu v architektuře Fundus GAN máme k dispozici dataset páru obrazů, kde každý pár se skládá z původního obrazu a odpovídajícího binárního obrazu cévního stromu získaného pomocí modelu P-Unet33. Tento dataset je následně využit ve druhé fázi, která se zaměřuje na trénování generátoru pro překlad obrazu na obraz (image-to-image translation). V této fázi autoři čerpají inspiraci z úspěchu modelu U-GAT-

IT [22], což je pokročilá technika generativních adversariálních sítí 3.1 určená k překladu mezi doménami obrazů. U-GAT-IT [22] demonstruje vynikající schopnost v překladu stylů a detailů mezi obrazy, což autoři "Fundus GAN" [31] využívají pro efektivní transformaci původních fundus obrazů na vylepšené verze s dobře definovanými cévními strukturami. Tento přístup umožňuje dosáhnout detailnějších a kvalitnějších výsledků, což je klíčové pro úspěšné trénování a vylepšení přesnosti klasifikátorů obrazů sítnice.

4.1 U-net

Architektura "U-Net" [25], navržená v roce 2015 pro segmentaci biomedicínských snímků, se stala revoluční díky své schopnosti přesně identifikovat objekty zájmu na snímku, a to i za přítomnosti malého množství trénovacích dat. Tato architektura se skládá ze dvou hlavních částí: zužující (kódování) a rozšiřující (dekódování)

1. Kódovací část (zúžení)

Kódovací část architektury U-Net [25] je navržena tak, aby postupně zužovala prostorové rozměry obrazu a zároveň zvyšovala hloubku (počet kanálů) map prvků. Toho je dosaženo střídáním konvolučních vrstev a vrstev s maximálním sdružováním. Každý blok v této části obsahuje:

- Dvě konvoluční vrstvy, z nichž každá používá konvoluční operaci k extrakci rysů z obrazu. Po konvoluci obvykle následuje aktivační funkce, například ReLU 3.4.1.
- Vrstvu max-pooling 3.3.2, která následuje po dvou konvolučních vrstvách a slouží ke zmenšení prostorové velikosti map příznaků, což umožňuje síti zaměřit se na obecnější příznaky.

2. Dekódovací část (rozšíření)

Dekódovací část odráží část kódovací tím, že postupně zvětšuje prostorovou velikost map prvků a snižuje jejich hloubku. Toho se dosahuje pomocí operací převzorkování (nebo zpětné konvoluce) a konkatenace s mapami prvků z kódovací části prostřednictvím průchozích vazeb. Každý blok obsahuje:

- Operaci převzorkování pro zvětšení velikosti mapy prvků. Konkatenaci zvětšené mapy prvků s odpovídající mapou prvků z kódovací části, což umožňuje obnovit prostorovou informaci ztracenou během procesu zužování.
- Dvě konvoluční vrstvy, které následují po konkatenaci a slouží k další extrakci prvků ze sloučených dat.

3. Finální vrstva

Na konci dekódovací části je závěrečná konvoluční vrstva, která počet kanálů mapy příznaků vyrovnává s počtem segmentačních tříd. Zde se k provedení této úlohy obvykle používá konvoluce 1x1.

4. Zvláštnosti

Propustná komunikace (Skip Connections)

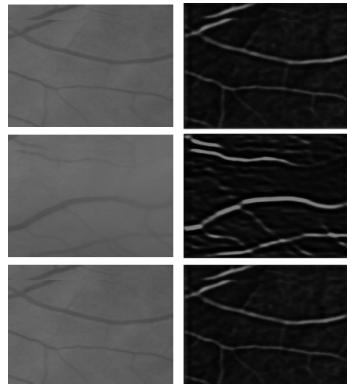
Propustná komunikace umožňuje přenáset mapy prvků přímo z kódovací části do dekódovací části, což pomáhá obnovit prostorové informace ztracené během procesu zužování v důsledku operací max-pooling. To má zásadní význam pro úlohy vyžadující přesnou lokalizaci při segmentaci, protože poskytuje podrobnější rekonstrukci výstupního obrazu.

4.1.1 P-Unet(33)

P-UNet, vylepšená verze U-Net[25], je klíčovým prvkem architektury "Fundus GAN"[31] pro segmentaci cévních stromů na snímcích sítnice. Hlavním rozdílem mezi P-UNet a původní U-Net je zvýšený počet konvolučních vrstev (33 vrstev) a zpracování obrazu na úrovni políček. To umožňuje podrobnou analýzu velkoformátových obrazů jejich rozdelením na menší části a trénováním na těchto fragmentech.

Při zpracování velkoformátových obrazů v architektuře P-UNet se používá metoda rozřezání obrazu na políčka (úseky) o velikosti 256x256 pixelů. Hlavní pozornost je věnována skutečnosti, že při krájení obrazů s výškou nebo šírkou, které nejsou dělitelné 256, může dojít ke ztrátě některých informací, což má za následek černé pruhy na okrajích políček. Tyto černé pruhy mohou signalizovat ztrátu informací a následně vést k nesprávnému výkonu modelu, protože model může tyto černé oblasti interpretovat jako součást dat, což negativně ovlivní kvalitu segmentace nebo klasifikace. Je proto důležité zajistit, aby proces rozřezání obrazu na políčka minimalizoval ztrátu informací a nevnášel zkreslení, která by mohla ovlivnit trénování a přesnost modelu P-UNet.

Zrcadlení se při zpracování obrazu, zejména při práci s políčky, obvykle používá k odstranění artefaktů na okrajích obrazu, pokud velikost obrazu neumožňuje vybrat políčka přesně definované velikosti bez ztráty informace. Pokud velikost obrazu není násobkem velikosti políček, mohou se na okrajích vyskytovat oblasti, které nejsou zcela pokryty políčky. V takových případech zrcadlení sousedních pixelů do chybějící oblasti zachovává kontextové informace a zabraňuje vzniku černých pruhů, které by síť mohla vnímat jako součást dat, což by negativně ovlivnilo proces učení a přesnost modelu.



Obrázek 4.2: Vstupní a výstupní políčka modelu P-Unet v průběhu 100 epoch.

4.2 U-GAT-IT

Architektura U-GAT-IT [22] představuje průlom v bezdozorovém převodu obrázků, který efektivně kombinuje hluboké učení a počítačové vidění pro transformaci obrázků mezi různými domény. Díky integraci pozornostního modulu a adaptivní normalizace AdaLIN[22] je model schopen flexibilně reagovat na různé styly a textury, čímž zvyšuje realističnost a detailnost převedených obrázků. Tato technologie otevírá nové možnosti v oblasti převodu obrázků a nabízí silný nástroj pro vývojáře a vědecké pracovníky.

Architektura U-GAT-IT [22] využívá šest neuronových sítí k jemnějšímu řízení procesu překladu obrazu a zlepšení kvality výsledků. Dvě z těchto sítí jsou generátory (jedna pro překlad z domény A do domény B a jedna pro zpětný překlad), které vytvářejí přeložené obrazy. Čtyři diskriminátory (dva pro každou doménu: jeden pro globální strukturu obrazu a jeden pro lokální detaily) umožňují přesnější posouzení realističnosti generovaných obrazů, což zlepšuje učení a zvyšuje kvalitu výsledných překladů.

Generátor v architektuře U-GAT-IT[22] je podobný klasickým generativním adversárním sítím (GAN3.1), ale má důležitou inovaci. Po procesu kódování jsou výsledkem aktivační mapy, které pak procházejí plně propojenou vrstvou. Cílem této vrstvy je extrahovat důležité detaily pro následné dekódování. V tomto procesu vrací plně propojená vrstva dva parametry: γ a β . Dekodér pak pomocí AdaLINova vzorce dynamicky vybírá typ normalizace pro zpracování dat. To umožňuje generátoru přizpůsobit se různým datům a zdůraznit důležité detaily domény.

Vzorec pro výpočet výběru:

$$\text{AdaLIN}(a, \gamma, \beta) = \gamma \cdot (\rho \cdot \hat{a}^I + (1 - \rho) \cdot \hat{a}^L) + \beta,$$

kde:

$$\hat{a}^I = \sqrt{\frac{a - \mu_I}{\sigma_I^2 + \epsilon}}, \quad \hat{a}^L = \sqrt{\frac{a - \mu_L}{\sigma_L^2 + \epsilon}},$$

$$\rho \leftarrow \text{clip}[0, 1](\rho - \tau \Delta \rho),$$

$\mu_I, \sigma_I, \mu_L, \sigma_L$ představují průměr a směrodatnou odchylku napříč kanály a vrstvami.

V pokročilých architekturách se často používají typy adaptivní normalizace, protože volba vhodného typu normalizace je často ponechána na experimentech. V takových případech mohou algoritmy kontrolovat průměr a směrodatnou odchylku parametrů aktivačních map dekodéru s odpovídající velikostí map kodéru napříč vrstvami a napříč instancemi. Na základě získaných hodnot může algoritmus pomocí klipsu určit, ke kterému typu normalizace se má v konkrétním případě uchýlit.

4.2.1 Funkce ztrát

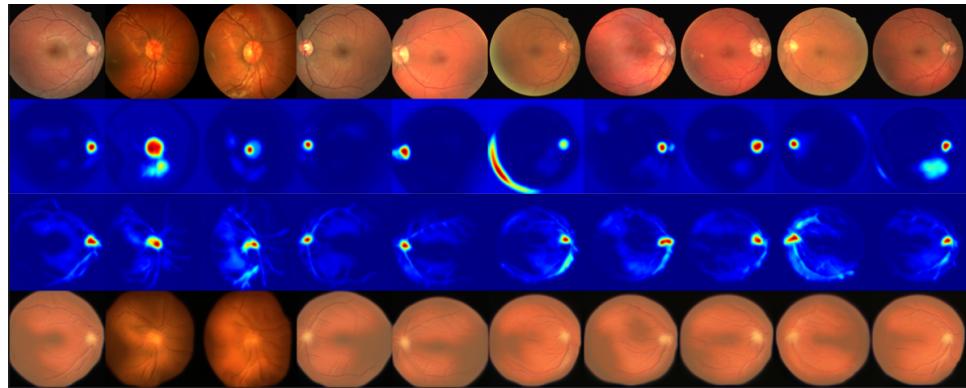
V naší architektuře je hlavní metrikou kvality soubor dříve popsaných chybových funkcí (ztráta identity3.2.2, ztráta cyklu3.2.1 a chyba diskriminátoru3.1). Zvláštní pozornost je však věnována metrice tepelné mapy, která je zodpovědná za důležité detaily obrazu, jako je umístění makuly, světelný odlesk a další. V kontextu naší úlohy je zásadní, aby se generátor, který převádí cévní strom na fundus, naučil rozpoznávat polohu, velikost a barvu zrakového nervu, protože to je klíčová vlastnost fundu, která hraje důležitou roli při diagnostice mnoha onemocnění. Tyto detaily lze sledovat v mapách pozornosti 4.3.

Matematický popis funkce "CAM-loss"[14] pro generátor a diskriminátor:

$$L_{s \rightarrow t}^{cam} = -(\mathbb{E}_{x \sim X_s}[\log(\eta_s(x))] + \mathbb{E}_{x \sim X_t}[\log(1 - \eta_s(x))]), \quad (4.2)$$

$$L_{Dt}^{cam} = \mathbb{E}_{x \sim X_t}[(\eta Dt(x))^2] + \mathbb{E}_{x \sim X_s}[(1 - \eta Dt(G_{s \rightarrow t}(x)))^2] \quad (4.3)$$

Matematicky se ztrátová funkce CAM podobá již popsané ztrátové funkci cyklu 3.2.1. V případě chyby CAM však výslednou chybu nepočítáme z konečného výsledku, ale z



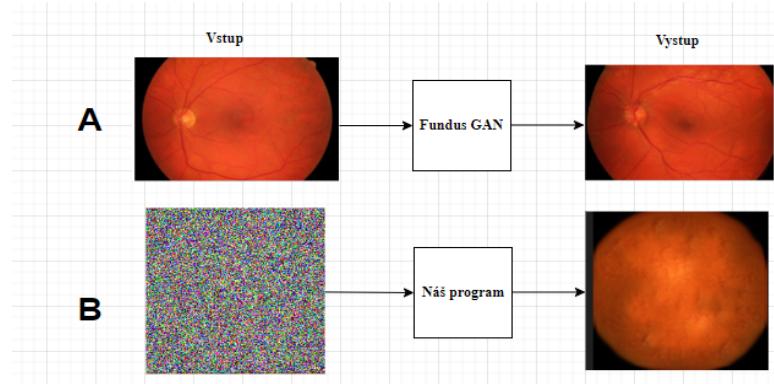
Obrázek 4.3: Původní a syntetická data, a mapy pozornosti diskriminátora k nim.

tepelných map, na které se diskriminátor orientuje. To umožnuje generátoru rychleji si uvědomit, kterým oblastem obrazu je vhodné věnovat při generování větší pozornost.

Výsledná metrika kvality je tedy kombinací generátorů, diskriminátorů a pomocných klasifikátorů vážených koeficienty λ_1 , λ_2 , λ_3 a λ_4 .

4.3 Výzvy Fundus GAN

Architektura "Fundus GAN"^[31] je účinným nástrojem pro vytváření různých stylistických obrazů očního fundu. Je však třeba upozornit na dvě hlavní nevýhody tohoto přístupu. Za prvé, generátor "Fundus GAN"^[31] používá pro generování obrazu strom cév odvozený od p-unetu. To má za následek minimální anatomické rozdíly mezi syntetickými a původními obrazy, což může snižovat jejich diagnostickou hodnotu. Zadruhé, navrhovaná architektura plně neřeší otázku ochrany osobních údajů, protože ke generování syntetických obrazů je třeba použít původní lékařské údaje. Tato omezení zdůrazňují potřebu najít nové metody, které by tyto problémy účinně řešily a zlepšily kvalitu syntetických obrazů fundu.



Obrázek 4.4: A) Fundus GAN [31], B) Upravený U-GAT-IT

Kapitola 5

Návrh vlastního algoritmu

Hlavním cílem této práce je zopakovat úspěch dosažený pomocí architektury "Fundus GAN"[\[31\]](#) při generování různých stylistických obrazů očního pozadí. Současně je záměrem provést významná vylepšení stávající architektury U-GAT-IT [\[22\]](#) a přizpůsobit ji tak, aby generátor odpovědný za generování cév nespoléhal na předem segmentovaná data, ale učil se z bílého šumu. Tento inovativní přístup umožní generátoru postupně generovat strukturu cév z abstraktních náhodných dat namísto použití přesných anatomických obrazů z původních snímků. Tato modifikace systému U-GAT-IT otevírá perspektivu vytváření unikátních syntetických snímků fundu a zároveň zlepšuje jejich diagnostickou hodnotu a poskytuje další výhody pro lékařské soukromí.

5.0.1 Datové sady

Trénování hlubokých neuronových sítí je náročný úkol, který vyžaduje značné výpočetní zdroje. Pro trénování generátoru převodu mezi obrazy používáme tři datové sady obsahující celkem 355 párů obrázků (fundus a cévní strom). Tyto datasety jsou DRIVE [\[12\]](#), obsahující 40 párů obrázků, "High-Resolution Fundus" HRF[\[4\]](#), obsahující 15 párů obrázků, a FIVES.rar[\[2\]](#), který obsahuje zbyvajících 300 párů obrázků.

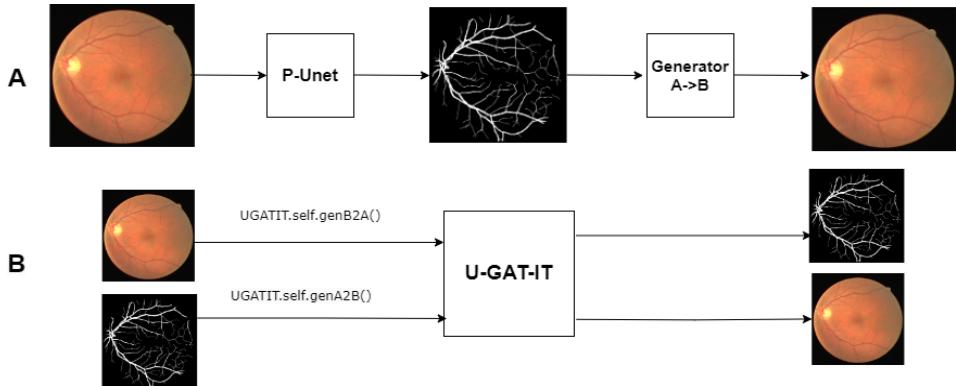
Vzhledem k tomu, že vývojáři architektury U-GAT-IT [\[22\]](#) kladou důraz na kvalitu překladu stylů a detailů ve výsledné doméně, může nastat určitá obtíž při překladu tvaru fundu, jako je například absolutní kruh. Proto budou všechny obrázky předem zpracovány, pečlivě vycentrovány a zmenšeny na celkovou velikost 256 x 256 pixelů. Tento předzpracovaný formát zajišťuje, že síť bude moci lépe zachytit potřebné detaily a styly při trénování na těchto datech.

Je důležité poznámenat, že výsledný dataset obsahuje rozmanité množství originálních medicínských dat z různých zdrojů a pacientů různého věku. Kromě toho tento dataset zahrnuje také určité množství patologických vzorků, jako jsou exudáty a retinopatie, což poskytuje širokou škálu variant pro trénování sítě.

5.0.2 Plán řešení

Náš soubor dat obsahuje 350 párů snímků ze dvou oblastí: strom cév (oblast A) a původní fundus (oblast B). Vývojáři použili kombinaci segmentátoru cév (P-Unet) a generátoru k implementaci generativní architektury "Fundus GAN"[\[31\]](#). Na rozdíl od původního U-GAT-IT[\[22\]](#), který se skládá ze dvou generátorů pro převod obrazů z domény A do domény B

a zpět, architektura "Fundus GAN"^[31] vyžaduje jako vstupní data původní lékařská data, aby bylo možné generovat vysoce kvalitní obraz.



Obrázek 5.1: A) Fundus GAN [31], B) Upravený U-GAT-IT[22]

V našem přístupu předpokládáme použití analogie U-net^[25] jako druhého generátoru U-GAT-IT^[22], který bude převádět obrazy z domény B do domény A. Na rozdíl od standardní architektury však plánujeme druhý generátor trénovat postupně přidáváním šumu do obrazů domény B a jejich přiváděním na vstup sítě. Tento proces bude pokračovat, dokud druhý generátor nebude schopen generovat strom nádob z čistě bílého šumu.

Nakonec budou výsledná data přivedena na vstup druhého generátoru, který bude schopen generovat syntetický fundus z cév.

5.1 Implementace

V průběhu této studie bylo pro provedení úlohy generování fundusu vybráno architektura "U-GAT-IT"^[22]. Klíčovým hlediskem při rozhodování o volbě této architektury byla existence licence MIT ^[7], která poskytuje svobodu v používání, úpravách a distribuci softwaru. Tato volba přináší možnost přizpůsobení "U-GAT-IT"^[22] našim specifickým potřebám, distribuce a používání v souladu s podmínkami uvedenými v licenci. Tím je zajištěna flexibilita a svoboda v rámci našeho projektu.

Hlavní části implementace jsou realizovány za pomoci framework pro strojové učení PyTorch verze 2.1.0 ^[9]. Trénování modelu probíhá na lokálním zařízení s grafickým procesorem (CUDA) GTX 1660 Super 8 GB.

Vývojáři architektury "U-GAT-IT"^[22] zdůraznili, že k dosažení kvalitních výsledků je třeba provést milion iterací trénování modelu. Během procesu trénování byla rychlosť učení nastavena na (10^{-4}) . Vzhledem k tomu, že grafický procesor má 8 gigabajtů videopaměti, je maximální velikost dávky, kterou lze přenést do neuronové sítě, omezena na 1. Překročení této hodnoty vede k přetečení videopaměti, což způsobí pád stroje.

Při této konfiguraci trvá trénování 20 000 iterací nebo 200 epoch přibližně 12 hodin. Konečný výpočet ukazuje, že k dokončení jednoho milionu iterací je zapotřebí přibližně 600 hodin povinného tréninku.

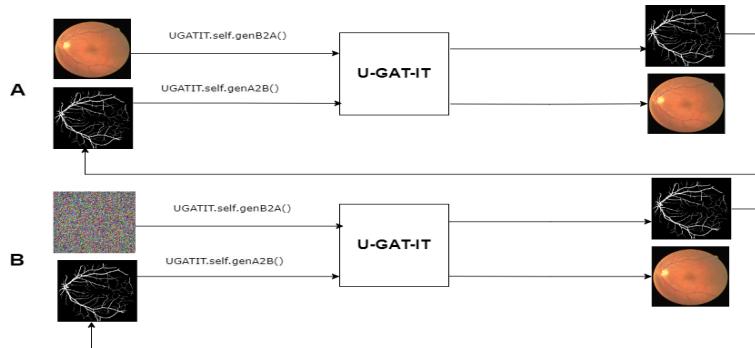
5.2 Struktura programu

Celá architektura se skládá z několika souborů:

- **main.py** - Vstupní bod programu . Obsahuje analyzátor uživatelských argumentů a volání hlavních metod modelu.
- **networks.py** - Soubor obsahující implementace hlavních bloků architektury . ResNet^{3.7}, blok adaptivní normalizace instancí^{4.2} , generátor ^{3.8} , diskriminátor a další.
- **UGATIT.py** - Soubor obsahující třídu architektury UGATIT se sadou atributů předaných uživatelem a základními metodami (trénování, načítání vah, testy atd.).
- **dataset.py** - Tento soubor poskytuje modulární nástroje pro načítání a předzpracování sad obrazových dat.
- **utils.py** - obsahuje další funkce pro práci s daty a jejich typy napsané pomocí modulu souborového systému OS, rámce strojového učení PyTorch a modulu Numpy pro práci s tenzory a vektory.
- **bp_bot.py** - obsahuje skript se serverovým řešením založeným na populárním messengeru pro pohodlí uživatelů
- Složka **datasets** obsahuje uspořádání složek **trainA** , **trainB**, **testA**, **testB** obsahující tréninková (350 snímků) a validační (20 snímků) data fundu a cévního stromu.

5.3 Trénování neuronové sítě

Pro trénování našeho modelu jsme provedli úpravy dat předávaných do tréninkových smyček třídou iterátorů domény B. V našem případě se jedná o část datové sady obsahující původní snímky . Model bude trénován každý den po dobu 12 hodin pomocí 20000 iterací. Každých 4000 iterací přidáme ke vstupním datům šum pomocí vrstvy rámce "PyTorch"^[9]. V průběhu jednoho dne tedy budeme naši architekturu trénovat tak, že jí budeme dodávat původní data a přijímat od ní syntetická data, což ztělesňuje myšlenku uvedenou v Shenkutově článku. Zároveň budeme trénovat druhý generátor reagovat na různé typy dat, především na "bílý šum", což je hlavním cílem našeho výzkumu.

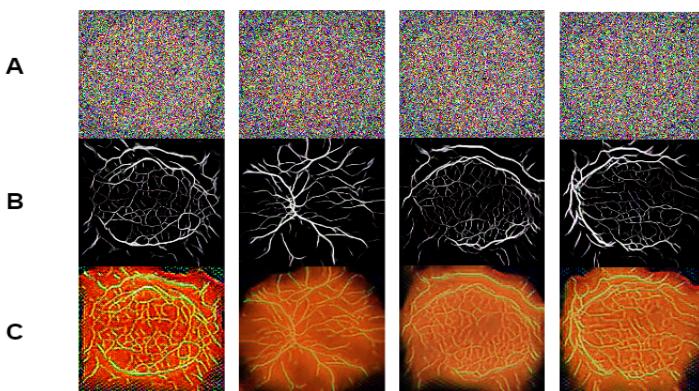


Obrázek 5.2: A) Generování syntetických dat ze skutečných lékařských dat , B) Generování syntetických dat z bílého šumu

Je třeba poznamenat, že tento přístup výrazně ovlivní výpočetní zdroje a dobu trénování celé architektury, protože složitost úlohy pro síť v takové situaci mnohonásobně vzroste.

5.4 Testování a validace

V procesu trénování naší sítě provedeme testování a validaci modelu, abychom vyhodnotili jeho výkonnost a kvalitu práce. Jak již bylo zmíněno, budeme postupně upravovat kód programu a parametry sítě, abychom zlepšili její výsledky. Testovací množina se bude skládat z 25 mapování obsahujících data ze všech datových sad ve správném poměru k trénovacím datům. Důležitou metrikou kvality bude chybovost sestavená podle výše popsaného vzorce, který zahrnuje soubor chyb, jako je ztráta cyklu, ztráta identity, chyba diskriminátoru a ztráta CAM [14]. V tomto případě bude hlavní složkou metrika CAM[14], jejíž koeficient při výpočtu hlavní pokuty je výchozí hodnotou 1000, zatímco pro ztrátu cyklu 3.2.1 a ztrátu identity 3.2.2 se použijí koeficienty 10 a pro diskriminátor 3.1 1. Po prvních 10000 iteracích (nebo 2 dnech tréninku na našem zařízení) jsme získali následující výsledek 6.1.



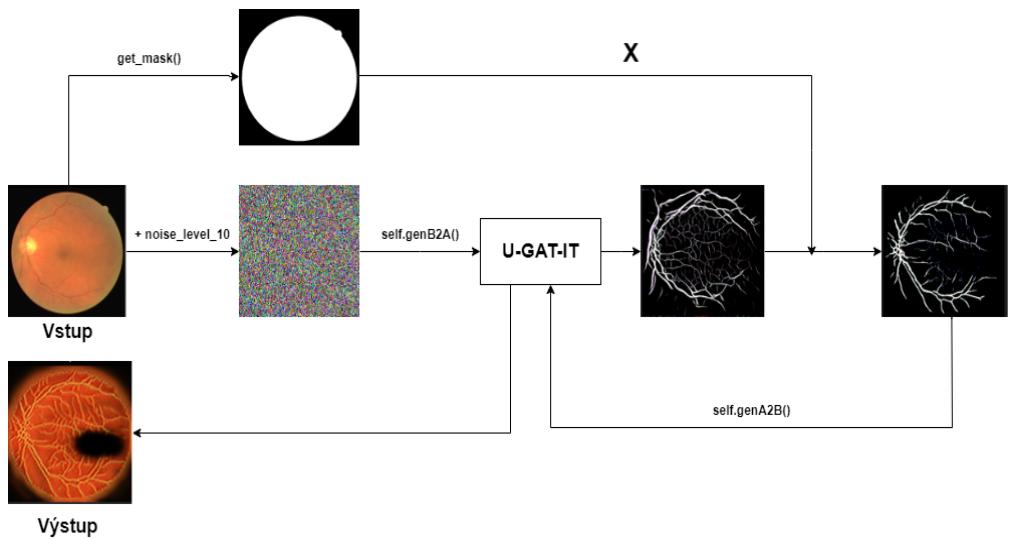
Obrázek 5.3: A - Vstupní data. B - Syntetický strom cév. C - Syntetický obraz fundu.

První výsledky již vzdáleně připomínají fundus, ale je zde řada problémů. Jak uvádějí vývojáři, architektura má potíže s generováním tvarů, jako jsou kruhy nebo čtverce. Pro naši úlohu je to kritické, protože obraz fundu má tvar kruhu. Problém je také částečně způsoben nedostatkem vstupních dat pro první generátor. V klasické Schenkuntové architektuře byl strom cév segmentován z původních lékařských dat, a proto si zachovával tvar kruhu. V našem řešení se však snažíme vygenerovat jedinečný cévní strom z čistého bílého šumu a udržení striktního tvaru syntetických dat je časově náročné na trénování generátoru složeného ze 6 vrstev.

Tyto problémy jsme vyřešili následujícím způsobem 5.4. Pomocí knihovny OpenCV[5] jsme pro každý vstupní snímek vytvořili masku, přičemž jsme všechny pixely matice obrazu, které nejsou nulové (černé), nahradili jedničkami. Výsledkem jsou binární data obsahující striktní obrys budoucího fundusového kruhu. Poté provedeme násobení matic mezi syntetickým nepravidelně tvarovaným stromem cév a binární maskou. Výsledkem maticového násobení je, že všechny pixely mimo obvod kruhu binární masky se změní na nulu. Obraz správného tvaru získaný jako výsledek těchto operací se předá na vstup druhého generátoru.

Po provedených úpravách budeme pokračovat v tréninku dalších 10 000 iterací a vyhodnotíme nárůst kvality pomocí dříve uvedeného vzorce pro výpočet chyby. Toto vyhodnocení nám umožní zjistit, jak účinné byly naše úpravy a nakolik dokázaly zlepšit kvalitu syntetizovaných snímků fundu.

Po provedených úpravách lze pozorovat výrazné zvýšení přesnosti sítě 6.2, a to přibližně o 40% ve srovnání s předchozími výsledky. Navzdory zlepšením však stále existují určité nedostatky. Obraz nyní správně zachovává tvar fundu, ale vzhledem k různorodosti datové



Obrázek 5.4: Algoritmus pro zpracování vstupních a výstupních dat

sady je pro generátor stále obtížné přesně určit, kde se nacházejí pixely pozadí, což může vést k určitým nedokonalostem syntetizovaných obrazů 5.4.

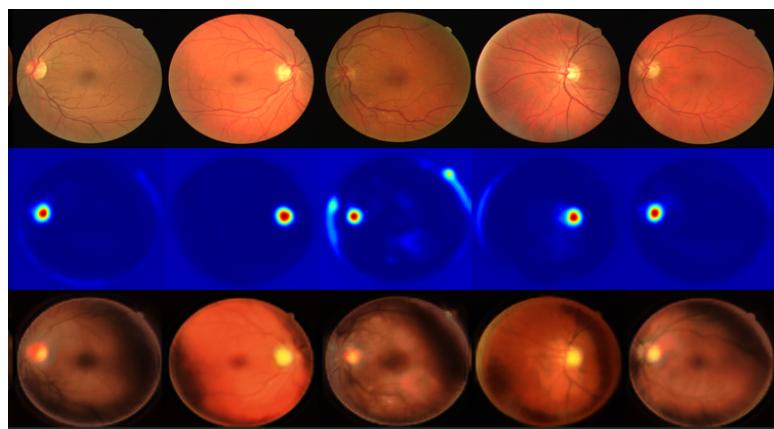
Kapitola 6

Experimenty a analýza výsledků

Celkem bylo provedeno přibližně 300 000 iterací k natrénování generátoru pro syntézu obrazů sítnice ze dvou různých typů vstupních dat. První přístup byl založen na použití původních dat z fundu, zatímco druhý využíval bílý šum. Každý z těchto přístupů má své vlastní výhody a nevýhody.

6.1 Syntéza z původních dat

Přístup se syntézou nových dat z původních lékařských dat, který popsal Shenkut [31], má ze všech tří navržených technik nejlepší matematický výkon. Tento přístup má však některé nevýhody. Model má stále omezené anatomické rozdíly se skutečnými daty. Vyházením různých trénovacích dat však může generátor potenciálně změnit stylistiku snímků, například stíny a barvy. Díky tomu může vytvářet jedinečné a poměrně realistické obrazy, které jsou podobné skutečným datům.



Obrázek 6.1: Původní vstupní data a syntetický snímek

Původní data na obrázku jsou součástí datové sady Drive, ale většinu dat při tréninku obsahovala datová sada FIVE [2]. Za zmínku stojí, že přibližně 25% dat obsahovalo patologické případy 2.2. Je možné, že některé vlastnosti patologických odlehlych dat se mohou odrážet v syntetických datech, což je činí cennějšími pro trénování budoucích architektur souvisejících s hodnocením kvality nebo klasifikací patologických odlehlych dat, což je hlavním cílem studované oblasti.

Z nevýhod je třeba uvést, že pokud jde o anatomickou strukturu zrakového nervu a cévního stromu, generátor kopíruje vstupní data.

Podle metrik kvality navržených tvůrci architektury má model dobrý výkon, ale takové srovnání není zcela korektní, protože předmětem zkoumání tvůrců byl překlad domény A - B (kůň na zebru, fotografie na portrét atd.), my syntetizujeme na principu A - A, úloha je mnohem jednodušší, ale proces trénování nám zabral více než třikrát méně času, než je popsáno v architektuře (300 000 vs. 1 000 000 iterací).

Tabulka 6.1: Kernel Inception Distance $\times 100 \pm \text{std.} \times 100$ pro režim rozdílového převodu obrazu. Nižší hodnota je lepší.

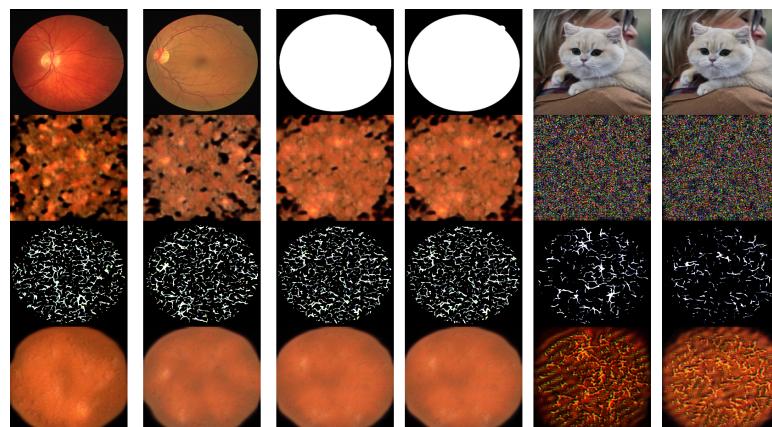
	selfie2anime	photo2portrait	fundus2fundus
U-GAT-IT	11.61 \pm 0.57	4.28 \pm 0.33	0.1206 \pm 0.04

6.2 Syntéza z bílého šumu

Druhým krokem byla analýza interního algoritmu, který pracoval nezávisle na formátu vstupních dat. Cílem bylo vycvičit generátor A2B k syntéze binárního stromu cév z bílého šumu, ze kterého mohl generátor B2A získat obraz fundu. Lze však pozorovat, že syntetický obraz se fundusu podobá jen vzdáleně. Mezi hlavní problémy patří nedostatek výpočetních zdrojů. Úkol vycvičit generátor o šesti vrstvách, aby vytvořil obraz 256×256 z ničeho, se již stává výpočetně náročným.

Dvěma klíčovými výzvami jsou nedostatek paměti RAM, která by umožnila zvýšit počet vrstev v generátoru a diskriminátoru, a také by umožnila natrénuvat generátor pro velkoformátové obrazy (3500×3500). Druhým problémem je nedostatek paměti grafického procesoru, který přímo ovlivňuje velikost datového paketu odesílaného do GPU během trénování, a tedy i rychlosť trénování.

Stojí za zmínku, že místo bílého šumu lze použít dříve získanou binární masku 5.4 . V důsledku toho bude obraz schopen zachovat správný tvar, ale kvůli nedotrénuvání generátoru A2B nebude obsahovat anatomicky správnou strukturu (cévy, zrakový nerv atd.). Technicky je síť schopna vygenerovat obraz z čehokoli, dokonce i z kočky 6.2.



Obrázek 6.2: Generování z náhodných snímků

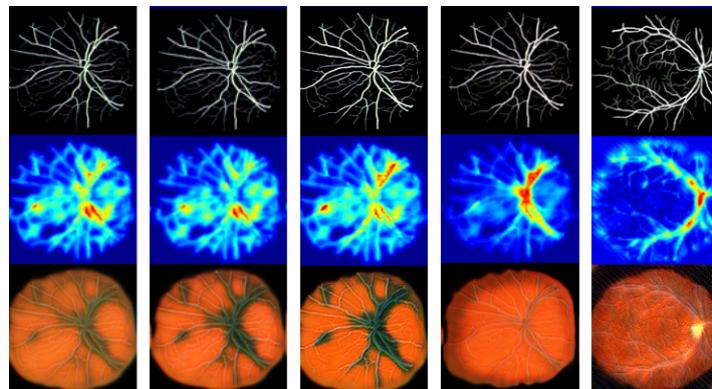
6.3 Analýza výsledků

Výsledky byly zaznamenávány denně, počínaje 50 000 iteracemi. Hlavním kritériem pro matematické vyhodnocení byla penalizace vypočtená podle dříve popsaného vzorce. Výpočet jsme provedli pro dva scénáře: když generátor obdrží jako vstupní data originální lékařské údaje a když jsou vstupní data bílým šumem.

Výběr objektivní metriky pro hodnocení kvality algoritmu syntézy dat je poměrně obtížný úkol. Standardní metriky, jako je přesnost(Accuracy [13]) a střední kvadratická chyba(MSE [13]), hodnotí pouze to, jak dobře syntetizovaná data kopírují vstupní data. Tento přístup však není vždy správný. Na příkladu obrázku výše 6.1 vidíme, že syntetizovaná data se od vstupních dat liší barvami a stíny, ale to neznamená, že výsledná data jsou nekvalitní. Proto jsme se pro vyhodnocení rozhodli vynechat všechny epochy a vypočítat průměrnou pokutu, která byla síti uložena během trénování. Tato pokuta se skládá z diskriminačních skóre, jako jsou CAM-loss^{4.2.1} a Cycle-loss^{3.2.1}, a také z dalších dříve popsaných metrik kvality.

Tabulka 6.2: Full loss = $1\lambda_1 + 10\lambda_2 + 10\lambda_3 + 1000\lambda_4$. [4.2.1](#)

	Chyba A->A	Chyba A->noise->B->A
50000 iterací	0.2339	4.0049
70000 iterací	0.2330	3.9974
90000 iterací	0.1606	3.9914
110000 iterací	0.1600	3.9872
130000 iterací	0.1780	3.9877
150000 iterací	0.1325	3.9995
170000 iterací	0.1833	3.9862
190000 iterací	0.1398	3.9866



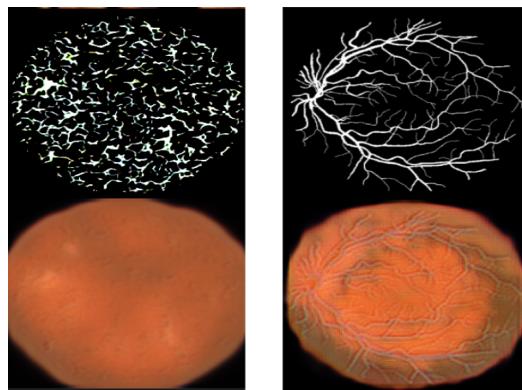
Obrázek 6.3: Výsledky generátoru A2B při 50,70,90,150,170 tisících iterací

Při analýze metrik lze pozorovat, že metoda generování pomocí "Fundus GAN"^[31] dosáhla lepších výsledků ve srovnání s naším přístupem. Celkový trend ukazuje pokles chyby generátoru, což naznačuje postupné zlepšování. V některých oblastech je však pozorováno prudké zhoršení výkonu. To naznačuje, že diskriminátor odhalil v aktivačních mapách nové rysy, které odlišují původní data od syntetických. V takových situacích se musí generátor přizpůsobit a najít nové strategie ke zlepšení kvality dat.

Díky zpětné vazbě z diskriminátoru se tak generátor A2B dokázal naučit přizpůsobit cévní strom a syntetizovat zrakový nerv na anatomicky správném místě [6.3](#).

6.4 Zjištěné problémy

Zatímco generátor A2B (strom cév do fundu) si vedl dobře, generátor B2A (fundus + šum - cévy) si vedl špatně. Jedním z hlavních problémů je zde pomalá změna celkové chyby architektury se vstupním bílým šumem. Nárůst kvality je extrémně pomalý, pouze tisícinu procenta za den. Výsledkem je, že strom cév neodpovídá očekávanému výsledku, což následně ovlivňuje kvalitu syntetického fundu.



Obrázek 6.4: A - Skutečný výsledek celé námi navržené architektury . B - Výsledky generátoru A2B se vstupní původní anotací cévního stromu

Je zřejmé, že generátor B2A je zjevně nedoškolen. 300 000 epoch pro architekturu se čtyřmi hlubokými sítěmi při náhodných vstupních datech je velmi málo.

Pro budoucí výzkumy v této oblasti je třeba bud hledat nový přístup k generování cévního stromu z bílého šumu, nebo využít naší myšlenku a zaměřit se na doškolení generátoru cév, což by mohlo vést k výraznému zlepšení výsledků [6.4](#).

Kapitola 7

Závěr

Cílem této práce bylo prostudovat a otestovat stávající algoritmy pro syntézu obrazu sítnice. Zhodnotili jsme nedávné pokroky v této oblasti, zopakovali předchozí výzkum a představili nové myšlenky. Studovali jsme algoritmy, implementovali je a trénovali na různých souborech dat a matematicky vyhodnotili výsledky pomocí základních metrik pro úlohy převodu obrazu na obraz. Přitom jsme identifikovali problémy a navrhli jejich řešení.

Výsledkem naší práce je archiv obsahující kód programu, složky s kolekcemi trénovacích epoch, z nichž každá obsahuje váhy modelu, a složku s testovacími obrázky vytvořenými během každé epochy.

Mezi hlavní problémy, které jsme identifikovali, patří výpočetní složitost úlohy a nedostatek výpočetních zdrojů. Paralelní trénování více algoritmů vyžaduje značné množství paměti GPU. Kromě toho se jako důležitý aspekt ukazuje také správný výběr a zpracování dat, protože nevyvážená data mohou vést k nepřesným výsledkům.

Navzdory těmto výzvám naše výsledky ukazují, že cíle bylo dosaženo. Navržený algoritmus úspěšně syntetizuje obrazy očního fundu jak z původních dat a dat s anotacemi, tak z náhodných dat.

Pro pohodlnost uživatelů a inženýrů, kteří se zajímají o tvorbu syntetických dat pomocí našeho řešení, jsme integrovali naši vyvinutou síť do bota pro jednoho z populárních open-source messengerů - Telegram.

Uživatelé tak mohou snadno získávat výsledky generace obrázků bez potřeby klonování a nasazení rozsáhlých dat, souborů, knihoven a vah modelu na svém lokálním počítači.

Návod k použití:

- Následujte odkaz na robota[3] a autorizujte se v aplikaci Telegram.
- Pošlete libovolná grafická data (.jpg, .png atd.), robot je převede na obrázek 256X256, aplikuje bílý šum, převede je na binární strom cév a následně na syntetický obraz fundu.

Literatura

- [1] *Diagnostika a laserová léčba diabetické makulopatie.* Tvrdé exsudáty.
- [2] *FIVES A Fundus Image Dataset for AI-based Vessel Segmentation.rar* [online]. Dostupné z: https://figshare.com/articles/figure/FIVES_A_Fundus_Image_Dataset_for_AI-based_Vessel_Segmentation/19688169.
- [3] *Fundus Generator VUT.* Dostupné z: t.me/fundus_gen_VUT_bot.
- [4] *High-Resolution Fundus (HRF) Image Database* [online]. Dostupné z: <https://www5.cs.fau.de/research/data/fundus-images>.
- [5] *OpenCV 4.9.0.80.* Dostupné z: https://kingmed.info/knigi/Oftalmologiya/book_2182/Oftalmoskopiya-Astahov_YuS_Dal_NYu_-2011-pdf.
- [6] *Understanding VGG16: Concepts, Architecture, and Performance* [online]. Dostupné z: <https://datagen.tech/guides/computer-vision/vgg16/>.
- [7] *Licence MIT* [online]. 2023. Dostupné z: https://ru.wikipedia.org/wiki/%D0%9B%D0%B8%D1%86%D0%B5%D0%BD%D0%B7%D0%B8%D1%8F_MIT.
- [8] *Understanding GAN Loss Functions.* 2023. Dostupné z: <https://neptune.ai/blog/gan-loss-functions>.
- [9] *PyTorch.* 2024. Dostupné z: <https://pytorch.org/>.
- [10] ARJOVSKY, M., CHINTALA, S. a BOTTOU, L. *Wasserstein GAN.* Courant Institute of Mathematical Sciences, 2017. Dostupné z: <https://arxiv.org/pdf/1701.07875.pdf>.
- [11] ASTACHOV a DAL'. *Oftalmoskopie* [online]. 2011.
- [12] BANERJEE, C. *Accuracy score and MSE and loss function* [online]. 2023. Dostupné z: <https://medium.com/@chandradip93/accuracy-score-and-mse-and-loss-function-23cabf70e2c4>.
- [13] BANERJEE, C. *Accuracy score and MSE and loss function* [online]. 2023. Dostupné z: <https://medium.com/@chandradip93/accuracy-score-and-mse-and-loss-function-23cabf70e2c4>.
- [14] CHAOFEI WANG, Y. H. Q. Y. S. S. G. H. *CAM-loss: Towards Learning Spatially Discriminative Feature Representations.* Department of Automation, Tsinghua University, 2022. Dostupné z: <https://arxiv.org/pdf/2109.01359.pdf>.

- [15] COSTA, P., GALDRAN, A., MEYER, M. I., NIEMEIJER, M., ABR'AMOFF, M. et al. *End-to-end Adversarial Retinal Image Synthesis*. journal, 2017 [cit. OI 10.1109/TMI.2017.2759102, IEEE Transactions on Medical Imaging].
- [16] FRED, A. *Deep Learning using Rectified Linear Units (ReLU)* [online]. 2019. Dostupné z: <https://arxiv.org/pdf/1803.08375.pdf>.
- [17] GESUNDHEIT.GV.AT. *Diabetická retinopatie*. Národní zdravotnický informační portál, 2018. Dostupné z: <https://www.nzip.cz/clanek/378-diabeticka-retinopatie>.
- [18] GOODFELLOW, I. J., POUGET ABADIE, J., MIRZA, M., XU, B., WARDE FARLEY, D. et al. *Generative Adversarial Nets* [online]. Departement d'informatique et de recherche op 'erationnelle ' Universite de Montr 'eal ' Montreal, QC H3C 3J7, 2019 [cit. 2019-10-02]. Dostupné z: <https://arxiv.org/pdf/1406.2661.pdf>.
- [19] HALLMAN, J. *A comparative study on Linear Regression and Neural Networks for estimating order quantities of powder blends*. DEGREE PROJECT IN COMPUTER SCIENCE AND ENGINEERING, SECOND CYCLE, STOCKHOLM, SWEDEN 2019. Dostupné z: <http://www.diva-portal.org/smash/get/diva2:1383464/FULLTEXT01.pdf>.
- [20] HE, K., ZHANG, X., REN, S. a SUN, J. *Deep Residual Learning for Image Recognition*. Microsoft Research, 2015. Dostupné z: <https://arxiv.org/pdf/1512.03385.pdf>.
- [21] JUN YAN ZHU, P. I. A. A. E. *Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks* [online]. 2017.
- [22] JUNHO KIM1, M. K. H. K. L. *U-GAT-IT: UNSUPERVISED GENERATIVE ATTENTIONAL NETWORKS WITH ADAPTIVE LAYER INSTANCE NORMALIZATION FOR IMAGE-TO-IMAGE TRANSLATION* [online]. 2020 [cit. Published as a conference paper at ICLR 2020]. Dostupné z: <https://arxiv.org/pdf/1907.10830.pdf>.
- [23] KOPAEVA. *Oční nemoci. Učebnice*. Kapitola 21 Změny ve zrakovém orgánu u obecných nemocí. Výuková literatura pro studenty lékařských vysokých škol., 2018. Dostupné z: <https://doi.org/10.25276/978-5-903624-36-2>.
- [24] NAN LIANG, X. W. H. X. a WANG, J. *End-To-End Retina Image Synthesis Based on CGAN Using Class Feature Loss and Improved Retinal Detail Loss*. IEEE, 2022. Dostupné z: <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9849644>.
- [25] OLAF RONNEBERGER, T. B. *U-Net: Convolutional Networks for Biomedical Image Segmentation* [online]. 2015 [cit. ISBI challenge for segmentation of neuronal structures in electron microscopic stacks]. Dostupné z: <https://arxiv.org/abs/1505.04597>.
- [26] O'SHEA, K. a NASH, R. *An Introduction to Convolutional Neural Networks* [online]. Department of Computer Science, Aberystwyth University, Ceredigion, SY23 3DB, 2015. Dostupné z: <https://arxiv.org/abs/1511.08458>.
- [27] PREIM, B. a BOTHA, C. *Visual Computing for Medicine*. 2014. Dostupné z: <https://www.sciencedirect.com/book/9780124158733/visual-computing-for-medicine>.

- [28] R., L. *Caplan's Stroke*. 4. vyd. Beth Israel Deaconess Medical Center, 2009. Dostupné z: <https://www.sciencedirect.com/topics/medicine-and-dentistry/fundus-eye>.
- [29] RADFORD, A., METZ, L. a CHINTALA, . *Supervised representation learning with deep convolutional generative adversarial networks*. arXiv:1511.06434, 2015.
- [30] SHAFKAT, I. *Intuitively Understanding Convolutions for Deep Learning* [online]. Dostupné z: <https://towardsdatascience.com/intuitively-understanding-convolutions-for-deep-learning-1f6f42faee1#:~:text=The%202D%20convolution%20is%20a,into%20a%20single%20output%20pixel>.
- [31] SHENKUT, D. a BHAGAVATULA, V. *Fundus GAN - GAN-based Fundus Image Synthesis for Training Retinal Image Classifiers*. 44th Annual International Conference of the IEEE Engineering in Medicine Biology Society (EMBC)., 2022. Dostupné z: <https://ieeexplore.ieee.org/document/9871771>.
- [32] SIMONYAN, K. a ZISSERMAN, A. *VERY DEEP CONVOLUTIONAL NETWORKS FOR LARGE-SCALE IMAGE RECOGNITION* [Published as a conference paper at ICLR 2015]. Visual Geometry Group, Department of Engineering Science, University of Oxford, 2015. Dostupné z: <https://arxiv.org/pdf/1409.1556.pdf>.
- [33] WANG, C.-F. *The Vanishing Gradient Problem* [online]. 2019. Dostupné z: <https://towardsdatascience.com/the-vanishing-gradient-problem-69bf08b15484>.
- [34] ÚZIS, M. Z. *Oči* [online]. 2009–2023.