



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV BIOMEDICÍNSKÉHO INŽENÝRSTVÍ

DEPARTMENT OF BIOMEDICAL ENGINEERING

METODY STROJOVÉHO UČENÍ HLUBOKÝCH NEURONOVÝCH SÍTÍ S OMEZENÝMI DATASETY

DEEP NEURAL NETWORK LEARNING METHODS WITH LIMITED DATASETS

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. Filip Németh

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Roman Jakubíček, Ph.D.

BRNO 2024



Diplomová práce

magisterský navazující studijní program **Bioinženýrství**

Ústav biomedicínského inženýrství

Student: Bc. Filip Németh

ID: 220488

Ročník: 2

Akademický rok: 2023/24

NÁZEV TÉMATU:

Metody strojového učení hlubokých neuronových sítí s omezenými datasety

POKYNY PRO VYPRACOVÁNÍ:

1) Prostudujte oblast strojového učení se zaměřením na hluboké neuronové sítě, možnosti učení a problémy vyvstávající z omezené velikosti trénovacího datasetu. 2) Proveďte průzkum a vypracujte rešerši o možnostech učení sítí na omezeném trénovacím datasetu a jejich vliv na úspěšnost. 3) Navrhněte neuronovou síť pro praktický experiment umožňující testování vybraných metod, a připravte vhodnou databázi z veřejně dostupných obrazů. 4) Připravte programové skripty pro práci se sadou obrazů pro navržené experimenty simulující omezení trénovací a validační množiny a proveďte standardní učení navrženého modelu na plném a omezeném datasetu. 5) Realizujte vybrané metody augmentace či metody učení s omezenými datasety; dle potřeby modifikujte. Navrhněte vhodnou metodiku vyhodnocení. 6) Proveďte srovnání implementovaných přístupů. Vyhodnoťte a diskutujte výsledky dosažené v experimentech a z nich vyvoďte závěry.

DOPORUČENÁ LITERATURA:

[1] SHURRAB, Saeed a DUWAIRI, Rehab, 2022. Self-supervised learning methods and applications in medical imaging analysis: a survey. Online. PeerJ Computer Science. Roč. 8. ISSN 2376-5992. Dostupné z: <https://doi.org/10.7717/peerj-cs.1045>. [cit. 2023-08-14].

[2] ZHANG, Man; ZHOU, Yong; ZHAO, Jiaqi; MAN, Yiyun; LIU, Bing et al., 2020. A survey of semi- and weakly supervised semantic segmentation of images. Online. Artificial Intelligence Review. Roč. 53, č. 6, s. 4259-4288. ISSN 0269-2821. Dostupné z: <https://doi.org/10.1007/s10462-019-09792-7>. [cit. 2023-08-14].

Termín zadání: 5.2.2024

Termín odevzdání: 22.5.2024

Vedoucí práce: Ing. Roman Jakubíček, Ph.D.

doc. Ing. Radim Kolář, Ph.D.
předseda rady studijního programu

UPOZORNĚNÍ:

Autor diplomové práce nesmí při vytváření diplomové práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

ABSTRAKT

Diplomová práca sa zameriava na preskúmanie efektívnosti hlbokých neurónových sietí pri spracovaní obrazových dát s obmedzeným množstvom tréningových dát. V rámci práce sa analyzovali vplyvy rôznych techník a prístupov na učenie týchto sietí, vrátane transferového učenia, dátovej augmentácie a metódy prenosu neurónového štýlu. Výsledky experimentov naznačujú, že transferové učenie s využitím pretrénovaných váh z veľkých datasetov ako ImageNet je efektívne pri zlepšovaní výsledkov na obmedzených dátach, dosahujúc vysoké F1-skóre. Využitie rozličných foriem dátovej augmentácie môže viesť k variabilným výsledkom, kde poskytuje rôzne výhody a nevýhody, ktoré majú významný vplyv na úspešnosť a efektívnosť modelov. Metóda využívajúca sieť na prenos neurónového štýlu nepriniesla vo všeobecnosti výraznejšie zlepšenie a hlavne sa ukázala ako menej účinná v prípade datasetu s veľkou diverzitou perspektív a geometrických vlastností.

KLÚČOVÉ SLOVÁ

hlboké učenie, obmedzený dataset, klasifikácia, transferové učenie, dátová augmentácia, prenos neurónového štýlu

ABSTRACT

The master thesis aims to investigate the effectiveness of deep neural networks in image processing with limited training data. As part of the work, the effects of various techniques and approaches on the learning of these networks were analyzed, including transfer learning, data augmentation, and neural style transfer method. Experimental results suggest that transfer learning using pre-trained weights from large datasets such as ImageNet is effective in improving results on limited data, achieving high F1-scores. The use of different forms of data augmentation can lead to variable results, where it provides different advantages and disadvantages that have a significant impact on the success and efficiency of the models. In general, the method using a neural style transfer network does not yield significant improvements and proved less effective for dataset with a large diversity of perspectives and geometric features.

KEYWORDS

deep learning, limited dataset, classification, transfer learning, data augmentation, neural style transfer

NÉMETH, Filip. *Metody strojového učení hlubokých neuronových sítí s omezenými datasy*. Diplomová práce. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav biomedicínského inženýrství, 2024. Vedúci práce: Ing. Roman Jakubíček, Ph.D.

Vyhlásenie autora o pôvodnosti diela

Meno a priezvisko autora: Bc. Filip Németh
VUT ID autora: 220488
Typ práce: Diplomová práca
Akademický rok: 2023/24
Téma záverečnej práce: Metody strojového učení hlubokých neuronových sítí s omezenými datasety

Vyhlasujem, že svoju záverečnú prácu som vypracoval samostatne pod vedením vedúcej/cého záverečnej práce, s využitím odbornej literatúry a ďalších informačných zdrojov, ktoré sú všetky citované v práci a uvedené v zozname literatúry na konci práce.

Ako autor uvedenej záverečnej práce ďalej vyhlasujem, že v súvislosti s vytvorením tejto záverečnej práce som neporušil autorské práva tretích osôb, najmä som nezasiahol nedovoleným spôsobom do cudzích autorských práv osobnostných a/alebo majetkových a som si plne vedomý následkov porušenia ustanovenia § 11 a nasledujúcich autorského zákona Českej republiky č. 121/2000 Sb., o práve autorskom, o právach súvisiacich s právom autorským a o zmene niektorých zákonov (autorský zákon), v znení neskorších predpisov, vrátane možných trestnoprávných dôsledkov vyplývajúcich z ustanovenia časti druhej, hlavy VI. diel 4 Trestného zákonníka Českej republiky č. 40/2009 Sb.

Brno

.....

podpis autora*

*Autor podpisuje iba v tlačenej verzii.

POĎAKOVANIE

Rád by som poďakoval vedúcemu diplomovej práce Ing. Romanovi Jakubíčkovi Ph.D. za odborné vedenie, konzultácie, trpezlivosť a cenné návrhy vypracovaniu k práci.

Obsah

Úvod	12
1 Konvolučné neurónové siete	13
1.1 Úvod do neurónových sietí	13
1.2 Základné prvky CNN	15
1.3 Učenie CNN	18
1.4 Klasifikácia obrazov	25
2 Problematika učenia s obmedzeným datasetom	28
2.1 Definícia problému	28
2.2 Riešenia problematiky obmedzeného datasetu	28
2.3 Techniky na úpravu datasetu	29
2.4 Techniky na úpravu algoritmu	33
3 Použité datasety	38
3.1 Dataset obrazov áut a motoriek	38
3.2 Dataset obrazov z histológie	39
4 Návrh experimentov	40
4.1 Príprava dát na učenie s obmedzeným datasetom	40
4.2 Popis klasifikačného modelu	41
4.3 Učenie s plným datasetom	45
5 Popis experimentov	47
5.1 Popis vplyvu architektúry siete ResNet	47
5.2 Popis transferového učenia	48
5.3 Popis dátovej augmentácie online	48
5.4 Popis dátovej augmentácie offline	49
5.5 Popis siete na prenos neurónového štýlu	50
6 Výsledky	56
6.1 Vplyv architektúry siete ResNet	56
6.2 Transferové učenie	58
6.3 Dátová augmentácia online	60
6.4 Dátová augmentácia offline	62
6.5 Prenos neurónového štýlu	64
6.6 Celkové porovnanie výsledkov	66
7 Diskusia	69

Záver	72
Literatúra	73

Zoznam obrázkov

1.1	Schéma základnej neurónovej siete	14
1.2	Viacvrstvomá neurónová sieť	14
1.3	Porovnanie medzi ANN a CNN	15
1.4	Posúvanie masky po obraze	16
1.5	Funkcia ReLU	16
1.6	Porovnanie typov združovania	17
1.7	Ilustračný príklad architektúry CNN	18
1.8	Matica zámien	23
1.9	Architektúra ResNet18	25
1.10	Architektúra VGG19	26
2.1	Ilustračný príklad základnej dátovej augmentácie	29
2.2	Grafické zobrazenie schémy siete typu GAN	30
2.3	Gramova matica	32
2.4	Výhody transferového učenia	34
2.5	Porovnanie štandardnej ANN s dropoutom	35
2.6	Schéma "bagging" metódy	36
3.1	Náhodne vybrané snímky z datasetu áut a motoriek	38
3.2	Náhodne vybrané snímky z datasetu obrazov histológie	39
4.1	Rozdelenie datasetu v množinách	41
4.2	Testovacia presnosť modelu s rôznymi semienkami	42
4.3	Testovacia presnosť modelu s rôznymi epochami	43
4.4	Celkové výsledky modelu s plným datasetom áut a motoriek	45
4.5	Celkové výsledky modelu s plným datasetom obrazov z histológie	46
5.1	Vývojový diagram dátovej augmentácia online	49
5.2	Vývojový diagram dátovej augmentácia offline	50
5.3	NST model	51
5.4	Vývojový diagram modelu NST pre nižší počet obrazov v triede	54
5.5	Generované obrazy modelom NST obrazov z histológie	55
5.6	Generované obrazy modelom NST obrazov motoriek	55
6.1	Porovnanie rôznych architektúr na datasete obrazov z histológie	56
6.2	Porovnanie rôznych architektúr na datasete áut a motoriek	57
6.3	Porovnanie transferového učenia na datasete obrazov histológie	58
6.4	Porovnanie transferového učenia na datasete áut a motoriek	59
6.5	Porovnanie dátovej augmentácie online na datasete obrazov histológie	60
6.6	Porovnanie dátovej augmentácie online na datasete áut a motoriek	61
6.7	Porovnanie dátovej augmentácie offline na datasete obrazov histológie	62
6.8	Porovnanie dátovej augmentácie offline na datasete áut a motoriek	63

6.9	Porovnanie s metódou NST na datasete obrazov histológie	64
6.10	Porovnanie s metódou NST na datasete áut a motoriek	65
6.11	Celkové porovnanie metód na datasete obrazov z histológie	66
6.12	Celkové porovnanie metód na datasete áut a motoriek	67
6.13	Časové porovnanie experimentov graficky	67
7.1	Príklad nepodarených áut a motoriek z modelu NST	70

Zoznam tabuliek

4.1	Znižovanie trénovacej a validačnej množiny dát	41
4.2	Výsledky Mann-Whitneyho testu	44
5.1	Výpočet výsledných obrazov po aplikácií augmentačným faktorom . .	49
5.2	Teoretické množstvo vytvorených obrazov pomocou modelu NST . . .	53
5.3	Časová náročnosť augmentácie pomocou NST	53
5.4	Dátová augmentácia pomocou modelu NST	54

Úvod

Strojové učenie sa v posledných dekádach stalo nevyhnutnou súčasťou ľudských životov, pretože prešlo neuveriteľným vývojom a ich schopnosť sa efektívne učiť viedlo k neobyčajným nástrojom v rôznych oblastiach ako rozpoznávanie vzorov v dátach, obrazoch či reči. Napriek týmto silným nástrojom, vďaka ktorým sa dokáže veľmi jednoducho zefektívniť pracovná efektivita, tak pre učenie modelov strojového učenia je potreba enormné množstvo dát, ktoré sú často aj tak neni k dispozícii v potrebnom rozsahu.

Učenie hlbokých neurónových sietí s obmedzeným alebo nedostatočným obsahom dát môže viesť nedostatočnej generalizácii modelu, čo v konečnom dôsledku obmedzuje schopnosť modelu sa efektívne adaptovať na neznáme a nové dáta.

Teoretická časť práce sa zaoberá stručným popisom neurónových sietí, následne prechádza do konvolučných neurónových sietí, ich základným prvkom, postupne nadväzuje na ich učenie spojené s problematikou z pohľadu klasifikačného problému a v neposlednom rade sa detailnejšie popisujú typy architektúr, ktoré sú používané v praktickej časti, predovšetkým siete ResNet18 a VGG19. Hlavný obsah teoretickej časti sa venuje detailnej problematike učenia hlbokých neurónových sietí s obmedzeným datasetom, začínajúca definíciou problému a nadväzuje sa na riešenie problematiky. Na záver sú popísané metódy spolu s ich základným princípom fungovania a aplikácií. Sú rozdelené do dvoch základných skupín - techniky na úpravu datasetu a techniky na úpravu algoritmu, ktorých hlavnou úlohou bolo preskúmať ich možnosti vplyvu na učenie s obmedzeným súborom dát, kde sa prechádza z jednoduchších metód až po zložitejšie.

Praktická časť práce sa začína kapitolou venujúca sa použitým datasetom založené na binárnu klasifikáciu. V ďalšej kapitole sa pokračuje s návrhom experimentov, kde sa popisuje príprava dát na učenie s obmedzeným datasetom, popis klasifikačného modelu spolu s nastavenými parametrami, ktoré vychádzajú predovšetkým zo sekcie venujúcej sa učeni s plným datasetom. Následne nadväzuje kapitola na popis experimentov, kde hlavnou úlohou je poskytnúť komplexný pohľad ich popisu. V posledných kapitolách sú zhrnuté a porovnané výsledky formou grafických zobrazení vykreslených metrík spolu s diskusiou, kde sú popísané možné vylepšenia použitých metód.

1 Konvolučné neurónové siete

V komunite strojového učenia (ML) sa považujú metódy hlbokého učenia za zlatý štandard, pretože v posledných rokoch sa stali najrozšírenejším výpočtovým prístupom v oblasti ML, pretože dosahujú výborné výsledky porovnajúce s výsledkom ľudského výkonu. Jedna z mimoriadne veľkých výhod je schopnosť učenia s obrovským množstvom údajom, vďaka čomu sa vo vysokej miere používa v širokom spektre aplikácií ako napríklad bioinformatika či spracovanie biomedicínskych obrazov. [1]

Konvolučné neurónové siete (CNN) sú špeciálnym typom neurónových sietí kde rovnako ako ostatných typov neurónových sietí základnou jednotkou sú neuróny, pričom je navrhnutá, aby sa automaticky a adaptívne dokázala učiť, čím sa eliminuje potreba ručnej extrakcie príznakov. [2]

Pre lepšie pochopenie CNN bude v skratke najprv vysvetlené fungovanie jednoduchochých neurónových sietí (ANN), následne sa bude pokračovať porovnaním medzi ANN a CNN, a potom bude venovať základnými prvkami CNN, v neposlednom rade spôsobom učenia CNN.

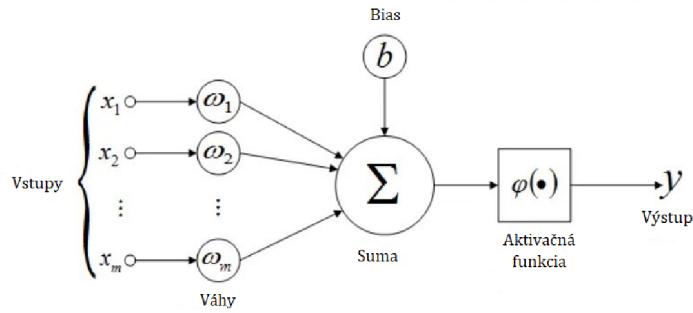
1.1 Úvod do neurónových sietí

Neurónové siete sú inšpirované spôsobom fungovania biologických neurónových sietí kde napodobňujú spôsob, akým si biologické neuróny predávajú signály v ľudskom mozgu. Označujú sa ako podmnožina strojového učenia a základom algoritmov hlbokého učenia. [3]

1.1.1 Matematický model neurónu

Základná jednotka umelej neurónovej siete, ktorý je inšpirovaný funkciou biologického neurónu je perceptrón, označovaný ako matematický model neurónu. Je popísaný len jedným neurónom, ktorý ma n -vstupov a prijíma informácie od každého spojenia. [4]

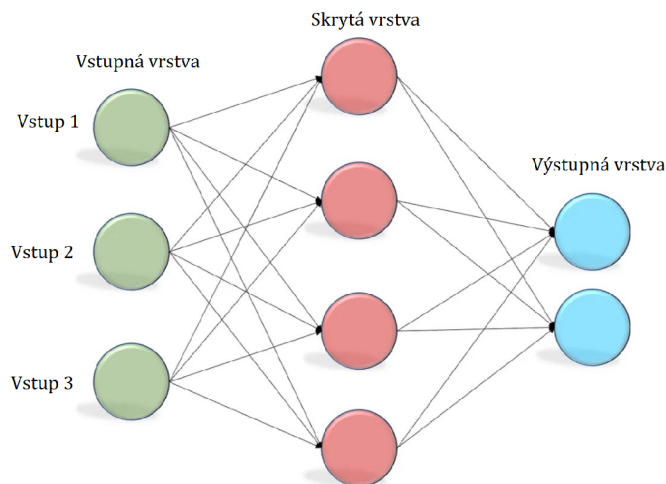
Obrázok 1.1 zobrazuje základný matematický model umelého neurónu, kde vstupy x_1, x_2, \dots, x_m sú reprezentované informáciami. Perceptrón upravuje svoje váhy pomocou vektoru váh w_1, w_2, \dots, w_m , ktoré sú kľúčový parameter pri správnom učení a fungovaní siete, pretože každý vstupný signál je násobený príslušnou váhou než sa dostane k neurónu. Váhované vstupy sú sčítané a kombinované so spoločným prahom, nazývaným ako "bias". Takáto kombinovaná hodnota je následne vložená do aktivačnej funkcie, ktorá rozhoduje či sa neurón stane aktívnym alebo nie. Výsledok z aktivačnej funkcie je na záver odoslaný na výstup. [4]



Obr. 1.1: Schéma základnej neurónovej siete [4]

1.1.2 Viacvrstvá neurónová sieť

Viacvrstvá neurónová sieť je špecifický typ umelej neurónovej siete, ktorá pozostáva minimálne z 3 vrstiev kde každá vrstva je reprezentovaná skupinkou neurónov. Bola vyvinutá z dôvodu, že jeden neurón nie je schopný riešiť nelineárne problémy. Tento typ sietí je základným kameňom pre hlboké učenie a ich architektúra začína vrstvou vstupných neurónov. Ich hlavná úloha je prijímať vstupné hodnoty. Následne pokračuje vrstva skrytých neurónov predstavujúca medzivrstvu medzi vstupnou a výstupnou vrstvou a slúžia na zložitejšie vzťahy v dátach. Poslednou časťou je skupina výstupných neurónov produkujúca výstup z neurónovej siete, ktorých počet závisí od typu problému - klasifikácia, regresia alebo iný typ úlohy. [4][5]

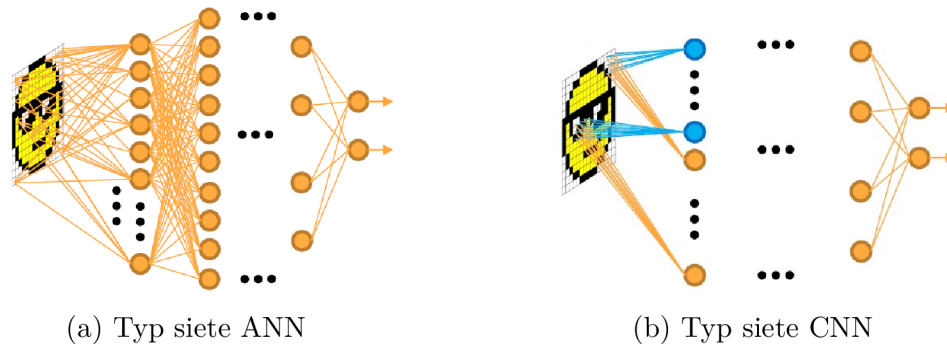


Obr. 1.2: Viacvrstvá neurónová sieť [6]

1.1.3 Rozdiel medzi ANN a CNN

Hlavným rozdielom je, že ANN majú štruktúru, v ktorej sú všetky neuróny v jednej vrstve priamo prepojené s každým neurónom v nasledujúcej vrstve. Táto štruktúra je vhodná pre všeobecné úlohy strojového učenia, ako sú klasifikácia či regresia. ANN neberú do úvahy priestorovú štruktúru vstupných dát, pretože každý prvok je spracovaný nezávisle od svojej polohy v priestore. [7]

Naopak, CNN využívajú vrstvy na detekciu vzorov a vzťahov v dátach. CNN obsahujú konvolučné filtre, pooling vrstvy a plne prepojené vrstvy. Tento typ siete sa často využíva pri spracovaní vizuálnych dát, ako sú obrázky a videá. Taktiež je efektívny pre úlohy ako je rozpoznávanie obrázkov, segmentácia či detekcia objektov. Konvolučné operácie umožňujú modelu zachytávať vzory bez ohľadu na ich presnú polohu v obraze. [7]



Obr. 1.3: Porovnanie medzi ANN a CNN [7]

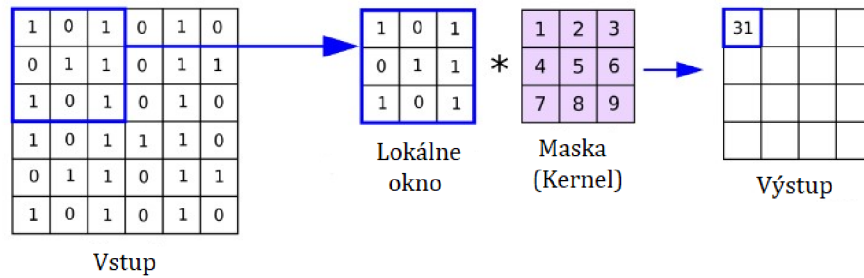
1.2 Základné prvky CNN

CNN má v závislosti od druhu požadovanej úlohy charakteristické základné dve až tri vrstvy - konvolučná, združovacia a plne prepojená vrstva. Bežná architektúra CNN sa skladá z opakovaných funkčných blokov tvorenými základnými vrstvami. Zaradenie do príslušnej triedy pri klasifikačnom probléme začína vstupným obrazom, ktorý prechádza navrhnutou neurónovou sieťou vrstvu po vrstve a postupne je transformovaný do výslednej hodnoty. [2]

1.2.1 Konvolučná vrstva

Najdôležitejšou zložkou v architektúre CNN je konvolučná vrstva a jej hlavnou úlohou je extrakcia typických príznakov zo vstupného obrazu. Je tvorená súborom

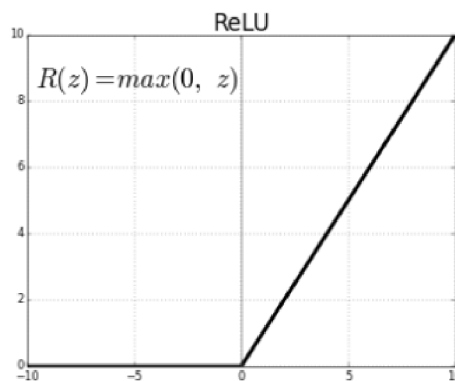
konvolučných filtrov - masiek, ktorú možno popísať ako mriežku diskretných čísel. Najčastejšie používané rozmery masiek sú 3 x 3 alebo 5 x 5 pixelov. [2]



Obr. 1.4: Posúvanie masky po obraze [8]

Proces konvolúcie slúži na získanie rôznych príznakov ("features") zo vstupného obrazu pomocou filtrov. Masky sú posúvané po vstupnom obraze a vykonáva sa konvolúcia filtrov s lokálnou oblasťou vstupného obrazu, ktorý je ohraničený veľkosťou masky, ako je znázornené na obrázku 1.4. V každom kroku sa násobí hodnota filtra s hodnotami vstupného obrazu v aktuálnom okne a výsledky sa sčítajú. Výstupom tejto matematickej operácie je tzv. príznaková mapa ("feature map"). Vďaka konvolúcii môže CNN efektívne získavať hierarchiu príznakov zo vstupného obrazu, napríklad jednoduché príznaky ako hrany či rohy. [9]

Po každej konvolyčnej vrstve nasleduje typ aktivačnej funkcie udávajúca prenos každého neurónu. Najčastejšie sa používa funkcia ReLU, ktorej hlavným cieľom je pridávať nelinearitu do neurónových sietí, čím umožňuje modelom učiť sa zložitejšie vzory v dátach. Priebeh prenosovej funkcie ReLU je zobrazený na obrázku 1.5. [10]



Obr. 1.5: Priebeh prenosovej funkcie ReLU [10]

1.2.2 Podvzorkovacia vrstva

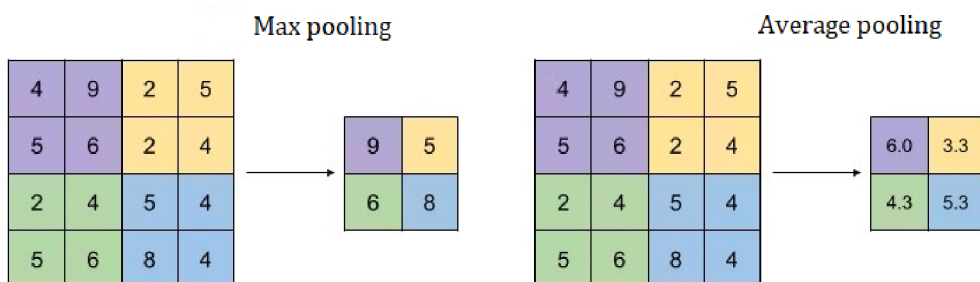
Ďalším stavebným blokom za konvolučnou vrstvou je združovacia vrstva, pričom hlavným cieľom je postupne zmenšovať príznakovú mapu, a tým znižovať počet parametrov v sieti. To pomáha zefektívniť tréning a zároveň zabezpečuje, že sieť je schopná generalizovať lepšie na nové dáta. Existujú dva najviac používané typy združovania zobrazené na obrázku 1.6 :

- **”Max pooling”**

Pri výbere maximálnej hodnoty v združovaní (”max pooling”) v definovanom bloku pixelov, najčastejšie 2x2, sa na príznakovej mape vyberie maximálna hodnota. Táto hodnota je následne zachovaná v novom združovanom obraze. Max pooling pomáha znižovať citlivosť na presnú polohu príznakov a pomáha zachovať dominantné príznaky. [11]

- **”Average pooling”**

Pri výbere priemernej hodnoty v združovaní (”average pooling”) sa v definovanom bloku pixelov vyberie priemerná hodnota z príznakovej mapy. Výsledkom je teda priemer hodnôt v danom bloku a táto hodnota je uložená v novom obraze. Výhodou je rovnomerné zmenšenie všetkých hodnôt v danej oblasti. [11]

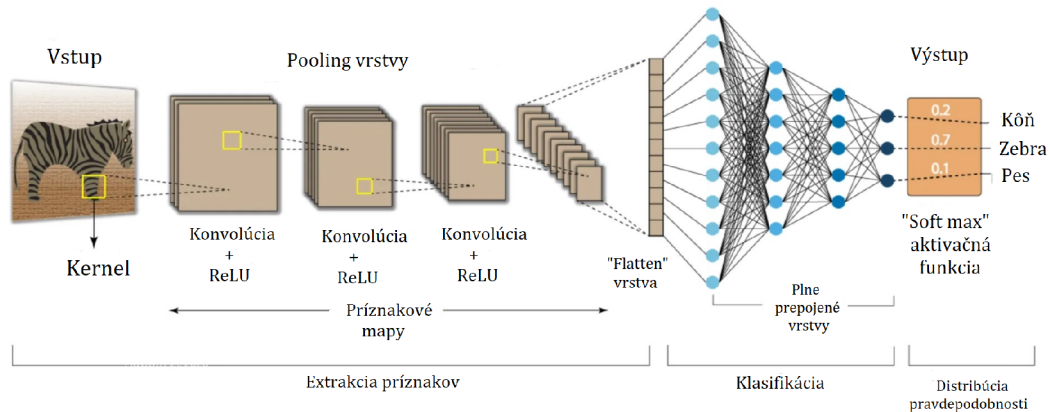


Obr. 1.6: Porovnanie typov združovania [11]

1.2.3 Plne prepojená vrstva

V CNN sú plne prepojené vrstvy obvykle umiestnené na konci siete, za konvolučnými a združovacími vrstvami. Charakteristickou črtou je, že každý neurón v danej vrstve je spojený s každým neurónom v predchádzajúcej a nasledujúcej vrstve. Pred vstupom do plne prepojenej vrstvy je príznaková mapa prevedená do jednorozmerného vektoru a ich hlavnou úlohou je transformovať vstupné príznaky do formy,

ktorá je vhodná na konečnú klasifikáciu, poprípade iný typ úlohy. Na obrázku 1.10 sa nachádza obecná architektúra CNN pre klasifikačnú úlohu s 3 výstupmi. [12]



Obr. 1.7: Ilustračný príklad architektúry CNN [12]

1.3 Učenie CNN

Ak bola zostavená architektúra CNN, nasleduje ďalšia veľmi dôležitá časť a to proces učenia siete. Medzi základné techniky učenia neurónovej siete patrí učenie s učiteľom ("supervised learning") a učenie bez učiteľa ("unsupervised learning"). Diplomová práca sa zaoberá technikou učením s učiteľom, ktorý pracuje s označenými údajmi, čo v praxi znamená, že pri tréningu sú dáta označené správnym výstupom. Následne pri testovaní by mala natrénovaná neurónovaná sieť s optimálnou presnosťou správne klasifikovať dáta bez označenia. [1]

1.3.1 Úprava váh

Bežne používaná metóda pre učenie CNN je algoritmus spätného šírenia chyby používané s výpočtom chybovej funkcie. V procese učenia sa počíta tzv. chybová funkcia, ktorá počíta odchýlku výstupu neurónu od očakávaného výstupu. Typom chybovej funkcie je možné ovplyvniť výsledok učenia a je jeden z hlavných parametrov, pričom sú popísané v nasledujúcej kapitole 1.3.2.

Na začiatku učenia sú nastavené váhy a prah náhodne. Následne na základe vypočítaných hodnôt dochádza k úprave váh, od poslednej vrstvy smerom k prvej vrstve. Váhy sú upravované na základe tzv. δ - pravidla popísané rovnicou 1.1 :

$$\vec{\omega}(t+1) = \vec{\omega}(t) - \mu_t \nabla \varepsilon. \quad (1.1)$$

Pričom $\vec{\omega}(t+1)$ označuje vektor váh v ďalšej iterácii, ktorý je počítaný pomocou vektoru váh v súčasnej iterácii $\vec{\omega}(t)$, a vypočítanej chybovej funkcie ε váhované parametrom učenia μ pri konkrétnej iterácii. [13]

1.3.2 Chybová funkcia

Stredná kvadratická odchýlka

Jedna z najviac používaných metrík hodnotenia chybovej funkcie je stredná kvadratická odchýlka popisujúca ako veľmi sa líši výstup siete od optimálneho výsledku, pričom je popísaná rovnicou 1.2 :

$$L(y, \hat{y}) = \frac{1}{m} \sum_{i=1}^m (y_i - \hat{y}_i)^2 \quad (1.2)$$

kde popisuje y_i skutočný výstup CNN, \hat{y}_i je požadovaný výstup CNN a m označuje počet vstupov. [5]

Typ metriky sa najviac používa keď je problém formulovaný ako regresia. To znamená, že predikovaná hodnota sa porovnáva so skutočnou hodnotou. Pri tréovaní modelu je možné vykresliť graf chybovej funkcie, vďaka ktorému sa dá efektívne sledovať tréovanie modelu. V ideálnom prípade by sa mala kvadratická odchýlka postupne znižovať a dosahovať svoje minimum. [14]

Vzájomná entropia

Vzájomná entropia známa ako "cross entropy loss" je typ chybovej funkcie popisujúca spôsob merania rozdielu medzi skutočnou a predpovedanou pravdepodobnosťou, vďaka čomu stanovuje rozdiel dvoch pravdepodobnostných rozdelení. Používa sa predovšetkým v kontexte klasifikácie, kde sa môže predpovedať príslušnosť každého vzorku do klasifikačnej triedy. Hlavným cieľom počas tréovania je minimalizovať hodnotu entropie medzi predikovanou a požadovanou hodnotou. Pre viacnásobnú klasifikáciu platí výpočet entropie H v rovnici 1.3:

$$H(P, Q) = - \sum_x^M P(x) \cdot \log(Q(x)) \quad (1.3)$$

kde M popisuje počet tried klasifikácie, $P(x)$ odpovedajúca požadovaná hodnota, ktorá je porovnávaná s predikovanou hodnotou $Q(x)$ pravdepodobnosti rozloženia. [14][15]

1.3.3 Optimalizačné algoritmy

Hlavnou úlohou optimalizačných algoritmov je nájsť optimálne hodnoty parametrov modelu, aby minimalizovali hodnotu chybovej funkcie počas tréningu neurónovej siete a najviac sa približovali k predikovaným výstupným hodnotám.

SGD

Stochastický gradientný zostup (SGD) je iteratívny optimalizačný algoritmus, ktorý je vďaka svojej ľahkej implementovateľnosti veľmi využívaná optimalizačná technika. Gradient udáva smer najprudšieho nárastu funkcie a je definovaný pomocou rovnice 1.4:

$$\text{grad}(E) = \frac{\partial E}{\partial w} \quad (1.4)$$

kde ∂E popisuje parciálnu deriváciu chybovej funkcie E podľa váh. w [13]

V každom kroku výpočtu gradientu sa získava podmnožina náhodne vybraných dát označujúca sa ako "mini-batch". Pre túto množinu sa vypočítava gradient chybovej funkcie vzhľadom k parametru modelu. Parametre sú následne aktualizované v opačnom smere ku gradientu. V porovnaní s metódou gradientného zostupu sa pri SGD aktualizujú váhy po každej iterácii, vďaka čomu je rýchlejšia, ale výpočetne náročnejšia. SGD je popísaný pomocou rovnice 1.5 :

$$w_{t+1} = w_t - \eta \nabla_w J(w_t; x_i, y_i) \quad (1.5)$$

pričom w_t je vektor parametrov v iterácii t , krok učenia popisuje η , následne $\nabla_w J(w_t; x_i, y_i)$ je gradient chybovej funkcie vzhľadom na vektor parametrov v iterácii t , vstupnom vektore x_i a cieľovom vektore y_i . [13][16]

Adam

Adaptívny odhad momentu, z anglického "adaptive moment estimation", skrátene Adam je v súčasnosti najviac využívaná optimalizačná technika, ktorá bola navrhnutá na tréning neurónových sietí a rieši niektoré nevýhody tradičného SGD, hlavne ľahké preskočenie minima funkcie. Adaptívny sa nazýva, pretože adaptívne upravuje veľkosť gradientu, ktorý súvisí s úpravou váh. Hlavnou výhodou algoritmu Adam je udržiavanie hodnôt gradientu z predchádzajúceho výpočtu a funguje na princípe, že počíta rýchlosti učenia rôznych parametrov. [17][18]

Zachováva exponenciálny kľzavý priemer prvého momentu m_t , ktorý zodpovedá priemeru gradientu, čo umožňuje algoritmu sledovať všeobecný trend gradientov počas tréningu. Momenty pre váhy premenných sú popísané rovnicou :

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t \quad (1.6)$$

Okrem toho zachováva exponenciálny kľzavý priemer druhého momentu v_t , ktorý zodpovedá priemeru štvorca gradientu. Tým sa zohľadňujú informácie o variabilite gradientov a je daný na základe vzťahu :

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2 \quad (1.7)$$

kde konštanty β_1 a β_2 v predošlých vzťahoch slúžia ako faktory zapomínania, t popisuje poradie tréningovej iterácie a parameter g_t udáva hodnotu gradientu chybovej funkcie. [17]

Keďže m_t a v_t sú inicializované ako vektory núl, tak je dôležité previesť upravenie koeficientu váh korekciou skreslenia pomocou nasledovných vzťahov :

$$\hat{m}_t = \frac{m_t}{1 - \beta_1^t} \quad (1.8)$$

$$\hat{v}_t = \frac{v_t}{1 - \beta_2^t} \quad (1.9)$$

V poslednom kroku hodnotu váh pre konkrétny parameter sa počíta pomocou vzťahu :

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\hat{v}_t} + \epsilon} \hat{m}_t \quad (1.10)$$

pričom η je veľkosť kroku a ϵ je konštatna na ochranu delenia pred 0 ako zanedbateľné číslo pri výpočte. [17]

1.3.4 Problémy pri učení

Miznúci gradient

Problém s miznúcim gradientom ("gradient vanishing problem") vzniká pri tréningu neurónových sietí, ovlyvňuje najmä CNN. Gradientsy stratovej funkcie vzhľadom na váhy siete sa stávajú extrémne malými, keď sa počas tréningového procesu spätne šíri z výstupnej vrstvy do vstupnej vrstvy. Čo znamená, že váhy v príslušných vrstvách sa aktualizujú len minimálne alebo vôbec, čo môže spôsobiť, že tieto vrstvy majú malý alebo nulový vplyv na výsledok. V prípade CNN môže nastať ešte komplikovanejší problém, pretože jednotlivé vrstvy siete majú špecifickú štruktúru a na extrakciu rôznych prvkov z obrazu využívajú jadrá filtrov. Následne ak sa gradient líši k nule, tak váhy filtrov nie sú dostatočne aktualizované, a preto nemôžu prispievať k učenie daných vlastností. [19][20]

Problémom je hlavne pri použití aktivačnej sigmoidálnej funkcie, tak malé derivácie sa spolu násobia, tak gradient následne exponenciálne klesá smerom k počiatočným vrstvám. Jedna z možností ako vyriešiť popísaný problém je použitie aktivačnej funkcie typu ReLU, ktorá nemá tendenciu vyvolávať miznúce gradienty. [20]

Ďalším spôsobom je využitie techniky na normalizáciu váh, ako napríklad "batch normalization", kde pomáhajú udržiavať stabilitu počas tréovania. Pri architektúrach sietí, ktoré obsahujú tzv. "skip connections" tak umožňujú gradientom prechádzať po určitých vrstvách, vďaka čomu dopomáhajú zmierniť problémy s miznúcim gradientom. [21]

Preučenie

Problém preučenia ("overfitting") nastáva, keď sieť sa naučí príliš dobre na tréovacie dáta a stráca generalizačnú schopnosť algoritmu na nové, neznáme dáta. To znamená, že sieť sa príliš presne prispôsobí šumom alebo konkrétnym vlastnostiam tréovacích vzorov a nedokáže efektívne zovšeobecniť na nové príklady. Problémom môžu byť práve malé zmeny v tréovacích dátach, kde dochádza k veľkým zmenám parametrom siete, čo môže naznačovať, že sieť zachytáva šum namiesto skutočného vzoru. [22]

Existuje niekoľko metód na riešenie problému preučenia. Jedna z nich je dátová augmentácia, popísaná v kapitole 2.3.2. Následne ďalším spôsobom je technika "dropout" fungujúca na princípe vypínania neurónov v priebehu tréovania. Použitie techník na regularizáciu váh ako L1 a L2 pomáha zabrániť príliš veľkým hodnotám váh, čo zlepšuje celkovú generalizáciu. [23]

Nevyrovnaný dataset

Problém nevyvážených súborov dát môže ovplyvniť schopnosť modelu správne zovšeobecniť a klasifikovať vzory vo všetkých kategóriách. Nevyvážený súbor dát znamená, že niektoré kategórie majú podstatne viac dát ako iné. V prípade menšinovej triedy nastáva pokles presnosti a nárast nadmerného prispôbenia. Model jednoducho nemusí byť schopný efektívne odhaliť vzory v menších triedach, pretože nie je dostatočne tréovaný na tieto vzory. [24]

Vyriešenie spomínaného problému môže nastať napríklad aplikovaním metódy SMOTE, čo je technika na vyváženie datasetov menšinovej triedy, pričom sa generujú syntetické príklady pre menšinovú triedu na základe existujúcich príkladov. Konkrétne, pre každý príklad v menšinovej triede vyberá susedné príklady a vytvára nové syntetické príklady v smere medzi nimi. [25]

Problém s nedostatkom dát

Učenie CNN s obmedzeným datasetom alebo nedostatok dát pri tréovaní je hlavným cieľom na skúmanie diplomovej práce, problém je popísaný v kapitole 2.

1.3.5 Hodnotenie predikcie modelu

Hodnotenie výkonnosti klasifikačného modelu je nevyhnutným krokom, pretože poskytuje informácie o tom, ako dobre model zvláda úlohu klasifikácie. Ich účinnosť a spoľahlivosť sú neoddeliteľne spojené s presnosťou ich predikcií a schopnosťou správne rozpoznať vzory v dátach. Existuje viacero metrík a ich výber závisí od konkrétneho kontextu. [26]

Matica zámien je nástrojom používaným na hodnotenie výkonnosti klasifikačného modelu. Ide o tabuľku, ktorá porovnáva skutočné triedy pozorovaní s predikovanými triedami modelu, pričom má formu dvojrozmerného poľa, kde jedna dimenzia predstavuje skutočné triedy a druhá dimenzia predstavuje predikované triedy. [27]

		Skutočná hodnota	
		Pozitívna	Negatívna
Predikovaná hodnota	Pozitívna	TP	FP
	Negatívna	FN	TN

Obr. 1.8: Matica zámien

Obsahuje 4 základné bunky, kde TP ("True positive") označuje prípady, kedy model správne predpovedal pozitívnu triedu. Napríklad, model správne diagnostikoval pacienta s rakovinou. Následne TN ("True negative") reprezentuje prípady, kedy model správne predpovedal negatívnu triedu, teda, že zdravý pacient naozaj nie je chorý. Ďalej FP ("False positive") identifikuje prípady, kedy model nesprávne predpovedal pozitívnu triedu. Týmto pádom by to znamenalo, že model mylne povedal, že pacient má chorobu, keď v skutočnosti nie je chorý. Posledným je FN ("False negative") označuje prípady, kedy model nesprávne predpovedal negatívnu triedu, čo znamená, že model prehliadol chorobu u pacienta, ktorý ju skutočne má. [26][27]

Pomocou matice zámien sa odvodzujú ďalšie metriky hodnotenia :

$$Presnosť = \frac{TP + TN}{TP + FP + FN + TN} \quad (1.11)$$

Presnosť vyjadrená vzorcom 1.11 popisuje celkový pomer správne klasifikovaných príkladov (TP a TN) k celkovému počtu príkladov. Je to jednoduchá a intuitívna metrika, ale môže byť nevhodná v prípade, že sa jedná o súbory dát s malým množstvom údajov. Ak je niekoľko chybných klasifikácií, ktoré majú výrazný vplyv na celkový pomer, tak to môže viesť k nekonzistentným alebo nereprezentatívnym výsledkom o výkonnosti modelu. Z tohto dôvodu sa používajú ďalšie metriky.

Sensitivita (SE) popisuje pomer správne klasifikovaných pozitívnych príkladov (TP) k celkovému počtu príkladov, ktoré sú skutočne pozitívne (TP + FN). Je dôležitá v prípade, keď je kritické minimalizovať falošne negatívne prípady. Vyjadruje koľko zo všetkých položiek, ktoré model predpovedal ako pozitívne boli skutočne pozitívne. Je vyjadrená vzorcom 1.12.

$$SE = \frac{TP}{TP + FN} \quad (1.12)$$

Špecificita (SP) definovaná vzorcom 1.13 vyjadruje pomer správne klasifikovaných pozitívnych príkladov (TP) k celkovému počtu príkladov, ktoré boli klasifikované ako pozitívne (TP + FP). Popisuje koľko zo všetkých položiek, ktoré sú skutočne pozitívne, boli správne predpovedané ako pozitívne.

$$SP = \frac{TP}{TP + FP} \quad (1.13)$$

Často sa ako dôležitá metrika používa F1-skóre vyjadrujúce harmonický priemer SE a SP, poskytuje vyvážený pohľad na výkonnosť klasifikačného modelu. Je popísaná vo vzorci 1.14.

$$F1 - skóre = \frac{2 \cdot SE \cdot SP}{SE + SP} \quad (1.14)$$

F1-skóre dosiahne svoju maximálnu hodnotu (1) v prípade, ak sú SE a SP rovnako maximálne, čo znamená, že model dosahuje rovnakú úspešnosť v identifikácii pozitívnych prípadov, minimalizácii falošných pozitívnych a falošných negatívnych príkladov. [26][27]

1.4 Klasifikácia obrazov

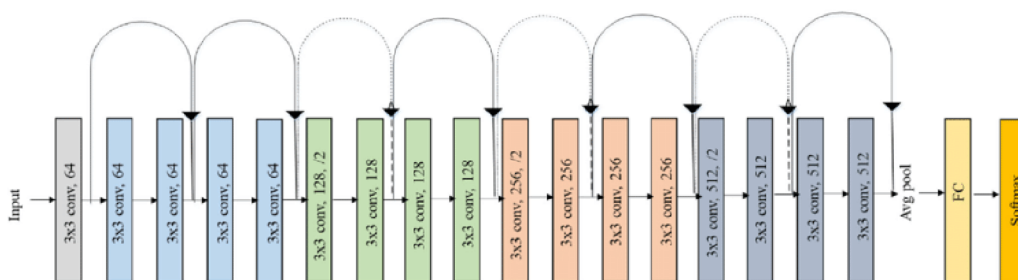
Klasifikácia obrazov predstavuje jedno z kľúčových odvetví v oblasti počítačového videnia a umelej inteligencie, kde hlavným cieľom je priradiť vstupný obraz do jednej alebo viacerých preddefinovaných kategórií. Existuje mnoho architektúr na klasifikáciu obrázkov, pričom jednou z najpopulárnejších sú CNN, pretože sa dokážu automaticky naučiť priestorovú hierarchiu prvkov, ako sú hrany, textúry a tvary, ktoré sú dôležité na rozpoznávanie objektov na obrázkoch. Dnes sa CNN používajú v širokom spektre aplikácií, od rozpoznávania objektov a tvárí, až po autonómne vozidlá či medicínsku diagnostiku. [28]

1.4.1 Typy architektúr CNN

Existuje veľké množstvo architektúr používaných v priemyselných aplikáciách na klasifikáciu obrazov. Avšak budú popísané typy používané v praktickej časti.

ResNet18

ResNet, skratka pre Residual Network, je rodina CNN navrhnutých tak, aby fungovali zavedením tzv. reziduálnych blokov, ktoré predovšetkým znižujú problém miznúcich gradientov pri trénovaní hlbokých sietí. Model ResNet bol prvýkrát navrhnutý v článku „Hlboké reziduálne učenie pre rozpoznávanie obrázkov“ od Kaiminga He v roku 2016 a odvtedy sa široko používa v mnohých oblastiach počítačového videnia. Architektúra ResNet18 je navrhnutá s ohľadom na kompaktnosť a efektívnosť pri zachovaní vysokého výkonu na menšom datasete. [29]



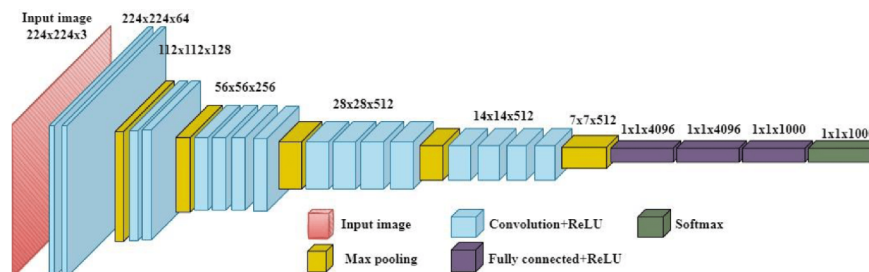
Obr. 1.9: Architektúra ResNet18 [30]

Architektúra siete ResNet18 pozostáva z niekoľkých komponentov. Na začiatku sa začína počiatočnou konvolučnou vrstvou so 7x7 filtrami, po ktorej nasleduje max-pooling na zníženie vzorkovania vstupného obrazu. Základnými stavebnými prvkami ResNet-18 sú reziduálne bloky, z ktorých každý pozostáva z dvoch konvolučných

vrstiev s 3x3 filtrami. Po každej konvolučnej vrstve sa aplikuje batch normalizácia a aktivačná funkcia ReLU. V sieti ResNet18 je zavedenie skrátенých spojení ("shortcut connections"), známych aj ako preskokové spojenia ("skip connections"). Tieto spojenia umožňujú vstupu obísť jednu alebo viac konvolučných vrstiev, čo pomáha zmierniť problém miznúceho gradientu. V sieti ResNet18 skrátенé spojenie preskakuje dve konvolučné vrstvy. Zvyšné bloky sú poskladané dohromady a tvoria hlavné telo siete. V sieti ResNet18 sú štyri takéto bloky, z ktorých každý obsahuje viacero zvyškových jednotiek. Na konci siete sa používa globálne priemerné združovanie na zníženie priestorových rozmerov na jedinú hodnotu pre mapu prvkov. To zjednodušuje následnú úlohu klasifikácie tým, že poskytuje vektor príznakov s pevnou veľkosťou bez ohľadu na veľkosť vstupného obrazu. Nakoniec je tu plne prepojená vrstva, ktorá vykonáva úlohu klasifikácie a mapuje príznaky extrahované predchádzajúcimi vrstvami na výstupné triedy. Na získanie pravdepodobností tried sa zvyčajne používa aktivačná funkcia softmax. [31]

VGG19

Architektúra siete VGG19 je významná svojou relatívne jednoduchou, avšak veľmi hlbokou architektúrou, ktorá obsahuje 19 vrstiev s váhami, pričom každá z nich využíva malé jadro veľkosti 3x3 s krokom 1, čo zabezpečuje zachovanie rozmerov obrazu. Táto vlastnosť umožňuje sieťam VGG učiť sa zložité a detailné vzory v dátach, čo prispieva k ich vysokému výkonu. [32]



Obr. 1.10: Architektúra VGG19 [33]

Po každej konvolučnej vrstve nasleduje aktivačná funkcia ReLU, ktorá zvyšuje nelineárnu schopnosť siete a zároveň pomáha predchádzať problému miznúcich gradientov. Funkcia je kľúčová pre udržanie dynamiky a efektívnosti učenia v celej sieti. Ďalej, architektúra obsahuje celkovo 5 max pooling vrstiev, ktoré sú rozmiesnené po určitých konvolučných vrstvách. Tieto vrstvy postupne redukujú priestorové rozmery vstupného obrazu, čím efektívne znižujú počet parametrov a výpočtovú náročnosť modelu, a súčasne zachytávajú a zvýrazňujú dominantné vlastnosti potrebné

pre klasifikáciu. Na koniec architektúry siete sú umiestnené 3 plne prepojené vrstvy, kde prvé dve disponujú 4096 kanálmi a tretia predstavuje výstupnú vrstvu s počtom kanálov odpovedajúcim počtu klasifikačných tried. [32]

Na úplnom závere architektúry je umiestnená softmax vrstva. Táto vrstva normalizuje výstupy z poslednej plne prepojenej vrstvy do pravdepodobnostného rozdelenia, čím umožňuje presnú klasifikáciu vstupného obrazu do jednej z preddefinovaných tried. Tento systém vrstiev a funkcionalít poskytuje VGG19 vynikajúce schopnosti v rozpoznávaní a klasifikácii obrazov, čo z neho robí jeden z pilierov v oblasti počítačového videnia. [32]

2 Problematika učenia s obmedzeným datasetom

2.1 Definícia problému

Na trénovanie neurónových sietí je potrebné veľké množstvo údajov, čo je veľakrát obmedzujúcim činiteľom v priemyselných aplikáciách, pretože proces zberu môže byť zdĺhavý, s čím súvisí aj nárast ceny na návrh modelu strojového učenia. Presné množstvo dát na proces trénovania je častokrát neznáme a je označované ako jeden z najviac dôležitých faktorov pri návrhoch optimálneho systému počítačom podporovanej diagnostiky. Malé množstvo vzoriek pri trénovaní neurónových sietí trpí prebytočným prispôbením údajov a majú oveľa nižšiu pravdepodobnosť, že by mali skutočné využitie, pričom negatívne ovplyvňujú schopnosť generalizácie na nové dáta. Nárast počtu vzoriek môže zvýšiť celkovú presnosť modelu, ale taktiež po určitom zvyšovaní nemusí nastať výrazné zlepšenie. [34][35]

Ďalším kľúčovým problémom je obmedzená rozmanitosť dát. V prípade, že trénovací dataset nedostatočne reprezentuje rôznorodosť vstupných dát, môže dôjsť k tomu, že model sa naučí len špecifické vzory a bude mať obmedzenú schopnosť adaptovať sa na nové situácie. Nižšia presnosť a obmedzená robustnosť modelu predstavujú ďalšie problémy spojené s obmedzeným datasetom. Model môže byť náchylný na šum a nejednoznačnosti v dátach, čo môže viesť k nižšej úspešnosti pri riešení nových úloh. [36]

Aplikácie medicínskych obrazov pre učenie neurónových sietí sú ideálnym príkladom z dôvodu vysokej kvality a manuálne anotovaných diagnóz špecialistom. Avšak súkromie pacientov je jeden z najzávažnejších problémov, pretože sťažuje prístup k potrebným dátam. Taktiež vysokokvalitné anotácie odborníkmi sú často nákladné a vo veľkom rozsahu takmer nezvládnuteľné. [37]

2.2 Riešenia problematiky obmedzeného datasetu

Učenie sa z obmedzených súborov údajov je veľmi náročné a stále je tento problém v podstate nevyriešený. Malé množstvo prác sa venuje učeniu s malým súborom údajov. Jedna z nich hodnotila skupinu 17 klasifikátorov aplikovaných na 121 datasetov, ale tento problém sa venuje práce s klasifikátormi strojového učenia, nie hlbokého učenia. Klasifikátor s najlepšimi výsledkami sa ukázal náhodný les, tesne za ním SVM. [38] V literatúre sa väčšina autorov nevenuje problému nedostatku dát v oblasti strojového učenia a hlbokého učenia zamerané na multidimenzionálne

vstupy - obrazy. Keď sa zvyšuje dimenzionaliza vstupných údajov sa úloha klasifikácie stáva oveľa zložitejšou. Jednoduchý prístup na zlepšenie generalizácie modelu je implementácia techník, ktoré sa snažia syntetizovať nové obrazy pomocou rôznych transformácií. Generovanie nových obrazov môže byť extrémne efektívne, ale na druhú stranu aj výpočetne náročné. Autori popisujú, že štandardné dátové augmentácie dokážu zlepšiť výkonnosť modelu do veľkej miery, avšak typ augmentácie by mal byť pečlivo navrhnutý, pretože je vysoko korelovaný s typom obrazu. [39]

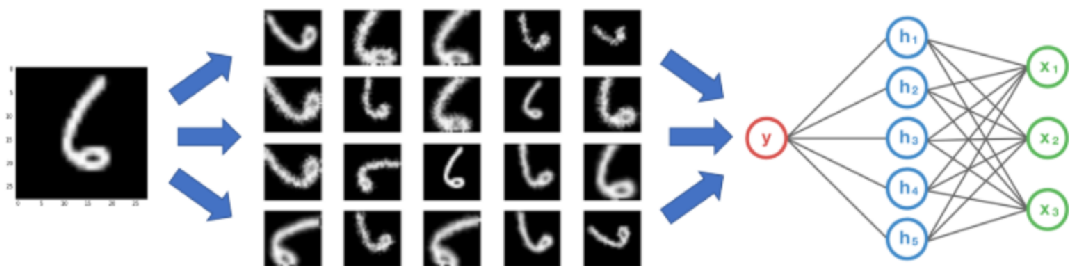
Z tohto dôvodu existujú metódy, ktoré pri malom množstve dát zlepšia celkovú presnosť modelu, vďaka čomu sú schopné šetriť čas a hlavne financie. Vo všeobecnosti prvá kategória implementuje techniky na úpravu datasetu popísané v kapitole 2.3. Následne druhá kategória sa venuje technikám na úpravu algoritmu popísané v kapitole 2.4.

2.3 Techniky na úpravu datasetu

2.3.1 Dátová augmentácia - základná

Dátová augmentácia je technika, ktorá sa používa na umelé zväčšenie veľkosti datasetu použitím rôznych transformácií z pôvodných dát. Hlavným cieľom je rozčleniť súbor dát, a tak zlepšiť robustnosť vytvoreného modelu. [40]

Hlavným cieľom dátovej augmentácie je zlepšiť generalizačnú schopnosť modelu tým, že ho vystavíme širšiemu spektru dátových vzorov počas tréningu. Z toho vyplýva, že ak je potrebné získať väčšie množstvo údajov stačí vykonať menšie zmeny, aby si vytvorená neurónová sieť myslela, že ide o odlišné obrázky, pretože ak sú objekty umiestnené v rôznych orientáciách, CNN je schopná robustne klasifikovať objekty pomocou vlastnosti nazývanej ako invariancia. [41]



Obr. 2.1: Ilustračný príklad základnej dátovej augmentácie [41]

Augmentácia sa zvyčajne vykonáva pred zadajím údajov do modelu strojového učenia. Medzi základné augmentačné techniky patrí [42] :

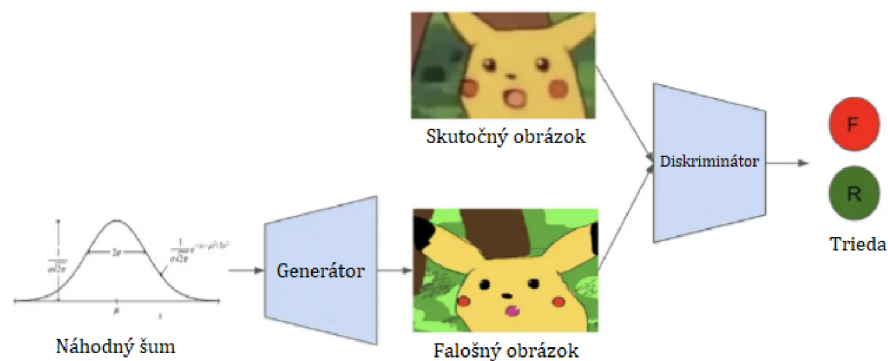
- **Rotácia:** rotácia o určitú hodnotu uhla
- **Preklopenie:** preklopenie horizontálnym alebo vertikálnym smerom
- **Škálovanie:** zväčšenie alebo zmenšenie obrázku, mení sa veľkosť
- **Orezanie:** vyberá sa časť pôvodného obrázku a mení sa na pôvodnú veľkosť
- **Translácia:** posun v smere osi X alebo Y (alebo oboch naraz)
- **Gausovský šum:** pridanie šumu rovnomerne všetkým frekvenciám

2.3.2 Dátová augmentácia - pokročilá

Niekedy pri tréňovaní CNN nestačia základné augmentačné techniky, a preto existujú pokročilé techniky, ktorým hlavným cieľom je zvýšiť rozmanitosť údajov generovaním jedinečných obrázkov s novou informáciou, čo môže pomôcť ku lepšej generalizácii modelu. Medzi najbežnejšie pokročilé techniky patria :

Generatívne adverzné siete

Generatívne adverzné siete (GANs) predstavujú pokročilý koncept v oblasti strojového učenia a umelého inteligencie, konkrétne v rámci generatívnych modelov. Táto technológia bola prvýkrát predstavená v roku 2014 Ianom Goodfellowom a jeho kolegami. [43]



Obr. 2.2: Grafické zobrazenie schémy siete typu GAN [44]

GANs sú modely založené na učení bez učiteľa skladajúce z dvoch typov neuronových sietí. Prvý z nich nazývaný ako generátor slúži k vytvoreniu údajov, ktoré sú na nerozoznanie od skutočných údajov. Na vstupe prijíma náhodný šum a generuje vzorky. Druhý s názvom diskriminátor má za úlohu rozlišovať medzi skutočnými údajmi a vzorkami vytvorené generátorom. Hlavným cieľom diskriminátora je správne klasifikovať či je vzorka skutočná alebo umelo vytvorená. Tento proces sa

odohráva ako hra s nulovým súčtom, kde zlepšenie jednej siete vynucuje zlepšenie druhej, čo vedie k rýchlemu a efektívnemu učeniu. [44]

Hlavným cieľom diskriminátora je správne označiť vygenerované obrázky ako falošné obrázky a empirické dátové body ako skutočné obrázky. Za diskriminátor sa považuje nasledujúca stratová funkcia, ktorá popisuje vzdialenosť alebo rozdiel medzi dvoma parametrami funkcie [45] :

$$L_D = \text{Error}(D(x), 1) + \text{Error}(D(G(z)), 0) \quad (2.1)$$

Na druhej strane, cieľom generátora je čo najviac zmiatť diskriminátor a falošne označiť vygenerované obrázky ako skutočné obrázky. Generátor by sa mal snažiť minimalizovať rozdiel medzi 1 (označené skutočné údaje) a falošnými údajmi generovanými vyhodnotením diskriminátora. Stratová funkcia je popísaná ako [45] :

$$L_G = \text{Error}(D(G(z)), 1) \quad (2.2)$$

Bežnou stratovou funkciou, ktorá sa používa v problémoch binárnej klasifikácie, je vzájomná entropia, ktorej vzorec je popísaný v rovnici 1.3 v kapitole 1.3.2. Stratová funkcia meria ako sa líšia dve rozdelenia, či je dátový bod pravdivý alebo nepravdivý. Aplikovaním sa dostanú 2 stratové funkcie pre diskriminátor L_D a generátor L_G [45] :

$$L_D = - \sum_{x \in \mathcal{X}, z \in \mathcal{Z}} \log(D(x)) + \log(1 - D(G(z))) \quad (2.3)$$

$$L_G = - \sum_{z \in \mathcal{Z}} \log(D(G(z))) \quad (2.4)$$

Optimalizácia modelu prebieha nájdením parametrov generátora a diskriminátora, aby boli stratové funkcie optimalizované, čo zodpovedá tréningu modelu. Výpočetná náročnosť GANs je jednou z hlavných výziev pri ich tréningu a využívaní. Táto náročnosť vychádza z viacerých faktorov, ktoré zahŕňajú architektúru modelu, veľkosť dátového setu, počet iterácií či epoch potrebných pre konvergenciu, a hardvérové požiadavky.

Prenos nerónového štýlu

Neurónové siete pre prenos štýlu (NST) predstavujú techniku v oblasti strojového učenia, ktorá umožňuje prenášať umelecký štýl z jedného obrazu tzv. štýlový obraz na iný obraz tzv. obsahový obraz, čím vytvára nový obraz, ktorý kombinuje obsah prvého obrazu so štýlom druhého. Táto technológia bola predstavená v práci Leona A. Gatysa a kolegov v roku 2015 a stala sa základom pre mnohé následné výskumy a aplikácie. [46]

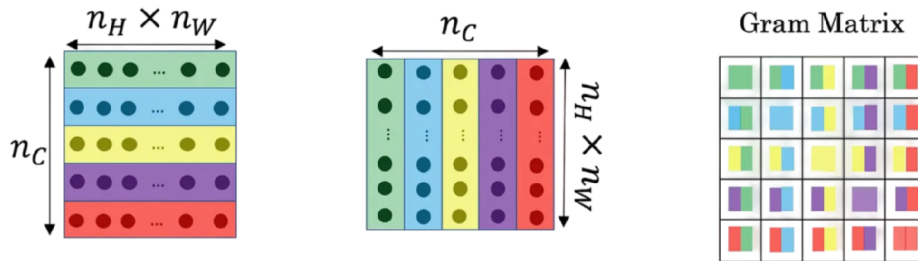
Architektúra NST využíva predtrénovanú konvolučnú neurónovú sieť, často VGG (napr. VGG19), ktorá bola pôvodne navrhnutá a trénovaná na klasifikáciu obrazov v datasete ImageNet. Proces prenosu štýlu zahŕňa tri hlavné obrázky: obsahový obraz, štýlový obraz a generovaný obraz, ktorý sa postupne upravuje tak, aby odrážal obsah z obsahového obrazu a štýl zo štýlového obrazu. Pomocou tejto techniky sa dosahuje podobný výsledok ako GAN, ale v ich porovnaní je NST výpočetne menej náročná, čo má ale za následok menšiu robustnosť algoritmu. [41]

Kľúčové komponenty architektúry siete NST zahŕňajú obsahovú stratu a štýlovú stratu, ako aj samotný optimalizačný proces. Obsahová strata zabezpečuje, že aktivácie vrstiev vyššej úrovne sú podobné medzi obrazom obsahu a generovaným obrazom, čím sa zachováva obsah pôvodného obrazu. Zvyčajne sa definuje ako euklidovská vzdialenosť medzi reprezentáciami prvkov obsahového obrazu a generovaného obrazu. Príznaková mapa generovaného obrazu označená ako F^l a príznaková mapa obsahového obrazu vo vrstve l označená P^l , kde následne je obsahová strata $L_{content}$ reprezentovaná rovnicou [47] [48] :

$$L_{content}(P^l, F^l) = \frac{1}{2} \sum_{i,j} (F_{ij}^l - P_{ij}^l)^2 \quad (2.5)$$

kde i a j sú indexy príznakovej mapy vo vrstve l .

Štýlová strata má za cieľ zladit štýl obrázku s vygenerovaným obrázkom. Definuje sa pomocou tzv. gramovej matice, ktorá zachytáva korelácie medzi mapami prvkov v rôznych vrstvách CNN. Gramova matica kóduje informácie o textúre a štýle obrazu.



Obr. 2.3: Vizualizácia matíc a výsledná Gramova matica [47]

Pre danú vrstvu l , kde A^l reprezentuje Gramovu maticu štýlového obrazu a G^l reprezentuje Gramovu maticu generovaného obrazu, štýlová strata L_{style} v tejto vrstve je popísaná rovnicou [48] :

$$L_{style}^l(A^l, G^l) = \frac{1}{4N_l^2 M_l^2} \sum_{i,j} (G_{ij}^l - A_{ij}^l)^2 \quad (2.6)$$

kde N_l je počet príznakových máp, M_l je veľkosť každej príznakovej mapy, i a j sú indexy elementov Gramovej matice.

Celková štýlová strata je následne daná ako vážený súčet strát štýlu v každej uvažovanej vrstve [48] :

$$L_{style}(A, G) = \sum_l w_l L_{style}^l(A^l, G^l) \quad (2.7)$$

kde w_l je váhovaný faktor je každú vrstvu l .

Finálna stratová funkcia, ktorá sa architektúra siete NST snaží minimalizovať je daná ako vážený súčet obsahovej straty $L_{content}$ a štýlovej straty L_{style} vyjadrená rovnicou [48] :

$$L_{total} = \alpha L_{content} + \beta L_{style} \quad (2.8)$$

kde α a β sú váhové faktory, ktoré vyrovnávajú príspevok každého člena.

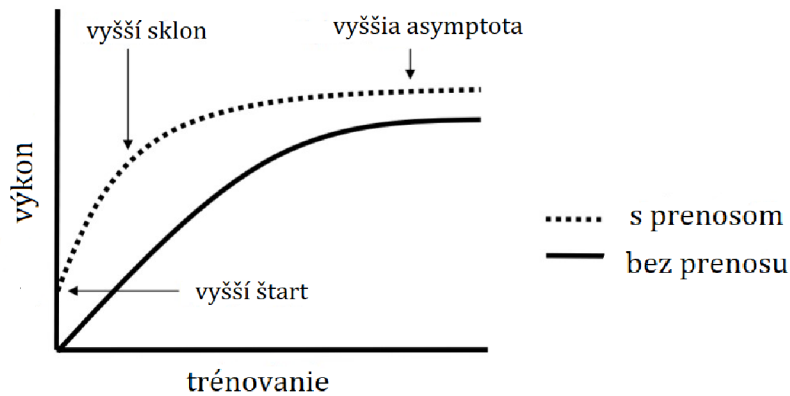
Proces NST zahŕňa viaceré kroky. Začína sa inicializáciou generovaného obrazu, ktorý je buď náhodný šum alebo kópia obsahového obrazu. V každej iterácii optimalizačného procesu sa vypočíta obsahová strata medzi generovaným a obsahovým obrazom, ako aj štýlová strata medzi generovaným a štýlovým obrazom. Použitím algoritmu, ako je gradientný zostup, sa potom pixely generovaného obrazu upravujú tak, aby sa celková strata minimalizovala. Tento proces iterácie pokračuje, až kým generovaný obraz nedosiahne uspokojivú podobnosť s obsahom obsahového obrazu a štýlom štýlového obrazu. [47]

2.4 Techniky na úpravu algoritmu

2.4.1 Transferové učenie

Transferové učenie ("Transfer learning") je technika hlbokého učenia, pri ktorej sa vytvorený model neučí od samého začiatku, ale využíva predtrénované modely s naučenými vlastnosťami na veľkých datasetoch, a preto si predtrénované modely vyžadujú menej času v porovnaní s tréňovaním od začiatku. Využívajú maximálne množstvo dostupných údajov z datasetu čo pomáha zlepšeniu presnosti pri konkrétnych úlohách. Tento typ techniky na zlepšenie presnosti modelu môže použiť predtrénovaný model, ktorý bol natrénovaný na veľkom datasete na úplne inej úlohe s rovnakým vstupom, ale iným výstupom. [49]

Hlavné výhody, ktoré ponúka transferové učenie je popísané na obrázku 2.4. Prvá možnosť zobrazuje začiatok s vyšším štartom, čo spôsobuje lepší výkon od začiatku. Ďalším prípadom je vyšší sklon, ktorý zabezpečuje rýchlejšie učenie modelu. Poslednou možnosťou je vyššia asymptota a označuje skutočnosť, že výkon modelu dosahuje vyšší maximálny bod, keď je predtrénovaný na veľkom a rôznorodom datasete, kde následne prenáša svoje naučené váhy na nový dataset. [50]



Obr. 2.4: Výhody transferového učenia [51]

Využívajú sa tri hlavné spôsoby transferového učenia :

CNN ako extrakcia príznakov

Model sa predtrénuje na veľkom datasete, často na úlohe klasifikácie obrázkov, a potom sa odstránia vrstvy na konci siete (plne prepojené vrstvy), ktoré boli naučené na konkrétnu úlohu. Zvyšné vrstvy sú potom ponechané a nové plne prepojené vrstvy sú pridané na konci siete. Tieto nové vrstvy sú inicializované náhodne a potom sú trénované na novej úlohe. [50]

Ladenie CNN

Model sa predtrénuje na veľkom datasete, ale namiesto toho, aby sa odstránili všetky vrstvy na konci siete, sa niektoré z nich môžu ponechať. Ponechané vrstvy môžu byť potom jemne ladené na nový dataset. To znamená, že váhy týchto vrstiev sú aktualizované na základe nových dát, zatiaľ čo váhy predošlých vrstiev zostávajú nezmenené alebo sa aktualizujú menej. [50]

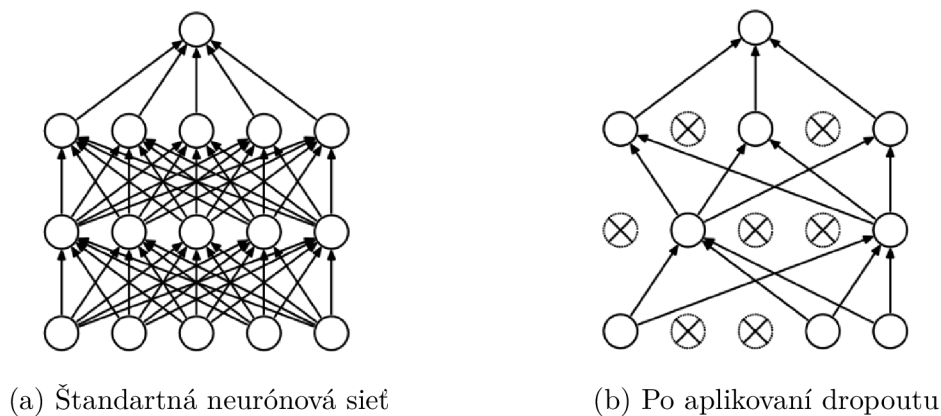
Predtrénované modely

Aj keď v dnešnej dobe sú pomerne vyvinuté GPU, tak časová náročnosť pri trénovaní CNN môže trvať aj týždne pri veľkom množstve obrazov. Mnoho ľudí zverejňuje svoje výsledky na úkor ostatných, aby mohli lepšie doladiť svoje modely a netrávili priveľa času manuálnym ladením konvolučnej siete. [50]

2.4.2 Dropout

Dropout je technika regularizácie používaná na tréning, aby sa predišlo ich pretrénovaniu modelu. V kontexte tréningu CNN s obmedzeným datasetom sa zvyčajne aplikuje na plne prepojené sieťové vrstvy na konci architektúry. Tieto vrstvy sú totiž vďaka väčšiemu počtu parametrov najviac náchylné na pretrénovanie. [52]

CNN môžu obsahovať milióny parametrov, sú obzvlášť náchylné na pretrénovanie, najmä ak sú tréňované s malými súbormi údajov. Dropout pomáha riešiť tento problém náhodným „vypínaním“ časti neurónov v sieti počas tréningu, čo núti sieť distribuovať „učenie“ medzi neuróny rovnomernejšie. To znamená, že siete musia vyvinúť výkonnejšie a všeobecnejšie reprezentácie, než sa spoliehať na malý počet neurónov na vykonávanie určitých úloh. [53]



Obr. 2.5: Porovnanie štandardnej ANN s dropoutom [54]

Hodnota dropoutu je hyperparameter, ktorý určuje pravdepodobnosť, s akou budú jednotlivé neuróny v sieti „vypnuté“ (t.j. ich výstupy budú dočasne nastavené na nulu) počas fázy tréningu. Táto pravdepodobnosť je vyjadrená ako desatinné číslo medzi 0 a 1, kde 0 znamená žiadny dropout a 1 znamená úplný dropout. Hodnota dropoutu nastavená na 0 znamená, že dropout nie je aplikovaný. To znamená, že žiadne neuróny nebudú náhodne „vypnuté“ počas tréningu. Sieť bude používať všetky dostupné neuróny pri každom prechode vpred a späť. Na druhej strane, hodnota dropoutu nastavená na 1 znamená, že všetky neuróny v danej vrstve budú „vypnuté“ počas tréningu. V praxi to znamená, že vrstva je úplne ignorovaná, keďže žiadne informácie neprechádzajú cez neuróny.

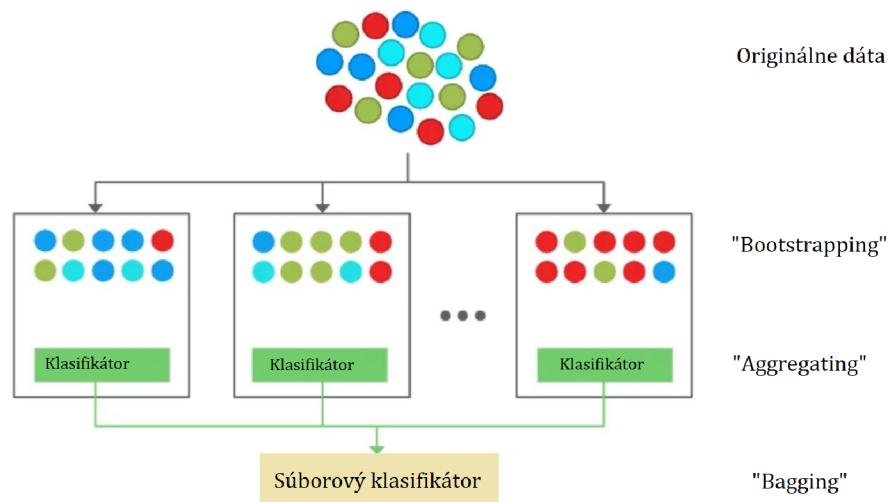
2.4.3 Súborové učenie

Súborové učenie ("Ensemble learning") zahŕňa kombináciu viacerých modelov strojového učenia s hlavným cieľom vytvoriť lepšiu presnosť modelu ako by to dokázal samostatne. CNN disponujú často vysokým rozptylom, čo môže viesť k ťažkostiam pri vytvorení finálneho modelu. Riešením je kombinácia viacerých dobrých modelov, ale zároveň aj rozdielnych, čo v praxi znamená, aby sa každý model dopúšťal iných chýb pri predikovaní. Taktiež je možné vytvoriť rôzne modely s odlišnými architekturami a následne ich kombinovať. [55]

Existujú dva hlavné prístupy k využitiu súborového učenia :

Bagging (Bootstrap Aggregating)

Metóda "bagging" sa používa na zostavenie viacerých modelov pomocou rôznych podmnožín tréningových údajov ako je zobrazené na obrázku 2.6. Následne každý model je trénovaný na inej podmnožine údajov a výsledky získané z týchto modelov sa potom kombinujú. Zlepšuje celkovú robustnosť, pretože model je trénovaný na viacerých podmnožinách a je viac pravdepodobné, že sa dopúšťajú rozdielnych chýb pri predikciách. [56]



Obr. 2.6: Schéma "bagging" metódy [56]

Boosting

Pri trénovaní CNN s obmedzeným datasetom sa metóda "boosting" používa na trénovanie viacerých modelov na rovnakom súbore dát a výsledky z týchto modelov sa kombinujú s cieľom zlepšenia celkového výkonu modelu. Známym prístupom na

kombináciu modelov je BoostCNN, kde hlavnou výhodou je využitie kombinácie silnejších stránok pri viacerých vytvorených slabších modeloch. [57]

BoostCNN funguje tak, že trénuje viacero CNN na rovnakom súbore údajov a potom kombinuje ich predpovede pomocou "boostovacích váh". Tieto váhy sa určujú pri každej iterácii a sieť, ktorá vedie k najväčšiemu zníženiu rizika pre klasifikáciu, sa pridá do výsledného súboru. [57]

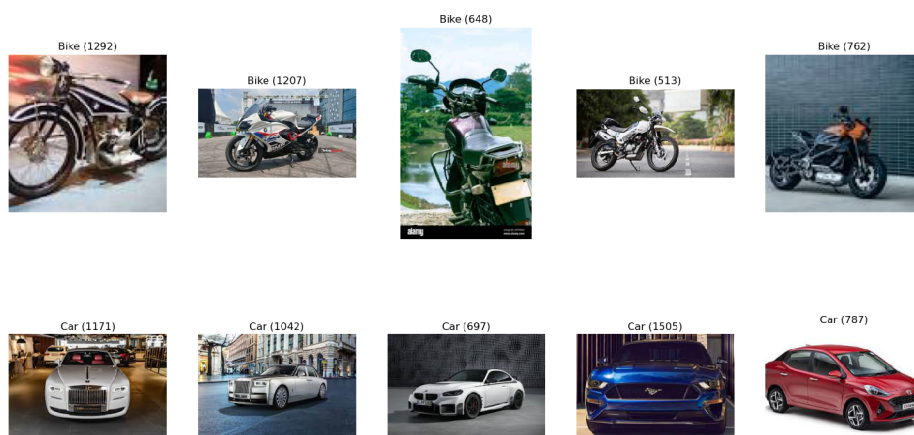
3 Použité datasey

3.1 Dataset obrazov áut a motoriek

Prvý dataset obsahuje jedinečné obrázky zobrazujúce rôzne typy automobilov a motoriek. Autor popisuje, že pri zbere dát bola zachovaná rôznorodosť, aby boli obe kategórie dostatočne prezentované vo všetkých možných podmienkach. Je verejne dostupný na internetovej stránke Kaggle. [58]

Rozmery každej fotografie sú odlišné, kde najmenšia fotografia dosahuje rozmery 100×100 pixelov a na druhej strane s najväčšími rozmermi 8192×5464 pixelov, čo nám potvrdzuje veľkú rôznorodosť. Snímky boli zaznamenané z rôznych uhlov, pri odlišných svetelných podmienkach a dokonca na niektorých záberov sa nachádza len časť automobilu či motoriky.

Dáta nie sú rozdelené na tréningovú a testovaciu množinu, pričom kategórie sú rovomerne rozdelené a obsahujú dokopy 4000 obrázkov pre binárnu klasifikáciu. Položky sú pomenované ako "bike", čiže motorika a "car" označenie pre automobil. Pre experimenty diplomovej práce bolo vybraných náhodne 1000 snímok z každej kategórie. Náhodné snímky z datasetu sú zobrazené na obrázku 3.1.

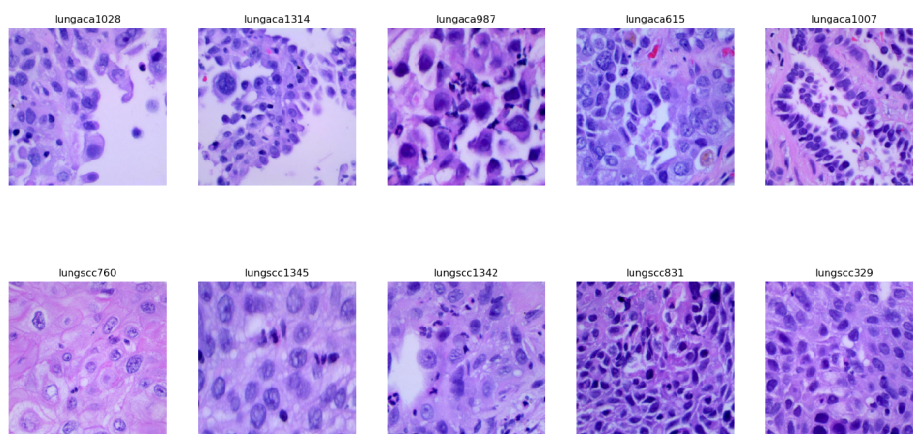


Obr. 3.1: Náhodne vybrané snímky z datasetu áut a motoriek

3.2 Dataset obrazov z histológie

Ako druhý dataset z medicínskej problematiky boli zvolené histologické snímky, primárne zamerané na prípady rakoviny pľúc, ktorý bol opäť stiahnutý na stránke Kaggle. Autor popisuje ako hlavný účel zverejnenia datasetu na vývoj pokročilých systémov umelej inteligencie. [59]

Pôvodne obsahuje dokopy 12 tisíc snímok v 3 kategóriách : adenokarcinóm ("lung aca"), neuroendokrinné nádory ("lung n") a spinocelulárny karcinóm ("lung scc"). Dáta boli vopred rozdelené na testovaciu a trénovaciu množinu, ale pre účely diplomovej práce a zachovanie rovnakého protokolu binárnej klasifikácie boli použité dve najviac podobné kategórie, ktoré sú zamerané na rozlíšenie medzi adenokarcinómom a spinocelulárnym karcinómom. Následne z každej kategórie bolo náhodne vybraných po 1000 snímok pre lepšiu reprodukovateľnosť. Rozmery snímok sú zachované všetky zhodne - 768×768 pixelov. Príklad snímok je zobrazený na obrázku 3.2.



Obr. 3.2: Náhodne vybrané snímky z datasetu obrazov histológie

4 Návrh experimentov

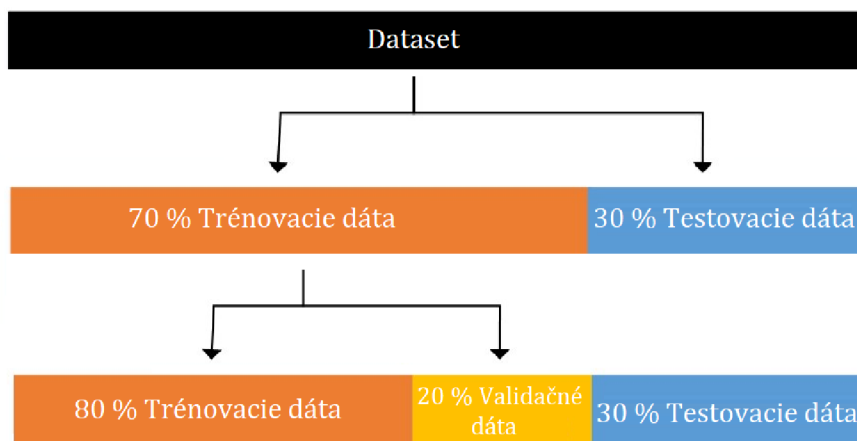
Hlavným cieľom diplomovej práce bola realizácia experimentov na učenie hlbokých neurónových sietí s obmedzeným datasetom a ich následne porovnanie. Experimenty boli navrhnuté a realizované v programovacom jazku Python vo verzii 3.11.8 spolu s využitím prostredia Spyder a knižnice pre hlboké učenie PyTorch, čo je jedna z hlavných knižníc pre hlboké učenie, ktorá poskytuje nástroje na efektívne trénovanie a nasadenie neurónových sietí s dôrazom na rýchlosť a flexibilitu. Jednou z jeho kľúčových predností je schopnosť rýchleho výpočtu tenzorov s využitím grafickej procesorovej jednotky prostredníctvom technológie CUDA, čo umožňuje výrazné zrýchlenie operácií. Bol použitý operačný systém Linux Ubuntu vo verzii 22.04 s grafickou kartou Nvidia Titan Xp 12 GB GDDR5. Zdrojové kódy s implementovanými experimentami sú dostupné na adrese : <https://github.com/DP-Nemeth>.

4.1 Príprava dát na učenie s obmedzeným datasetom

V experimentoch diplomovej práce zameranej na učenie hlbokých neurónových sietí s obmedzeným datasetom je k dispozícii súbor obrázkov, označený ako \mathcal{X} , a označenie tried ako \mathcal{Y} . Hlavnou úlohou je navrhnúť klasifikátory f_θ , popísané vektorom parametrov θ . Klasifikátory sa navrhujú tak, aby predpovedali označenie y pre ľubovoľný obrázok x , kde $x \in \mathcal{X}$ a y je predpovedané tak, že $y = f_\theta(x)$. Predpokladá sa, že súbor údajov predstavuje K jedinečných kategórií a tréningová množina vybraná z \mathcal{X} a \mathcal{Y} , obsahuje N obrázkov pre každú z týchto kategórií. Cieľom je naučiť f_θ presne rozlišovať medzi dostupnými kategóriami.

Oba datasety sú popísané v kapitole 3, počty tried boli nastavené $K = 2$, čo sa označuje ako binárna klasifikácia. V oboch prípadoch obsahovali po 1000 snímok na jednu triedu. Rozdelenie prebehlo v pomere 70:30, kde 70 percent reprezentuje tréningová množina a 30 percent testovacia množina. Tréningová množina bola následne rozdelená v pomere 80:20, kde 80 percent bolo použité na trénovanie a 20 percent na validáciu. Zastúpenie tried v tréningovej, validačnej aj testovacej množine bolo v každom kroku 1:1, aby sa predišlo problému s nevyrovnaným datasetom. Vizualne zobrazenie rozdelenia datasetu je zobrazené na 4.1.

Znižovanie datasetu prebiehalo v tréningovej množine N a validačnej množine V , kde hodnota N a V korešponduje s počtom obrázkov v danej kategórii. Hodnotenie metód bude prebiehať na nezávislých testovacích obrazoch z množiny T , ktoré do tréningového procesu nezasahovali a ich počet sa nemenil. Počty obrazov v daných množinách sú zobrazené v tabuľke 4.1.



Obr. 4.1: Rozdelenie datasetu v množinách

Tab. 4.1: Znižovanie trénovacej a validačnej množiny dát

Počet obrazov na triedu		
N (Trénovanie)	V (Validácia)	T (Testovanie)
560	140	300
320	80	300
160	40	300
80	20	300
40	10	300
20	5	300
10	2	300
5	1	300

4.2 Popis klasifikačného modelu

4.2.1 Nastavené parametre

Architektúra

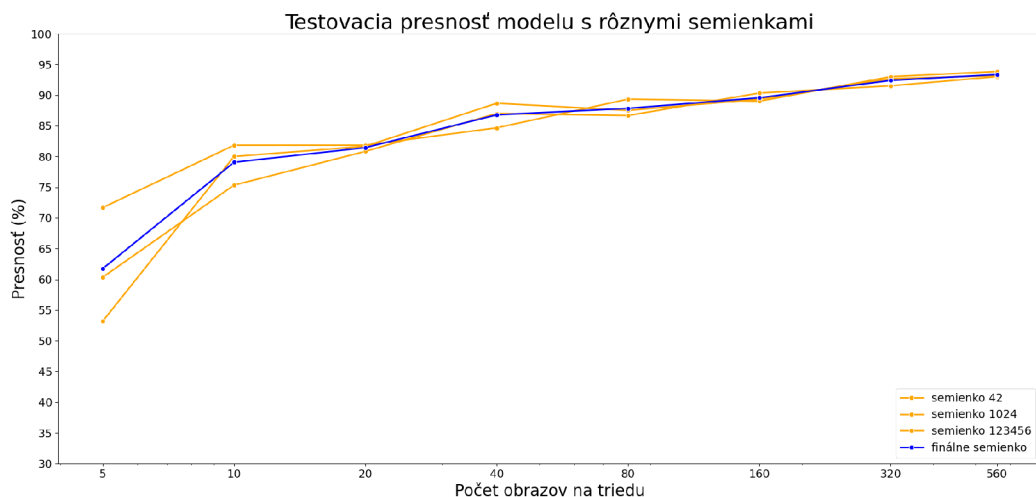
Na navrhnuté experimenty boli použité architektúry siete ResNet a ich modifikácie. Následne sa realizovali experimenty predovšetkým s architektúrou siete ResNet18 popísanou v kapitole 1.4.1, pretože funguje na princípe reziduálnych blokov, ktoré pomáhajú bojovať s problémom miznúceho gradientu tým, že umožňujú prúdenie gradientov cez sieť priamo, bez prechodu cez nelineárne aktívácie, a taktiež z dôvodu časovej realizácie experimentov. [29]

”Batch size”

Velkosť dávky, vyjadrená parametrom ”batch size”, je kritickým hľadiskom efektivity tréovania. Zvolená hodnota bola 8, čo znamená že model neurónovej siete bude aktualizovať svoje váhy na základe chyby výpočtu, ktorý bude vychádzať z 8 tréovacích príkladov naraz.

”Seed”

Pre reprodukovateľnosť siete bol použitý parameter ”seed”, alebo aj semienko, ako generátor náhodných čísel. Rôzne nastavenie môže mať za následok mierne odlišné výsledky tréovania modelu v dôsledku inicializácie váh a stochastickej povahy algoritmov. Minimálna hranica tréovacej množiny bola nastavavená na 5 obrazov z každej triedy, tak sa používali 3 odlišné semienka, aby sa aspoň čiastočne zabránilo stochasticite. Výsledná hodnota bola vypočítaná ako priemer zo všetkých 3 semienok zobrazená na obrázku 4.2 modrou farbou.

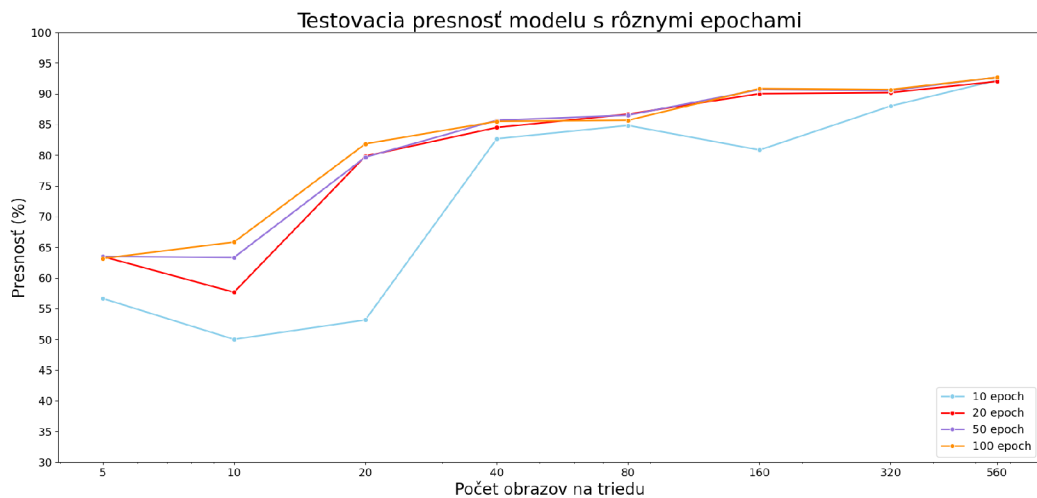


Obr. 4.2: Graf v závislosti dosiahnutej testovacej presnosti na počet obrazov na triedu. V grafe sú zobrazené pseudokrivky vplyvu počtu obrazov na triedu pre 3 semienka znázornené oranžovou farbou s architektúrou ResNet18. Výsledná hodnota braná ako ich priemer zobrazená modrou farbou.

Počet epoch

Nastavenie počtu epoch bol kľúčovým parametrom pri návrhu experimentov, pretože vyšší počet epoch môže viesť k lepšiemu výkonu za predpokladu, že sa primerane

zabrání nadmernému prispôsobeniu pomocou regularizačných techník, ale na druhej strane môže dlhší tréning viesť k nadmernému preučeniu, čo znamená, že model bude veľmi dobre fungovať na tréningových dátach, ale jeho schopnosť generalizovať na nové, nevidené dáta bude slabá. Okrem toho dlhší tréningový proces zvyšuje časovú a výpočtovú náročnosť, čo môže byť nevýhodné v prípade obmedzených zdrojov alebo keď je potrebné rýchlo iterovať a experimentovať s rôznymi modelmi. Preto bolo potrebné nájsť ideálnu rovnováhu medzi počtom epoch a výkonom modelu. Testovacia presnosť modelu s rôznym nastaveným epoch so znižujúcimi sa obrazmi na jednu triedu je zobrazená na obrázku 4.3.



Obr. 4.3: Graf v závislosti dosiahnutej testovacej presnosti na počet obrazov na triedu na základe zmeny počtu epoch. V grafe sú zobrazené pseudokrivky vplyvu počtu obrazov na triedu pre 4 rôzne hodnoty epoch s architektúrou ResNet18.

Pre výber optimálneho hyperparametru počtu epoch sa zrealizoval experiment, v ktorom sa model učil na rôznych parametroch s hodnotami $E = \{10, 20, 50, 100\}$. Na základe použitia štatistiky pomocou analýzy medzi skupinami testovacích presností pomocou vykonania Kruskal-Wallis testu, ktorý sa používa na zistenie, či existujú štatisticky významné rozdiely medzi skupinami testovacích presností modelu naprieč rôznymi počtami epoch. Kruskal-Wallis test je neparametrická alternatíva jedno-smerného rozptylovej analýzy (ANOVA) a je vhodný, keď sa nemôže predpokladať normálne rozdelenie dát. Po vykonaní testu na hladine významnosti $\alpha = 0,05$ vyšla p-hodnota = 0,55, na základe ktorej vyplýva, že zmeny v počte epoch (10, 20, 50, 100) nemali štatisticky významný vplyv na presnosť testovania modelu. To znamená, že zvyšovanie počtu epoch v tomto prípade nepredstavuje významné zlepšenie alebo zhoršenie výkonu modelu. Toto môže naznačovať, že presnosť modelu nemá zásadný

vplyv na počet nastavených epoch v rámci testovaného rozsahu.

Následne bol ešte vykonaný Mann-Whitneyho test na porovnanie jednotlivých párov, ktorý porovnáva výsledky medzi každým možným párom epoch (10 a 20, 10 a 50, atď.). Tento test porovnáva dve nezávislé vzorky a zisťuje, či pochádzajú z rovnakého rozdelenia. Výsledky každého testu sú interpretované na základe p-hodnoty vo vzťahu k vopred určenej úrovni významnosti $\alpha = 0,05$, čím sa rozhodne, či medzi skupinami existujú štatisticky významné rozdiely. Výsledky s p-hodnotami sú zaznačené v tabuľke 4.2.

Tab. 4.2: Výsledky Mann-Whitneyho testu

Porovnanie	p-hodnota
Epoch 10 vs. Epoch 50	0,279
Epoch 10 vs. Epoch 100	0,234
Epoch 20 vs. Epoch 50	0,834
Epoch 20 vs. Epoch 100	0,721
Epoch 50 vs. Epoch 100	1,000

Výsledky naznačujú, že zvyšovanie počtu epoch od 10 po 100 nemá štatisticky významný vplyv na výkonnosť modelu. Toto môže poukazovať na to, že model možno dosiahol svoju asymptotickú úroveň učenia skôr, než bolo dosiahnuté 100 epoch, a ďalšie tréningy nad tento počet neprináša zlepšenie. Na základe čoho bolo rozhodnuté, že v použitých experimentoch sa bude používať 50 epoch.

4.2.2 Optimalizačný algoritmus a chybová funkcia

Pre optimalizačný algoritmus bol zvolený Adam popísaný v kapitole 1.3.3 pomocou vzorca 1.10 na úpravu váh, ktorý zohľadňuje informácie o gradiente z minulých iterácií a je relatívne jednoduchý algoritmus na implementáciu v knižici PyTorch. [17]

Na aplikáciu chybovej funkcie bola použitá vzájomná entropia popísaná v kapitole 1.3.2 pomocou vzorca 1.3. Bola vybraná z dôvodu jej efektívnosti v kontexte klasifikácie, ľahkej implementovateľnosti a väčšiemu robustnosti voči preučeniu. Krok učenia bol nastavený experimentálne na hodnotu 0,0001 spolu s parametrom "weight decay", ktorý funguje tak, že sa do stratovej funkcie pridá ďalší člen a penalizuje veľké váhy vo vytvorenom modeli, čo pomáha znižovať nadmerné preučenie modelu.

Krok učenia bol optimalizovaný pomocou knižnice "StepLR" poskytujúci jednoduchý, ale účinný spôsob ako upravovať rýchlosť učenia počas tréningu, čo môže

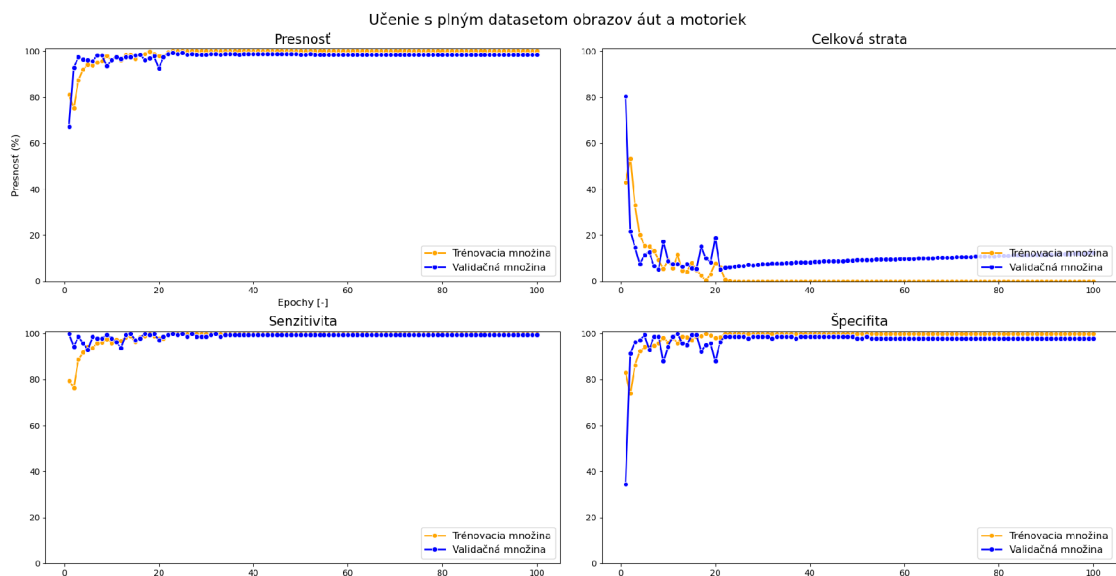
viest k lepšej výkonnosti a konvergencii modelu. Nastavenie spočívalo v znížení rýchlosti učenia o faktor gama (hodnota 0,5) každých 5 epoch. Prístup vedie k efektívnejšej dynamike učenia tým, že optimalizátoru umožní robiť väčšie kroky v počítačových fázach (keď je ďaleko od minima) a menšie kroky neskôr (na doladenie svojho priblíženia k minimu).

4.3 Učenie s plným datasetom

Na začiatku bolo potrebné zrealizovať učenie siete na plnom datasete, aby sa overila efektívnosť a správnosť nastavenia parametrov siete a samozrejme aj overeniu samotných datasetov pred tým, ako sa prejde k experimentom s modifikovanými dátovými sadami. Bola použitá architektúra ResNet18 bez inicializovaných váh. Nastavenie parametrov na "batch-size", optimalizačný algoritmus a chybovej funkcie zostali nezmenené. Počet epoch bol v tomto prípade nastavený na hodnotu 100, aby bolo zvýraznené, že po 50. epochách nenastáva v žiadnom parametri významná zmena a všetky vykreslené hodnoty zostávajú konštantné. Delenie datasetu na tréningovanie, validáciu a testovanie prebehlo rovnako ako je znázornené na obrázku 4.1.

4.3.1 Učenie s datasetom áut a motoriek

Výsledky učenia s datasetom na binárnu klasifikáciu áut a motoriek je popísané metrikami zobrazené na grafoch 4.4.

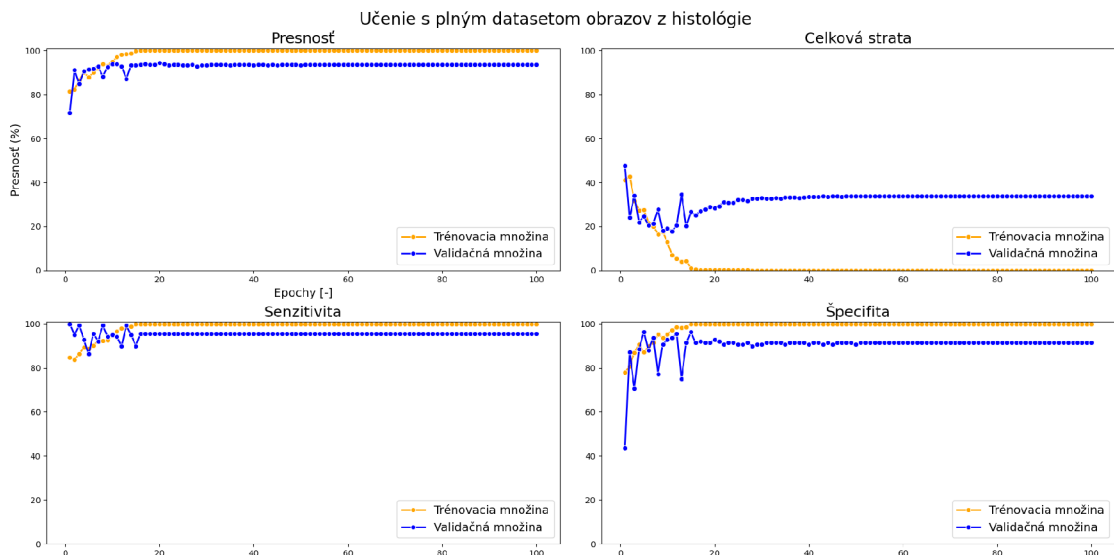


Obr. 4.4: Celkové výsledky modelu s plným datasetom áut a motoriek

Pre dosiahnutú presnosť, sensitivitu aj špecificitu sú vypočítané hodnoty na validačnej množine približne na rovnakej úrovni (okolo 95 %). Výsledky učenia modelu sú vypočítané na testovacej množine dát. Z vypočítaných metrík sa ukazuje, že model je schopný presne identifikovať obe skupiny, čo je možné vidieť na vypočítanej hodnote F1 skóre dosahujúcej takmer 93 % a, presnosti na úrovni 92 %, senzitivite rovnkej 92 % a špecificite takmer 93 %.

4.3.2 Učenie s datasetom obrazov histológie

Učenie na plnom datasete obrazov z histológie je vykreslené pomocou grafou znázornené na obrázku 4.5. Senzitivita, špecificita aj presnosť sa pohybujú na úrovni 90-95 % validačnej množine, čo naznačuje, že model má vysokú schopnosť správne klasifikovať pozitívne aj negatívne prípady a zároveň udržiavať vysokú úroveň celkového výkonu na nezávislých dátach.



Obr. 4.5: Celkové výsledky modelu s plným datasetom obrazov z histológie

Naučený model otestovaný na množine dát preukázal dobrú generalizačnú schopnosť, kde sa presnosť pohybuje na úrovni 92,66 %, F1-skóre rovnkej 92,67 % a metriky senzitivita so špecificitou ukazujú zhodne po 92,66 %.

5 Popis experimentov

Kapitola poskytuje komplexný pohľad popisu navrhnutých experimentov. V prvej časti sa venuje vplyvu architektúry siete, konkrétne siete typu ResNet a ich modifikácie. Ďalej sa popisuje transferové učenie ako metóda pre zlepšenie výsledkov. Tretia a štvrtá sekcia sa zaoberajú dátovou augmentáciou, kde sa rozlišuje medzi online a offline prístupmi. Nakoniec, v poslednej časti kapitoly sa zameriava na techniku využívajú sieť typu NST a jej aplikácie v kontexte učenia pri obmedzenom množstve dát.

5.1 Popis vplyvu architektúry siete ResNet

ResNet modely dosahujú vynikajúce výsledky v mnohých oblastiach videnia počítačov vďaka svojej inovatívnej architektúre s reziduálnymi spojeniami, ktoré umožňujú signálom prechádzať cez viaceré vrstvy bez straty informácií. V našom experimente na začiatku prebehne porovnanie 3 architektúr ResNet - ResNet18, ResNet50 a ResNet152. Stručné porovnanie použitých modelov [60] :

ResNet18

Má približne 11 miliónov parametrov. Je to najjednoduchší a najľahší model v rade ResNet, vhodný pre menšie datasety a menej výpočtovo náročné úlohy. Vzhľadom na menší počet vrstiev a parametrov je ResNet18 rýchlejší na trénovanie a menej náchylný na pretrénovanie, čo je vhodné pri obmedzenom množstve dát. Môže však dosahovať nižšiu presnosť v porovnaní s hlbšími modelmi, ak je k dispozícii veľa dát.

ResNet50

Obsahuje okolo 23,5 miliónov parametrov. Je to stredná varianta, ktorá poskytuje dobrý kompromis medzi výkonom a počtom parametrov. Tento model poskytuje lepšiu schopnosť zovšeobecnenia na rozmanitejších alebo zložitejších dátových sadách. Je schopný naučiť sa zložitejšie vzory, čo môže viesť k vyššej presnosti klasifikácie.

ResNet152

Je tam približne 60 miliónov parametrov. Tento model je najhlbší z uvedených, vhodný pre veľmi komplexné úlohy klasifikácie, kde je dostupných veľa trénovacích dát. Ideálny pre veľmi rozsiahle a komplexné datasety, kde je dôležitá maximálna možná presnosť. Jeho hĺbka umožňuje modelu zachytiť veľmi jemné a zložité vzory v dátach, ale za cenu dlhšieho trénovania a vyššieho rizika pretrénovania, ak nie sú opatrenia, ako regulácia alebo dostatočne veľký dataset.

5.2 Popis transferového učenia

Transferové učenie je technika, pri ktorej sa model, ktorý bol vopred natrénovaný na veľkom a všeobecnom súbore údajov, upravuje sa a doladuje pre konkrétnu úlohu alebo menší súbor údajov. V našom prípade sa používala architektúra siete ResNet18 s nastaveným parametrom váh na hodnotu *ImageNet-1K*, čo je veľký súbor údajov obsahujúci približne 1,3 milióna obrázkov rozdelených do 1000 rôznych tried. Modely natrénované na tomto súbore údajov dokážu rozpoznať a klasifikovať širokú škálu objektov a scén, vďaka čomu sú veľmi užitočné pre mnohé úlohy klasifikácie.

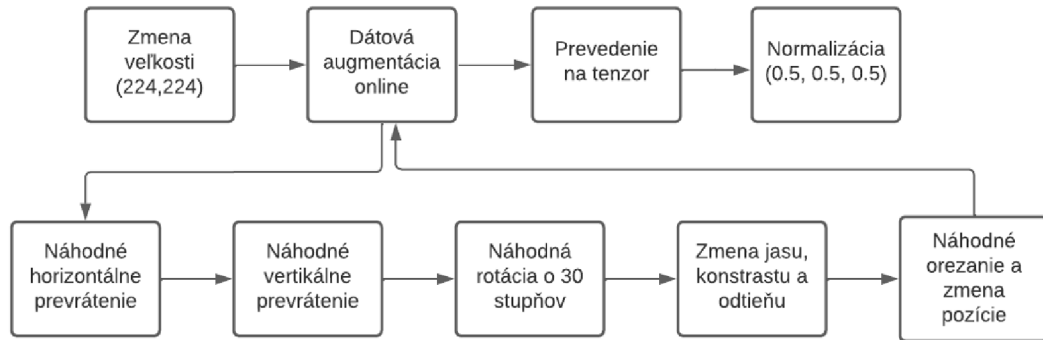
Následne sa získava počet vstupných prvkov poslednej vrstvy modelu, ktorý je dôležitý pre následné prispôsobenie modelu cieľovému počtu tried. V poslednom kroku nastáva nahradenie poslednej vrstvy novou vrstvou, ktorá má toľko výstupov, koľko je tried v cieľovom súbore údajov - v našom prípade 2.

5.3 Popis dátovej augmentácie online

Implementácia dátovej augmentácie označená ako online, bola aplikovaná na tréningovú množinu za účelom zväčšenia počtu obrazov pomocou rôznych transformácií, ktoré sú definované pomocou knižnice PyTorch a jej modulu *transforms*. Tento prístup znamená, že transformácie obrazov sa vykonávajú dynamicky na každý obrázok v tréningovej množine v reálnom čase, teda práve v momente, keď je obrázok načítavaný do modelu na spracovanie.

Na začiatku prebehlo zmenšenie alebo zväčšenie obrázkov, aby mali pevne stanovenú veľkosť - 224×224 pixelov. Tento krok je dôležitý pre modely, ktoré vyžadujú pevne stanovenú veľkosť vstupu ako je ResNet18. Následne prebehla samotná dátová augmentácia pozostávajúca z viacerých transformácií. Prvá z nich náhodne prevráti obrázky horizontálne, čo môže pomôcť modelu naučiť sa rozpoznávať objekty bez ohľadu na ich orientáciu. Následne sa náhodne rotujú obrázky o určitý počet stupňov (± 30 stupňov). Táto variabilita pomáha modelu zvládať rôzne uhly pohľadu objektov. Ďalej sa pokračuje s náhodnou zmenou jasnosti, kontrastu, sýtosti a odtieňu obrázka, čo simuluje rôzne svetelné podmienky a nastavenia. V poslednom kroku sa náhodne orezá a mení veľkosť obrázkov na rozmery 224×224 pixelov. Toto je silný spôsob augmentácie, pretože mení aspekty ako pozícia a mierka objektov v obrázku.

Nakoniec po dátovej augmentácii sa konvertujú obrázky na tenzory, čo je vhodný formát pre spracovanie neurónovými sieťami v knižnici PyTorch. V poslednom kroku sa normalizujú hodnoty pixelov obrázkov tak, že upraví hodnoty každého kanálu (červený, zelený a modrý), aby mali nulový priemer a jednotkovú štandardnú odchýlku, čo pomáha stabilizovať učenie upravením rozsahu vstupných dát.



Obr. 5.1: Vývojový diagram dátovej augmentácia online

5.4 Popis dátovej augmentácie offline

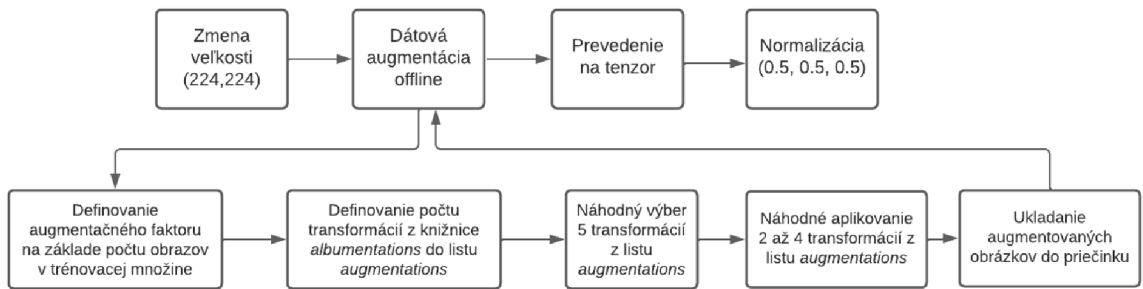
Dátová augmentácia offline spočíva vo vlastnej vytvorenej funkcii, ktorá definuje prácu s augmentáciou obrázkov v datasete. V skratke povedané, najprv sa vytvoria augmentované obrázky, uložia sa a následne sa učí sieť. Definoval sa tzv. augmentačný faktor popisujúci koľkokrát sa má aplikovať sada augmentačných transformácií na každý obrázok v datasete. Tento faktor je kľúčový pre rozhodnutie o tom, koľko augmentovaných kópií každého obrázka sa vygeneruje, čo umožňuje zvýšiť počet vzoriek v tréningovom datasete, a tým zlepšiť robustnosť a generalizačnú schopnosť modelu. Je navrhnutý tak, aby v prípade menšieho počtu pôvodných dát vytvoril viac augmentovaných kópií, čím kompenzuje nižší počet pôvodných obrázkov. Definované hodnoty augmentačného faktoru sú zobrazené v tabuľke 5.1.

Tab. 5.1: Výpočet výsledných obrazov po aplikácii augmentačným faktorom

Počet obrazov na triedu		
N (Tréningovanie)	A (Augmentačný faktor)	$N \times A$ (Tréningovanie po augmentácií)
560	0	560
320	1	560
160	3	560
80	7	560
40	14	560
20	28	560
10	56	560
5	112	560

Pre definovanie transformácií sa využíva knižnica *alumentations*, ktorá posky-

tuje rôzne transformácie a ich počet sa líši v závislosti od použitého datasetu.



Obr. 5.2: Vývojový diagram dátovej augmentácia offline

Dataset obrazov áut a motoriek

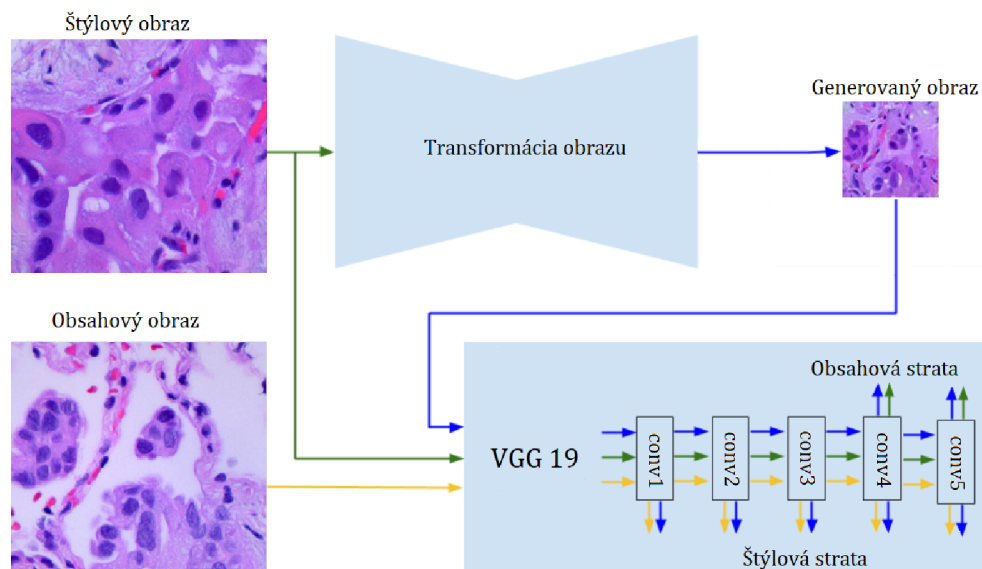
Dataset obsahuje rozličné obrázky z oboch skupín, vďaka čomu je zachovaná vysoká rôznorodosť. Obrázky sú zaznamenané z odlišných svetelných podmienok či z rôznych uhlov. Z tohto dôsledku sa definovalo 16 transformácií od transformácie šumu a rozmazania, cez geometrické a farebné transformácie, lokálne zlepšenie obrazu až cez simuláciu poškodenia pixelov.

Dataset obrazov z histológie

V prípade druhého datasetu zameraného na medicínske obrázky z histológie sa sada transformácií prispôbila daným podmienkam a celkovo 7 definovaných transformácií predstavuje jednoduchší, a možno menej intenzívny prístup k augmentácii oproti predošlému datasetu. Zameriava sa na základné geometrické transformácie, zmiernenie šumu, farebné úpravy a simuláciu vypadnutí pixelov, ale s miernejšími limitmi a pravdepodobnosťami. Celkovo boli transformácie vyberané, aby nezmenili obrázky príliš drasticky, čo môže pomôcť zachovať originálne vlastnosti a kvalitu obrázkov, zatiaľ čo stále poskytuje rozmanitosť potrebnú na tréningovanie modelu.

5.5 Popis siete na prenos neurónového štýlu

Typ dátovej augmentácie využívajúci sieť prenos neurónového štýlu, skrátene NST, je pokročilejšou formou augmentácie, ktorý generuje nové jedinečné obrázky zložené z vizuálneho štýlu jedného obrázka a obsahu iného obrázka. [46] Tento prístup je založený na použití siete VGG19 na oddelenie a následné spojenie týchto dvoch aspektov jedného obrázka. Celkový proces je znázornený na obrázku 5.3.



Obr. 5.3: Grafické zobrazenie tvorby jedinečných generovaných obrazov zo štýlového a obsahového obrazu pomocou transformácie obrazu modelom NST.

Predspracovanie obrázkov

Obrázky sú načítané a predspracované pomocou transformácií, ktoré zahŕňajú zmenšenie veľkosti, centrováný orez a konverziu na tenzor. Proces zabezpečuje, že vstupné obrázky sú kompatibilné s architektúrou CNN.

Vrstvy pre stratové funkcie obsahu a štýlu

- **Obsahová strata**
Počíta strednú kvadratickú chybu (mean-squared error, MSE) medzi cieľovými a vstupnými vlastnosťami, čo umožňuje zachovanie obsahu základného obrázka.
- **Štýlová strata**
Používa gramovú maticu na kvantifikáciu štýlu a počíta MSE medzi cieľovými a vstupnými štýlovými vlastnosťami.

Normalizačná vrstva

Normalizuje vstupné obrázky na štandardizované hodnoty, ktoré sú očakávané predtrénovaným CNN modelom, čo zlepšuje výkonnosť a stabilitu učenia.

Model a optimalizácia

Model je zostavený ako sekvencia vrstiev, pričom do nej sú vkladané vrstvy pre stratu obsahu a štýlu na špecifických miestach podľa architektúry CNN. Používa sa model VGG19, popísaný v kapitole 1.4.1, ktorý je populárny v úlohách NST kvôli svojej efektívnosti v extrakcii vlastností obrázkov. Na úpravu vstupného obrázka, tak aby minimalizoval celkovú stratu štýlu a obsahu, sa používa optimalizačný algoritmus L-BFGS. Tento krok iteratívne upravuje obrázok s cieľom dosiahnuť želaný vizuálny efekt. Na úpravu vstupného obrázka, tak aby minimalizoval celkovú stratu štýlu a obsahu, sa používa optimalizačný algoritmus L-BFGS používajúci sa pre metódy, kde je potrebné efektívne riešiť problémy s veľkým počtom parametrov. Počas každej iterácie sa vypočíta strata, vykoná spätné šírenie chyby a aktualizuje obrázok tak, aby sa zlepšila jeho kvalita vzhľadom na definované kritériá štýlu a obsahu. Počas procesu sa sleduje kombinovaná strata štýlu a obsahu, ukladá sa najlepší dosiahnutý výsledok, čo zabezpečuje, že výstupný obrázok je ten s najnižšou celkovou stratou.

Nastavenie hyperparametrov

Jedným zo základných parametrov je výstupná veľkosť generovaného obrázka, ktorá je nastavená na 112×112 pixelov. Veľkosť bola nastavená experimentálne na základe rýchlosti výpočtu, avšak sa berie do úvahy strata detailov v obrázku. Ďalším dôležitým aspektom sú váhy straty, kde váha straty štýlu je nastavená na 1 milión, čo značne zvyšuje dôležitosť vizuálneho štýlu obrázka v porovnaní so zachovaním jeho obsahu, pre ktorý je váha straty obsahu nastavená len na 1. Nastavenie reflektuje prioritu štýlových charakteristík vo výslednom obrázku.

Na výpočet straty obsahu a štýlu sú vybrané konkrétne vrstvy z modelu VGG19. Vrstvy pre obsah, typicky hlbšie vrstvy ako "conv4", zachytávajú detaily obsahu, zatiaľ čo vrstvy pre štýl, rozsahujúce sa od "conv1" do "conv5", sú zodpovedné za zachytenie rôznych úrovní textúr a vzorov štýlu.

Počet generovaných obrazov

Pomocou modelu NST by sa dalo vygenerovať obrovské množstvo jedinečných obrázkov. V prípade ak je k dispozícií N obrazov na tréning, označí sa obsahový obraz A a štýlový obraz B , tak sa dokáže vygenerovať $\frac{N \cdot (N-1)}{2}$. Taktiež ak sa zoberie tých istých N obrazov a obráti sa ich poradie, tak obsahový obraz bude B a štýlový obraz A , opäť sa dokáže vygenerovať $\frac{N \cdot (N-1)}{2}$. Čo vedie k záveru, že ak sa zoberie N obrazov, tak celkovo je model NST schopný vygenerovať $N \cdot (N - 1)$ obrazov. Teoretické množstvo vytvorených obrazov podľa tohto výpočtu je uvedené v tabuľke 5.2.

Tab. 5.2: Teoretické množstvo vytvorených obrazov pomocou modelu NST

Počet obrazov na triedu		
N (Trénovanie)	NST (Teoretický počet vytvorených obrazov)	N + NST (Trénovanie po augmentácií)
560	313 040	313 600
320	102 090	102 410
160	25 540	25 700
80	6 320	6 400
40	1 560	1 610
20	380	400
10	90	100
5	20	25

Avšak pri počte vytvorených obrazov pomocou neurónovej siete typu NST sa brala do úvahy časová náročnosť jedného generovaného obrazu, ktorá veľmi úzko súvisí s veľkosťou výstupného obrazu. Časová náročnosť na základe veľkosti výstupného obrazu je uvedená v tabuľke 5.3. Na základe tohto výstupu sa rozhodlo, že generovaný obraz bude mať veľkosť 112×112 pixelov.

Tab. 5.3: Časová náročnosť augmentácie pomocou NST

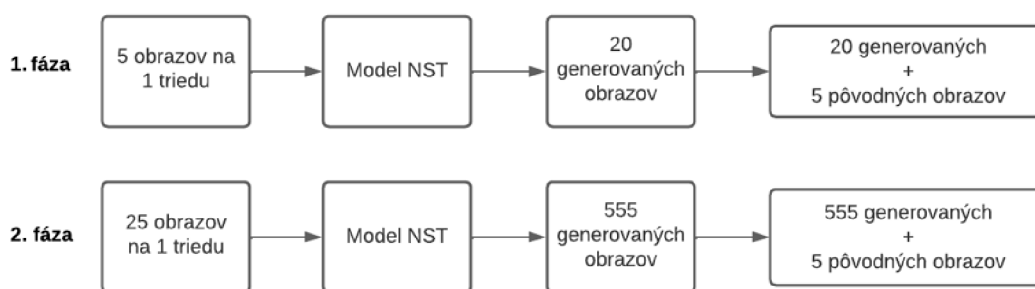
Veľkosť obrazu [pixel x pixel]	Čas na 1 obrázok [sekundy]
112	40
224	60
768	90
1024	140

V prípade, ak by sa zobralo teoretické množstvo všetkých vytvorených obrazov, čo zodpovedá 449 040 obrazom, tak generovanie obrazov na jednu triedu by trvalo približne 4990 hodín, zodpovedajúce 209 dní. Na základe čoho sa rozhodlo, že experiment sa bude realizovať do maximálneho počtu 560 obrazov na jednu triedu, označené v tabuľke 5.4. Hodnoty označené hviezdičkou v tabuľke 5.4 v stĺpci počet vytvorených obrazov bolo potrebné rozdeliť na 2 fázy, pretože ich počet neumožňoval doplniť obrazy do maximálneho limitu. Platí to pre triedy 5, 10 a 20 obrazov. V prvej fáze nastalo generovanie maximálneho počtu obrazov, ktoré potom vstupovali do 2. fázy kde sa z nich opäť generovali výstupné obrazy. Konkrétny príklad pre

triedu s 5 obrazmi je znázornený na vývojovom diagrame 5.4.

Tab. 5.4: Dátová augmentácia pomocou modelu NST

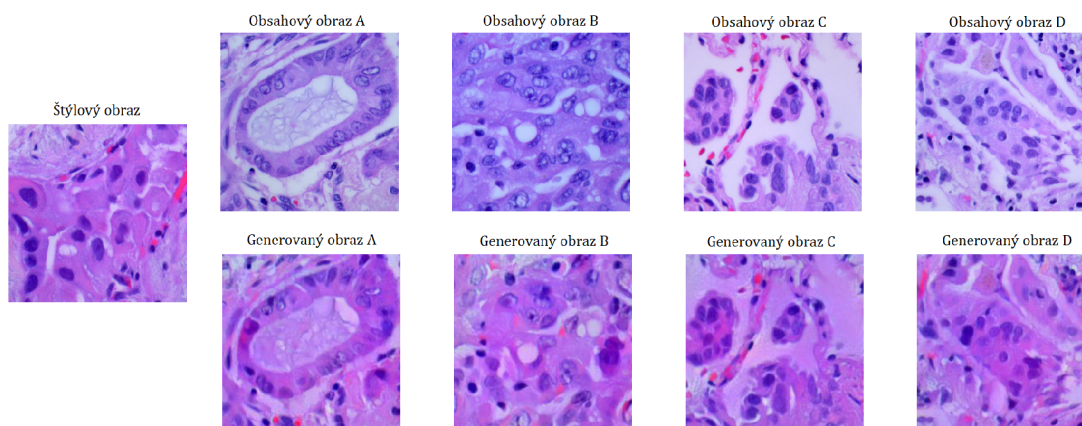
Počet obrazov na triedu		
N (Trénovanie)	NST (Počet vytvorených obrazov)	N+NST (Trénovanie po augmentácii)
560	0	560
320	240	560
160	400	560
80	480	560
40	520	560
20	540*	560
10	550*	560
5	555*	560



Obr. 5.4: Vývojový diagram modelu NST pre nižší počet obrazov v triede

Generované obrazy

Príklad generovaných obrazov z histológie pre triedu adenokarcinóm je viditeľný na obrázku 5.5. Zobrazenie je organizované do dvoch radov, pričom každý obsahuje štyri obrázky označené písmenami od A do D. Vrchný riadok s názvom "Obsahový obraz" predstavuje pôvodné obsahové obrázky, ktoré slúžia ako šablóny pre proces obsahovej transformácie. Každý z týchto obrázkov ilustruje rozličné histologické vzory a charakteristiky. Spodný riadok s názvom "Generovaný obraz" predstavuje výsledky modelu NST. Tieto obrázky sú výsledky modelu replikovať obsahové charakteristiky príslušných obrázkov nad nimi, spolu so štýlovým obrazom.



Obr. 5.5: Náhodne vybrané obrázky datasetu obrazov histológie z triedy adenokarcinóm. V prvej rade obsahuje 4 náhodne vybrané obsahové obrázky, kde s kombináciou so štýlovým obrazom vzniknú generované obrázky. Napríklad obsahový obraz A s kombináciou so štýlovým obrazom vznikne generovaný obraz A.

Rovnaký proces je možné sledovať na druhom datasete s triedou obrazov motoriek na obrázku 5.6.



Obr. 5.6: Náhodne vybrané obrázky datasetu obrazov áut a motoriek z triedy motoriek. V prvej rade obsahuje 4 náhodne vybrané obsahové obrázky, kde s kombináciou so štýlovým obrazom vzniknú generované obrázky. Napríklad obsahový obraz A s kombináciou so štýlovým obrazom vznikne generovaný obraz A.

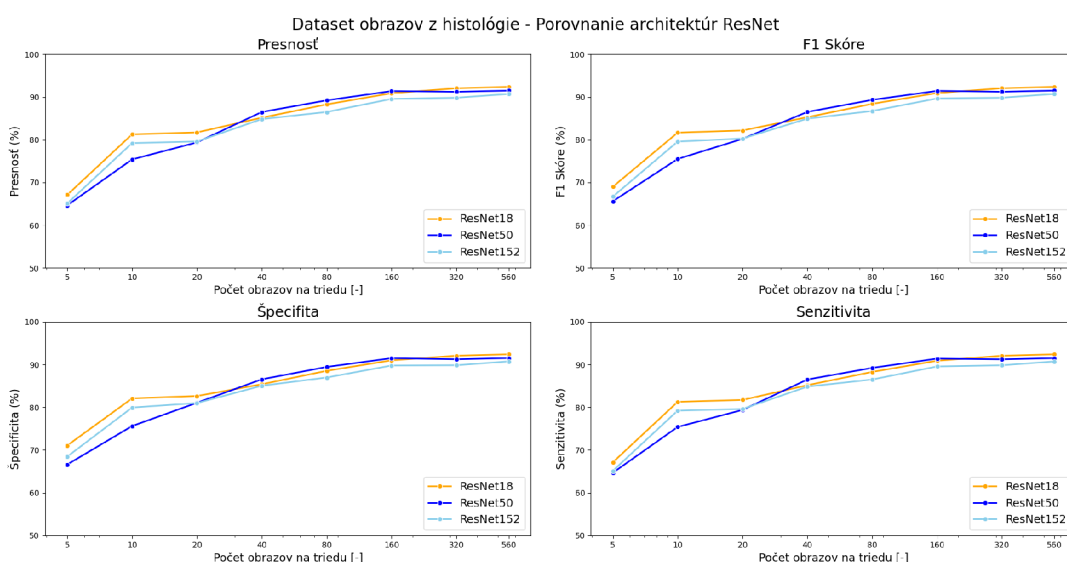
6 Výsledky

Na štatistické vyhodnotenie výsledkov pre oba datasety boli použité metriky popísané v kapitole 1.3.5, ktoré sa hodnotili na nezávislom testovacom datasete.

6.1 Vplyv architektúry siete ResNet

6.1.1 Učenie s datasetom obrazov z histológie

Grafické porovnanie na obrázku 6.1 vykresľuje vyhodnocovanie metriky. Vo vyšších počtoch obrazov na triedu ($N > 20$) není možno jednoznačne určiť, ktorá architektúra ukazuje najlepšie výsledky. V prípade menších počtoch ($N < 20$) nastávajú viditeľne viacej odlišné rozdiely, pri ktorých najlepších výsledkov pri všetkých metrikách dosahuje architektúra siete ResNet18, kde v prípade $N = 10$ architektúra ResNet50 dosiahla presnosť 75,3 % a ResNet18 približne o 6 % viacej.

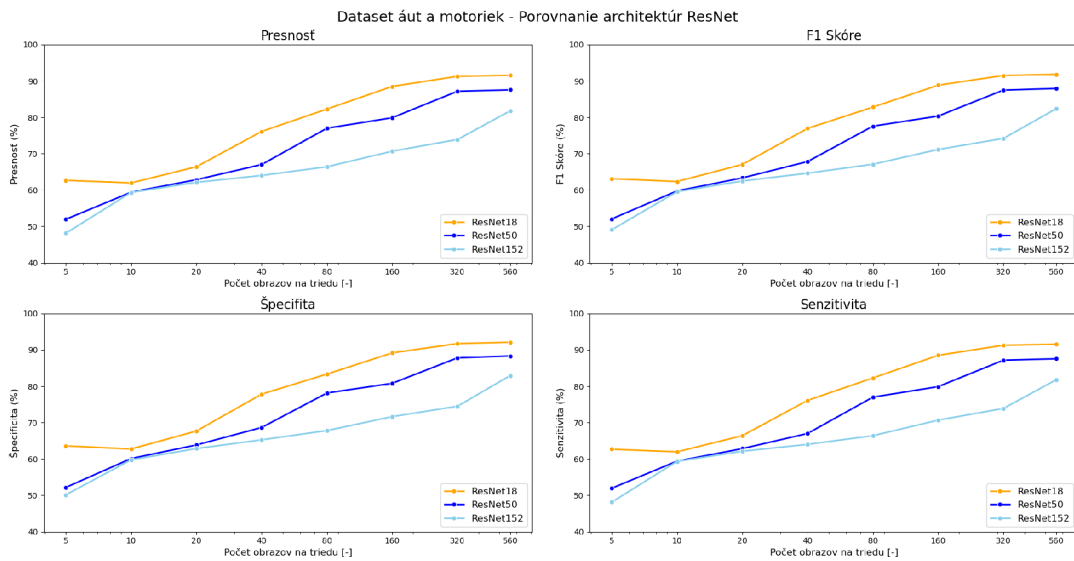


Obr. 6.1: Grafické porovnanie na testovacej množine pomocou rozdielnych architektúr ResNet na datasete obrazov z histológie

6.1.2 Učenie s datasetom áut a motoriek

Na základe grafického porovnania na obrázku 6.2 sú oveľa viacej viditeľné rozdiely v porovnaní architektúr, kde ResNet18 dosahuje najlepšie výsledky v každom modeli pri všetkých metrikách. V prípade najmenšieho počtu obrazov na triedu sa líšia

architektúry ResNet18 a ResNet152 až o necelých 15 %. Druhý extrém počtu obrazov na triedu (najvyšší počet) ukazuje taktiež rozdiel približne 10 % v prospech architektúry ResNet18.



Obr. 6.2: Grafické porovnanie na testovacej množine pomocou rozdielnych architektúr ResNet na datasete áut a motoriek

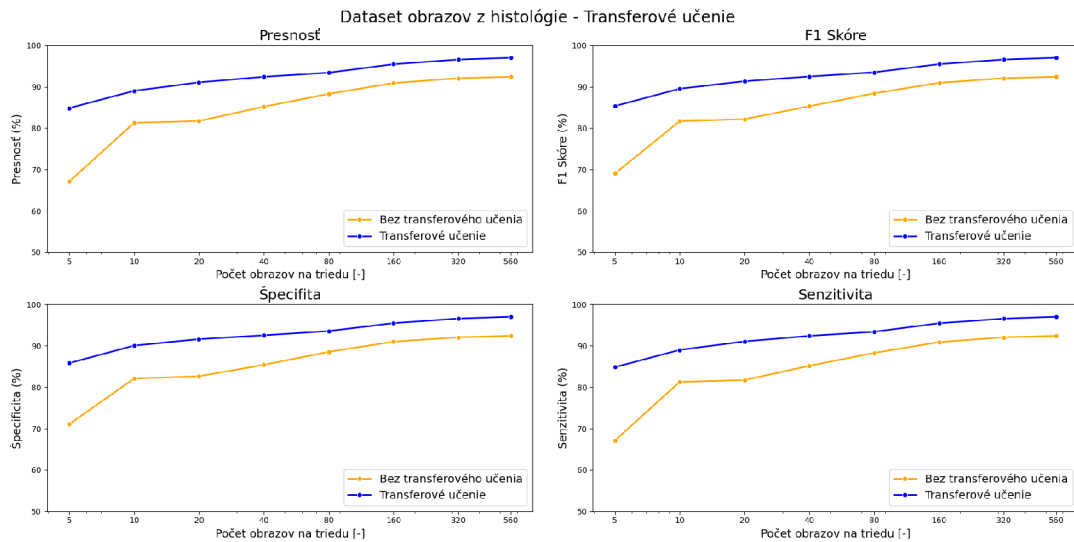
Záver

V tomto prípade možno konštatovať, že vysoký počet parametrov v neurónovej sieti nie je nevyhnutne spojený s vyššou schopnosťou efektívne riešiť klasifikačné úlohy, obzvlášť ak sa jedná o jednoduchý klasifikačný problém. Dĺžka trvania ResNet18 pri 1 datasete na všetkých 3 semienkach sa pohybovala na úrovni 3 hodín, následne s architektúrou ResNet50 to bolo 3,5 hodiny a s výpočetne najnáročnejšou ResNet152 na úrovni 4,5 hodiny. Na základe týchto porovnaní a výsledkov sa ďalšie experimenty realizovali s architektúrou siete ResNet18.

6.2 Transferové učenie

6.2.1 Učenie s datasetom obrazov z histológie

Vypočítané metriky modelov na testovacej množine sú znázornené na obrázku 6.3. Každá metrika je porovnávaná s modelom bez použitia dátovej augmentácie a bez naučených váh, pričom bola použitá rovnaká architektúra - ResNet18. Aj v prípade nižších obrazov na triedu boli schopné modely ukázať testovaciu presnosť 84,8 % z pôvodných 67,1 %, čo predstavuje zlepšenie až o 17,7 %. V tomto prípade rovnaké zlepšenie nastalo aj u F1-skóre. V prípade väčších počtov obrazov na triedu sa všetky testovacie metriky výrazne zlepšili, približne o 5 %. Predtrénovanie modelu na inom, väčšom datasete môže zvýšiť jeho schopnosť efektívne sa učiť špecifické úlohy, ktoré môžu byť relevantné aj pre obrazy z histológie, vďaka čomu sa zvýšila výrazne jeho presnosť vo všetkých metrikách.

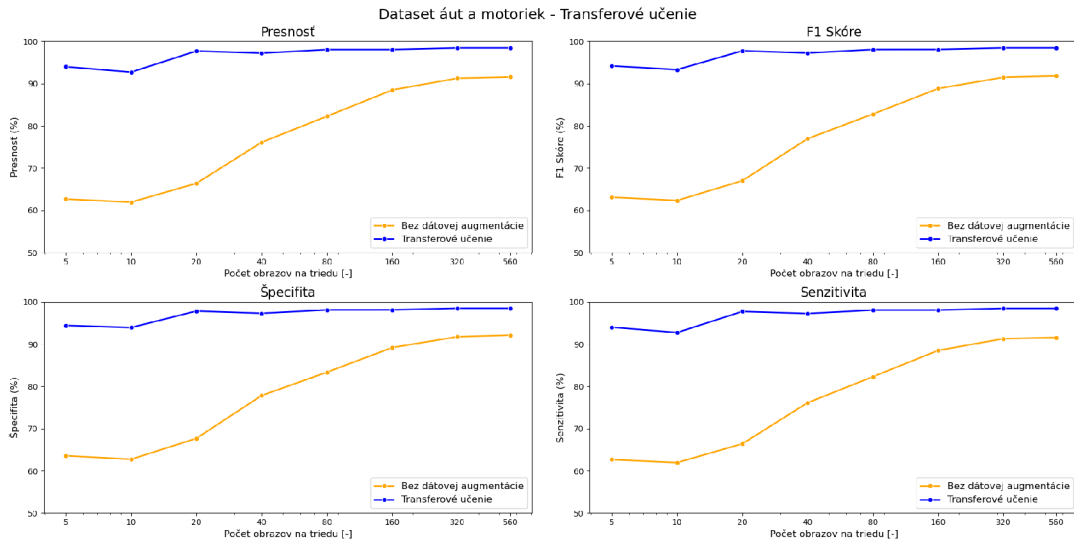


Obr. 6.3: Grafické porovnanie na testovacej množine pomocou transferového učenia na datasete obrazov histológie

6.2.2 Učenie s datasetom áut a motoriek

Dosiahnuté metriky transferového učenia na grafoch 6.4 dosahujú pri všetkých metrikách konzistentne lepšie výsledky. V prípade najnižšieho počtu obrazov na triedu ($N = 5$) boli schopné modely zlepšiť svoje metriky až o viac než 30 %. Metriky dosahujú skoro konštatné výsledky pohybujúce sa na úrovni približne 95 %. Treba byť opatrný pri hodnotení týchto výsledkov, pretože použité modely boli predtrénované s pomocou ImageNetu, ktorý obsahuje aj kategórie s autami a motorkami.

Keďže ImageNet je veľký dataset s mnohými rôznymi kategóriami obrázkov, môže to znamenať, že modely už mali nejaké základné informácie o autách a motorkách ešte pred tým, ako boli testované na datasete s vozidlami. To by mohlo byť dôvodom ich výrazne lepších výsledkov.



Obr. 6.4: Grafické porovnanie na testovacej množine pomocou transferového učenia na datasete áut a motoriek

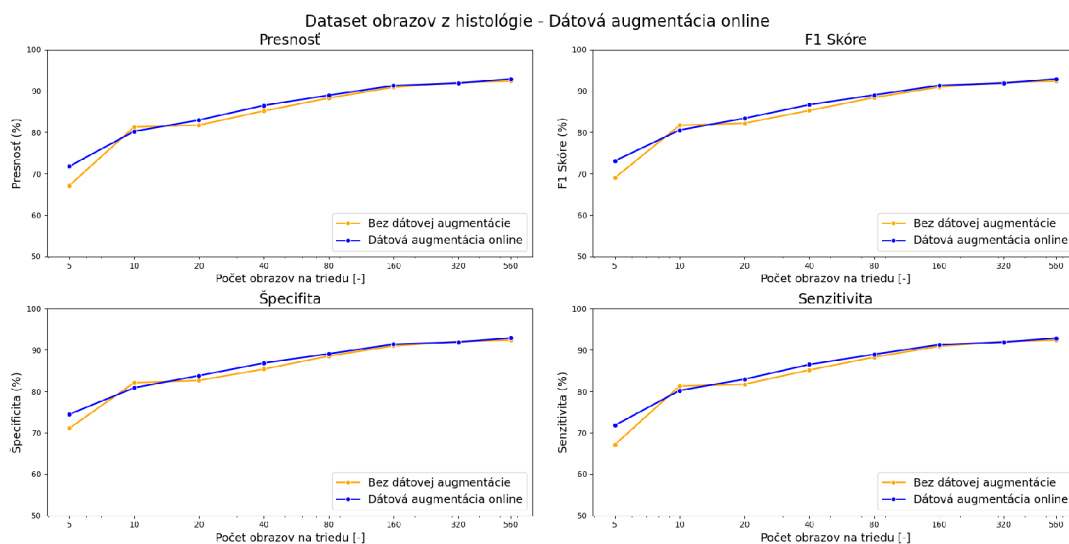
Záver

Pretrénované modely, ktoré využívajú váhy získané z tréningu na rozsiahlych a rozmanitých dátových súboroch, majú tendenciu lepšie zvládať aj menej reprezentatívne a špecifické dátové množiny. To im umožňuje zachovať vysokú úroveň presnosti a efektivity, dokonca aj pri zmenšených alebo špecificky distribuovaných dátových množinách. Tento prístup umožňuje modelom získať dôležité informácie, ktoré by v špecializovaných datasetoch chýbali.

6.3 Dátová augmentácia online

6.3.1 Učenie s datasetom obrazov z histológie

Na výsledných grafoch zobrazené na obrázkoch 6.5 je možné vidieť implementáciu dátovej augmentácie, označená ako online. Preukázala mierne zlepšenie takmer v každom modeli oproti použitia bez dátovej augmentácie.



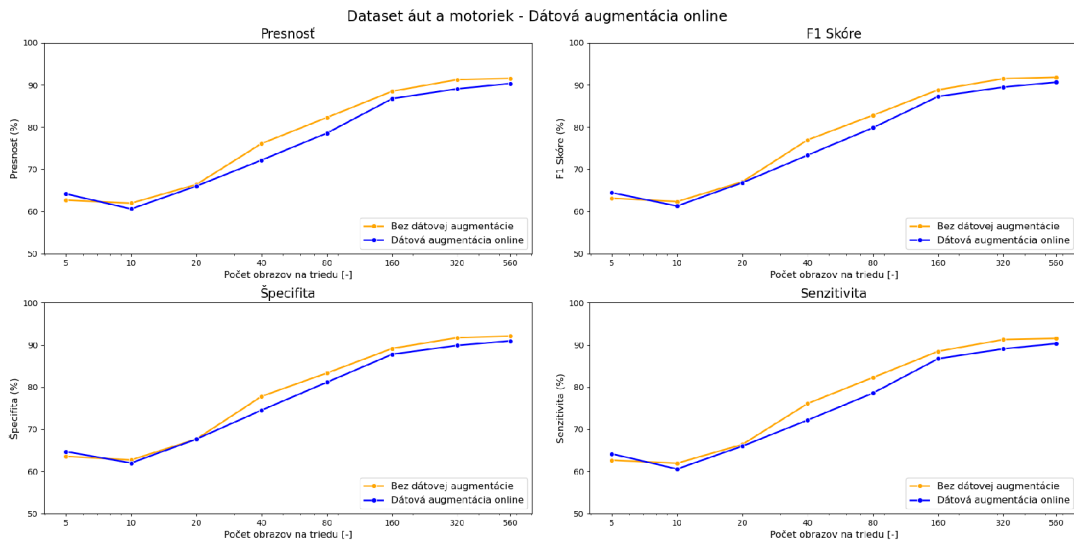
Obr. 6.5: Grafické porovnanie na testovacej množine s dátovou augmentáciou online na datasete obrazov histológie

Vo zvolenom datasete hrá významnú úlohu vykonaná dátová augmentácia predovšetkým v minimálne nastavenom počte obrazov na triedu ($N = 5$), kde sa testovacia presnosť zlepšila zo 67,1 % na 71,8 %, celkovo teda o necelých 4,7 %. V ostatných metrikách tiež nastalo zlepšenie v minimálne nastavenom počte obrazov na triedu, v priemere približne o 4 %. V hodnotách vyšších ako 10 obrazov na triedu vykazujú metriky porovnateľné výsledky, jedine v prípade 10 obrazov na triedu zaznamenali metriky minimálne zhoršenie približne o 1 %, čo môže byť ale ovplyvnené vybranými vstupnými obrazmi v trénovacej množine.

6.3.2 Učenie s datasetom áut a motoriek

Zobrazené grafy na obrázku 6.6 vykazujú variabilné výsledky. Jediné zlepšenie predstavuje dátová augmentácia iba v prípade najmenšieho počtu obrazu na triedu ($N=5$) z 62,5 % na 64,5 %. Ostatné modely ukazujú mierne zhoršenie, čo môže byť zapríčinené nadmerným prispôbením modelov, známym ako overfitting, ktoré môže vzniknúť, keď dátová augmentácia vytvára príliš špecifické alebo irelevantné varianty

trénovacích príkladov, čo vedie k zlepšeniu výkonu na trénovacej množine, ale zhoršeniu generalizácie na nevidených dátach.



Obr. 6.6: Grafické porovnanie na testovacej množine s dátovou augmentáciou online na datasete áut a motoriek

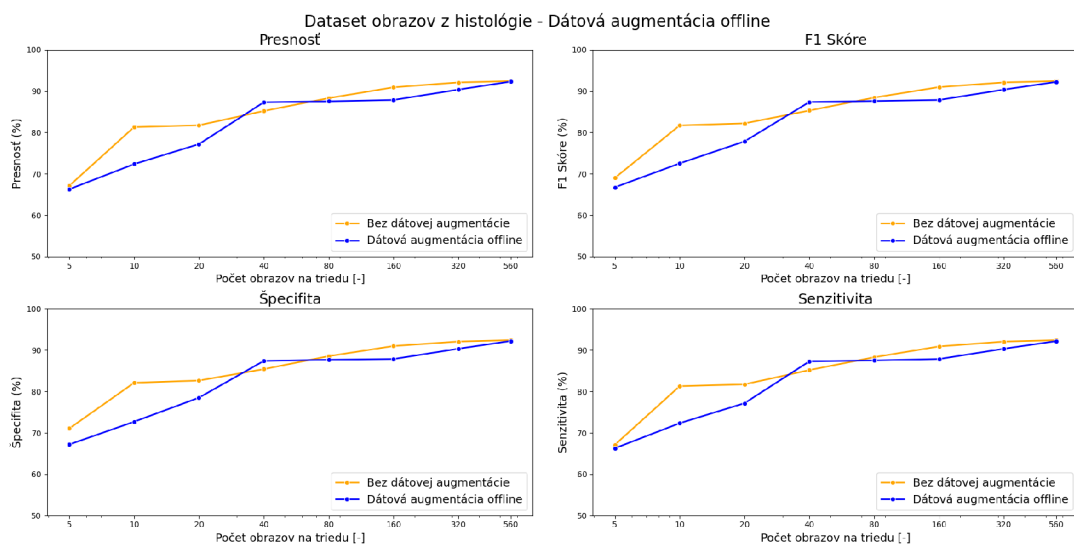
Záver

Analýza výsledkov získaných z implementácie online dátovej augmentácie na dvoch odlišných poukázala na potenciál, ktoré táto metóda prináša. V prípade histologických obrazov dátová augmentácia online viedla k miernemu zlepšeniu presnosti a ostatných metrík vo väčšine prípadov, najvýraznejšie pri nižšom počte obrazov na triedu. Zvýšenie presnosti o 4,7 % pri najmenšom počte obrazov je významné, najmä v kontexte medicínskych aplikácií. Naopak, v datasete áut a motoriek online dátová augmentácia nepriniesla očakávané zlepšenie, s výnimkou najmenšieho počtu obrazov na triedu, kde došlo k zlepšeniu presnosti o 2 %. Tieto výsledky poukazujú na to, že efektívnosť dátovej augmentácie môže závisieť od charakteristík konkrétneho datasetu a že nadmerné prispôbenie môže viesť k zhoršenej generalizácii na nevidených dátach.

6.4 Dátová augmentácia offline

6.4.1 Učenie s datasetom obrazov z histológie

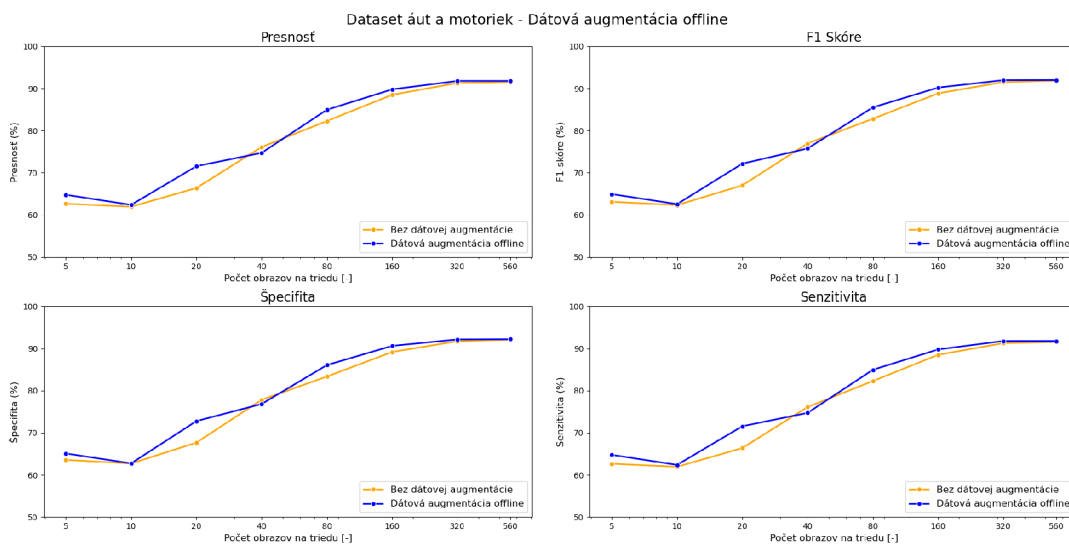
Vytvorené grafy na 6.7 ukazujú na vylepšenie modelu iba v jednom prípade, kde počet obrazov na triedu ukazuje 40, kde nastalo zlepšenie presnosti z 85,1 % na 87,1 %. V ostatných modelov metóda nevykazuje žiadne zlepšenie. Je možné, že použité augmentácie nie sú dostatočne relevantné pre konkrétny typ úloh, akým je analýza histologických obrazov. Následne to mohlo byť, pretože ak sú augmentačné techniky príliš špecifické alebo extrémne, môžu skresliť dôležité informácie na obrazoch, čo vedie k zníženiu presnosti modelu namiesto jej zvýšenia. Pri histologických obrazoch môže byť dôležitý každý detail a drastické zmeny môžu byť kontraproduktívne.



Obr. 6.7: Grafické porovnanie na testovacej množine s dátovou augmentáciou offline na datasete obrazov histológie

6.4.2 Učenie s datasetom áut a motoriek

Výsledky dátovej augmentácie označenej ako offline na obrázkoch 6.8 preukázali mierne pozitívnejšie výsledky oproti predošlému datasetu. Takmer v každom modeli priniesli zlepšenie vykreslených metrik. Najväčšie zmenu zaznamenala technika v počte obrazov na triedu rovnej 20, kde sa polepšila presnosť z pôvodných 66,4 % na 71,6 % a podobné zlepšenia v tomto prípade je možné pozorovať aj v ostatných metrikách. V ostatných modeloch a odlišnom nastavenom počtoch obrazov na triedu



Obr. 6.8: Grafické porovnanie na testovacej množine s dátovou augmentáciou offline na datasete áut a motoriek

Záver

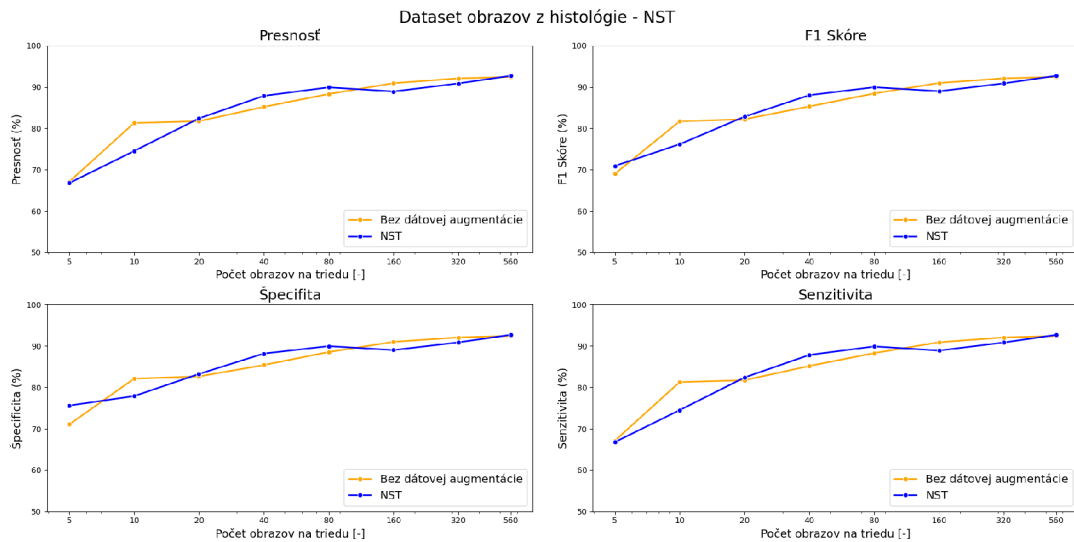
Druhý typ dátovej augmentácie ukazuje mierne opačné výsledky a ukazuje na ľahké zlepšenie v datasete áut a motoriek. Histologické snímky často vyžadujú zachovanie presnej štruktúry a morfológie, ktoré môžu byť augmentáciami narušené. Úspešnosť tejto metódy môže výrazne závisieť od typu datasetu a špecifikácií úloh, na ktorých je model trénovaný. Kým v prípade datasetu obrazov z histológie offline augmentácia nepriniesla významné zlepšenie a v niektorých prípadoch mohla dokonca zhoršiť výkon modelu kvôli možnému skresleniu dôležitých detailov, v prípade datasetu áut a motoriek sa ukázala ako mierne efektívnejší spôsob na zlepšenie presnosti a ďalších metrik. Toto naznačuje, že pri implementácii dátovej augmentácie je kľúčové brať do úvahy charakteristiky konkrétnych dát a ich relevanciu pre danú úlohu.

6.5 Prenos neurónového štýlu

6.5.1 Učenie s datasetom obrazov z histológie

Na základe grafického porovnia grafov vykreslených metrikou pomocou metódy NST na obrázku 6.9 je možné vidieť miernu kolísavú tendenciu vo všetkých grafoch. Hoci je to pokročilá metóda, ktorá zabezpečuje jedinečnosť každého obrazu, tak bola úspešná iba dvoch prípadoch ($N=40$ a $N=80$). Rozšírenie datasetu v týchto prípadoch prinieslo aj tak iba minimálne zlepšenie vo všetkých metrikách, približne o 4 % resp. 2 %.

Zhoršenie výsledkov nastalo v prípade nižších počtov obrazov na triedu ($N < 10$) a taktiež v prípade veľmi vysokých počtov ($N > 160$). Tento jav môže byť odôvodnený tým, že pri menšom počte obrazov nie je dostatočná variabilita v dátach na to, aby metóda NST mohla efektívne extrahovať a naučiť sa relevantné vzory. Naopak, pri príliš vysokom počte obrazov môže dôjsť k zvýšenej špecificite modelu, ktorá znižuje jeho schopnosť generalizácie na nové, nevidené dáta.

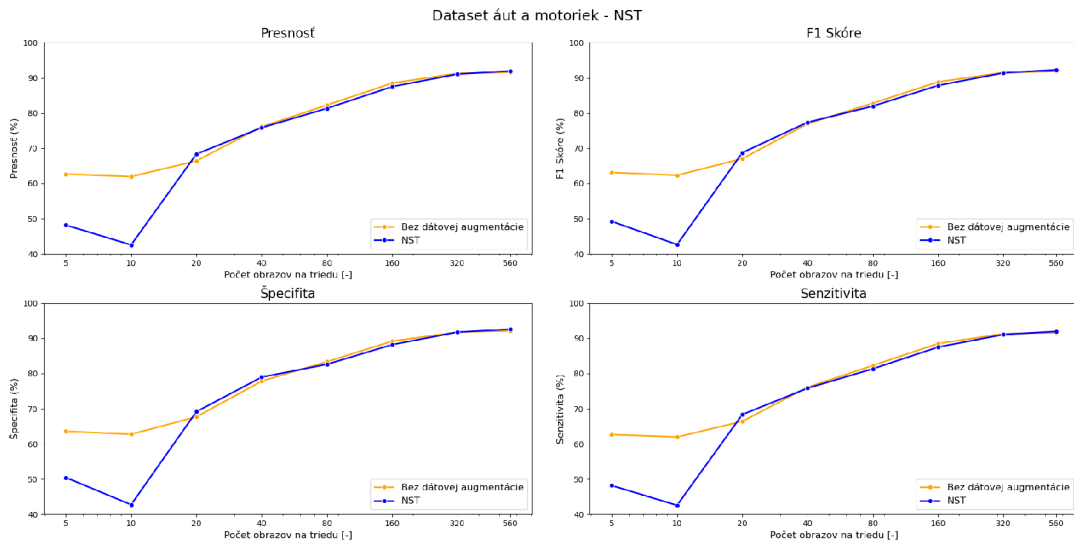


Obr. 6.9: Grafické porovnanie na testovacej množine s dátovou augmentáciou online na datasete obrazov histológie

6.5.2 Učenie s datasetom áut a motoriek

Na základe vyhotovených grafov pomocou metódy NST na obrázku 6.10 vidieť masívne negatívny dopad na testovacie metriky v prípade nižších obrazov na triedu ($N < 10$), čo naznačuje, že aplikácia metódy môže viesť k zníženej schopnosti modelu generalizovať, keď je k dispozícii len obmedzené množstvo tréningových dát. To

môže byť výsledkom prekomplikovania modelu, ktorý sa príliš špecializuje na štýlové charakteristiky tréningových dát, pričom stráca schopnosť efektívne spracovávať a klasifikovať nevidené vzorky. Fenomén sa obzvlášť prejavuje v scenároch s malým počtom dát, kde každý príklad má nepomerne veľký vplyv na učenie modelu. V prípade väčšieho počtu dát obrazov na jednu triedu nenastáva výrazné zlepšenie.



Obr. 6.10: Grafické porovnanie na testovacej množine s dátovou augmentáciou online na datasee áut a motoriek

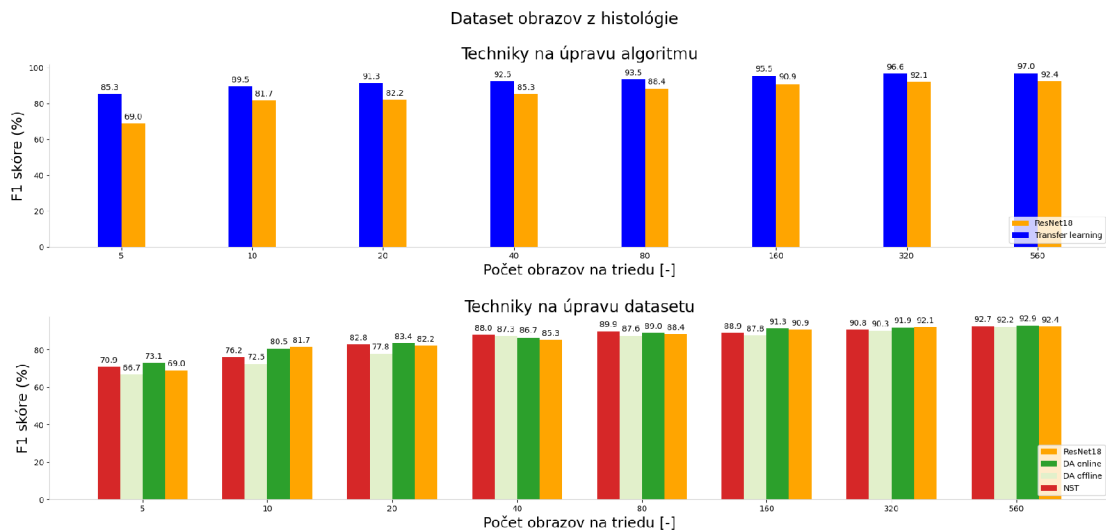
Záver

Metóda NST napriek svojej inovatívnosti a schopnosti generovať unikátne obrázky, vykazuje výrazné obmedzenia, ktoré sú závislé od množstva a variability dát. V prípade malého počtu obrázkov na triedu dochádza k pretrénovaniu modelu a straty generalizačnej schopnosti, zatiaľ čo pri veľmi veľkých datasetoch sa model stáva príliš špecifickým, čo tiež nepriaznivo ovplyvňuje jeho výkonnosť na nových, nevidených dátach. Tieto výsledky poukazujú na dôležitosť optimalizácie veľkosti a diverzity tréningových datasetov pri využívaní metód hlbokého učenia, ako je NST, aby sa zabezpečila ich efektívnosť a široká aplikovateľnosť v rôznych scenároch.

6.6 Celkové porovnanie výsledkov

6.6.1 Dataset obrazov z histológie

Prehľadné grafické porovnanie výsledkov pre dataset obrazov z histológie je znázornené na obrázku 6.11, kde kľúčovým parametrom hodnotenia je F1-skóre. Najlepšie výsledky sa nachádzajú u modeloch transferového učenia ukazujúce konštatné zlepšenie v každom modeli, kde v prípade 5 obrazov na triedu sa zlepšila až o viac než 15 %. Použitie rôznych foriem dátovej augmentácie neprinášajú významné zlepšenie, v niektorých modeloch dokonca mierne zhoršenie, čo je možné interpretovať ako indikáciu toho, že niektoré formy dátovej augmentácie môžu byť pre tento konkrétny dataset kontraproduktívne, alebo že modely sú už dostatočne robustné a nevyžadujú ďalšie zložité metódy augmentácie. Toto poukazuje na dôležitosť výberu vhodných techník augmentácie špecificky prispôbených pre každý typ údajov a úlohy.

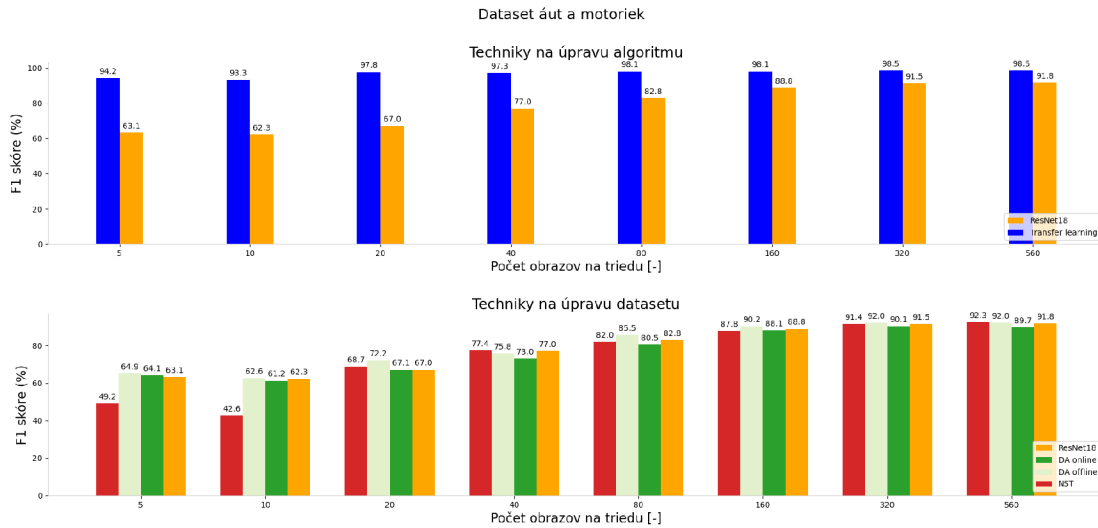


Obr. 6.11: Celkové porovnanie metód na dataseete obrazov z histológie

6.6.2 Dataset áut a motoriek

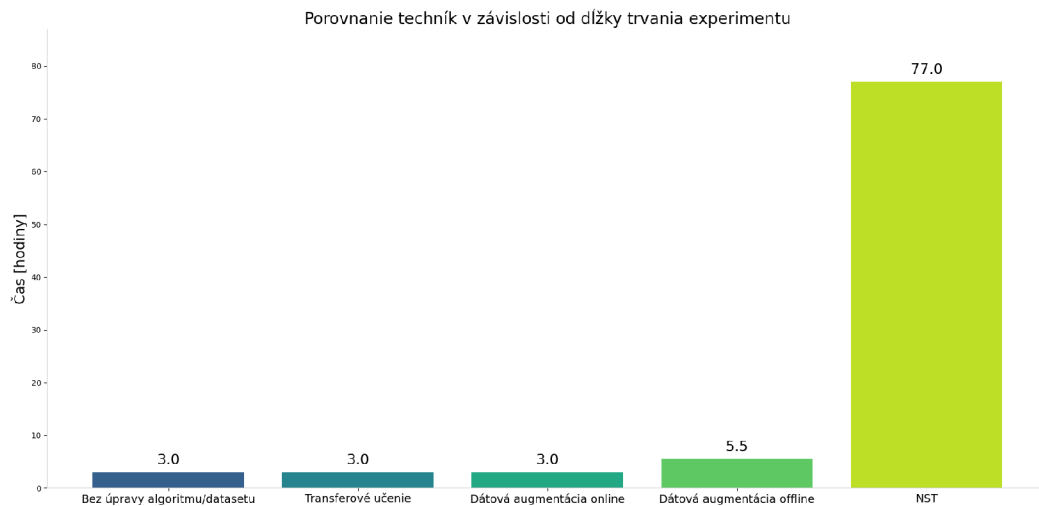
Na obrázku 6.12 je zobrazené grafické porovnanie výsledkov rôznych metód učenia pre dataset obsahujúci obrázky áut a motoriek, pričom kľúčovým kritériom hodnotenia je F1-skóre. Najlepšie výsledky preukázali opäť modely transferového učenie, pretože priniesli konštatné zlepšenie výsledkov v každom modeli, kde sa bola schopná zlepšiť metrika až o viac než 30 %. Za ňou nasledujú použitie techník dátovým augmentácií online a offline, ktoré sú viac menej porovnateľné, ale priniesli minimálne

zlepšenie v priemere o 2-3 %. V niektorých prípadoch minimálne zhoršenie. Najhoršie výsledky ukazovali modely metód NST, ktoré nepriniesla výrazné zlepšenie v žiadnom modeli a ako jediná v menších počtoch obrazov na triedu ($N < 10$) priniesla výrazné zhoršenie výsledkov, kde nastal pokles približne o 15-20 %.



Obr. 6.12: Celkové porovnanie metód datasetu áut a motoriek

6.6.3 Časová analýza experimentov



Obr. 6.13: Časové porovnanie experimentov graficky.

Rozdiely v čase trvania experimentov sú priamo ovplyvnené komplexnosťou a výpočtovými požiadavkami každej techniky. Časové porovnanie je znázornené na grafe 6.13, kde sú časové údaje uvádzané na všetky 3 semienka pre 1 dataset s použitou architektúrou siete ResNet18 nastavené na počet epoch rovnej 50. Implementované techniky transferové učenie a dátová augmentácia online majú rovnaký čas, ktorý je približne 3 hodiny. Tieto metódy nevyžadujú komplexné úpravy v algoritme, čo umožňuje rýchle spracovanie. Dátová augmentácia offline má čas približne 5,5 hodiny, čo je takmer dvojnásobok predchádzajúcich metód. Tento rozdiel je hlavne spôsobený potrebou ukladať dát pred ich použitím v modeli, čo vyžaduje viac času na prípravu a spracovanie. NST vyžaduje výrazne najviac času, až 77 hodín. Táto technika zahŕňa veľmi komplexné spracovanie obrazov. Takýto proces je výpočtovo náročný a zahŕňa iteratívne úpravy obrazu, čo významne zvyšuje celkový čas spracovania. Najviac náročnejšia časť samotného experimentu NST je generovanie obrazov, ktoré zabralo pri jednom datasete až 73 hodín pri celkovom počte 6570 obrazov.

7 Diskusia

V diplomovej práci bolo zrealizovaných viacero experimentov s cieľom preskúmať možnosti učenia hlbokých neurónových sietí v prípade učenia s obmedzeným súborom dát.

Vysoký počet parametrov v neurónovej sieti pri použití rôznych architektúr siete nie je vždy zárukou lepších výsledkov, najmä pri jednoduchších klasifikačných problémoch alebo menších datasetoch. Na základe týchto výsledkov sa rozhodlo pokračovať v experimentoch s použitím architektúry ResNet18, ktorá poukazuje na vhodnosť použitia v zvolených datasetoch.

Najlepšie výsledky dosahovala v oboch prípadoch technika transferového učenia potvrdzujúce jeho použitie vo viacerých článkoch [62, 63, 64], kde dosahovala sľubné výsledky. Metrika využívajúca naučené váhy z ImageNet, kde v prípade najmenšieho počtu obrazov v triede ($N=5$) bol schopný dosiahnuť F1-skóre až 94,2 % v prípade dataset áut a motoriek. Tieto výsledky treba brať mierne s rezervou, pretože je dôležité zohľadniť, že pretrénované váhy z ImageNet zahŕňajú aj kategórie obsahujúce autá a motorky. Tento fakt môže ovplyvniť výkonnosť modelov, keďže ImageNet je obsiahly dataset, ktorý obsahuje rozsiahle spektrum obrazových kategórií, vrátane tých, ktoré sú priamo relevantné pre dataset áut a motoriek. Táto špecifikácia môže poskytnúť modelu prednostné informácie, ktoré by inak v menej reprezentatívnych alebo špecializovaných datasetov neboli dostupné. Tento fakt potvrdzuje výsledok transferového učenia v prípade datasetu obrazov z histológie, kde úroveň vypočítanej metriky pre rovnaký počet obrazov na triedu sa pohybuje na úrovni 85,3 %, čo vykazuje zníženie približne o 9 % oproti predošlému datasetu. Avšak stále sa považuje tento progres veľmi významný.

Použitie rôznych foriem dátovej augmentácie môže priniesť variabilné výsledky a prinášajú rôzne výhody, ktoré môžu významne ovplyvniť úspešnosť a efektivitu modelov strojového učenia. Dátová augmentácia poukazuje že niektoré metódy augmentácie nemusia byť vhodné pre konkrétny dataset alebo že modely sú už dostatočne adaptabilné a nepotrebujú ďalšie komplexné techniky augmentácie. To zdôrazňuje význam správneho výberu techník augmentácie, ktoré sú na mieru prispôsobené pre konkrétne typy údajov a špecifické úlohy. V prípade mierne zhoršených hodnôtach mohlo nastať výber nedostatočne reprezentovateľnej trénovacej množiny údajov alebo nadmerná aplikácia augmentácie, čo mohlo viesť k pretrénovaniu modelu na neadekvátne zmenené dáta. Tento problém môže byť obzvlášť závažný, keď augmentácia vytvára neexistujúce vzorce alebo charakteristiky, ktoré v skutočnom svete nie sú relevantné alebo pravdepodobné.

Metóda NST, ako forma dátovej augmentácie, môže byť využitá na zvýšenie množstva a rozmanitosti trénovacích dát pre modely strojového učenia, čím by sa

potenciálne mohla zlepšiť ich schopnosť generalizácie. V našom navrhnutom experimente ilustruje limitácie spojené s množstvom a diversitou dát, kde optimalizácia veľkosti a rôznorodosti dátových súborov je kľúčová pre zachovanie generalizačnej schopnosti modelu. Pri metóde NST je výsledná kvalita obrázkov veľmi závislá na charaktere vstupných obrazov, pretože ak sú dva obrázky snímané z rôznych uhlov, môžu mať odlišné perspektívy a geometrické vlastnosti. Prenos štýlu z jedného obrázku na druhý bez zohľadnenia týchto rozdielov môže viesť k nesprávnej alebo vizuálne neuspokojivej integrácii štýlových prvkov. V niektorých prípadoch môže byť výsledok vizuálne nepríjemný alebo málo realistický, čo môže mať za následok zavádzajúce alebo nekvalitné dáta pre tréning modelu. Toto hralo v našom prípade hralo významnú úlohu, pretože pri datasete áut a motoriek boli obrázky snímané z rôznych uhlov či ukazovali len časť vozidla. Príklady nepodarených snímok, ktoré mohli negatívne ovplyvňovať model sú zobrazené na obrázku 7.1,

Príklad nepodarených áut a motoriek z metódy NST



Obr. 7.1: Príklad nepodarených vytvorených obrazov z modelu NST z datasetu áut a motoriek

V prípade datasetu obrazov z histológie, ktoré boli nasnímané z rovnakého uhla, model NST generoval na oko krajšie nové obrázky, avšak ich unikátnosť nebola natoľko dostatočná, aby sa zvýšila celková presnosť modelov o viac ako 3 %. Ďalším faktorom neúspešnosti nastáva pri veľkosti generovaného obrazu, ktorá bola v našom prípade nastavená na 112 pixelov, kedy menšie obrázky môžu stratiť detaily, pretože menej pixelov znamená menej priestoru na zachytenie jemných štýlových prvkov. V našom prípade hralo významnú úlohu časová analýza celého experimentu, kedy generovanie 6570 obrazov na jeden dataset zabralo celkovo približne 73 hodín.

Metóda NST nebola príliš veľmi úspešná a z pohľadu teoretickej roviny by mohla

byť viac prospešná metóda GAN, ktorou kľúčovou výhodou GAN je ich schopnosť generovať vysoko kvalitné realistické obrázky a sú často ťažko odlíšiteľné od skutočných fotografií. Tento fakt potvrdujú aj viaceré články, kde nastalo zlepšenie na známych datasetoch o 5-10 %. [66, 67] Avšak metóda GAN je výpočetne veľmi náročná, čo poukazuje aj článok, v ktorom autori poukazujú na dlhý čas tréovania GAN sietí, kde tréovanie siete si vyžaduje veľký počet obrazov a čas generovaných obrazov o veľkosti 256 pixelov popisujú na približne 14 dní. [65]

Vo všeobecnosti nižší počet obrazov na triedu vo výsledných metrikách poukazoval variabilné výsledky, pričom mohli byť ovplyvnené stochasticitou, ktorej sa snažilo zabrániť aspoň použitím 3 rôznych semienok v každom experimente. Avšak experiment bol nastavený, aby sa preskúmalo učenie na veľmi nízkom počte obrazov na triedu, ktoré môže byť výzvou pre modely v učení z obmedzených dát. Napriek použitiu rôznych semienok na zmiernenie stochasticity, variabilita výsledkov naprieč rôznymi pokusmi naznačuje, že s menším množstvom dát sa zvyšuje vplyv náhodných faktorov na výsledky. Toto poukazuje na potrebu efektívnejších stratégií pre učenie s malým množstvom dát. Na úplne zbavenie sa vplyvu stochasticity, by bolo preskúmať oveľa väčší počet nastavených semienok, ktoré bohužiaľ z časovej náročnosti by neboli príliš uskutočniteľné.

Záver

Strojové učenie a hlbokové neurónové siete predstavujú mocný nástroj pre riešenie rôznorodých úloh, predovšetkým v oblasti spracovania obrazov. Sú schopné učiť sa zložité vzory zo vstupných dát, čo umožňuje aplikáciu v mnohých odvetviach.

Diplomová práca sa zamerala na preskúmanie možnosti učenia na obmedzenom trénovacom datasete, ktorý môže negatívne ovplyvniť výkon modelu a schopnosti generalizácie, hlavne pri riešení komplexnejších úloh ako spracovanie biomedicínskych obrazov. Boli preskúmané viaceré techniky a prístupy, ktoré pozitívne aj negatívne ovplyvňujú učenie hlbokých neurónových sietí na obmedzenom súbore údajov.

Implementácia experimentov diplomovej práce ukázala, že transferové učenie môže byť účinné pri učení s obmedzeným počtom dát, najmä ak sú k dispozícii pretrénované váhy z rozsiahlych datasetov ako ImageNet. Tento prístup sa osvedčil, keďže dosiahol vysoké F1-skóre, avšak výsledky treba interpretovať s istou mierou opatrnosti, najmä pri datasetoch s priamou relevantnosťou kategórií z ImageNetu.

Použitie dátovej augmentácie môže prinášať rozmanité výsledky a je dôležité správne vybrať augmentačné techniky prispôbené konkrétnym dátovým sadám a úlohám. Nesprávny výber techník alebo ich nadmerné použitie môže viesť k pretrénovaniu modelu na nevhodne modifikované dáta, čo môže mať za následok zníženú schopnosť modelu generalizovať na reálne údaje. To zdôrazňuje dôležitosť cielenej aplikácie augmentácie, aby sa zabránilo vytváraniu irelevantných alebo nepravdepodobných vzorcov v tréningových dátach.

Metóda NST, aj keď poskytla v niektorých prípadoch minimálne zlepšenie, mala svoje veľké obmedzenia, najmä pri zachovaní kvality a reprezentatívnosti obrázkov. Pri menšom množstve pixelov a rozličných uhloch snímania sa ukázala ako menej efektívna. Zdá sa, že pre dosiahnutie lepšej generalizácie a vyššej presnosti modelu by boli vhodnejšie metódy ako GAN, ktoré však vyžadujú väčšie množstvo dát a dlhší čas na tréningovanie.

Pri písaní záverečnej práce boli využívané nástroje generatívnej umelej inteligencie výhradne na základné vyhľadávanie informácií a konzultáciu, ako aj pri tvorbe a úprave zdrojových skriptov, čím sa urýchlil proces programovania a zvýšila efektivita. Tieto nástroje pomohli získať základné informácie súvisiace s témou práce, ktoré následne boli overované a validované z dôveryhodných zdrojov. Tiež slúžila ako konzultačný nástroj pri riešení odborných problémov a otázok, ktoré sa vyskytli počas spracovania práce. Použitie generatívnej umelej inteligencie bolo v súlade s akademickými štandardmi, pričom všetky získané a generované informácie boli overované.

Literatúra

- [1] ALZUBAIDI, Laith; ZHANG, Jinglan; HUMAIDI, Amjad J; AL-DUJAILI, Ayad; DUAN, Ye et al., 2021. Review of deep learning: Concepts, CNN architectures, challenges, applications, future directions: Concepts, CNN architectures, challenges, applications, future directions. *Journal of big Data*. Roč. 8, s. 1-74.
- [2] YAMASHITA, Rikiya; NISHIO, Mizuho; TOGASHI, Kaori; et al. Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. Online. Dostupné z: <<https://insightsimaging.springeropen.com/articles/10.1007/s13244-018-0639-9>>. [cit. 2023-11-10].
- [3] What are neural networks? IBM. 2021. Online. Dostupné z: <<https://www.ibm.com/topics/neural-networks>> [cit. 2023-10-25].
- [4] NIELSEN, Michael A, 2015. *Neural networks and deep learning*. 25. Determination press San Francisco, CA, USA.
- [5] BEŇUŠKOVÁ, Lubica; *Umelá inteligencia siete*. 2.vyd. Bratislava. Slovenská technická univerzita, 2007. ISBN 978-80-227-2629-0.
- [6] MOHANTY. A.; Multi layer Perceptron (MLP) Models on Real World Banking Data. Online. Dostupné z: <<https://becominghuman.ai/multi-layer-perceptron-mlp-models-on-real-world-banking-data-f6dd3d7e998f>> [cit. 2023-10-30].
- [7] CHMELIK. Jiří; *Deep learning 1. Machine learning. Department of Biomedical Engineering*. Brno, 2023. Vysoké učení technické v Brně.
- [8] REYNOLDS, A.; Convolutional Neural Networks (CNNs). Online. Dostupné z: <<https://anhreynolds.com/blogs/cnn.html>> [cit. 2023-11-10].
- [9] CHEN, Leiyu; LI, Shaobo; BAI, Qiang; YANG, Jing; JIANG, Sanlong et al., 2021. Review of image classification algorithms based on convolutional neural networks. *Remote Sensing*. Roč. 13, č. 22, s. 4712. ISSN 2072-4292.
- [10] THAKUR, Ayush; ReLU vs. Sigmoid Function in Deep Neural Networks. Online. Dostupné z: <<https://wandb.ai/ayush-thakur/dl-question-bank/reports/ReLU-vs-Sigmoid-Function-in-Deep-Neural-Networks--VmlldzoyMDkOMzI>> [cit. 2023-11-12].

- [11] SHIN, Eric; Machine Learning - Final Week. Online. Dostupné z: <<https://developmentblog.hashnode.dev/machine-learning-final-week>>[cit. 2023-11-15].
- [12] SHAHRIAR, Nafiz; What is Convolutional Neural Network — CNN (Deep Learning) Online. Dostupné z: <<https://nafizshahriar.medium.com/what-is-convolutional-neural-network-cnn-deep-learning-b3921bdd82d5>> [cit. 2023-11-15].
- [13] KOZUMPLÍK, Jiří; PROVAZNÍK. Ivo; Umělá inteligence v medicíně. Brno, 2007. Skripta. Fakulta elektrotechniky a komunikačních technologií - Vysoké učení technické v Brně
- [14] TUČKOVÁ, Jana; Vybrané aplikace umělých neuronových sítí při zpracování signálů. Praha: České vysoké učení technické v Praze, 2009. isbn 978-80-01-04229-8.
- [15] BROWNLEE. Jason; A Gentle Introduction to Cross-Entropy for Machine Learning. [online]. [cit. 2023-11-15]. Dostupné z URL: <<https://machinelearningmastery.com/loss-and-loss-functions-for-training-deep-learning-neural-networks/>> [cit. 2023-11-15].
- [16] Stochastic Gradient Descent. Online. Dostupné z: <<https://paperswithcode.com/method/sgd>>[cit. 2023-11-15].
- [17] Adaptive Moment Estimation (Adam). Online. Dostupné z: <<https://ml-explained.com/blog/adam-explained>>[cit. 2023-11-15].
- [18] ADAM. Online. Dostupné z: <<https://medium.com/@nerdjock/deep-learning-course-lesson-7-4-adam-adaptive-moment-estimation-e23434850bfc>> [cit. 2023-11-15].
- [19] WANG. Chi-Feng; The Vanishing Gradient Problem. Online. Dostupné z: <<https://towardsdatascience.com/the-vanishing-gradient-problem-69bf08b15484>> [cit. 2023-11-15].
- [20] LI. Katherine; Vanishing and Exploding Gradients in Neural Network Models: Debugging, Monitoring, and Fixing. Online. Dostupné z: <<https://neptune.ai/blog/vanishing-and-exploding-gradients-debugging-monitoring-fixing>> [cit. 2023-11-15].

- [21] DUTTA. Bhumika; What are Skip Connections in Neural Networks? Online. Dostupné z: <<https://www.analyticssteps.com/blogs/what-are-skip-connections-neural-networks>> [cit. 2023-11-15].
- [22] THANAPOL, Panissara; LAVANGNANANDA, Kittichai; BOUVRY, Pascal; PINEL, Frédéric a LEPRÉVOST, Franck, 2020. Reducing overfitting and improving generalization in training convolutional neural network (CNN) under limited sample sizes in image recognition. In: 2020-5th International Conference on Information Technology (InCIT). IEEE, s. 300-305. ISBN 1728166942.
- [23] CHUAN-EN. Lin; 8 Simple Techniques to Prevent Overfitting. Online. Dostupné z: <<https://towardsdatascience.com/8-simple-techniques-to-prevent-overfitting-4d443da2ef7d>> [cit. 2023-12-15].
- [24] WERNER DE VARGAS, Vitor; SCHNEIDER, Aranda; JORGE, Arthur; DOS SANTOS COSTA, Ricardo; DA SILVA PEREIRA, Paulo Ricardo a VICTÓRIA BARBOSA, JORGE, Luis; 2023. Imbalanced data preprocessing techniques for machine learning: a systematic mapping study: a systematic mapping study. Knowledge and Information Systems. Roč. 65, č. 1, s. 31-57. ISSN 0219-1377.
- [25] JOLOUDARI, Javad Hassannataj; MAREFAT, Abdolreza; NEMATOLLAHI, Mohammad Ali; OYELERE, Solomon Sunday a HUSSAIN, Sadiq, 2023. Effective Class-Imbalance Learning Based on SMOTE and Convolutional Neural Networks. Applied Sciences. Roč. 13, č. 6, s. 4006. ISSN 2076-3417.
- [26] Evaluating Classification Models Using Accuracy, Precision, Recall, and More. Online. Dostupné z: <<https://frankcorso.dev/classification-machine-learning-metrics.html>> [cit. 2023-11-28].
- [27] F1 Score vs ROC AUC vs Accuracy vs PR AUC: Which Evaluation Metric Should You Choose? Online. Dostupné z: <<https://neptune.ai/blog/f1-score-accuracy-roc-auc-pr-auc>> [cit. 2023-11-28].
- [28] TRIPATHY, Sushreeta a SINGH, Rishabh, 2022. Convolutional neural network: an overview and application in image classification: an overview and application in image classification. In: Proceedings of Third International Conference on Sustainable Computing: SUSCOM 2021. Springer, s. 145-153.
- [29] A quick overview of ResNet models. Online. Dostupné z: <<https://lekhuyen.medium.com/a-quick-overview-of-resnet-models-f8ed277ae81e>> [cit. 2023-4-10].

- [30] A Deep Learning Approach for Automated Diagnosis and Multi-Class Classification of Alzheimer’s Disease Stages Using Resting-State fMRI and Residual Neural Networks. Online. Dostupné z: <https://www.researchgate.net/figure/Original-ResNet-18-Architecture_fig1_336642248> [cit. 2023-4-10].
- [31] RATH, Sovit; 2023. Implementing ResNet18 in PyTorch from Scratch. Online. Dostupné z: <<https://debuggercafe.com/implementing-resnet18-in-pytorch-from-scratch/>> [cit. 2023-11-23].
- [32] VGG-Net Architecture Explained Online. Dostupné z: <<https://medium.com/@siddheshb008/vgg-net-architecture-explained-71179310050f>> [cit. 2023-4-10].
- [33] NGUYEN, Thanh-Hai; NGUYEN, Thanh-Nghia a NGO, Ba-Viet, 2022. A VGG-19 Model with Transfer Learning and Image Segmentation for Classification of Tomato Leaf Disease. Online. AgriEngineering. Roč. 4, č. 4, s. 871-887. ISSN 2624-7402.<<https://doi.org/10.3390/agriengineering4040056>> . [cit. 2024-04-10].
- [34] BALKI, Indranil; AMIRABADI, Afsaneh; LEVMAN, Jacob; MARTEL, Anne L; EMERSIC, Ziga et al., 2019. Sample-size determination methodologies for machine learning in medical imaging research: a systematic review: a systematic review. Canadian Association of Radiologists Journal. Roč. 70, č. 4, s. 344-353. ISSN 0846-5371.
- [35] RONNEBERGER, Olaf; FISCHER, Philipp a BROX, Thomas, 2015. U-net: Convolutional networks for biomedical image segmentation: Convolutional networks for biomedical image segmentation. In: Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18. Springer, s. 234-241. ISBN 3319245732.
- [36] SHAHINFAR, Saleh; MEEK, Paul a FALZON, Greg, 2020. “How many images do I need?” Understanding how sample size per class affects deep learning model performance metrics for balanced designs in autonomous wildlife monitoring. Ecological Informatics. Roč. 57, s. 101085. ISSN 1574-9541.
- [37] LITJENS, Geert; KOOI, Thijs; BEJNORDI, Babak Ehteshami; SETIO, Arnaud Arindra Adiyoso; CIOMPI, Francesco et al., 2017. A survey on deep learning in medical image analysis. Medical image analysis. Roč. 42, s. 60-88. ISSN 1361-8415.

- [38] FERNÁNDEZ-DELGADO, Manuel; CERNADAS, Eva; BARRO, Senén a AMORIM, Dinani, 2014. Do we need hundreds of classifiers to solve real world classification problems? The journal of machine learning research. Roč. 15, č. 1, s. 3133-3181. ISSN 1532-4435.
- [39] BRIGATO, Lorenzo a IOCCHI, Luca, 2021. A close look at deep learning with small data. In: 2020 25th International Conference on Pattern Recognition (ICPR). IEEE, s. 2490-2497. ISBN 1728188083.
- [40] MIKOŁAJCZYK, Agnieszka a GROCHOWSKI, Michał, 2018. Data augmentation for improving deep learning in image classification problem. In: 2018 international interdisciplinary PhD workshop (IIPhDW). IEEE, s. 117-122. ISBN 1538661438.
- [41] GANDHI, Arun; 2023. Data Augmentation | How to use Deep Learning when you have Limited Data | Part 2. Nanonets. Online. Dostupné z: <<https://nanonets.com/blog/data-augmentation-how-to-use-deep-learning-when-you-have-limited-data-part-2>> [cit. 2023-11-15].
- [42] SHALUL, Es; 2023. Data Augmentation in NLP: Best Practices From a Kaggle Master. Neptune. Online. Dostupné z: <<https://neptune.ai/blog/data-augmentation-nlp>> [cit. 2023-11-21].
- [43] GILES, Martin; 2018. The GANfather. Online. Dostupné z: <<https://www.technologyreview.com/2018/02/21/145289/the-ganfather-the-man-whos-given-machines-the-gift-of-imagination/>> [cit. 2023-11-23].
- [44] NOLEN, Sam; 2019. GANs for Data Augmentation. Online. Dostupné z: <<https://medium.com/abacus-ai/gans-for-data-augmentation-21a69de6c60b>> [cit. 2023-11-23].
- [45] TAE, Jake; 2020. The Math Behind GANs. Online. Dostupné z: <<https://jaketae.github.io/study/gan-math/>> [cit. 2024-3-15].
- [46] FERLATTI, Aldo; 2021. Neural Style Transfer (NST) — theory and implementation. Dostupné z: <<https://medium.com/@ferlatti.aldo/neural-style-transfer-nst-theory-and-implementation-c26728cf969d>> [cit. 2024-3-15].

- [47] RAJJA, Shangeth; 2018. Neural Style Transfer - the math and code. Dostupné z: <<https://medium.com/@shangethrajaa/neural-style-transfer-the-math-code-71ef46200ca8>> [cit. 2024-3-15].
- [48] SINGH, Ayush; 2019. How Do Neural Style Transfers Work? Dostupné z: <<https://towardsdatascience.com/how-do-neural-style-transfers-work-b76de101eb3>> [cit. 2024-3-15].
- [49] JAIN, Sarthak; 2021. How to use Deep Learning when you have Limited Data. Online. Dostupné z: <<https://nanonets.com/blog/how-to-use-deep-learning-when-you-have-limited-data/>> [cit. 2023-11-28].
- [50] Convolutional Neural Networks for Visual Recognition. Online. Dostupné z: <<https://cs231n.github.io/transfer-learning/>> [cit. 2023-11-28].
- [51] BROWNLEE, Jason.; 2019. A Gentle Introduction to Transfer Learning for Deep Learning. Online. Dostupné z: <<https://machinelearningmastery.com/transfer-learning-for-deep-learning/>> [cit. 2023-11-28].
- [52] NEALIS, Nisha.; 2020. A Simple Introduction to Dropout Regularization Online. Dostupné z: <<https://medium.com/analytics-vidhya/a-simple-introduction-to-dropout.>> [cit. 2023-11-28].
- [53] BROWNLEE, Jason.; 2019. A Gentle Introduction to Dropout for Regularizing Deep Neural Networks. Online. Dostupné z: <<https://machinelearningmastery.com/dropout-for-regularizing-deep-neural-networks/>> [cit. 2023-11-28].
- [54] AJAY, Simon.; 2020. Dropout: A Powerful Regularization Technique for Deep Neural Networks. Online. Dostupné z: <<https://www.linkedin.com/pulse/dropout-powerful-regularization-technique-deep-ajay-singh/>> [cit. 2023-11-28].
- [55] BROWNLEE, Jason; 2019. Ensemble Learning Methods for Deep Learning Neural Networks. Online. Dostupné z: <<https://machinelearningmastery.com/ensemble-methods-for-deep-learning-neural-networks/>> [cit. 2023-11-28].
- [56] ANUSHA, Hema; Bootstrapped Aggregation (Bagging). Online. Dostupné z: <<https://medium.com/@hemaanushatangellamudi/bootstrapped-aggregation-bagging-481f4812e3ea>> [cit. 2023-11-29].

- [57] MOGHIMI, Mohammad; BELONGIE, Serge J; SABERIAN, Mohammad J; YANG, Jian; VASCONCELOS, Nuno et al., 2016. Boosted convolutional neural networks. In: BMVC. S. 6.
- [58] *Car vs Bike Classification Dataset*. Kaggle, 2022. Online. Dostupné z: <<https://www.kaggle.com/datasets/utkarshsaxenadn/car-vs-bike-classification-dataset>> [cit. 2023-11-30].
- [59] BHAVESH, M.; *Lung cancer images*. Kaggle, 2021. Online. Dostupné z: <<https://www.kaggle.com/datasets/bhaveshmisra/lung-cancer-images12000-imagesmostly/data>> [cit. 2023-11-29].
- [60] ResNet, 2017. Online. Dostupné z: <<https://pytorch.org/vision/stable/models/resnet.html>> [cit. 2023-11-29].
- [61] BRIGATO, Lorenzo a IOCCHI, Luca, 2021. A close look at deep learning with small data. In: 2020 25th International Conference on Pattern Recognition (ICPR). IEEE, s. 2490-2497. ISBN 1728188083.
- [62] KIM, Hee E; COSA-LINAN, Alejandro; SANTHANAM, Nandhini; JANNE-SARI, Mahboubeh; MAROS, Mate E et al., 2022. Transfer learning for medical image classification: a literature review: a literature review. BMC medical imaging. Roč. 22, č. 1, s. 69. ISSN 1471-2342.
- [63] KAUR, Taranjit a GANDHI, Tapan Kumar, 2020. Deep convolutional neural networks with transfer learning for automated brain image classification. Machine vision and applications. Roč. 31, č. 3, s. 20. ISSN 0932-8092.
- [64] ALZUBAIDI, Laith; AL-AMIDIE, Muthana; AL-ASADI, Ahmed; HUMAIDI, Amjad J; AL-SHAMMA, Omran et al., 2021. Novel transfer learning approach for medical imaging with limited labeled data. Cancers. Roč. 13, č. 7, s. 1590. ISSN 2072-6694.
- [65] StyleGAN — Official TensorFlow Implementation. 2023. Online. Dostupné z: <<https://github.com/NVlabs/stylegan>> [cit. 2024-3-15].
- [66] ANTONIOU, Antreas; STORKEY, Amos a EDWARDS, Harrison, 2017. Data augmentation generative adversarial networks. ArXiv preprint arXiv:1711.04340.

- [67] ZHU, Xinyue; LIU, Yifan; LI, Jiahong; WAN, Tao a QIN, Zengchang, 2018. Emotion classification with data augmentation using generative adversarial networks. In: Advances in Knowledge Discovery and Data Mining: 22nd Pacific-Asia Conference, PAKDD 2018, Melbourne, VIC, Australia, June 3-6, 2018, Proceedings, Part III 22. Springer, s. 349-360. ISBN 3319930397.