



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

RE-IDENTIFIKACE GRAFFITI TAGŮ

GRAFFITI TAGS RE-IDENTIFICATION

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. JAN PAVLICA

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. JAKUB ŠPAŇHEL

BRNO 2020

Zadání diplomové práce



Student: **Pavlica Jan, Bc.**
Program: Informační technologie Obor: Informační systémy
Název: **Re-identifikace graffiti tagů**
Graffiti Tags Re-Identification
Kategorie: Zpracování obrazu

Zadání:

1. Seznamte se s principy re-identifikace objektů a vyhledávání obrazů na základě vizuálních podobností.
2. Obstarejte si vhodnou datovou sadu pro re-identifikaci graffiti tagů.
3. Vyberte vhodné metody založené na konvolučních neuronových sítích pro re-identifikace objektů / vyhledávání obrazů na základě podobností (*image retrieval*) a experimentujte s nimi.
4. Navrhněte úpravy těchto neuronových sítí a řádně otestujte.
5. Vyhodnoťte vybrané metody a navržené úpravy a diskutujte dosažené výsledky.
6. Vytvořte plakát a video prezentující vaši práci, její cíle a výsledky.

Literatura:

- Kumar et al. "Vehicle re-identification: An efficient baseline using triplet embedding". *IJCNN*, 2019.
- Špaňhel et al. "Learning Feature Aggregation in Temporal Domain for Re-Identification." *arXiv preprint:1903.05244* (2019).
- Ghosh, Neha, Shikha Agrawal, and Mahesh Motwani. "A survey of feature extraction for content-based image retrieval system." *ICRAC*, 2018.
- Masood et al.. "Content-Based Image Retrieval Features: A Survey." *IJANA*, 2018
- Dále dle pokynů vedoucího.

Při obhajobě semestrální části projektu je požadováno:

- Splnění prvních tří bodů zadání.

Podrobné závazné pokyny pro vypracování práce viz <https://www.fit.vut.cz/study/theses/>

Vedoucí práce: **Špaňhel Jakub, Ing.**
Vedoucí ústavu: Černocký Jan, doc. Dr. Ing.
Datum zadání: 1. listopadu 2019
Datum odevzdání: 31. července 2020
Datum schválení: 1. listopadu 2019

Abstrakt

Tato práce se zaměřuje na možnost využití současných metod na poli počítačového vidění k re-identifikaci graffiti tagů. V práci je zkoumána možnost využití konvolučních neuronových sítí k re-identifikaci graffiti tagů, které jsou nejrozšířenějším typem graffiti. V rámci práce bylo experimentováno s různými modely konvolučních neuronových sítí z nichž nejvhodnější byl MobileNet s využitím triplet loss funkce, kdy se podařilo dosáhnout mAP 36.02%.

Abstract

This thesis focuses on the possibility of using current methods in the field of computer vision to re-identify graffiti tags. The work examines the possibility of using convolutional neural networks to re-identify graffiti tags, which are the most common type of graffiti. The work experimented with various models of convolutional neural networks, the most suitable of which was MobileNet using the triplet loss function, which managed to achieve a mAP of 36.02%.

Klíčová slova

graffiti tagy, re-identifikace objektů, konvoluční neuronové sítě, MobileNet, ResNet, VGG16, triplet loss, circle loss

Keywords

graffiti tags, object re-identification, convolutional neural networks, MobileNet, ResNet, VGG16, triplet loss, circle loss

Citace

PAVLICA, Jan. *Re-identifikace graffiti tagů*. Brno, 2020. Diplomová práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Jakub Špaňhel

Re-identifikace graffiti tagů

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně pod vedením pana Ing. J. Špaňhela. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....

Jan Pavlica

30. července 2020

Poděkování

Rád bych poděkoval Ing. Jakubovi Špaňhelovi za cenné rady, věcné připomínky a vstřícnost při konzultacích a vypracování diplomové práce.

Obsah

1	Úvod	2
2	Analýza problému	3
2.1	Graffiti	3
2.2	Proces re-identifikaci objektů	5
3	Možné přístupy při re-identifikaci	10
3.1	CBIR	10
3.2	Klasifikace s využitím konvolučních neuronových sítí	10
3.3	Siamská neuronová síť	10
3.4	Triplet loss	12
3.5	Circle loss	14
3.6	Re-identifikace graffiti tagů	15
4	Trénování sítě	18
4.1	Dataset	19
5	Experimenty	22
5.1	Metriky pro vyhodnocování úspěšnosti	22
5.2	Modifikace trénovací množiny	23
5.3	Neuronové sítě s využitím triplet loss	23
5.4	Neuronové sítě s využitím circle loss	35
5.5	Re-identifikace	39
5.6	Shrnutí	39
6	Implementace	41
6.1	Aplikace	41
7	Závěr	43
	Literatura	44
A	Ukázka testovací sady	47

Kapitola 1

Úvod

Graffiti se již staly neodmyslitelnou součástí každodenního života. Lze je vidět na každém kroku a to nejen na mostech a betonových oploceních, ale naneštěstí i na fasádách domů, vozech veřejné dopravy a památkách. Ačkoliv policejní statistiky ukazují každoroční pokles sprejství, škody se stále pohybují v desítkách miliónů korun. Hlavním problémem spjatým se sprejstvím je nízká objasněnost jednotlivých případů, která je mírně přes 20%. Sprejství se špatně dokazuje, a tak pokud se pachatel nepřizná nebo není přistižen přímo při činu, zůstane skutek neobjasněn.

Cílem této práce je nalezení efektivního způsobu pro vyhledávání graffiti na základě podobnosti. Kvůli rozmanitosti jednotlivých graffiti je velmi náročné vyhledávat podobné vzorky. Při vhodném propojení s dalšími systémy by mohlo dojít k detailnímu mapování aktivit jednotlivých sprejerů při značném ušetření zdrojů.

V kapitole 2 této práce bude čtenáři přiblížena problematika sprejství a detekce objektů. Sekce 2.1 se věnuje graffiti subkultuře. Bude v ní objasněna základní terminologie a detailněji rozebrán speciální případ graffiti, kterým jsou tagy. V kapitole 3 budu popisovat možné možné přístupy k re-identifikaci při využití konvolučních neuronových sítí. Kapitola 4 se bude věnovat přípravě pro účely re-identifikace graffiti tagů. V rámci kapitoly 5 se pak nachází vyhodnocení jednotlivých experimentů. Poslední kapitola 6 se pak věnuje implementačním detailům a ukázce grafického rozhraní finální aplikace.

Kapitola 2

Analýza problému

V následující kapitole bych rád přiblížil subkulturu graffiti a vyjasnil některé pojmy s ní spjatou.

2.1 Graffiti

Graffiti, v Česku též známé pod ekvivalentem sprejerství, je druh výtvarného vyjádření na principu nanášení barev, nejčastěji fixem nebo sprejem, na veřejná místa. Ačkoliv se v rámci boji proti graffiti snaží města vyhradit legální plochy pro tvorbu, je většina tvorby nelegální. Informace uvedené v této kapitole byly čerpány z bakalářské práce věnující se graffiti [33].

2.1.1 Autoři

Autoři jsou dle policejních statistik nejčastěji muži v rozmezí 16-24 let. V subkultuře je pak sprejer označován jako *writer*. Je možno setkat se i s pojmem *toy*, jež označuje člověka, který s graffiti teprve začíná a je bez zkušeností. Tito lidé vystupují anonymně a užívají nejrůznější přezdívky. Tyto přezdívky se zpravidla skládají pouze z písmen a čísel, která se potom dobře píšou. Příkladem takových přezdívek mohou být například LAFOR, PAUSER nebo ASH721.

V zahraničí je činnost sprejerů v nemalé míře úzce spojená s gangy. Slouží k označování kontrolovaného území či k předávání pokynů ostatním členům. V Česku se writeři taktéž shlukují do skupin, takzvaných *crew*, ale účel těchto skupin je odlišný. Na rozdíl od gangů se nepodílí na organizovaném zločinu. Příkladem takových skupin může být *crew* z Brna FET - Fatal Error Team, jejíž tag je možné vidět na obrázku 2.1b nebo celorepubliková WATT s tagem na obrázku 2.1a.

Pro policii je velmi obtížné dopadnout jednotlivé členy. Sprejování totiž probíhá často nárazově a je předem důkladně plánováno. Samotná akce, pak nezahrnuje pouze jednotlivé *writery*, ale taktéž *checkery*, kteří mají za úkol hlídat okolí a případně varovat ostatní.

Počínání těchto skupin se často označuje jako *bombing*. Cílem je posprejovat co největší plochu. Většinou jsou proto voleny jednodušší formy graffiti, jako jsou tagy nebo tzv. throw-upy. Graffiti mají více projevů než pouze tagy a throw-upy, nicméně jejich rozbor je nad rámec této práce.



(a) WATT



(b) F.E.T.

Obrázek 2.1: Tagy crew

2.1.2 Tagy a throw-upy

Obecně platí, že čím menší náročnost na tvorbu graffiti, tím častěji se bude vyskytovat. Vytvoření tagu či throw-upu zabere pár desítek vteřin. S přihlédnutím k faktu, že v Česku je možné činnost gangů vyloučit, připadá na tento typ graffitů až 95% ze všech případů. Proto je potřeba se na tento typ zaměřit nejvíce.

Throw-up je zpravidla graffiti větších rozměrů jedné barvy. Často se jedná pouze o konturu prováděnou jedním tahem. Není ale výjimkou, že graffiti obsahuje výplň. Vytvoření throw-upu nezabere většinou více než 2 minuty. Námětem bývají nejčastěji přezdívky *writers* případně *crew*. Cení se zde převážně jednoduchost, rychlost a kontinuita čar. Jednotlivá díla jednoho autora se již vyznačují vizuální podobností, a proto je má smysl taktéž detekovat.



Obrázek 2.2: Throw-up AL

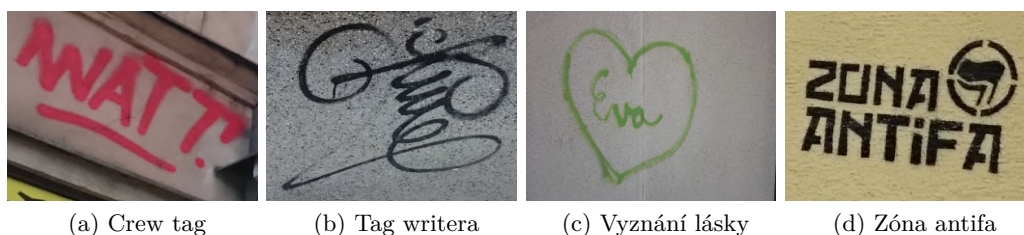
Tag je nejzákladnější a nejjednodušším prvkem graffiti. V reálném životě by se dal nejvíce připodobnit k podpisu. Taggovaním o sobě dává *writer* vědět okolí. Tagy se vyskytují povětšinou samostatně, případně mohou být společně s větším dílem (*piece*), kde slouží jako podpis autora.

Dle **U.S. Department of Justice Office of Community Oriented Policing Services**¹ lze tagy rozdělit do čtyř kategorií. Jednotlivé příklady je možno vidět na obrázcích 2.3.

- **Crew tagy** - Tyto nelze přisuzovat jednotlivci, jelikož je může psát kterýkoliv člen *crew*. Často se skládají z dvou až čtyř písmen a jsou dobře čitelná.

¹<http://www.popcenter.org/problems/pdfs/Graffiti.pdf>

- **Tagy writera** - Jedná se o osobitý podpis autora. Každý *writer* má svůj jedinečný tag, který by si nikdo neměl dovolit kopírovat. Některé tagy může být velmi obtížné přečíst nebo dokonce určit jejich orientaci.
- **Konvenční** - Nejčastěji jde o ojedinělý, či spontánní akt jedince. Může se jednat například o vyznání lásky.
- **Ideologické** - Do poslední skupiny pak spadají politické či nenávistné graffiti. Obsahem bývá vyjádření ideologického postoje. Často se zde však vyskytují rasistické, náboženské či etnické urážky.



(a) Crew tag (b) Tag writera (c) Vyznání lásky (d) Zóna antifa

Obrázek 2.3: Jednotlivé druhy tagů

2.2 Proces re-identifikaci objektů

Re-identifikace objektů patří v oblasti počítačového vidění mezi jeden z klíčových problémů. Pro člověka není problém rozlišit, zda se jedná o stejný objekt jako na jiné fotografii a to i v případě, kdy došlo ke změně měřítka, pozice či barevného schématu. V počítačovém vidění však zůstává tento problém i nadále výzvou. V následující kapitole přiblížím aktuální metody pro re-identifikaci, jež se užívají v poslední dekádě.

Jako re-identifikaci můžeme označit proces, kdy dojde k opětovné identifikaci objektu při změně pozorovacích podmínek. Tím může být myšleno vše od drobných změn jako například pozice nebo natočení, přes změnu světelných podmínek, až po kompletní změnu pozadí. Aktuálně se snažení zaměřuje převážně na re-identifikaci osob a automobilů z kamerových záznamů nicméně potenciál lze najít i v mnoha dalších odvětvích jako armáda, inteligentní bezpečnostní systémy nebo medicína. Na poli v oblasti re-identifikace došlo v posledních letech k značnému posunu. Hlavní zásluhu má na tom především rozvoj strojového učení a to primárně konvolučních neuronových sítí.

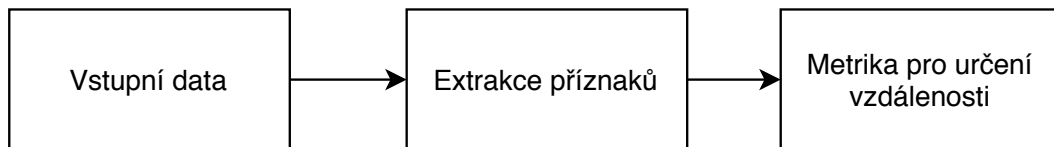


Obrázek 2.4: Ilustrace re-identifikace osoby. Převzato z publikace [31].

Proces re-identifikace se obvykle skládá ze tří základních fází, jak je možno vidět na obrázku 2.5. První fází je předzpracování vstupních dat. Předzpracování může zahrnovat mnoho operací a závisí na použitých metodách, které úpravy budou žádoucí. U strojového učení hrozí při malém množství vstupních dat například přeučení sítě, kdy se stává, že síť reaguje správně jenom na data z trénovací množiny nebo nedochází ke konvergenci dané sítě. Abychom tomuto stavu předcházeli dochází k augmentování dat. Uplatňovanými operacemi jsou pak změna kontrastu, saturace, jasu, rozměrů, měřítko nebo různé rotace. Další metody mohou zahrnovat gamma korekci nebo korekci barev a potlačení šumu.

Ve druhé fázi pak dochází k extrakci příznaků. Tradiční metody pro extrakci příznaků zahrnují například Histogram of Oriented Gradient (HOG) [7], scale-invariant feature transform (SIFT) [16], Speeded Up Robust Features (SURF) [3], Local Binary Pattern (LBP) [2]. Metody, které využívají hluboké učení jsou konvoluční neuronové sítě případně rekurentní neuronové sítě.

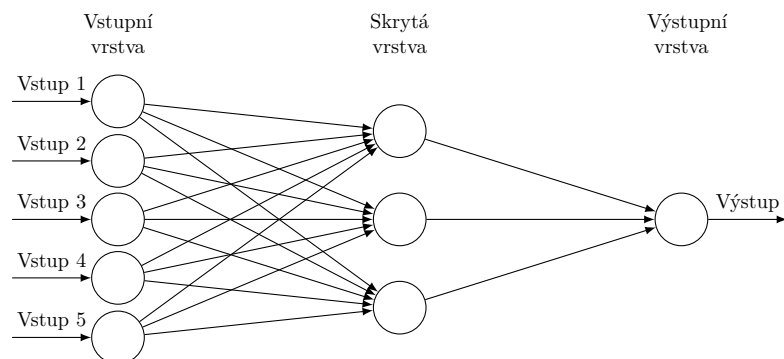
V závěrečné fázi se vypočítá vzdálenost mezi vstupním obrazem a ostatními obrazy v databázi. Pro posouzení podobnosti příznaků lze použít mnoho metrik. V případě měření vzdálenosti lze využít například euklidovskou metriku nebo manhattanskou metriku. Pro porovnání podobnosti vektorů existuje kosinova podobnost nebo Jaccardův koeficient.



Obrázek 2.5: Schéma obecného procesu re-identifikace

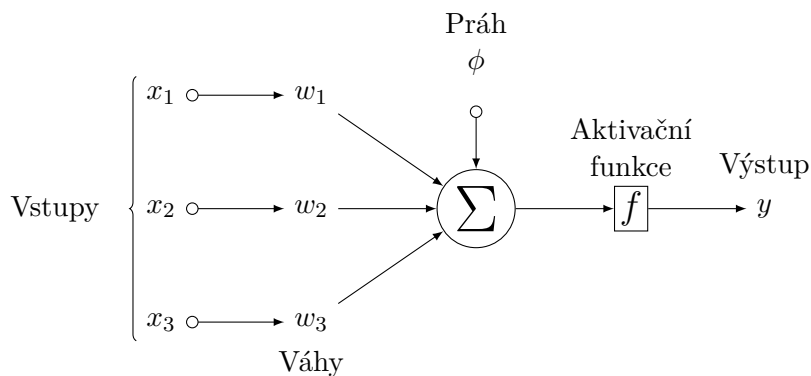
2.2.1 Neuronové sítě a hlubkové učení

Neuronové sítě patří mezi modely inspirované biologickým neuronem. Jednotlivé modely se skládají z několika vrstev, jak je možno vidět na obrázku 2.6, které se skládají z určitého počtu neuronů. Minimální počet vrstev jsou dvě, jež se označují jako vstupní a výstupní. Obvykle se však síť skládá z dalších skrytých vrstev, aby bylo možno síť naučit zvládat složitější úkoly jako například detekci objektů. Informace v této sekci byly převzaty z mé bakalářské práce [19].



Obrázek 2.6: Obecný model neuronové sítě

Všechny neurony jedné vrstvy jsou napojeny na neurony vrstvy následující. Vzniká tak rozsáhlá síť obsahující velké množství parametrů. Všechna jednotlivá spojení mají váhu, která je postupně upravována během učící fáze. Neurony jsou poté aktivovány v případě, kdy součet vstupních signálů přesáhne určitou hranici a aktivační funkce spustí výstup. Ukázkou neuronu je možné vidět na obrázku 2.7.



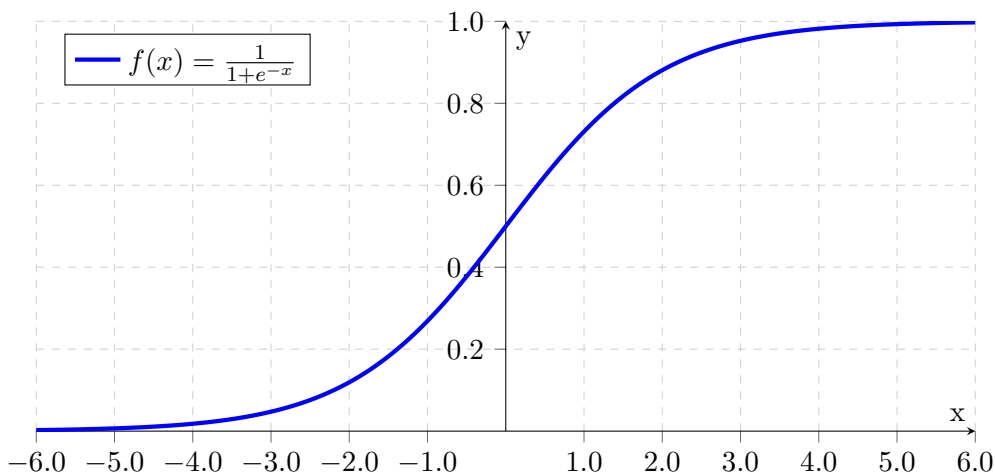
Obrázek 2.7: Model umělého neuronu

Výstup jednoho neuronu lze vypočítat jako:

$$y_k = \varphi \left(\sum_{i=1}^N (w_i x_i) + \phi \right) \quad (2.1)$$

kde y_k je výstup neuronu, φ aktivační funkce, x_i vstup a w_i jeho váha. Čím vyšší je hodnota váhy, tím důležitější je vstup. Součet prahu ϕ , který nabývá záporných hodnot, a součtu všech vstupů vynásobených jejich vahami se stává argumentem přenosové funkce.

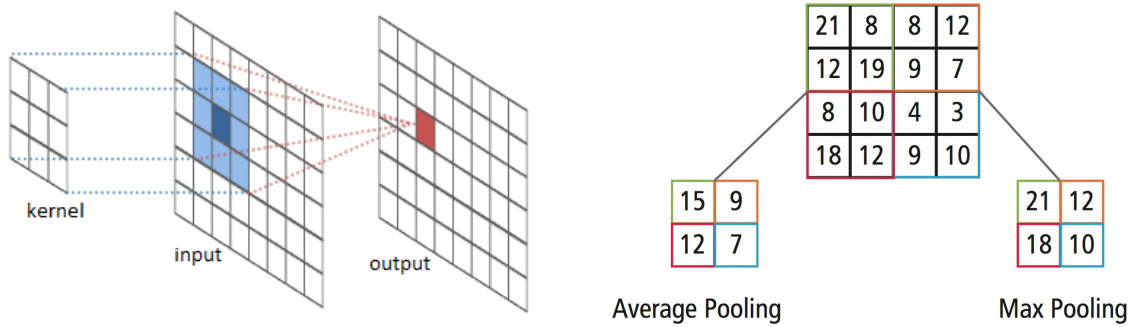
Přenosových funkcí je užíváno více druhů. Jedna z nich je sigmodiální přenosová funkce, kterou můžeme vidět na grafu 2.8.



Obrázek 2.8: Průběh sigmodiální přenosové funkce.

Obecný princip neuronových sítí jsme nastínili. V oblasti grafiky se uplatňují především (hloubkové) konvoluční neuronové sítě (CNN) [24]. Hloubkové značí, že síť bude obsahovat

alespoň jednu skrytou vrstvu. Tyto skryté vrstvy jsou pak konvoluční nebo pooling vrstvy. Ukázkou principu konvolučních vrstev je možné vidět na obrázku 2.9a. Na obrázku 2.9b lze vidět max a average pooling. Základní principy konvolučních neuronových sítí jsou inspirovány částí lidského mozku, která má za úkol zpracovávat vizuální vjemy. Převzato z [26].

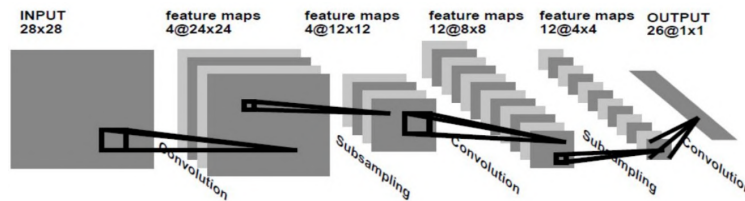


(a) Ukázkou principu konvoluční vrstvy. Na vstupní data je aplikována konvoluční maska. Jednotlivé pixely jsou vynásobeny příslušným koeficientem v dané buňce konvoluční masky a posčítány, čímž vznikne hodnota výstupního pixelu.

(b) Ukázkou pooling. Při subsamplingu 2×2 je vstupní obraz o velikosti 4×4 rozdělen na 4 matice velikosti 2×2 . Při average pooling je vypočítán průměr jednotlivých pixelů v 2×2 matici. U max-poolingu je pouze vybrána největší hodnota a zbylé se zahodí. Převzato z [11].

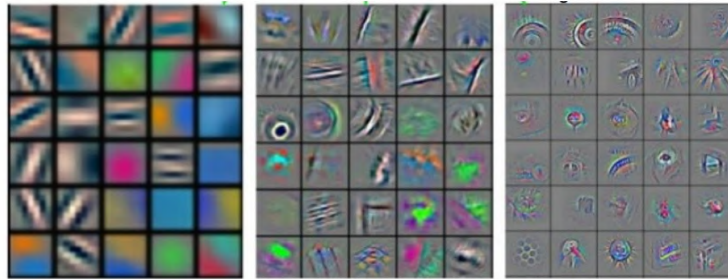
Obrázek 2.9: Ukázkou konvoluce a pooling.

Architekturu konkrétní sítě je možné vidět na obrázku 2.10. Vstupní obrázky s velikostí 28×28 pixelů jsou zpracovány konvolučním filtrem, který získává 3D mapy příznaků. Následují pak tzv. *pooling* vrstvy, kde dochází k zmenšení objemu dat. Stejná procedura se opakuje tak dlouho, dokud nedojde k vytvoření jednorozměrného vektoru, který obsahuje reprezentaci jednotlivých tříd, které v obraze zjistil.



Obrázek 2.10: Příklad sítě užívající skryté konvoluční a pooling vrstvy. Vstupní obrázek o velikosti 28×28 pixelů prochází přes tři konvoluční a dvě pooling vrstvy pro dosažení výsledného jednorozměrného vektoru. Převzato z [15].

Jednotlivé extrakce příznaků, ke kterým dochází během konvoluce, by se daly rozdělit do tří kategorií. Nízko-úrovňové příznaky, kde spadají barvy, čáry nebo kontrast. Do střední úrovně by se dalo zařadit rozeznání hran a rohů. Poslední úroveň příznaků už zahrnuje rozpoznání jednotlivých tříd či oblastí. Mezivýsledky jednotlivých kategorií je možno vidět na obrázku 2.11 níže.



Obrázek 2.11: Mezivýsledky skrytých vrstev od nejnižších příznaků po nejvyšší. Převzato z [29].

Podobně jako velké množství detekčních algoritmů i konvoluční nerunové sítě potřebují natrénovat jednotlivé váhy všech neuronů. Cílem algoritmu učení je nastavení vah w , aby se výstup sítě, co nejvíce přibližoval požadovanému výstupu z trénovací množiny. Tato skutečnost je charakterizována chybovou funkcí ERR . Ta pracuje s výstupy sítě, které porovnává s výstupy požadovanými. Cílem učení je tuto chybovou funkci minimalizovat.

Je k tomu užíván například algoritmus zpětného šíření chyby. Jedná se o jeden ze základních algoritmů pro učení vícevrstvých sítí. Tato gradientní metoda iterativně adaptuje jednotlivé váhy na základě parciální derivace chybové funkce viz rovnice 2.2.

$$w_{ij}(t+1) = w_{ij}(t) - \alpha \frac{\partial ERR}{\partial w_{ij}(t)} \quad (2.2)$$

Parametr α zde představuje učící krok a t příslušnou iteraci.

Kapitola 3

Možné přístupy při re-identifikaci

3.1 CBIR

Content-Based Image Retrieval [12] systémy jsou využívány pro vyhledávání relevantních obrázků napříč různými oblastmi výzkumu v oblasti počítačového vidění. Tyto systémy využívají příznaků daných tvarem, texturou nebo například barvami. Extrakce a vyhodnocení těchto příznaků je klíčovou částí na které závisí proces určení podobných obrázků. Vyhledávání tedy neprobíhá na základě dodatečných informací jako třeba klíčová slova nebo popis obrázku, ale je závislé pouze na samotném obrázku a informací v něm.

3.2 Klasifikace s využitím konvolučních neuronových sítí

Jedním z možných přístupů je využití nějakého z aktuálních systémů pro klasifikaci obrázků. Obecné schéma klasifikačních systémů lze vidět na obrázku 3.1. Vhodným kandidátem by mohl být nejmodernější systém pro klasifikaci BiT-L [13]. Aktuálně vykazuje 87,8% top 1 accuracy na datasetu ImageNet.

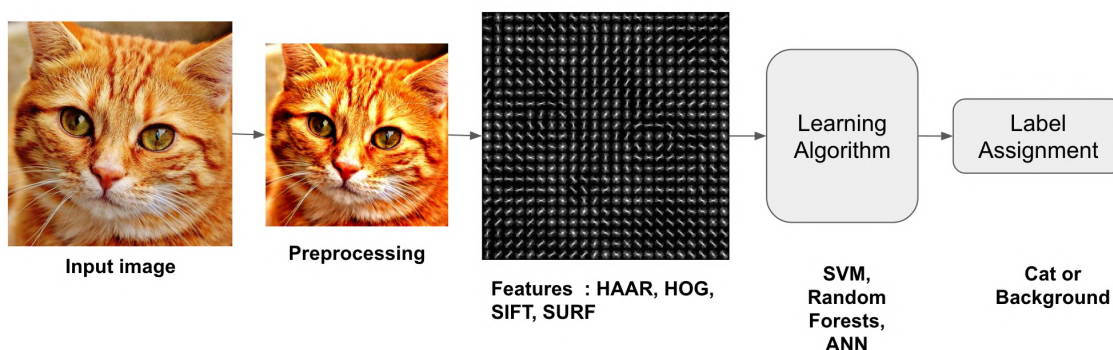
Zjevnou nevýhodou těchto systémů je fakt, že vyžadují znalost všech možných výstupů. Systémy totiž využívají učení s učitelem. Při přidání nového graffiti, pro které nebyla síť trénována, by muselo dojít k přetrénování sítě, aby byla schopna zpracovat novou třídu graffiti.

3.3 Siamská neuronová síť

Dašlí možností je využití siamské neuronové sítě, jejíž schéma je na obrázku 3.2. Tato metoda byla například využita při re-identifikaci osob [27]. Standardní siamská neuronová síť se skládá ze dvou totožných podsítí, jež sdílí parametry. Vstupem jsou tudíž dva rozdílné obrazy, které prochází neuronovou sítí, kde dochází k extrakci výstupních vektorů [12], které je možno porovnat. K tomu slouží spojovací funkce. Jako spojovací funkce je možné například využít:

Euclidovu vzdálenost, která je definována takto:

$$S_{euc}(x, y) = - \sum_i (x_i - y_i)^2 \quad (3.1)$$



Obrázek 3.1: Obecné schéma klasických klasifikačních systémů. Vstupní obraz je předzpracován pro následnou extrakci příznaků. Příznaky získané vhodným algoritmem jsou dále postoupeny učicímu algoritmu. V poslední fázi je vyhodnocen typ a pozice objektu v původním obraze. Převzato z [1].

Cosinovu vzdálenost:

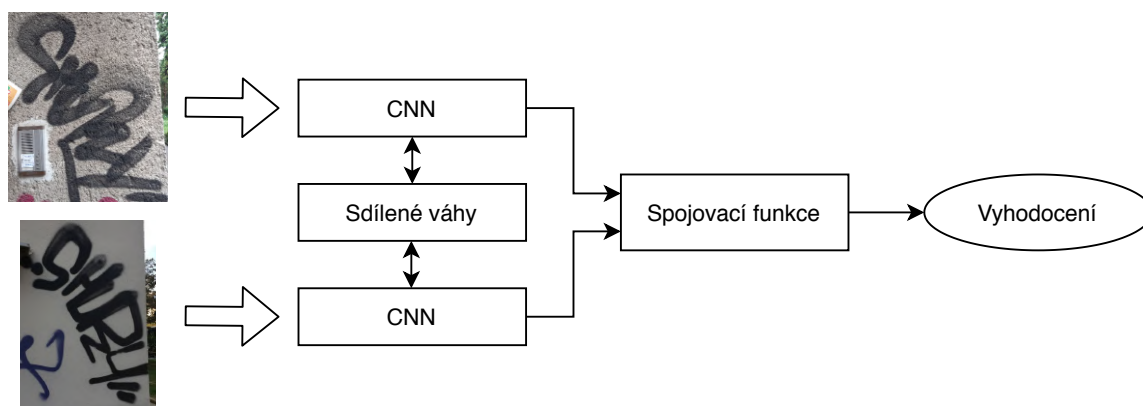
$$S_{cos}(x, y) = \frac{\sum_i(x_i y_i)}{\sqrt{\sum_i(x_i x_i) \sum_i(y_i y_i)}} \quad (3.2)$$

nebo rozdíl absolutních hodnot:

$$S_{abs}(x, y) = - \sum_i |x_i - y_i| \quad (3.3)$$

kde:

- x a y - výstupní vektory



Obrázek 3.2: Schéma siamské neuronové sítě. Dva rozdílné vstupy prochází sítí se stejnou architekturou a stejnými váhami. Na základě výstupu je rozhodnuto o shodě či neshodě vstupních obrázků.

3.4 Triplet loss

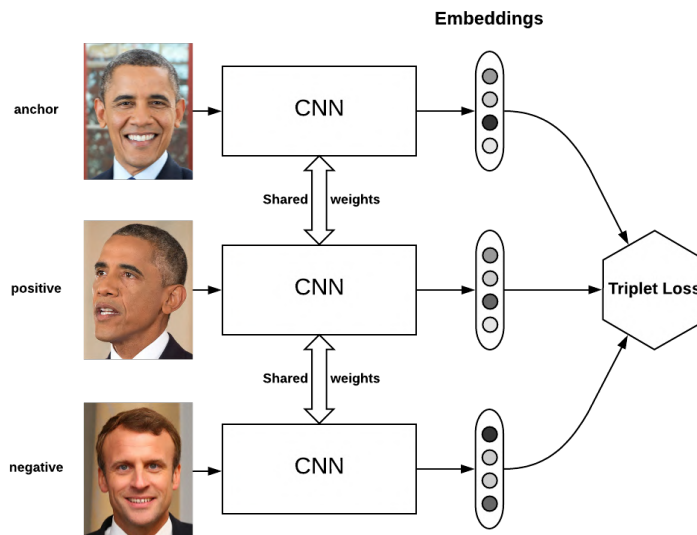
Triplet loss byla poprvé použita pro porovnávání obličejů ve výzkumu společnosti Google [21]. Standardně u učení s učitelem máme předem daný počet výstupů a síť je trénována s pomocí SoftMax loss funkce. Nicméně konkrétně pro praktické určení podobnosti obličejů potřebujeme mít proměnný počet tříd. Musíme být schopni porovnat dva obličeje a určit, zda se jedná o stejnou osobu či nikoliv.

Triplet loss tedy nabízí výbornou alternativu, kdy jsme schopni pracovat s novými třídami bez nutnosti přetrénování celé sítě. Dvě stejné tváře budou mít výsledné vektory produkované neuronovou sítí blízko u sebe a tudíž budou formovat shluky. Zatímco rozdílné tváře budou od sebe vzdálené.

Cílem triplet loss funkce je zajistit, že výsledné vektory dvou rozdílných obrázků budou od sebe velmi vzdálené, zatímco dva podobné obrázky budou u sebe blízko. Nicméně není žádoucí, aby vzdálenost mezi podobnými vstupy byla velmi malá. Taktéž není žádoucí, aby byly jednotlivé shluky blízko u sebe, a proto zavádíme minimální vzdálenost mezi nimi.

Jak již název napovídá, tak triplet loss funkce vyžaduje trojici obrázků. K referenčnímu vstupu je třeba ještě zvolit pozitivní vstup, který odpovídá obrázku ze stejné kategorie. Vzdálenost mezi výstupními vektory těchto dvou vstupů se snažíme minimalizovat. Třetím vstupním obrázkem je pak obrázek z jiné kategorie. Vzdálenost výstupních vektorů referenčního vstupu a negativního vstupu se snažíme maximalizovat.

Triplet loss je tedy často využívána v oblasti re-identifikace a to v různých oblastech. Příkladem může být re-identifikace aut [14], osob [10] nebo třeba delfínů [4]. Nabízí se tedy její využití při re-identifikaci graffiti tagů. Schematická ukázka systému, který využívá triplet loss k re-identifikaci obličejů lze vidět níže na obrázku 3.3.



Obrázek 3.3: Ukázka triplet loss s využitím obličejů. Referenční a pozitivní obličej (Obama) a k němu negativní (Macron). Převzato z [17].

Definici takové triplet loss funkce tedy vypadá následovně:

$$\mathcal{L} = \max(d(a, p) - d(a, n) + m, 0) \quad (3.4)$$

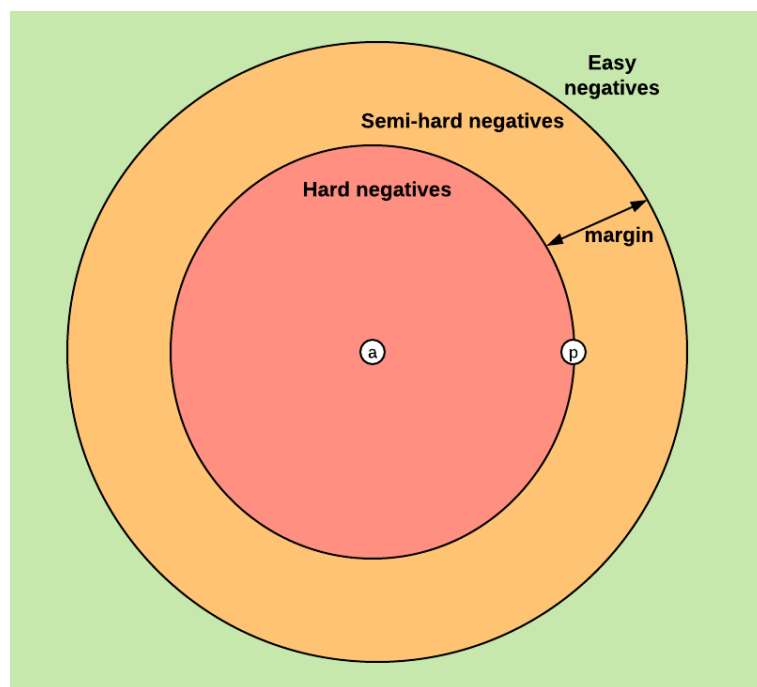
kde:

- d - funkce pro výpočet vzdálenosti mezi vektory
- a - referenční obrázek
- p - pozitivní obrázek vzhledem k referenčnímu
- n - negativní obrázek vzhledem k referenčnímu
- m - margin

3.4.1 Triplet mining

Na základě minimálního rozestupu mezi shluky jednotlivých tříd dělíme trojice do tří kategorií:

- **easy triplets** - mají loss 0, jelikož $d(a, p) + margin < d(a, n)$
- **hard triplets** - negativní obrázek je blíže referenčnímu než pozitivní, tudíž $d(a, n) < d(a, p)$
- **semi-hard triplets** - negativní je vzdálenější než pozitivní, ale $d(a, p) < d(a, n) < d(a, p) + margin$



Obrázek 3.4: Možné trojice s ohledem k loss funkci. Převzato z [17].

3.4.2 Online a offline triplet mining

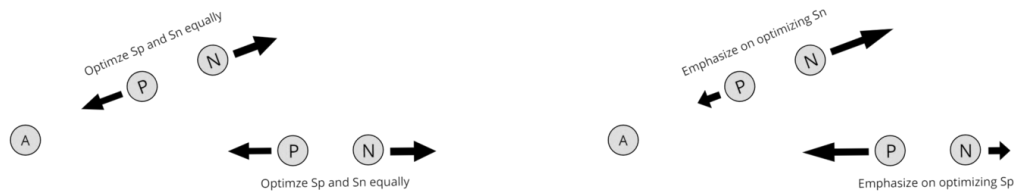
Správná volba kategorie trojic se kterou budeme trénovat je zásadní pro správné fungování celého systému. Pro možné získání jednotlivých vstupních dvojic a jejich kategorií lze využít dva možné přístupy.

Prvním přístupem je **offline mining**, kdy je na začátku každé iterace vyhodnocen trénovací dataset na trénované síti. Při získání výsledných vektorů je již možné vypočítat jejich vzdálenosti a určit, které z nich spadají do jaké kategorie. Tato metoda však není velice efektivní jelikož je nezbytné často přepočítat výsledné vektory trénovacích dat, aby bylo možné vytvořit trojice daných kategorií.

Jako alternativou se poté nabízí **online mining**, který byl představen v práci [21]. Principem je výpočet jednotlivých tripletů přímo během procesu trénování z konkrétní trénovací dávky a s ní pak nadále pracovat. V případě, že budeme mít velikost vstupní dávky B , tak po výpočtu výstupních vektorů pro dané obrázky, jsme schopni najít až B^3 trojic se kterými můžeme pracovat. Pochopitelně většina z nich nebude mít platné vstupy. Těmi jsou myšleny jeden obrázek ze stejné kategorie jako referenční a jeden negativní. Nicméně je tato metoda stále efektivnější, jelikož jsme schopni získat trojice bez nutnosti zpracování celé testovací sady. Online mining nabízí pak více variant pro výběr vhodných trojic z nichž neefektivnější se jeví *batch hards* jak bylo představeno v práci [10].

3.5 Circle loss

Jako alternativa k triplet loss funkci je možno využít také její robustnější variantu, kterou je circle loss [25]. Zatímco triplet loss funkce minimalizuje vzdálenost mezi referenčním a pozitivním obrázkem a maximalizuje vzdálenost mezi referenčním a negativním obrázkem stejnou měrou, tak u circle loss je možno tuto míru upravit. Situace, kdy dva rozdílné páry mají mezi sebou stejnou vzdálenost, ale mají odlišnou vzdálenost od referenčního obrázku je ilustrována na obrázcích 3.5 níže.



(a) Optimalizace vzdálenosti s využitím triplet loss. (b) Optimalizace vzdálenosti s využitím circle loss.

Obrázek 3.5: Rozdílná optimalizace pro optimalizace podobnosti obrázků u dvou rozdílných párů s využitím triplet a circle loss funkcí. S_p (značeno šipkou) značí podobnost referenčního a pozitivního obrázku P . S_n (značeno šipkou) pak podobnost referenčního a negativního obrázku N z páru. Triplet loss nebere v potaz vzdálenost páru od referenčního obrázku A . Circle loss se však snaží přistupovat k optimalizaci s rozdílnou mírou, která je úměrná vzdálenosti. Obrázky převzaty z [5].

Toto řešení je obzvlášť výhodné, jelikož v případě, kdy se pozitivní obrázek nachází blízko referenčnímu obrázku je vhodné zaměřit se spíše na oddálení negativního referenčního obrázku. To samé pochopitelně platí i pro situaci, kdy se snažíme zaměřit na přiblížení velmi vzdáleného pozitivního obrázku na úkor oddálení již velmi vzdáleného negativního obrázku.

Cirle loss funkce, která bere v potaz podobnosti v rámci třídy a mimo třídu, tedy vypadá následovně:

$$\mathfrak{L} = \log\left[1 + \sum_{j=1}^L \exp(\gamma a_j^n (s_j^n - \Delta_n)) \sum_{i=1}^K \exp(-\gamma a_i^p (s_i^p - \Delta_p))\right] \quad (3.5)$$

kde:

- L - celkový počet negativních obrázků v dávce
- a_j^n - váhový faktor pro negativní obrázek j , který odpovídá $a_j^n = [s_j^n - O_n]_+$
- s_j^n - podobnost negativního obrázku j a referenčního obrázku
- Δ_n - margin pro negativní obrázky
- K - celkový počet negativních obrázků v dávce
- a_i^p - váhový faktor pro pozitivní obrázek i , který odpovídá $a_i^p = [O_p - s_i^p]_+$
- s_i^p - podobnost pozitivního obrázku i a referenčního obrázku
- Δ_p - margin pro pozitivní obrázky
- γ - měřítko

Hyperparametry O_p , O_n , γ , Δ_p a Δ_n je možné zredukovat jejich nastavením na $O_p = 1 + m$, $O_n = -m$, $\Delta_p = 1 - m$ a $\Delta_n = m$. Tudíž stačí nastavit pouze vhodné hodnoty pro γ a margin m .

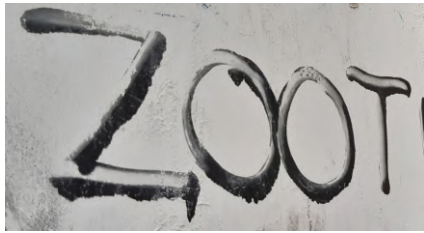
3.6 Re-identifikace graffiti tagů

Úloha porovnávání graffiti tagů na základě podobnosti je svou povahou velice specifická a jedinečná. Svým způsobem tagy fungují jako podpis, nicméně narozdíl od podpisu se graffiti vyskytují na různých podkladech a i úprava takového textu se může více či méně lišit v závislosti na autorovi, jak lze vidět na obrázku 3.6. V tomto ohledu k nim tedy nelze přistupovat stejně jako ke klasickým podpisům.

Na druhou stranu srovnáme-li re-identifikaci graffiti tagů s re-identifikací osob, tak se problém jeví jako méně složitý. U tagů máme například téměř garantováno, že budou zapsány barvou, jež je v kontrastu s podkladem. Dalším téměř jistým předpokladem je stálá orientace textu. Text by měl být totiž vždy čitelný, a tak můžeme předpokládat, že bude ve většině případů zapsán zleva doprava. Nicméně i zde existují výjimky, kde je text orientován například shora dolů. Ukázka různé orientace stejného graffiti tagu je na obrázku 3.7.

Dalším problémem, který danou úlohu komplikuje, je často rozdílné barevné schéma vstupních obrázků. V případě re-identifikace osob nebo automobilů předpokládáme, že daný objekt je barevně shodný. Tagy však často bývají kresleny různými barvami s rozdílným pozadím. Změna fixu nebo spreje také dost často znamená změnu tloušťky čar, čímž se opětovně snižuje podobnost těchto tagů. Vliv zvoleného prostředku pro vytvoření tagu na cílový graffiti lze vidět na obrázcích 3.8 níže.

Co se samotných graffiti týče nelze taky opomenout špatnou čitelnost mnohých graffiti tagů. Graffiti se často vyskytují na starých omítkách, které jsou ve špatném stavu. Nebývá



(a) ZOOT tag zapsán pouze velkými písmeny.



(b) ZOOT tag zapsán s užitím velkých i malých písmen.

Obrázek 3.6: Rozdílné druhy zápisu stejného graffiti tagu. Jednou je tag zapsán za pomoci velkých psacích písmen, v druhém případě je pak velikost písmen rozdílná. Mimoto je tag zdoben dalšími prvky kupříkladu kolečkem nad textem, podtržením nebo hvězdičkou za textem.



(a) Orientace zleva doprava.



(b) Orientace shora dolů.

Obrázek 3.7: Rozdílná orientace stejného tagu SHURY.



(a) RR psán fialovým sprejem na šedém podkladu.



(b) RR psán stříbrným fixem.



(c) RR psán černým sprejem na béžovém podkladu.

Obrázek 3.8: Rozdílné tagy v závislosti na použitém fixu či spreji.

také vyjímkou, že povrch na kterém jsou nakresleny je nějakým způsobem strukturovaný. V neposlední řadě bývají graffiti rozteklé nebo se je někdo snažil neúspěšně odstranit či přemalovat barvou. Zmíněné problémy lze vidět na ukázkách 3.9 níže



Obrázek 3.9: Samotné tagy jsou často špatně čitelné z rozličných důvodů. Tyto tagy jsou pak obtížněji re-identifikovatelné.

Ačkoliv existují práce, které se zabývají srovnáváním graffiti tagů na základě podobnosti [22, 23, 8], tak navržená řešení v těchto pracích vycházejí z předpokladu, že tagy jsou si skutečně velice podobné a ignorují výše zmíněné problémy. K řešení využívají tradiční metody pro extrakci příznaků jako jsou SURF a SIFT.

V posledních letech se graffiti dostává více pozornosti současně s rozvojem počítačového vidění. Neuronové sítě však byly primárně využity pro detekci graffiti ve scéně a to například v pracích [18, 20]. Mimoto, že samotná detekce graffiti je odlišnou oblastí počítačového vidění, než na kterou tato práce cílí, jsou v těchto případech brány všechny typy graffiti. Zahrnuty jsou tedy kromě tagů také piece a throw-up typy popsány v sekci 2.1.2.

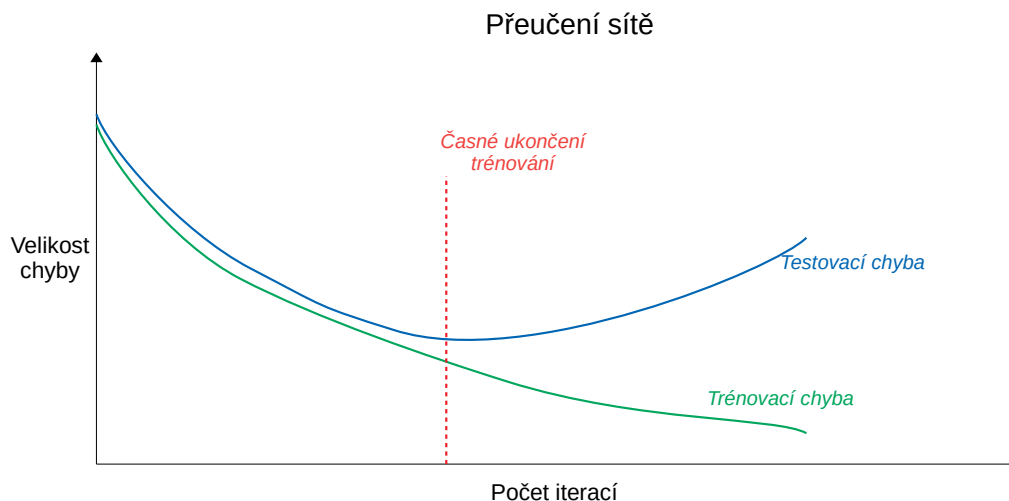
Vzmemme-li pak v potaz aktuální nejúspěšnější přístupy pro re-identifikaci osob, tak zjistíme, že mnohé z principů jež využívají není možné aplikovat při re-identifikaci graffiti tagů. Často je například využíváno dodatečných informací, které nabízí znalost kamery, kterou byl pořízen obrázek. Tyto kamery jsou zpravidla statické a tudíž zabírají stále stejnou oblast. Společně s informací o čase, kdy byl snímek pořízen, je této informace využíváno například v článku [28], který je *state of the art*. O tuto možnost však u re-identifikace graffiti přicházíme. Další práce [32] pak například využívá informace o natočení objektu v prostoru, což u tagů nelze.

Kapitola 4

Trénování sítě

Neuronové sítě byly trénovány na grafické kartě NVIDIA GeForce 1660 Ti disponující 6GB pamětí. Z důvodu výpočetní náročnosti celého procesu bylo třeba nalézt vhodné parametry, které zajistí konvergenci sítě. Nejzásadnějším parametrem se ukázala velikost jedné učící dávky a to hlavně z důvodu výpočetní náročnosti u větších modelů. Další parametry jsou pak například velikost učícího kroku či změna během pozdějších iterací. Vzhledem k tomu, že nebylo třeba trénovat celý model od začátku, nýbrž byly využity již předtrénované modely, které byly doučeny na cílových datech, došlo k značnému úbytku potřebných iterací.

Jistým problémem, který bylo třeba ohlídat, bylo přeučení. Z důvodu malé velikosti trénovacích dat totiž v pozdějších fázích učení dochází ke ztrátě generalizace. Výsledný model je pak specializován na trénovací data a ztrácí schopnost pracovat s objekty mimo datovou sadu. Princip přeučení je ukázán na obrázku 4.1.



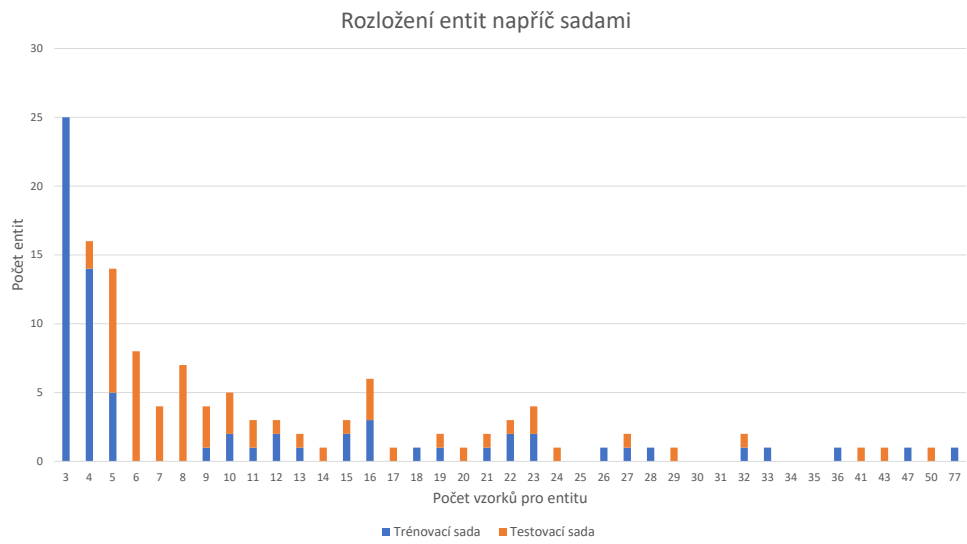
Obrázek 4.1: Zelená křivka znázorňuje trénovací chybu a modrá křivka odpovídá validační chybě. Obě chyby napříč iteracemi klesají, dokud nenarazí do bodu, kdy začne validační chyba narůstat z důvodu přeučení. Je důležité proto vybrat vhodné natrénované váhy, které odpovídají minimu u validační chyby.



Obrázek 4.3: Ukázka datasetu.



Obrázek 4.4: Počet tříd disponující daným počtem vzorků.



Obrázek 4.5: Rozložení datasetu na trénovací a testovací sadu.

Kapitola 5

Experimenty

Cílem práce je najít nejvhodnější přístup pro re-identifikaci graffiti tagů. Tím je myšlena vhodná kombinace předzpracování dat, modelu neuronové sítě, loss funkce a funkce pro výpočet vzdálenosti mezi jednotlivými výslednými vektory. Jako loss funkce byly zvoleny triplet a circle loss. Jako metriky pak kosinova podobnost, Euklidovská vzdálenost a Manhattanová vzdálenost.

5.1 Metriky pro vyhodnocování úspěšnosti

Jedním ze sledovaných ukazatelů při porovnávání efektivity jednotlivých modelů je **mean average precision** (dále jen mAP). Tato metrika je užívána napříč všemi oblastmi počítačového vidění, ať už se jedná o klasifikaci, detekci, tak i re-identifikaci.

Pro výpočet mAP je důležité nejprve definovat, co přesně je myšleno přesností. Pro účely re-identifikace je přesnost poměr vzorků, které jsou relevantní vzhledem k query ku celkovému počtu získaných. Tento vzorec se dá tedy zapsat jako:

$$precision = \frac{|\{\text{relevant images}\} \cap \{\text{retrieved images}\}|}{|\{\text{retrieved images}\}|} \quad (5.1)$$

Obecně je přesnost vyhodnocena pro všechny získané vzorky, nicméně lze brát v potaz pouze jistý počet získaných vzorků. Model je tedy vyhodnocen pouze pro několik prvních nejrelevantnějších výsledků. Tato obdoba je známa jako cut-off a metrika *precision at k* zkráceně pak $P@k$.

Výše definovanou přesnost dále použijeme pro výpočet $AP@k$. Tato metrika umožňuje lepší způsob jak měřit schopnost modelu v řazení relevantních výsledků pro danou query.

$$AP@k = \frac{1}{GTP} \sum_k^n P@k \times rel@k \quad (5.2)$$

GTP (*ground truth positives*) v dané rovnici značí celkový počet pozitivních vzorků, $P@k$ je přesnost vydefinována výše a $rel@k$ označuje funkci relevance. Funkce relevance nabývá hodnotu 1 pokud je získaný vzorek na pozici k relevantní a hodnotu 0 v opačném případě.

Výpočet mAP je již jednoduchý. Vypočteme odpovídající AP pro každý query obrázek. Query obrázků může být libovolný počet a měl by korespondovat s třídami v testovací sadě. Samotné mAP je pak aritmetickým průměrem nad všemi vypočtenými AP.

$$mAP = \frac{1}{N} \sum_{i=1}^N AP_i \quad (5.3)$$

Druhou velmi často užívanou metrikou je **cumulative matching characteristics** (dále jen CMC). Která posuzuje zda, se v dané seřazené množině vzorků vyskytuje, alespoň jeden pozitivní. CMC je pak pouze aritmetický průměr nad všemi query obrázky.

$$Acc_k = \begin{cases} 1 & \text{pokud se v top-}k \text{ vzorků nachází alespoň jeden relevantní} \\ 0 & \text{jinak} \end{cases} \quad (5.4)$$

5.2 Modifikace trénovací množiny

Z důvodu malého počtu vzorků pro trénování byla trénovací množina různými způsoby modifikována. Operace použité pro úpravu dat byly ořez, převrácení obrazu, změna saturace a změna jasu. Ukázky dopadu jednotlivých operací lze vidět na následujících obrázcích. Tedy ukázka 5.1 pro ořez, ukázka 5.2 pro převrácení obrazu, ukázka 5.3 pro změnu saturace obrazu a ukázka 5.4 pro změnu jasu.



Obrázek 5.1: Ukázka ořezu vzorku.



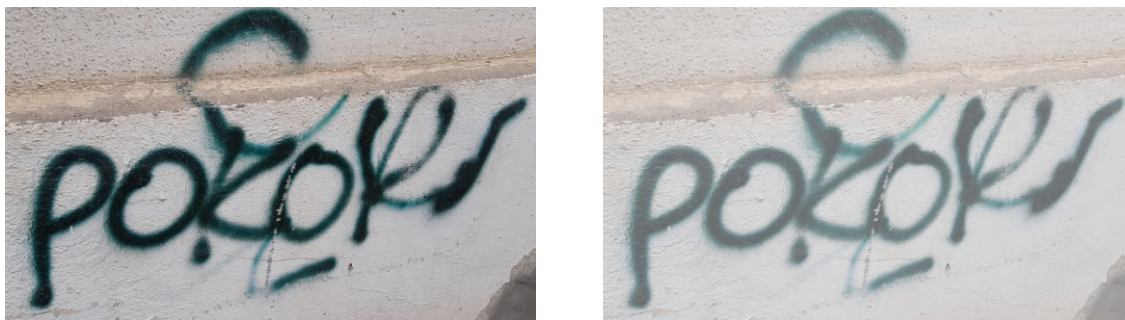
Obrázek 5.2: Úprava dat s využitím převrácení.

5.3 Neuronové sítě s využitím triplet loss

Jako stěžejní princip pro re-identifikaci graffiti tagů byla zvolena triple loss funkce. Loss funkce byla aplikována společně s různými modely neuronových sítí, které byly laděny s využitím předtrénovaných vah pro jednotlivé sítě. Těmito sítěmi jsou ResNet, MobileNet



Obrázek 5.3: Úprava dat při změně saturace obrazu.



Obrázek 5.4: Úprava dat při změně jasu.

a VGG16. U jednotlivých modelů bylo experimentováno s parametry při trénování, augmentací datové sady a různých metod pro zjištění vzdálenosti jednotlivých vektorů. Při prvotních experimentech bylo využito aktivační funkce Softplus namísto práce s hyperparametrem margin.

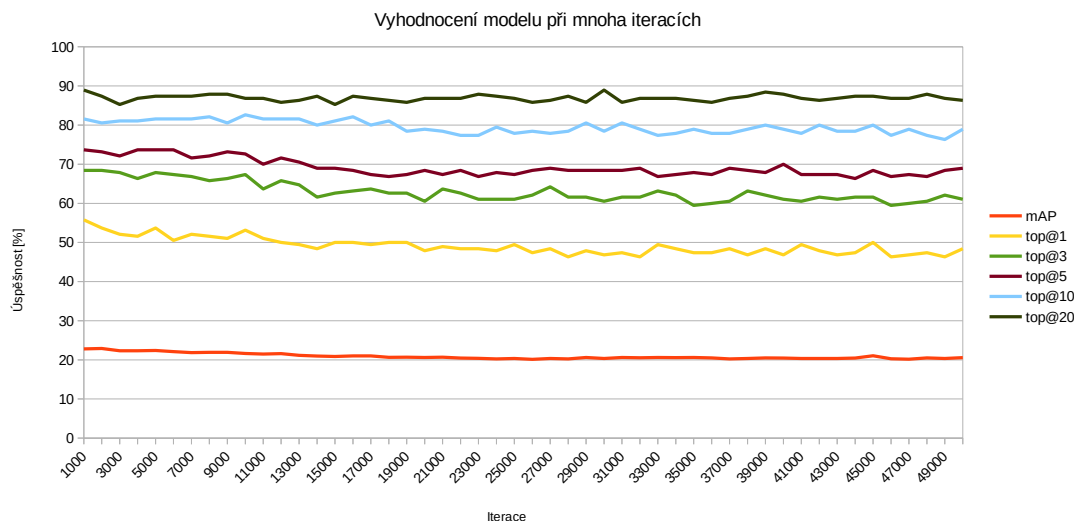
Jako směrodatný ukazatel pro vyhodnocení efektivity jednotlivých modelů byl použit top@10 rank s využitím CMC, jelikož je předpoklad u výsledné aplikace, že bude zobrazovat deset nejpodobnějších obrázků oproti query. Dále pak mAP, které lépe reflektuje efektivitu modelu napříč všemi tagy v datasetu. Všechny výsledky jsou vyhodnoceny na testovacím datasetu.

5.3.1 ResNet

Jako prvním testovaným modelem je síť ResNet [9]. Model pracuje s předposlední vrstvou sítě a výsledné vektory mají velikost 128. Z průběhu experimentů a povahy trénovací množiny dat bylo vyzorováno, že stačí pouze malý počet iterací při trénování modelu. Trénování bylo nastaveno s následujícími parametry:

- learning rate: 0.0003
- batch KP: 4 * 18
- iteration decay: 35000
- mining strategy: batch-hard
- augmentation: none

Ukázka vývoje přesnosti modelu lze vidět na grafu níže, kdy model prošel 50000 iteracemi.



Obrázek 5.5: Vyhodnocení modelu při velkém množství iterací.

Stejně tak byla experimentálně zjištěna ideální velikost dávky pro trénování. Ta byla stanovena na $P = 18$ a $K = 4$, kde P je počet identit a K počet jednotlivých vzorků z dané identity. Velikost dávky je tedy $P \times K$ ze které jsou následně dopočítány nejtěžší pozitivní dvojice a nejllehčí negativní dvojice pro jednotlivé vzorky. Tímto způsobem jsou dopočítány hard-triplets, nicméně pouze v rámci dané dávky. Dochází tak však k efektivnějšímu trénování.

Prvotní experimenty probíhaly nad zjištěním ideálního koeficientu učení (*learning rate*). Ideální koeficient se pohybuje mezi 0.0001 až 0.0007. Obzvláště u vyšších koeficientů dochází k velké ztrátě efektivity modelu. Jako nejvhodnější koeficient byl pak vybrán 0.0002 z důvodu nejvyššího dosaženého mAP, což činí 25.46%, jak lze vidět v tabulce 5.1 níže.

Další experimenty pak byly provedeny nad zjištěním vhodné metriky pro určení vzdálenosti výsledných vektorů. Jako nejvhodnější se ukázalo využití Euklidovské vzdálenosti, kdy se podařilo dosáhnout mAP 25.46%. Při využití ostatních metrik došlo k poklesu mAP a to na 23.49% v případě kosinovy podobnosti a 24.11% s Manhattanskou vzdáleností. Obdobně tomu bylo i v případě top@10 rank, což lze vidět v tabulce 5.2.

K dalšímu možnému navýšení přesnosti modelu lze dospět úpravami trénovacích dat. Pro úpravu dat byly využity operace zmíněné v sekci 5.2. Napříč trénovací dávkou jsou data vždy náhodně upravena. V tabulce 5.3 je možno vidět vyhodnocení vybraných kombinací při augmentaci dat. U prováděných experimentů došlo při náhodném ořezu pokaždé k zlepšení přesnosti modelu. Proto jsou všechny kombinace, které využívají více způsobů augmentace, kombinovány s ořezem. V nejlepším případě došlo k navýšení mAP o takřka 2.5% a to na 27.93% při kombinaci všech způsobů úpravy dat.

Jak lze vidět v grafu 5.6, tak top@1 rank dosahuje úspěšnosti přes 60%. Stejně tak jako u mAP, které se pohybuje pouze kolem 28%, je to dáno převážně různorodostí jednotlivých entit. Na obrázcích níže lze vidět vyhodnocení query pro dvě rozdílné entity.

Tag RMNK je povětšinou velice podobný a všechny vzorky v testovací sadě jsou stejně orientované, snadno čitelné a s drobnými změnami rukopisu. Přehled všech obrázků v tes-

Učící koeficient	Metrika	Trénovací dávka	Augmentace	top@10 rank	mAP
0.00008	E	72	-	80.00%	22.61%
0.00009	E	72	-	79.47%	22.45%
0.0001	E	72	-	82.63%	23.44%
0.0002	E	72	-	82.11%	25.46%
0.0003	E	72	-	83.68%	24.60%
0.0004	E	72	-	80.00%	22.37%
0.0005	E	72	-	84.21%	25.27%
0.0006	E	72	-	81.05%	23.63%
0.0007	E	72	-	82.11%	25.21%
0.0008	E	72	-	49.70%	8.88%
0.0009	E	72	-	58.95%	11.96%
0.001	E	72	-	28.42%	4.22%
0.002	E	72	-	23.16%	4.08%

Tabulka 5.1: Vyhodnocení modelu využívající sítě ResNet s triplet loss funkcí při různých variantách učícího koeficientu. Metrika E reprezentuje Euklidovskou vzdálenost.

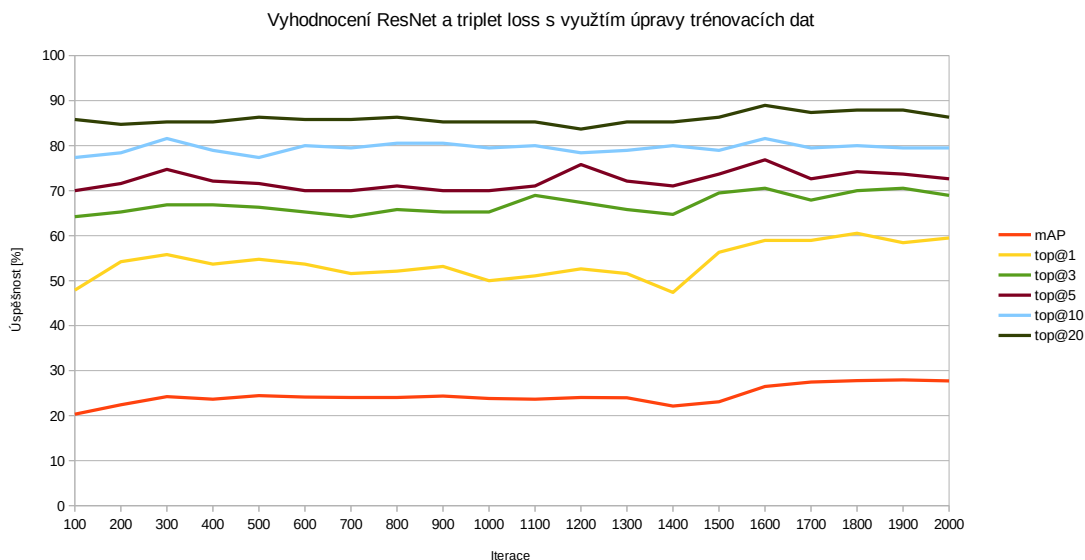
Učící koeficient	Metrika	Trénovací dávka	Augmentace	top@10 rank	mAP
0.0002	E	72	-	82.11%	25.46%
0.0002	K	72	-	81.23%	23.49%
0.0002	M	72	-	80.00%	24.11%

Tabulka 5.2: Nejvyšší dosažené hodnoty modelu ResNet s triplet loss funkcí při využití různých metrik pro určení vzdálenosti(podobnosti) vektorů. Metrika E reprezentuje Euklidovskou vzdálenost, K kosinovu podobnost a M Manhattanskou vzdálenost.

Učící koeficient	Metrika	Trénovací dávka	Augmentace	top@10 rank	mAP
0.0002	E	72	O	80.53%	24.57%
0.0002	E	72	P	79.47%	25.67%
0.0002	E	72	S	83.16%	24.77%
0.0002	E	72	J	82.11%	24.76%
0.0002	E	72	O + P	84.21%	25.99%
0.0002	E	72	O + S	83.16%	27.12%
0.0002	E	72	O + J	82.68%	26.51%
0.0002	E	72	O + P + S	83.68%	26.21%
0.0002	E	72	O + P + J	80.53%	25.36%
0.0002	E	72	O + S + J	82.11%	24.26%
0.0002	E	72	O + P + S + J	81.58%	27.93%

Tabulka 5.3: Vyhodnocení modelu ResNet s triplet loss funkcí při úpravě trénovací množiny. O, P, S a J pak značí jednotlivé operace nad datasetem, či-li ořez, přetočení, změnu saturace a jasů.

tovací sadě je možno vidět v příloze A.1. Výsledná query nejlépe si stojícího modelu je na obrázku 5.7.



Obrázek 5.6: Vyhodnocení efektivity modelu při využití všech možných úprav datasetu.

Nadruhou stranu ZOOT je ukázkovým příkladem, kdy jsou tagy velmi rozličné, což je vidět v příloze A.2. Samotný tag je psán různými způsoby. Liší se rukopisem, kombinují se náhodně malá a velká písmena, kapitálka a psací písmo. Často je tag doplněn interpunkčními znaménky nebo jinými malůvkami. A i orientace občas bývá nestandardně z levého horního rohu do pravého spodního. Z daného důvodu je úspěšnost daleko horší, jak jde vidět z vrácených výsledků na obrázku 5.8.

5.3.2 MobileNet

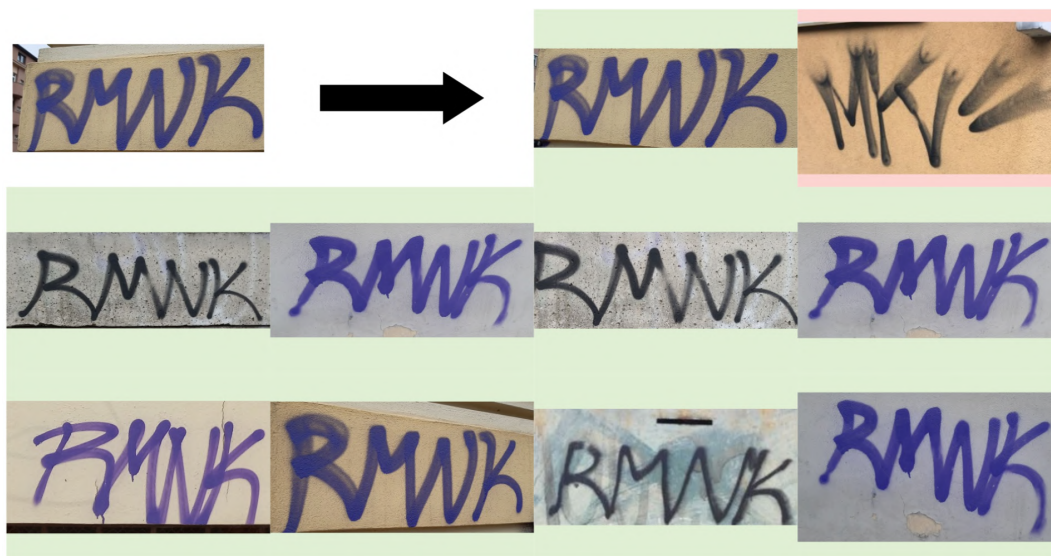
Dalšího možného navýšení efektivity můžeme docílit změnou neuronové sítě. Proto byl zvolen model MobileNet. Průběh experimentů probíhal obdobně jako u předchozího modelu ResNet. Z tohoto důvodu jsou vybrány stejné experimenty jako byly provedeny v předchozí kapitole.

Již při prvních experimentech bylo evidentní zvýšení přesnosti modelu. Navíc se ukázalo, že síť konverguje i při vyšším učícím koeficientu. Nejvhodnější koeficient pak odpovídá hodnota 0.0009 a povedlo se při něm dosáhnout mAP 30.01%. Povedlo se tak dosáhnout dvouprocentního nárůstu oproti ResNet a to bez využití augmentace dat. Kompletní vyhodnocení vybraných učících koeficientů je v tabulce 5.4.

Obdobně jako u sítě ResNet nejlépe se opět projevila Euklidovská vzdálenost. Srovnání je pak možno vidět v tabulce 5.5.

I s MobileNet byly provedeny experimenty a augmentací dat, které vedly k navýšení přesnosti modelu. Podařilo se dokonce dosáhnout top@10 rank 92.11% a mAP 33.13%. Efektivita při využití jednotlivých augmentací je shrnuta v tabulce 5.6.

Obdobně jako u modelu s ResNet i zde uvádím dvě ukázkové query. Pro srovnání byly vybrány stejné vstupy. Jak lze vidět na obrázku 5.9, tak u tagu, jež není moc rozmanitý,



Obrázek 5.7: Ukázka prvních 10 vrácených obrázků z testovací sady graffiti tagu RMNK. Kvůli podobnosti jednotlivých vzorků je shoda v devíti případech.



Obrázek 5.8: Ukázka prvních 10 vrácených obrázků z testovací sady graffiti tagu ZOOT. Kvůli rozmanitosti jednotlivých vzorků je shoda pouze ve čtyřech případech.

Učící koeficient	Metrika	Trénovací dávka	Augmentace	top@10 rank	mAP
0.00008	E	72	-	82.63%	25.27%
0.00009	E	72	-	82.11%	25.36%
0.0001	E	72	-	82.63%	26.19%
0.0002	E	72	-	82.63%	26.93%
0.0003	E	72	-	83.16%	28.57%
0.0004	E	72	-	85.26%	28.28%
0.0005	E	72	-	87.37%	28.65%
0.0006	E	72	-	86.84%	29.97%
0.0007	E	72	-	86.84%	29.37%
0.0008	E	72	-	85.79%	28.73%
0.0009	E	72	-	86.32%	30.01%
0.001	E	72	-	87.89%	29.65%
0.002	E	72	-	82.63%	28.66%

Tabulka 5.4: Vyhodnocení modelu využívající síť MobileNet při různých variantách učícího koeficientu. Metrika E reprezentuje Euklidovskou vzdálenost.

Učící koeficient	Metrika	Trénovací dávka	Augmentace	top@10 rank	mAP
0.0002	E	72	-	86.32%	30.01%
0.0002	K	72	-	84.52%	28.82%
0.0002	M	72	-	85.26%	29.55%

Tabulka 5.5: Nejvyšší dosažené hodnoty MobileNet při využití různých metrik pro určení vzdálenosti(podobnosti) vektorů. Metrika E reprezentuje Euklidovskou vzdálenost, K kosinovu podobnost a M Manhattanskou vzdálenost.

Učící koeficient	Metrika	Trénovací dávka	Augmentace	top@10 rank	mAP
0.0009	E	72	O	86.84%	31.56%
0.0009	E	72	P	87.89%	31.34%
0.0009	E	72	S	87.89%	30.66%
0.0009	E	72	J	85.79%	28.36%
0.0009	E	72	O + P	83.68%	31.44%
0.0009	E	72	O + S	87.37%	33.13%
0.0009	E	72	O + J	87.89%	31.50%
0.0009	E	72	O + P + S	92.11%	31.59%
0.0009	E	72	O + P + J	91.05%	30.04%
0.0009	E	72	O + S + J	86.11%	30.28%
0.0009	E	72	O + P + S + J	84.58%	30.93%

Tabulka 5.6: Vyhodnocení modelu při úpravě trénovací množiny. O, P, S a J pak značí jednotlivé operace nad datasetem, či-li ořez, přetočení, změnu saturace a jasů.

bylo pozitivních všech 10 nejbližších vzorků nalezených modelem. Tag ZOOT má opět pět shodných vzorků. Zajímavé je, že ve čtyřech případech, kdy byl navrácen špatný tag, tak se jedná o tag CLOE. Ukázku pro tag ZOOT je k vidění na obrázku 5.10. Kompletní vyhodnocení pro neefektivnější model je možno vidět v grafu 5.12.

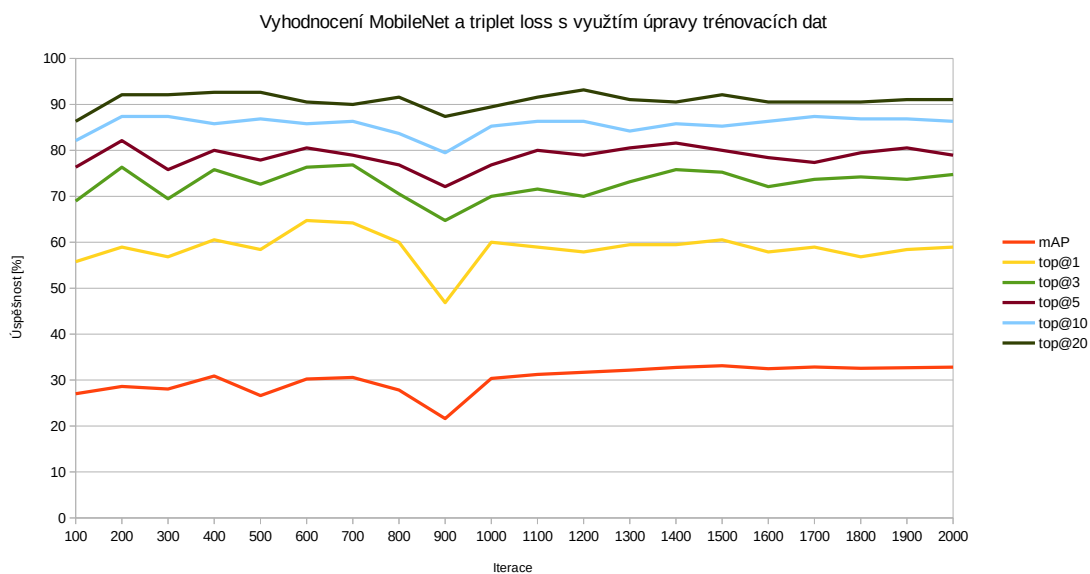


Obrázek 5.9: Ukázka prvních 10 vrácených obrázků z testovací sady graffiti tagu RMNK. Shoda je ve všech 10 případech, jelikož tag RMNK není rozmanitý.



Obrázek 5.10: Ukázka prvních 10 vrácených obrázků z testovací sady graffiti tagu ZOOT. Kvůli rozmanitosti jednotlivých vzorků je shoda pouze v pěti případech.

V předchozích experimentech bylo využito aktivační funkce Softplus. Při další iteraci experimentů byl zaveden margin. Jako optimální velikost se ukázala 0.6 a povedlo se při ní dosáhnout mAP 33.23%. To bez využití augmentace dat. Kompletní vyhodnocení vybraných velikostí margin se nachází v tabulce 5.7.



Obrázek 5.11: Vyhodnocení efektivity modelu MobileNet s triplet loss funkcí při využití úprav datasetu.

Margin	Metrika	Trénovací dávka	Augmentace	top@10 rank	mAP
0.1	E	72	-	87.96%	31.39%
0.15	E	72	-	90.58%	29.00%
0.20	E	72	-	88.48%	30.04%
0.25	E	72	-	87.96%	32.99%
0.35	E	72	-	87.96%	31.19%
0.5	E	72	-	86.39%	30.61%
0.6	E	72	-	87.58%	33.23%
0.7	E	72	-	86.91%	31.63%
0.8	E	72	-	87.43%	31.72%
1	E	72	-	88.48%	31.85%
1.5	E	72	-	85.86%	30.57%

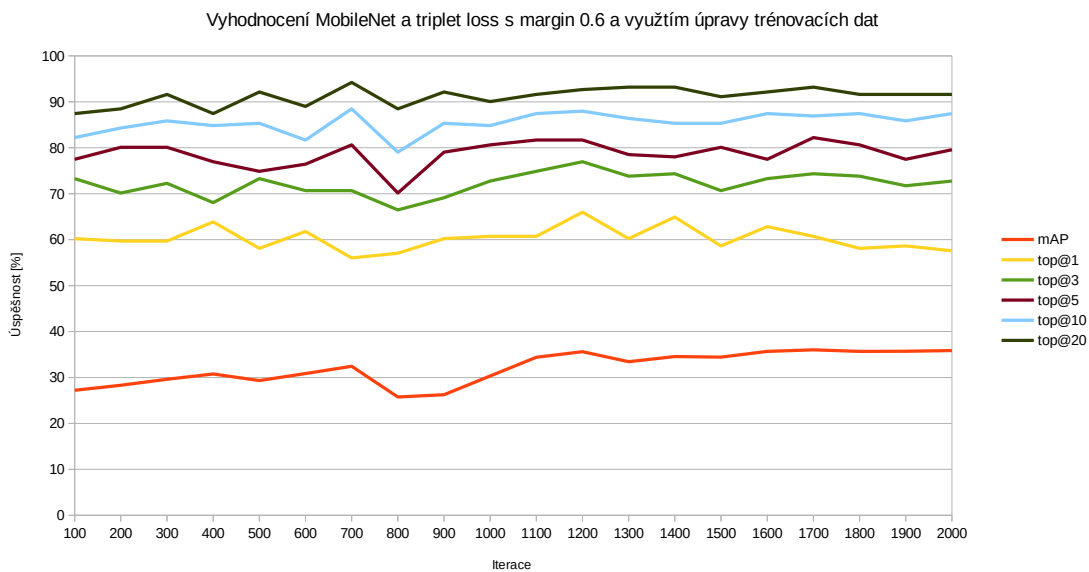
Tabulka 5.7: Výsledky experimentů pro zjištění velikosti margin u triplet loss funkce.

Opětovně došlo k trénování modelu s využitím augmentace dat, kdy se podařilo opětovně drobně navýšit mAP. Vyhodnocení dílčích experimentů lze vidět v tabulce 5.8. Nejvyšší dosažené mAP dosahuje 36.02%, což je maximální dosažené napříč všemi konfiguracemi.

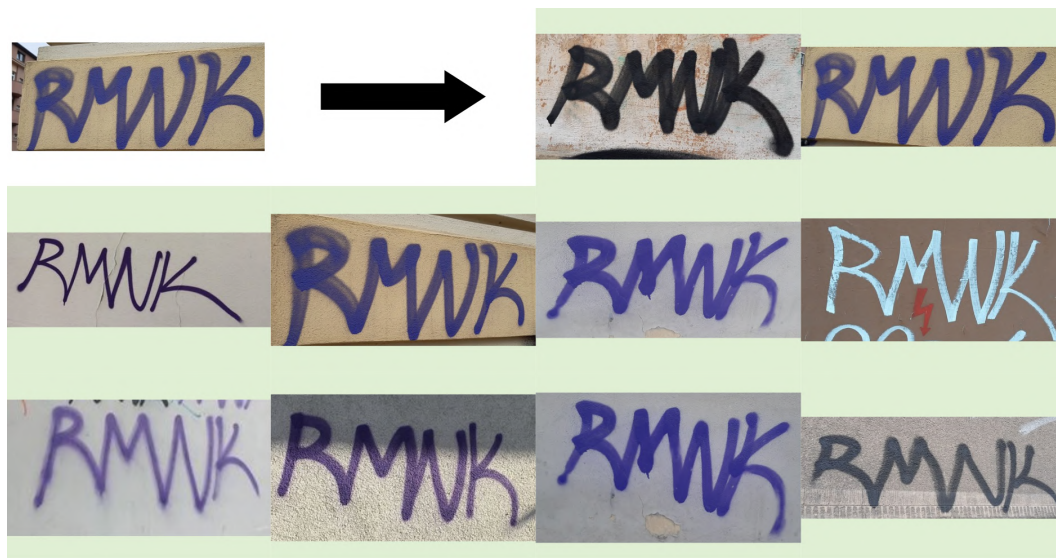
Učící koeficient	Metrika	Trénovací dávka	Augmentace	top@10 rank	mAP
0.0009	E	72	O	89.01%	34.4%
0.0009	E	72	P	87.96%	30.81%
0.0009	E	72	S	89.53%	34.07%
0.0009	E	72	J	87.96%	33.08%
0.0009	E	72	O + P	86.91%	31.67%
0.0009	E	72	O + S	86.91%	36.02%
0.0009	E	72	O + J	90.58%	33.72%
0.0009	E	72	O + P + S	90.58%	32.91%
0.0009	E	72	O + P + J	86.91%	32.57%
0.0009	E	72	O + S + J	86.11%	33.6%
0.0009	E	72	O + P + S + J	87.96%	32.09%

Tabulka 5.8: Vyhodnocení modelu při úpravě trénovací množiny s využitím margin 0.6. O, P, S a J pak značí jednotlivé operace nad datasetem, či-li ořez, přetočení, změnu saturace a jasů.

Kompletní vyhodnocení modelu napříč iteracemi je na obrázku 5.12, kdy nejvyšší hodnoty mAP dosáhne při iteraci 1700 během trénování modelu. Ukázkové query jsou posléze na obrázku 5.13 pro tag RMNK a na obrázku 5.14 pro tag ZOOT.



Obrázek 5.12: Vyhodnocení efektivity modelu MobileNet s triplet loss funkcí při využití úprav datasetu s nastaveným margin 0.6.



Obrázek 5.13: Ukázka prvních 10 vrácených obrázků z testovací sady graffiti tagu RMNK. Shoda je ve všech 10 případech, jelikož tag RMNK není rozmanitý. Model MobileNet s triplet loss funkcí a margin 0.6.



Obrázek 5.14: Ukázka prvních 10 vrácených obrázků z testovací sady graffiti tagu ZOOT. Kvůli rozmanitosti jednotlivých vzorků je shoda pouze v pěti případech. Model MobileNet s triplet loss funkcí a margin 0.6.

5.3.3 VGG16

Do série experimentů byla taktéž zahrnuta síť VGG16 s využitím Triplet loss funkce. I přes množství provedených experimentů se nepovedlo dosáhnout vyšší přesnosti jako u předchozích modelů. Konkrétně se podařilo dosáhnout mAP 22.36%. Nutno však podotknout, že kvůli paměťové náročnosti dané sítě bylo třeba pracovat s nižšími trénovacími dávkami. Efektivita sítě je například na klasifikačních úlohách nižší¹, a tak byla tato síť pro danou úlohu vyhodnocena jako nevhodná.

5.4 Neuronové sítě s využitím circle loss

Jako další loss funkce, která se jeví jako vhodná pro práci s graffiti tagy byla zvolena circle loss. Obdobně jako u triplet loss byly prováděny experimenty k zjištění optimálních parametrů pro nejefektivnější fungování sítě. Na základě předchozích experimentů bylo experimentováno pouze s MobileNet modelem neuronové sítě, jelikož vykazovala nejlepší výsledky.

Jako metrika pro měření podobnosti vektorů byla zvolena kosinová podobnost. Vycházím tak z článku [25], kde je využita.

Model stejně jako v předchozích případech vykazuje nejlepší výsledky do prvních 2000 iterací s velikostí trénovací dávky 72. Z následující série experimentů pak vyšlo najevo, že nejlepší učící koeficient je 0.0009. Efektivitu modelu při různých koeficientech je zobrazeno v tabulce 5.9. Již bez augmentace se podařilo dosáhnout velmi dobrých hodnot mAP 28.59% a top@10 rank 86.68%.

Učící koeficient	Metrika	Trénovací dávka	Augmentace	top@10 rank	mAP
0.0001	C	72	-	83.16%	23.47%
0.0002	C	72	-	86.68%	24.21%
0.0003	C	72	-	83.68%	25.05%
0.0004	C	72	-	84.21%	25.39%
0.0005	C	72	-	83.68%	26.61%
0.0006	C	72	-	85.79%	26.29%
0.0007	C	72	-	82.11%	25.51%
0.0008	C	72	-	81.05%	26.77%
0.0009	C	72	-	83.68%	28.59%
0.001	C	72	-	81.21%	26.71%
0.002	C	72	-	82.16%	25.02%

Tabulka 5.9: Výsledky experimentů pro zjištění nejvhodnějšího učícího koeficientu. *C* značí kosinovu podobnost.

Dalším hyperparametrem, který má zásadní vliv na efektivitu circle loss funkce je margin. Nejlépe si vedl margin nastaven na hodnotu 0.5 a 0.6, jak lze vidět v tabulce 5.10 níže. Tímto krokem se podařilo dosáhnout dalšího drobného zlepšení modelu a to na mAP 29.21% a top@10 rank 87.89%.

¹<https://github.com/tensorflow/models/tree/master/research/slim>

Margin	Metrika	Trénovací dávka	Augmentace	top@10 rank	mAP
0.1	C	72	-	86.32%	27.57%
0.15	C	72	-	83.16%	26.54%
0.20	C	72	-	83.68%	24.88%
0.25	C	72	-	86.84%	28.12%
0.35	C	72	-	84.72%	26.68%
0.5	C	72	-	87.89%	28.88%
0.6	C	72	-	86.32%	29.21%
0.7	C	72	-	80.53%	23.48%
0.8	C	72	-	73.68%	19.04%

Tabulka 5.10: Výsledky experimentů pro zjištění velikosti margin.

Na závěr byla využita augmentace dat stejně jako v předchozích případech. Experimenty byly provedeny pro margin 0.5 a 0.6 s učícím koeficientem 0.0009. Dle očekávání došlo k vylepšení přesnosti modelu a to na mAP 34.39%.

Margin	Metrika	Trénovací dávka	Augmentace	top@10 rank	mAP
0.6	C	72	O	84.21%	28.99%
0.6	C	72	O + P	87.89%	33.22%
0.6	C	72	S	85.79%	27.63%
0.6	C	72	J	90%	34.39%
0.6	C	72	O + S	85.79%	26.68%
0.6	C	72	O + S + P	87.89%	30.5%
0.6	C	72	O + S + P + J	85.79%	30.34%
0.5	C	72	O	87.37%	30.73%
0.5	C	72	O + P	85.26%	28.64%
0.5	C	72	S	84.72%	27.24%
0.5	C	72	J	87.89%	29.61%
0.5	C	72	O + S	87.37%	27.89%
0.5	C	72	O + S + P	84.72%	27.51%
0.5	C	72	O + S + P + J	85.26%	29.97%

Tabulka 5.11: Výsledky experimentů s využitím náhodné augmentace dat.

Následuje demonstrace na dvou ukázkových query, které byly využity v předchozích kapitolách. U tagu RMNK došlo opět ke 100% ve všech 10 případech, jak je možno vidět níže na obrázku 5.15. U ZOOT se podařilo nalézt shodné tagy v 6 případech, jak je patřičné z ukázky 5.16.

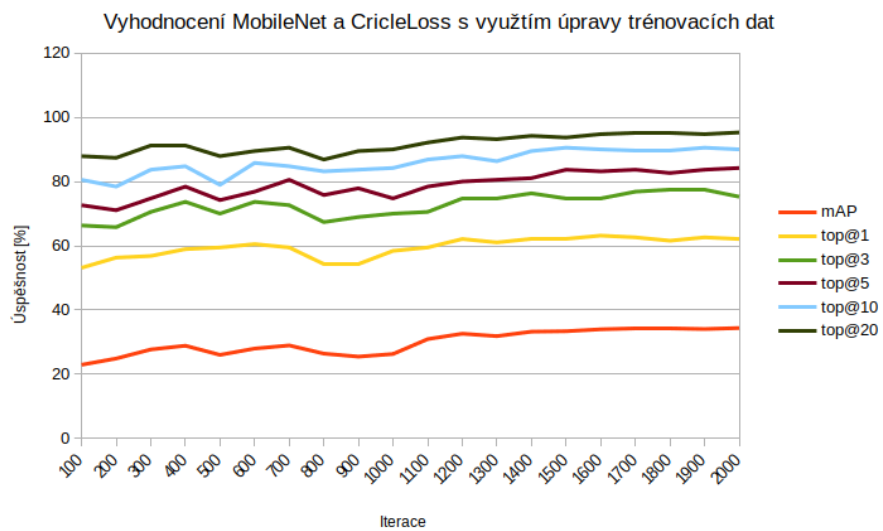
Kompletní vyhodnocení pro nejefektivnější model je možno vidět v grafu 5.17.



Obrázek 5.15: Ukázka prvních 10 vrácených obrázků z testovací sady graffiti tagu RMNK při využití MobileNet a circle loss. Shoda je ve všech 10 případech, jelikož tag RMNK není rozmanitý.



Obrázek 5.16: Ukázka prvních 10 vrácených obrázků z testovací sady graffiti tagu ZOOT při využití MobileNet a circle loss. Kvůli rozmanitosti jednotlivých vzorků je shoda v šesti případech.



Obrázek 5.17: Vyhodnocení efektivity modelu při využití MobileNet s circle loss a různých úprav datasetu.

5.5 Re-identifikace

Podíváme-li se na úlohu re-identifikace v oblasti počítačového vidění, tak je snaha o opětovnou identifikaci objektu. Příkladem může být obličej konkrétní osoby, člověk zachycen na různých kamerách nebo například automobil. V každém případě se jedná o stále stejnou entitu. Pokud bychom stejný princip aplikovali na graffiti tagy, tak je natrénovaný model velice úspěšný. Konkrétní ukázkou je možno vidět na obrázku 5.18, kdy všechny stejné graffiti tagy jsou navraceny jako první. Stejný tag je v tomto případě zachycen pod různými úhly.



Obrázek 5.18: Ukázka re-identifikace stejného graffiti tagu. V testovací sadě se nachází třikrát vyfocený stejný graffiti tag, který je akorát pod různými úhly. Všechny tři obrázky jsou navraceny na prvních třech pozicích pro daný query obrázek.

Je pochopitelné, že v takovém případě bude úroveň re-identifikace výborná, jelikož je tag focen vždy zepředu. Není třeba řešit například natočení v prostoru oproti re-identifikaci osob a podobné problémy. Takový přístup však není prakticky přínosný, proto je v této práci na re-identifikaci nahlíženo jako na tagy stejného nápisu.

5.6 Shrnutí

Jako nejlepší varianta pro re-identifikaci graffiti tagů vyšla neuronová síť MobileNet s využitím triplet loss funkce, pevného margin a Euklidovskou metrikou. V tabulce 5.12 níže jsou vypsány všechny měřené metriky pro nejefektivnější modely u kterých byly provedeny experimenty s úpravou datové sady.

Akčkoliv se dosažené přesnosti mohou jevit jako nízké je to primárně způsobeno povahou úlohy a rozmanitostí datasetu. Přesnost výrazně ovlivňují tagy, které jsou velice rozdílné v rámci svojí vlastní třídy. Ukázkou takového tagu je například SYT, jehož ukázky je možné vidět na obrázku 5.19. Všechny nasbírané tagy mají pouze pramalou podobnost, a proto se ani v jednom experimentu nepodařilo dohledat tag ve stejné třídě v rámci top@10 rank. S ohledem na velikost query sady, pak takové obrázky vcelku významně ovlivňují dosažené

Sít	Loss funkce	Metrika	mAP	R@1	R@3	R@5	R@10	R@20
ResNet	Triplet+SP	E	27.93	60.53	70.53	76.84	81.58	88.95
MobileNet	Triplet+SP	E	33.13	64.72	76.84	82.11	87.37	93.16
MobileNet	Triplet+M	E	36.02	60.73	74.35	82.2	86.91	93.19
MobileNet	Circle+M	K	34.39	63.16	77.37	84.21	90.53	95.26

Tabulka 5.12: Srovnání nejefektivnějších modelů s využitím optimálních trénovacích argumentů. SP reprezentuje SoftPlus aktivační funkci a M fixní margin.

přesnosti. Pokud by byl dataset vhodně profiltrován určitě by bylo možno dosáhnout daleko lepších výsledků.

V rámci trénování různých konfigurací bylo vyzkoušeno, že trénování modelu se občas stává náhodným, jelikož záleží na obrázcích, jež vstupují do trénovací dávky. Obdobně pak augmentace dat může být v některých případech spíše na škodu.

Ukázkové query pro vybrané modely, jež jsou ukázány v předchozích kapitolách, také ukazují, že pro stejný vstup bývá navraceno vícero různých obrázků v rámci stejné třídy. Evidentní je to u tagu ZOOT, kdy se napříč jednotlivými modely objevují různé tagy třídy ZOOT jako nejpodobnější.



Obrázek 5.19: Tag SYT, který nemá v rámci datasetu podobné tagy. Nejlepší tag je součástí query sady. Zbývající tagy pak náleží testovací sadě. Mezi tagy je pouze minimální podobnost.

Jako nejsnazší způsob vylepšení přesnosti tedy aktuálně vnímám rozšíření datové sady. Vezmu-li v potaz například dataset Market-1501 [30], který je využíván pro re-identifikaci osob, tak je téměř dvacetkrát obsáhlejší. V případě graffiti je však získání tak rozsáhlého datasetu takřka nemožné. Zmíněný dataset obsahuje 1501 tříd, zatímco mi se podařilo získat pouze 249 použitelných entit na poměrně rozsáhlém území města Brno. Ačkoliv se graffiti tagy vyskytují prakticky všude, tak velké množství jich má na svědomí úzká skupina lidí. Naopak nezanedbatelná část graffiti jsou pak často ojedinělé případy jednotlivce, a tak je problém nalézt podobné tagy. Tyto extrémy tedy tvorbu datasetu zásadně ztěžují.

Další možností je pak vhodné profiltrování datasetu. Graffiti tagy, které jsou velmi rozdílné v rámci jedné třídy, mohou způsobovat horší trénování sítě. Ačkoliv se odhadem jedná pouze o jednotky procent, kdy jsou graffiti velice rozdílné, tak může dojít k zhoršení efektivity sítě v důsledku toho, že se síť snaží natrénovat na rozpoznávání těchto případů.

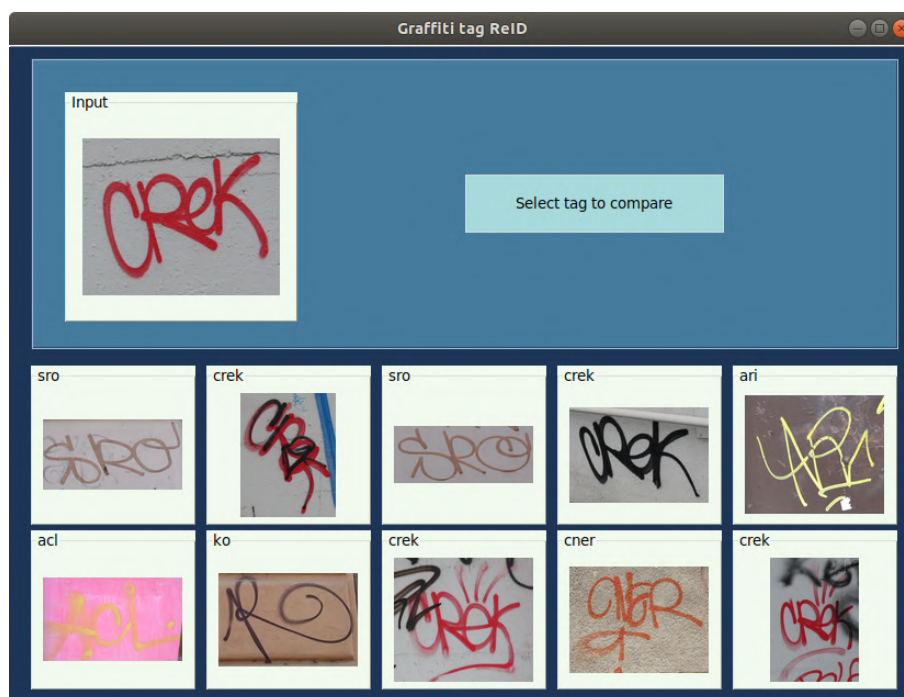
Minimálně u vhodně zvolené query sady by došlo k výraznému navýšení přesnosti v případě, kdy by dataset nebyl tvořen náhodným rozřazením na testovací a query sadu.

Kapitola 6

Implementace

6.1 Aplikace

V rámci práce bylo také naimplementováno jednoduché grafické rozhraní implementováno s využitím knihovny Tkinter, které umožňuje vyhledávání top 10 tagů v rámci testovací sady. Aplikace slouží k demonstrativním účelům funkčnosti natrénovaného modelu. Pro jednoduchost využívá předpočítaných vektorů uložených v souboru formátu *.h5*. Výslednou aplikaci lze vidět na obrázku 6.1.



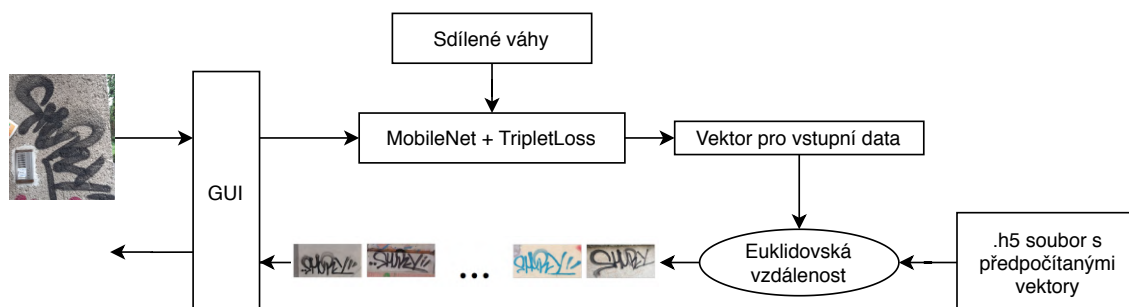
Obrázek 6.1: Ukázka GUI s prvními deseti vrácenými obrázky. Každý obrázek je nadepsán příslušnou třídou.

Pro implementaci grafického rozhraní byl využito aplikace Page¹, která slouží jako editor pro tvorbu grafického rozhraní pro jazyk Python. Samotný jazyk Python byl pak zvolen s ohledem na implementaci skriptů, jež pracují s neuronovými sítěmi a také přenositelnost

¹<http://page.sourceforge.net/>

mezi operačními systémy. Barevné schéma aplikace bylo vybráno na základě nejpoužívanějších barevných palet².

Princip fungování aplikace je pak triviální, jak lze vidět na schématu 6.2 níže.



Obrázek 6.2: Schéma aplikace pro vyhledání 10 nejpodobnějších graffiti tagů.

Jak již bylo řečeno, tak zobrazená aplikace slouží pouze k demonstrativnímu účelu pro snadnou ukázkou funkčnosti natrénovaného modelu. V kontextu systému, kde by byl zmíněný model využit, by byla zvolena implementace klientské části ve vhodnějším jazyce například JavaScript s využitím React knihovny. Ať už z důvodu rozšiřitelnosti nebo protože existují knihovny, jež nabízí požadované funkcionality uživatelského rozhraní, které jsou řádně odladěné s ohledem na UX.

²<https://colors.co/palettes/trending>

Kapitola 7

Závěr

Hlavním výsledkem této práce je jednoduchá grafická aplikace, která umožňuje vyhledávání podobných graffiti tagů s využitím přístupu re-identifikace. Jádrem aplikace je model využívající MobileNet neuronové sítě a triplet loss funkce, který dosahuje mAP 36.02%. Tento model byl vybrán jako nejefektivnější na základě experimentů, které byly prováděny s dalšími neuronovými sítěmi, různými metrikami pro porovnávání vzdáleností vektorů, loss funkcí a v neposlední řadě augmentací dat.

Dále v rámci práce vznikl dataset graffiti tagů, který obsahuje 2715 obrázků. Tyto obrázky jsou v původní kvalitě a jsou vhodné i pro další využití, kterým je například detekce graffiti tagů. Tento dataset byl dále upraven pro potřeby re-identifikace, kdy bylo získáno 1682 výřezů tagů vhodných k re-identifikaci. Tyto výřezy byly rozřazeny do příslušných tříd.

Dataset zároveň vnímám jako nejproblematictější část této práce. Povaha řešené úlohy a graffiti vyžadují větší množství dat a to jak pro trénování, tak pro vyhodnocení modelu. Nicméně tato data není možné získat statickými kamerami, jak je tomu často u obdobných re-identifikačních úloh. Dále pak rozmanitost graffiti tvorbu datasetu dělá obtížnější.

Práce částečně navazuje na mou bakalářskou práci a bylo k ní přistupováno, jako by se jednalo o součást systému, jež by umožňoval snažší správu zaznamenaných graffiti. Aplikace by sloužila pro snažší dohledání graffiti tagů vedoucích k jedné osobě. V ideálním případě by uživatel vybral obsahově stejný tag, což by vedlo ke snažšímu mapování aktivity konkrétního writera či skupiny.

Literatura

- [1] *Image Recognition and Object Detection : Part 1*. Satya Malick, 2016. Dostupné z: <http://www.learnopencv.com/image-recognition-and-object-detection-part1/>.
- [2] AHONEN, T., HADID, A. a PIETIKÄINEN, M. Face Recognition with Local Binary Patterns. In: *ECCV*. 2004.
- [3] BAY, H., TUYTELAARS, T. a VAN GOOL, L. Surf: Speeded up robust features. *Computer vision–ECCV 2006*. Springer. 2006, s. 404–417.
- [4] BOUMA, S., PAWLEY, M. D. M., HUPMAN, K. a GILMAN, A. Individual common dolphin identification via metric embedding learning. *CoRR*. 2019, abs/1901.03662. Dostupné z: <http://arxiv.org/abs/1901.03662>.
- [5] FATHY RASHAD, M. Siamese Net, Triplet Loss, and Circle Loss Explained. In: *Medium*. Dostupné z: <https://medium.com/vitrox-publication/understanding-circle-loss-bdaa576312f7>.
- [6] FISCHER, M. *Detekce graffiti tagů v obraze*. Brno, 2019. Bakalářská práce. Vysoké učení technické v Brně. Fakulta informačních technologií. Ústav počítačové grafiky a multimédií. Dostupné z: <http://hdl.handle.net/11012/180108>.
- [7] FREEMAN, W. T. a ROTH, M. *Orientation Histograms for Hand Gesture Recognition*. TR94-03. Cambridge, MA 02139: MERL - Mitsubishi Electric Research Laboratories, prosinec 1994. Dostupné z: <http://www.merl.com/publications/TR94-03/>.
- [8] GRÜNSEISEN, V. *Vyhledávání graffiti tagů podle podobnosti*. Brno, 2013. Diplomová práce. Vysoké učení technické v Brně. Fakulta informačních technologií. Ústav počítačové grafiky a multimédií. Dostupné z: <http://hdl.handle.net/11012/53556>.
- [9] HE, K., ZHANG, X., REN, S. a SUN, J. Deep Residual Learning for Image Recognition. *CoRR*. 2015, abs/1512.03385. Dostupné z: <http://arxiv.org/abs/1512.03385>.
- [10] HERMANS, A., BEYER, L. a LEIBE, B. *In Defense of the Triplet Loss for Person Re-Identification*. 2017.
- [11] HIJAZI, S., KUMAR, R. a ROWEN, C. *Using Convolutional Neural Networks for Image Recognition*. 2015.
- [12] KHOKHER, A. a TALWAR, R. Content-based Image Retrieval: Feature Extraction Techniques and Applications. In: Březen 2012.

- [13] KOLESNIKOV, A., BEYER, L., ZHAI, X., PUIGSERVER, J., YUNG, J. et al. *Large Scale Learning of General Visual Representations for Transfer*. 2019.
- [14] KUMAR, R., WEILL, E., AGHDASI, F. a SRIRAM, P. Vehicle Re-Identification: an Efficient Baseline Using Triplet Embedding. *CoRR*. 2019, abs/1901.01015. Dostupné z: <http://arxiv.org/abs/1901.01015>.
- [15] LECUN, Y., BENGIO, Y. et al. Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks*. 1995, roč. 3361, č. 10, s. 1995.
- [16] LOWE, D. G. Object Recognition from Local Scale-Invariant Features. In: *ICCV*. 1999, s. 1150–1157. Dostupné z: <http://dx.doi.org/10.1109/ICCV.1999.790410>.
- [17] MOINDROT, O. Triplet Loss and Online Triplet Mining in TensorFlow. In: *Olivier Moindrot blog*. 2018. Dostupné z: <https://omindrot.github.io/triplet-loss>.
- [18] NOVACK, T., VORBECK, L., LOREI, H. a ZIPF, A. Towards Detecting Building Facades with Graffiti Artwork Based on Street View Images. *ISPRS International Journal of Geo-Information*. Únor 2020, roč. 9, s. 98.
- [19] PAVLICA, J. *Detekce graffiti tagů v obraze*. 2016. Diplomová práce. Vysoké učení technické v Brně, Fakulta informačních technologií.
- [20] SANTOS, G. R., BALLESTER, P., BIRCK, M., CORRÊA, U., ANDERSSON, V. et al. Towards Graffiti Classification in Weakly Labeled Images Using Convolutional Neural Networks. In: Listopad 2017, s. 39–48. ISBN 978-3-319-71010-5.
- [21] SCHROFF, F., KALENICHENKO, D. a PHILBIN, J. FaceNet: A unified embedding for face recognition and clustering. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE. Jun 2015. Dostupné z: <http://dx.doi.org/10.1109/CVPR.2015.7298682>.
- [22] SCHWARZ, P. B. Recognition of Graffiti Tags. In: 2006.
- [23] SEMERÁK, V. *Vyhledávání graffiti tagů podle podobnosti*. Brno, 2013. Diplomová práce. Vysoké učení technické v Brně. Fakulta informačních technologií. Ústav počítačové grafiky a multimédií. Dostupné z: <http://hdl.handle.net/11012/53431>.
- [24] SRINIVAS, S., SARVADEVABHATLA, R. K., MOPURI, K. R., PRABHU, N., KRUTHIVENTI, S. S. S. et al. A Taxonomy of Deep Convolutional Neural Nets for Computer Vision. *Front. Robotics and AI*. 2016, roč. 2016. Dostupné z: <http://dx.doi.org/10.3389/frobt.2015.00036>.
- [25] SUN, Y., CHENG, C., ZHANG, Y., ZHANG, C., ZHENG, L. et al. Circle loss: A unified perspective of pair similarity optimization. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, s. 6398–6407.
- [26] TAQI, A. M., AL-AZZO, F., MARIOFANNA, M. a AL-SAAD, J. M. Classification and discrimination of focal and non-focal EEG signals based on deep neural network. In: *2017 International Conference on Current Research in Computer Science and Information Technology (ICCIT)*. 2017, s. 86–92.

- [27] VARIOR, R. R., HALOI, M. a WANG, G. Gated Siamese Convolutional Neural Network Architecture for Human Re-identification. In: LEIBE, B., MATAS, J., SEBE, N. a WELLING, M., ed. *Computer Vision – ECCV 2016*. Cham: Springer International Publishing, 2016, s. 791–808. ISBN 978-3-319-46484-8.
- [28] WANG, G., LAI, J., HUANG, P. a XIE, X. Spatial-Temporal Person Re-identification. *CoRR*. 2018, abs/1812.03282. Dostupné z: <http://arxiv.org/abs/1812.03282>.
- [29] ZEILER, M. D. a FERGUS, R. Visualizing and Understanding Convolutional Networks. In: FLEET, D. J., PAJDLA, T., SCHIELE, B. a TUYTELAARS, T., ed. *Computer Vision - ECCV 2014 - 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part I*. Springer, 2014, s. 818–833. Lecture Notes in Computer Science, sv. 8689. Dostupné z: http://dx.doi.org/10.1007/978-3-319-10590-1_53. ISBN 978-3-319-10589-5.
- [30] ZHENG, L., SHEN, L., TIAN, L., WANG, S., WANG, J. et al. Scalable Person Re-identification: A Benchmark. In: *2015 IEEE International Conference on Computer Vision (ICCV)*. 2015, s. 1116–1124.
- [31] ZHENG, W.-S., GONG, S. a XIANG, T. Transfer re-identification: From person to set-based verification. In: červen 2012, s. 2650–2657. ISBN 978-1-4673-1226-4.
- [32] ZHU, Z., JIANG, X., ZHENG, F., GUO, X., HUANG, F. et al. Aware Loss with Angular Regularization for Person Re-Identification. In: *AAAI*. 2020, s. 13114–13121.
- [33] ŠŤASTNÝ, M. *Graffiti a sprejersství*. 2010. Diplomová práce. Bankovní institut vysoká škola Praha, Katedra ekonomických a sociálních věd.

Příloha A

Ukázka testovací sady



Obrázek A.1: Ukázka všech RMNK tagů v testovací sadě.



Obrázek A.2: Ukázka všech ZOOT tagů v testovací sadě.