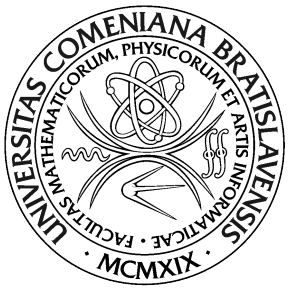
UNIVERZITA KOMENSKÉHO V BRATISLAVE
FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY



REIDENTIFIKÁCIA VOZIDIEL V SNÍMKACH Z DOPRAVNÝCH KAMIER

Diplomová práca

2021 Bc. Richard Dominik

UNIVERZITA KOMENSKÉHO V BRATISLAVE
FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY

REIDENTIFIKÁCIA VOZIDIEL V SNÍMKACH Z DOPRAVNÝCH KAMIER

Diplomová práca

Študijný program:	Aplikovaná informatika
Študijný odbor:	2511 Aplikovaná informatika
Školiace pracovisko:	Katedra aplikovanej informatiky
Školiteľ:	Ing. Viktor Kocur, PhD.

Bratislava, 2021

Bc. Richard Dominik

Univerzita Komenského v Bratislave
Fakulta matematiky, fyziky a informatiky

ZADANIE ZÁVEREČNEJ PRÁCE

Meno a priezvisko študenta: Bc. Richard Dominik
Študijný program: aplikovaná informatika (Jednoodborové štúdium, magisterský II. st., denná forma)
Študijný odbor: informatika
Typ záverečnej práce: diplomová
Jazyk záverečnej práce: slovenský
Sekundárny jazyk: anglický

Názov: Reidentifikácia vozidiel v snímkach z dopravných kamier
Re-identification of vehicles captured by traffic cameras

Anotácia: Inteligentný dopravný systém (IDS) je pokročilý systém integrujúci rôzne informačné technológie s cieľom poskytnúť nástroje pre efektívejšie, informovanejšie a bezpečnejšie využitie a návrh dopravných sietí. Dôležitou súčasťou IDS je zber dát. V kontexte cestnej dopravy je často vhodné zbierať dáta o pohybe vozidiel po rôznych cestách. Schopnosť reidentifikovať vozidlá v snímkach z rôznych dopravných kamier môže byť pri takomto zbere veľmi prospešná.

Cieľ: Cieľom práce je navrhnúť, implementovať a otestovať algoritmus založený na princípoch hlbokého učenia pre účely reidentifikácie vozidiel v snímkach z dopravných kamier. Súčasťou práce bude prehľad s moderných techník reidentifikácie obecne ako aj konkrétne v kontexte sledovania dopravy. Na vyhodnotenie budú využité verejne dostupné datasety a výsledky riešenia budú porovnané s existujúcimi prístupmi.

Vedúci: Ing. Viktor Kocur
Katedra: FMFI.KAI - Katedra aplikovanej informatiky
Vedúci katedry: prof. Ing. Igor Farkaš, Dr.

Dátum zadania: 07.10.2020

Dátum schválenia: 08.10.2020 prof. RNDr. Roman Ďurikovič, PhD.
garant študijného programu

.....
študent

.....
vedúci práce

Čestne prehlasujem, že túto diplomovú prácu som vypracoval samostatne len s použitím uvedenej literatúry a za pomoci konzultácií u môjho školiteľa.

. . . . . .

Bratislava, 2021 Bc. Richard Dominik

Poďakovanie

Poďakovanie

Abstrakt

Abstrakt

Kľúčové slová:

Abstract

Abstrakt EN

Keywords:

Obsah

Úvod

Úvod

Kapitola 1

Prehľad problematiky

V tejto kapitole sa budeme venovať prehľadu pojmov potrebných pre problematiku, ktorej sa táto práca venuje. Predstavíme si úlohu reidentifikácie vozidiel a jej využitie v praxi, následne si zadefinujeme dôležité pojmy spojené s neurónovými sietami v počítačovom videní a na záver tejto kapitoly si popíšeme metriky, pomocou ktorých sa vyhodnocuje úspešnosť neurónových sietí pri úlohe reidentifikácie vozidiel.

1.1 Reidentifikácia vozidiel

Vozidlá ako také sú v dnešnej dobe neoddeliteľnou súčastou našich životov a umožnujú nám dopraviť sa z našej východzej pozície do nami zvoleného cieľa. K úspešnému a pohodlnému dopraveniu ale potrebujeme aj kvalitnú a vhodne navrhnutú cestnú komunikáciu. Pri výstavbe alebo úprave cestných úsekov je informácia o dopravnom toku veľmi prínosná. Práve reidentifikácia vozidiel nám umožnuje získať informácie o dopravnom toku. Takéto riešenie môžeme následne využiť v inteligetných dopravných systémoch pre efektívnejšie navrhovanie dopravných sietí.

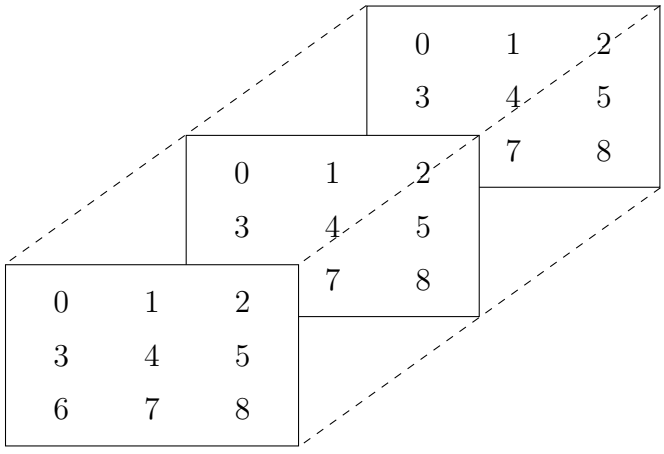
V článku [KU19] kolektív autorov uvádza, že výskumu ohľadom reidentifikácie vozidiel sa venovali rôzne publikácie už od roku 1990. Metódy publikované v tomto roku sa snažili problém reidentifikácie vozidiel riešiť pomocou metód založených na rôznych senzoroch (magnetické senzory, indukčné slučky...). V roku 2003 prišli metódy, ktoré sa problém snažili rešiť pomocou GPS (Globálny loakalizačný systém) a RFID (Vysokofrekvenčná identifikácia) a neskôr aj hybridné metódy, ktoré sa snažili kombinovať senzory a počítačové videnie. Spomínané metódy mali ale viaceré nedostatky. Medzi hlavné nedostatky patrila hlavne cena, náročnosť inštalácie, presnosť výsledkov bola závislá od rýchlosti vozidla atď. S príchodom roku 2012 sa začali objavovať prvé publikácie, ktoré sa tejto problematike začali venovať iba pomocou metód počítačového videnia. V dnešnej dobe sa pri riešení týchto úloh využíváju prístupy založené na metódach hlbokého učenia, čoho dôkazom je aj táto diplomová práca. Zdrojom dát pre metódy založené na počítačovom videní a hlbokom učení sú hlavne dopravné kamery z cestných komunikácii.

Presnejšie by sme úlohu reidentifikácie vozidiel v snímkach z dopravných kamier mohli zadefinovať ako nájdenie zhody rovnakého vozidla vo veľkom datasete obrázkov. Obrázky môžu byť nasnímané z rôznych kamier, orientácií, lokácií, ale aj času. Obrazové dáta môžu taktiež obsahovať oklúzie, šum a nezaostrenosti. Narozdiel od úlohy rozpoznávania vozidiel, ktorá sa venuje iba rozpoznaniu konkrétnej značky alebo modelu mvozidla je reidentifikácia väčšou výzvou, nakoľko 2 rôzne vozidlá si môžu byť vizuálne veľmi podobné. Môže ísť o rovnaký model, značku, alebo vozidlo, ktoré má rovnaký typ kolies, farbu, alebo iné časti vozidla (nárazníky, svetlá a podobne). Najaktuálnejšie prístupy sa venujú riešeniu tejto úlohy pomocou konvolučných neurónových

sietí a postupne sa začínajú objavovať aj riešenia pomocou vizuálnych transformerov. Mnohé existujúce riešenie profitujú z ponatkov nadobudnutých z odborných publikácii, ktoré sa venujú reidentifikácii osôb.

1.2 Konvolučné neurónové siete

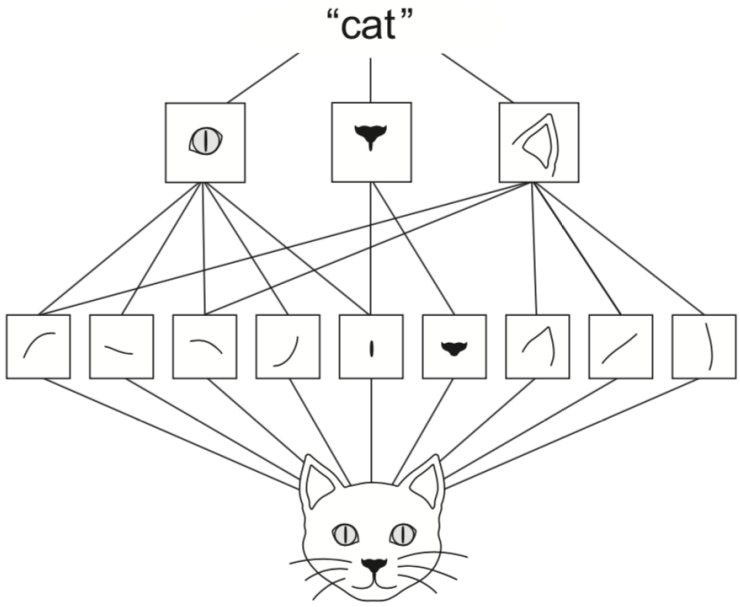
Pri riešení úloh spojených s počítačovím videním (detekcia objektov, určovanie veku osôb na obraze, rozpoznávanie tváre...) sa v posledných rokoch stali konvolučné neurónové siete (v literatúre označované aj ako ConvNets alebo CNN) veľmi populárne. Ide o typ doprednej neurónovej siete, ktorá v oblasti počítačového videnia pracuje s obrázkami alebo videom (postupnosť obrázkov). Takýto typ dát vieme veľmi jednoducho reprezentovať pomocou dátovej štruktúry tenzor. V prípade obrázkov ide o tenzor s rozmerom *šírka* $\times$ *výška* $\times$ *hĺbka*, kde *hĺbka* repzezentuje počet kanálov (v prípade RGB obrázkov je počet kanálov 3, naopak šedotónový obraz má kánal iba jeden). Na obrázku č. 1.1 môžeme vidieť grafickú reprezentáciu 3 rozmerného tenzora.



Obr. 1.1: Ukážka 3 rozmerného (3x3x3) tenzora

Pomocou konvolučných neurónových sietí vieme priradiť dátam význam na základe vzorov, ktoré vieme získať aplikovaním matematickej operácie

konvolúcia. Medzi vzory, ktoré môžeme získať patria na vyšších úrovniach rôzne typy hrán a čím ideme vo vrstvách siete hlbšie, tým vieme získavať komplexnejšie vzory (oči, ucho...) viď obrázok č. 1.3.

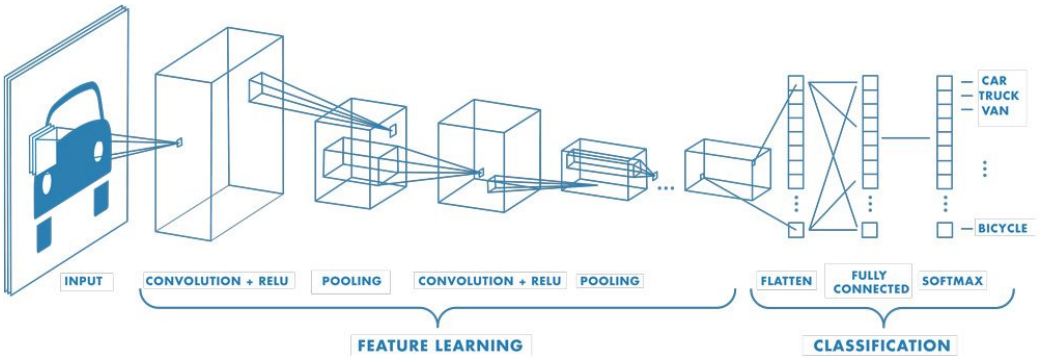


Obr. 1.2: Príklad detekcie vzorov v konvolučných neurónových sieťach [Cho17].

Jednou z hlavných výhod týchto sietí je ekvivariancia voči translácií vstupného obrazu. Pri vhodne zvolenom postupe trénovania sietí je možné napr. pomocou augmentácie sieť natrénovať tak, aby sa výstup siete nemenil aj pri škálovaní a rotácii vstupného obrazu. Ďalšie využitie konvolučných neurónových sietí vieme nájsť aj v spracovaní zvuku alebo v spracovaní prirodzeného jazyka.

Súčasťou architektúr, ktoré využúvajú konvolučné neurónové siete sú viaceré skryté konvolučné vstvy, ktoré sa kombinujú s vrstami, ktoré nazývame pooling vrstvy, aktivačné vrstvy a plne prepojené vrstvy. Príklad architek-

túry s viacerými vrstami môžeme vidieť na obrázku č. 1.3.



Obr. 1.3: Príklad neurónovej siete s viacerými vrstvami [mat].

1.2.1 Konvolučná vrstva

Ako už názov napovedá, táto vrstva pracuje s matematickou operáciou, ktorú nazývame konvolúcia. Podľa [GBC16] je konvolúcia operácia, ktorá pracuje s dvomi funkciami f a g a je definovaná nasledovne:

$$(f * g)(t) = \int_{-\infty}^{\infty} f(x)g(t-x)dx \tag{1.1}$$

V našom prípade si ešte zadefinujeme disktétnu časť konvolúcie, nakoľko sieť trénujeme na počítačoch, a tie ako vieme pracujú s dikrétnymi hodnotami. Konvolúcia je v diksrétnej verzii definovaná pre f a g nasledovne:

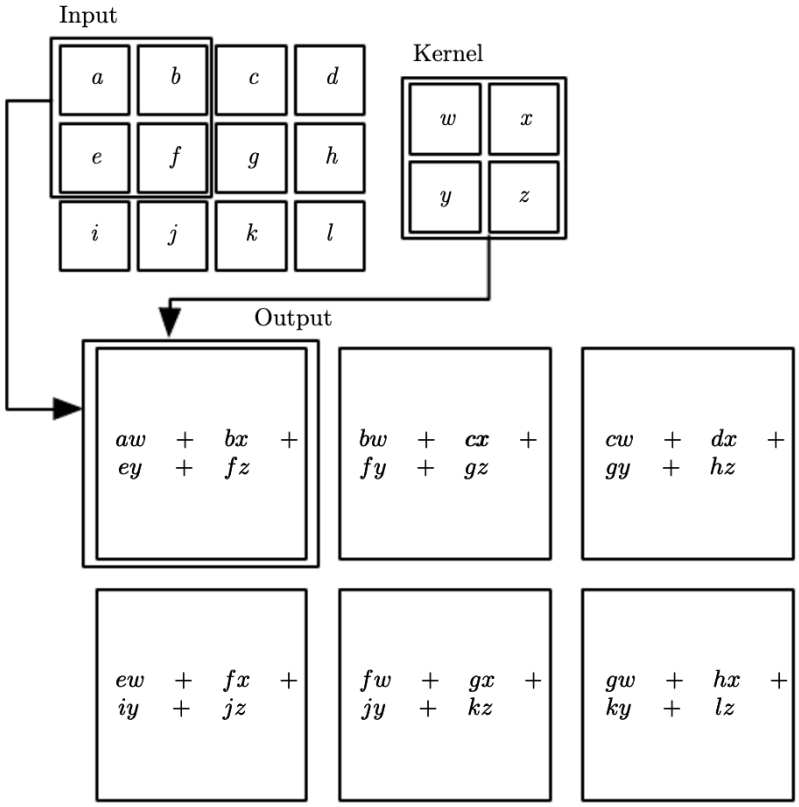
$$(f * g)(t) = \sum_{x=-\infty}^{\infty} f(x)g(t-x) \tag{1.2}$$

Ak pracujeme s obrazom, tak za funkciu f považujeme vstupný obraz a za funkciu g jadro (taktiež aj kernel). V tomto prípade hovoríme o dvojroz-

mernej konvolúcii, ktorá je definovaná predpisom:

$$(f * g)(i, j) = \sum_m \sum_n f(m, n)g(i - m, j - n) \tag{1.3}$$

Jadro si v našom prípade volíme samostatne a výstupom 2 rozmernej konvolúcie je takzvaná mapa príznakov. Konvolučná vrstva znižuje šírku a výšku obrazu oproti vstupu. Grafickú reprezentáciu 2 rozmernej konvolúcie môžeme vidieť na obrázku č. 1.4



Obr. 1.4: Príklad 2 rozmernej konvolúcie [GBC16]. Ako môžeme vidieť, na vstupe sme získali maticu o veľkosti 3x4 a aplikovaním jadra 2x2 sme získali výstupnu mapu príznakov s rozmerom 3x3.

Príklad jadra môže byť Sobelov hranový dektor, ktorý je definovaný pre

hľadanie hrán v horizontálnom smere nasledovne:

$$S_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

a pre vertikálny smer je Sobelov hranový detektor definovaný ako:

$$S_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

Výstup konvolučnej vrstvy (mapa príznakov) je následne vstupom pre ďalšiu vrstvu a to konkrétne aktivačnú vrstvu.

1.2.2 Aktivačná vrstva

Úlohou aktivačnej vrsty je rozhodovanie, ktoré neuróny budú počas priebehu trénovania aktivované a ktoré nie. Týmto prístupom pridávame na mapu príznakov z konvolučnej vrstvy nelinearitu za pomoci vhodnej aktivačnej funkcie. Pri konvolučných neurónových sieťach je najčastejšie ako aktivačná funkcia používana ReLU (Rectified linear unit), ktorej definíciu môžeme vidieť na rovnici (1.4)

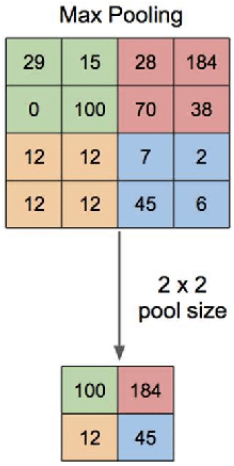
$$f(x) = max(0, x) \tag{1.4}$$

Táto funkcia nemení veľkosť vstupu a medzi jej výhody patrí napríklad efektívnosť jej výpočtu. Ak by sme vynechali túto vrstvu (pridanie aktivačnej funkcie), tak by sa naša sieť v priebehu trénovania nedokázala zlepšovať a stal by sa z nej iba lineárny klasifikátor. Výstup z aktivačnej vrstvy je následne vstupom pre pooling vrstvu.

1.2.3 Pooling vrstva

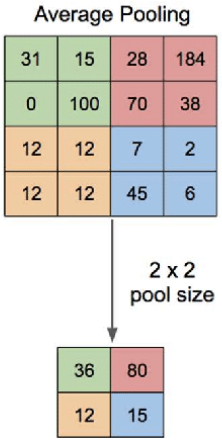
Problémom mapy príznakov z predchádzajúcich vrstiev je citlivosť na pozíciu príznakov v obraze. To znamená, že zmena veľkosti, otočenie alebo posun môžu výrazne zmeniť výslednu mapu príznakov. O riešenie tohto problému sa stará pooling vrstva a to tak, že zmenšuje mapu príznakov. Týmto prístupom zaručíme, že ďalšie vrstvy sa budú pozerať už iba na najvýznamnejšie príznaky a tie menej dôležité (akými sú napríklad veľmi podrobné detaily), sa znížením rozmerov odstránia. Ďalšími výhodami tejto vrstvy sú zníženie počtu trénovacích parametrov, čo znižuje počet výpočtov a teda aj času potrebného pre trénovanie siete a predchádza javu, ktorý nazývame overfitting (sieť je až veľmi dobré natrénovaná na trenovacích dátach). Najznámejšie funkcie využívané pre operáciu poolingu sú max pooling a priemerovací pooling.

Princípom max poolingu je výber maximálnej hodnoty podľa zvoleného rozmeru filtra. Zvyčajne sa volí veľkosť filtra *2×2*. Princíp max poolingu môžeme vidieť na obrázku č. 1.5.



Obr. 1.5: Príklad operácie max pooling. Prevzaté z [YSS19].

Princípom priemerovacieho poolingu je výber priemeru hodnôt podľa zvoleného rozmeru filtra. Zvyčajne sa volí veľkosť filtra *2×2*. Princíp average poolingu môžeme vidieť na obrázku č. 1.6.



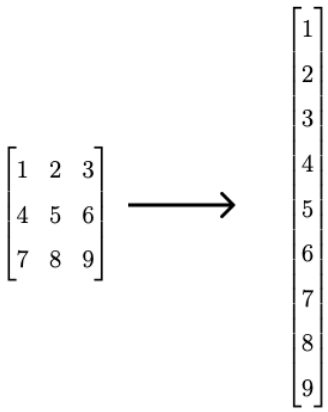
Obr. 1.6: Príklad operácie priemerovacieho poolingu. Prevzaté z [YSS19].

Po poslednej pooling vrstve väčšinou nasleduje plne prepojená vrstva. Vstupom pre plne prepojenú vrstvu nie je ale n rozmerný vektor, ale iba 1 rozmerný vektor. Z tohto dôvodou potrebujeme na výstup poslednej pooling vrstvy aplikovať jednu z nasledujúcich operácií a to buď operáciu flatten alebo globálny pooling.

Princípom operácie flatten je vytvoriť 1 rozmerný vektor pomocou sploštenia viacrozmerného vstupného tenzora. Výstupom operácie flatten bude teda 1 rozmerný vektor s počtom prvkov podľa vzorca (1.5) kde x počet prvkov výstupného 1 rozmerného vektora, n je počet dimenzií a S_i je veľkosť dimenzii i.

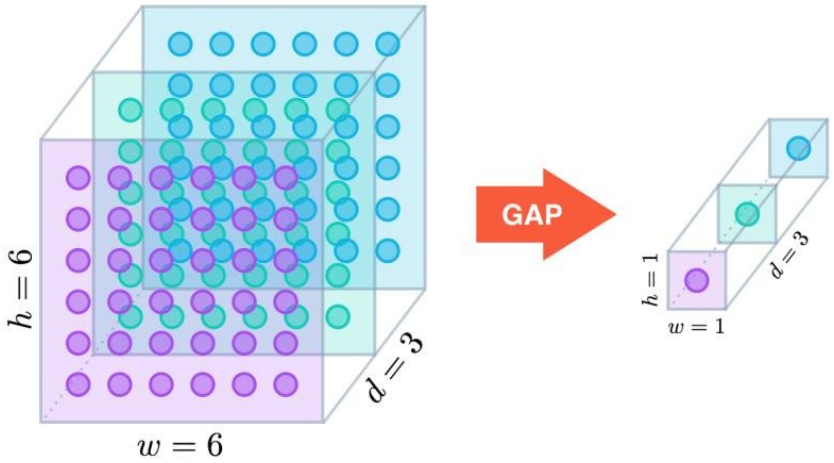
$$x = \prod_{i=0}^{n} S_i \tag{1.5}$$

Ilustráciou operácie flatten môžeme vidieť na obrázku č. 1.7.



Obr. 1.7: Príklad operácie flatten.

V praxi sa ale odporúča využívať namiesto operácie flatten globálny pooling. Princípom globálneho poolingu je aplikovať pooling na každú dimenziu viacrozmerného vstupného tenzora. Výsledný 1 rozmerný vektor bude mať teda taký počet prvkov, koľko rozmerný bol vstupný tenzor. Ilustráciu globálneho priemerovacieho poolingu (GAP) môžeme vidieť na obrázku č. 1.8.



Obr. 1.8: Príklad globálneho priemerovacieho poolingu. Prevzaté z [SY19]

Po aplikácii operácie flatten alebo globálneho priemerovacieho poolingu po poslednej pooling vrstve je 1 rozmerný vektor vstupom pre plne prepojenú vrstvu.

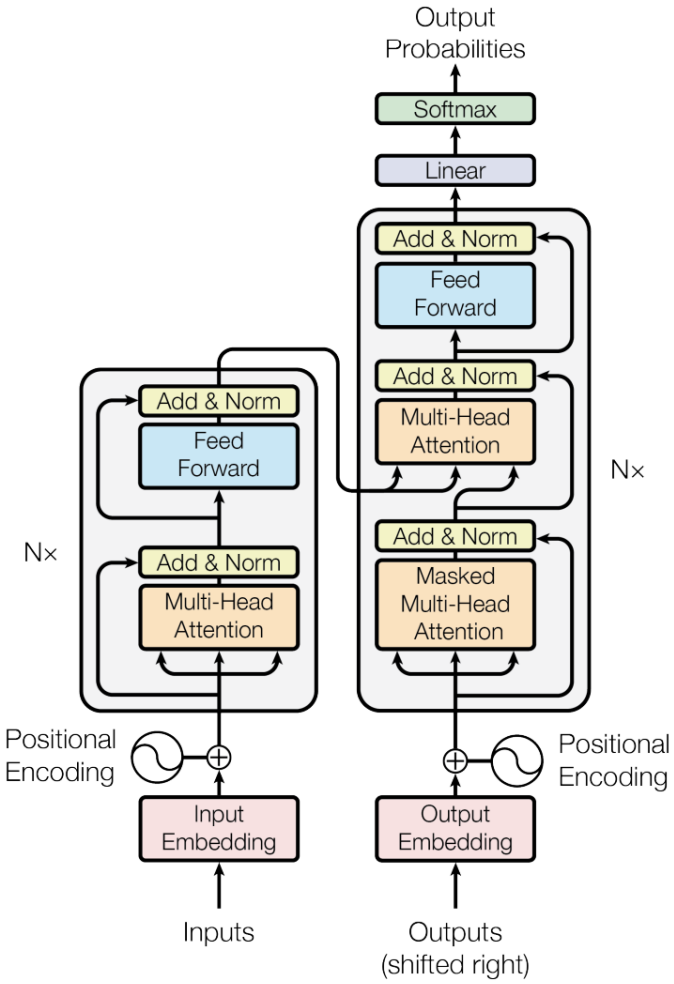
1.2.4 Plne prepojená vrstva

Poslednou fázou v konvolučných neurónových sieťach väčšinou býva plne prepojená vrstva. Ako už názov tejto vrstva napovedá, všetky neuróny tejto vrstvy sú navzájom plne prepojené ako tomu je aj v klasických neurónových sieťach. Úlohou tejto vrstvy je na základe príznakov naučených z predchádzajúcich vrstiev zaradiť objekt, ktorý bol na vstupnom obraze do správnej výstupnej triedy.

1.3 Transformery

Transformer je názov pre architektúru neurónovej siete, ktorá v posledných rokoch nabrala na popularite pri riešeniach problémov v doméne spracovania prírodzeného jazyka. V doméne spracovania prirodzéneho jazyka sa objavovali častokrát aj riešenia založené na konvolučných neurónych sieťach a tak sa výskumíci začali zaoberať aj snahou využiť transformery v doméne úloh pre počítačové videnie ako náhradu za konvolučné neurónové siete. Pojem transformer bol prvý krát predstavený v publikácii z roku 2017 [VSP+17], ktorej autori sú hlavne výskumníci zo spoločnosti Google. Autori (Waswani a kol.) sa pri návrhu Transformera zamerali na riešenie úloh v doméne prirodzeného spracovania jazyka. Konkrétne sa venovali prekladu textov z anglického jazyka do nemeckého jazyka ako aj prekladom z anglického jazyka do francúzskeho jazyka. V čase uvedenia publikácie táto architektúra dosiahla lepšie výsledky ako predošlé existujúce metódy, podporovala paralelizáciu a vyža-

dovala si výrazne menej času na tréning. Transformer pracuje na vstupe so sekvenciou dát. V prípade úlohy prekladanie viet ide o vektor s číselnými hodnotami slov v danej vete. Architektúru transformera môžeme vidieť na obrázku č. 1.9.



Obr. 1.9: Architektúra transformera [VSP+17].

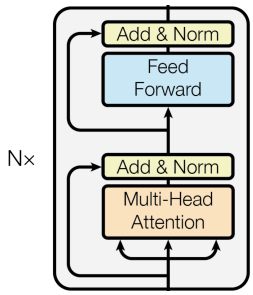
Ako môžeme z obrázku č. 1.9 vidieť, architektúra transfomera dostane na vstup sekvenciu dát, následne sa tieto dáta spracujú pomocou postupnosti enkóderov a dekóderov (ktoré používajú techniku attention) a na záver

nasleduje za dekóderom lineárna vstva (plne prepojená sieť) za ktorou nasleduje vsrtva softmax. Následne na výstupe dostaneme znova sekvenciu dát (v tomto prípade preložený vstupný text).

Zadefinujme si jednotlivé komponenty dôležité pre pochopenie fungovania tranformera, ako aj ich jednotlivé súčasti:

1.3.1 Enkóder

Podľa [VSP+17] si enkóder môžeme predstaviť ako komponent, ktorý získava na vstupe postupnosť $x = (x_1, x_2, ..., x_n)$ a na výstupe produkuje postupnosť $z = (z_1, z_2, ..., z_n)$. Enkóder sa v prípade tranformera skladá z troch hlavných vrstiev a to Multi-head attention vrstva, normalizačná vrstva a plne prepojená vrstva. Medzi vstupom a normalizačnou vrstvou a následne aj medzi výstupom prvej normalizačnej vrstvy a druhou normalizačnou vrstvou sa nachádzajú reziduálne prepojenie ako to môžeme vidieť znázornené na obrázku č. 1.10.



Obr. 1.10: Vizualizácia enkódera [VSP+17].

Multi-head attention

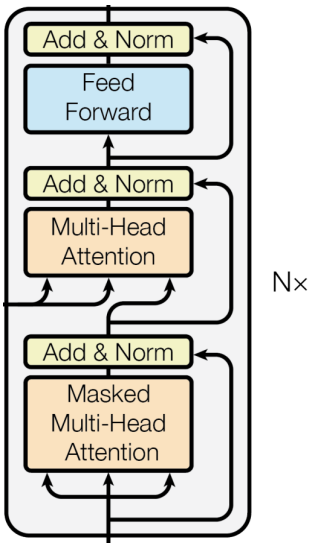
Úlohou Multi-head attention vrstvy je nájsť závislosti mezdi jednotlivými prvkami vstupnej postupnosti. V prípade prekladu ide o nájdenie závisloti

medzi každým slovom vo vstupom texte.

V publikácii [VSP+17] sa komponent enkóderov skladal z postupnosti 6tich identických enkóderov, kde výstup jedného enkódera bol vstupom nasledújuceho enkódera. Výstup posledného enkódera bol naslédne vstupom pre komponent dekóderov.

1.3.2 Dekóder

Podľa [VSP+17] si dekóder môžeme predstaviť ako komponent, ktorý získava na vstupe postupnosť $z = (z_1, z_2, ..., z_n)$ vygenerovanú enkóderom a na základe tejto postuponosti vygeneruje požadovanú výstupnú postupnosť $y = (y_1, y_2, ..., y_n)$ (napríklad vektor čísel, ktoré reprezentujú preložený text do požadovaného jazyka). Dekóder sa v prípade transformera skladá z rovnakých častí ako enkóder, ale má navyše pridanú ešte jednu Multi-head attention vrstvu. Vizualizáciu dekódera môžeme vidieť na obrázku č. 1.11.



Obr. 1.11: Vizualizácia dekódera [VSP+17].

Rovnako ako v prípade enkóderov sa aj komponent dekóderov skladá z postupnosti 6tich identických dekóderov. Výstupom dekódera je vektor desatinných čísel.

1.3.3 Lineárna a Softmax vrstva

Ako posledné 2 vrstvy v architektúre transfomera sa nachádzajú lineárna a softmax vsrtva. Linearná vrstva je v tomto prípade plne plepojená sieť, ktorej úlohou je projekcia výstupného vektora desatinných čisel z komponentov enkódera do množiny slov, ktoré naša sieť pozná (slová z jazyka). Následne softmax vrstva prevedie tento vektor do formy pravdepodobností. Následne na základe týchto pravdepodobností sú priradené výsledné slová výstupného textu.

Narozdiel od predstaveného príkladu s prekladom textu sa v doméne počítačového videnia ako vstup pre tranformer nevyužívajú 1 rozmerné vektory ale 2 rozmerné vektory. Tie vieme získať rozdelením vstupného obrazu na viacerené podčasti rovnakého rozmeru. V dobe písania tejto práce sú najvýznamejšie transformery pre prácu s obrazom Vision Transformer (ViT) [DBK+21] a Shifted window Tranformer (Swin tranformer) [LLC+21] , ktoré si bližšie predstavíme v kapitole 3.

Kapitola 2

Analýza datasetov

V tejto kapitole si predstavíme sériu datasetov, ktoré sa aktuálne využívajú v počítačovom videní pri úlohe reidentifikácie vozidiel. Popíšeme si ako vybrané datasety vznikli, aké dáta obsahujú a vzájomne ich porovnáme. Niektoré z vybraných datasetov sú aj voľne prístupné, iné si však vyžadujú individuálne vyžiadanie a podpísanie súhlasu s podmienkami, ktoré si určili autori. Medzi podmienky môže patriť súhlas s využitím dát iba na nekomerčné účely a zákaz šírenia dát mimo organizácie pre ktorú je žiadosť schválená.

2.1 VeRi-776

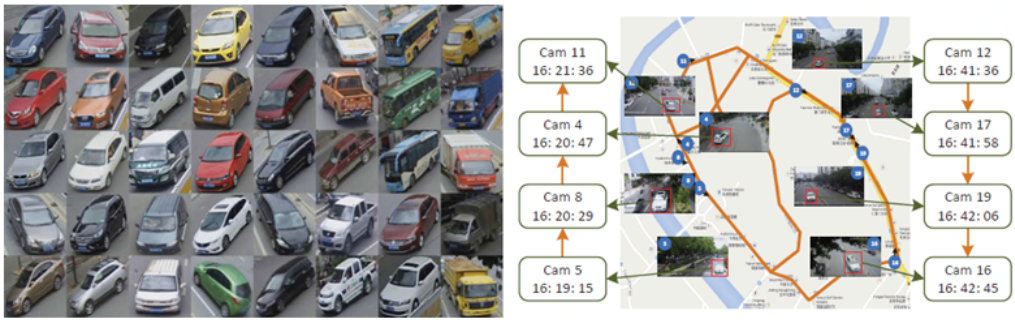
VeRi-776 [LLMM18] je dataset, ktorý bol prvý krát uvedený v [LLMM16]. Dataset obsahuje:

1. viac ako 50 000 obrázkov
2. 776 rôznych vozidiel
3. dáta nasnímané v meste pomocou 20tich kamier

Každé vozidlo v tomto datasete je nasnímané 2 $\sim$ 18 kamerami z rôznych uhlov, svetelných podmienok, oklúzií a v rôznych rozlíšeniach. Vozidlá disponujú aj nasledujúcimi atribútmi:

1. farba
2. typ
3. značka
4. bounding box

VeRi-776 nie je voľne dostupný. Je určený iba pre výskumné účely a pre prácu s ním je potrebné podať elektronickú žiadosť autorom datasetu. Ukážku obrázkov z tohto datasetu môžeme vidieť na obrázku 2.1



Obr. 2.1: Ukážka dát z datasetu VeRi 776.

2.2 Stanford Cars

Stanford Cars je voľne dostupný dataset poskytovaný Stanfordskou univerzitou. Prvý krát bol uvedený v [KSDFF13].

Dataset obsahuje:

1. 16 185 obrázkov

2. 8144 trénovacích a 8041 testovacích obrázkov

3. 196 rôznych vozidiel

Dáta sú nasnímané v meste pomocou 20tich kamier. Stanford Cars je síce voľne dostupný, ale rovnako slúži iba na výskumné účely. Tento dataset ale neslúži na aplikácie pri sledovaní dopravy. Ukážku dát z tohto datasetu môžeme vidieť na obrázku č. 2.2

Obr. 2.2: Ukážka dát z datasetu Stanford Cars.

2.3 AI City Challenge dataset

AI City Challenge dataset [NWA+21] je dataset, ktorý je súčasťou výzvy AI City Challenge . Je rozdelený na viacero stôp z ktorých je pre reidentifikáciu určená stopa 2 s názvom City-Scale Multi-Camera Vehicle Re-Identification. Tento dataset obsahuje:

1. 85 058 obrázkov

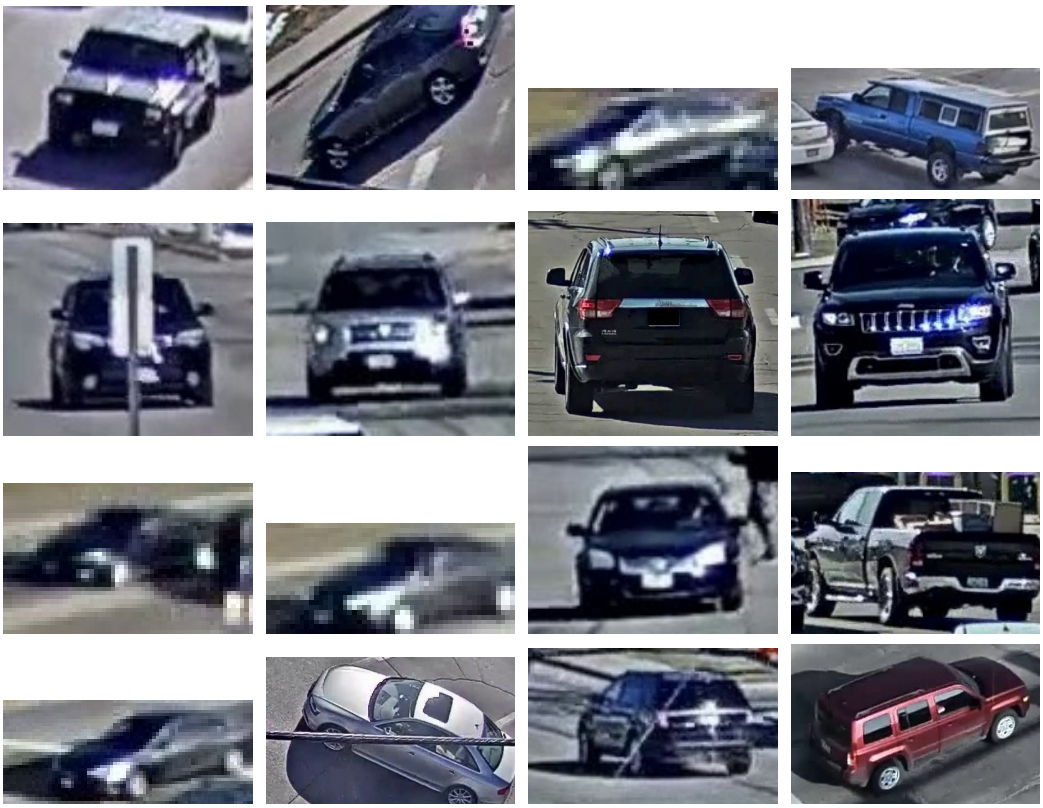
2. 52 717 trénovacích a 31 238 testovacích obrázkov

3. 440 rôznych vozidiel

4. 1103 obrázkov pre dopytovanie

5. syntetické dáta

Dáta sú nasnímane z rôznych dopravných kamier zo štátu Iowa v Spojených štátoch amerických. Obrázky boli anotované ľudmi a poskytujú atribúty ako typ, farba vozidla a vzťahy medzi ostanými vozidlami na obrázku. Ukážku dát z tohto datasetu môžeme vidieť na obrázku 2.3



Obr. 2.3: Ukážka dát z datasetu AI City Challenge.

2.4 Porovnanie

TODO: Porovnanie

Kapitola 3

Súvisiace práce

V tejto kapitole si predstavíme súvisíace práce, ktoré sa zaoberajú významými architektúrami konvolučných neurónových sietí, architektúrami transformerov, ktoré pracúju s obrazom a následne publikáciami z oblasti reidentifikáciou vozidiel a jednu prácu, ktorá sa zaoberá reidentifikáciou osôb. Dôvodom výberu práce zaoberajúcej sa reidentifikácii ôsob, je dôvod, ktorý sme už spomínali v predchádzajúcej kapitole, a to, že poznatky nadobudnuté z publikácií, ktoré sa venujú reidentifikácia ôsob priniesli významné poznatky aj pre tému reidentifikácie vozidiel. Popíšeme rôzne prístupy autorov vybraných prác a výsledky ktoré dosiahli.

3.1 Architektúry konvolučných neurónových sietí

V predchádzajucej časti práce sme si predstavili základné princípy fungovania konvolučných neurónových sietí. V tejto časti práce sa preto zameráme na uvedenie architektúr hlbokých neurónových sietí, ktoré v čase ich predstavenia dosahovali najlepšie výsledky a využívajú sa ako základ pri trénovaní modelov pre úlohu reidentifikácie vozidiel.

3.1.1 VGG

VGG [SZ15] je skratka pre Visual Geometry Group.

3.1.2 ResNet

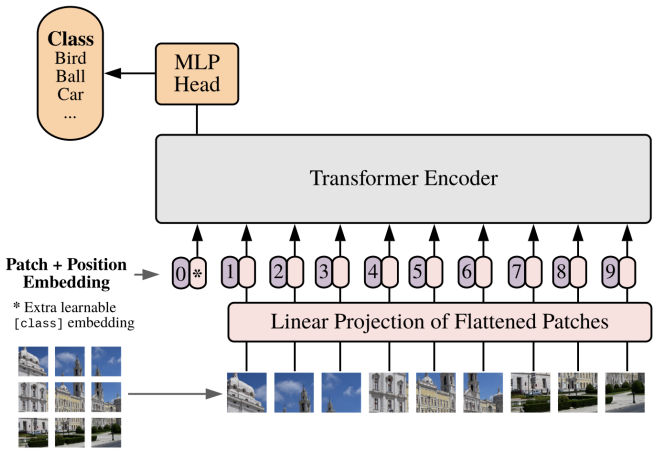
ResNet [HZRS15] je skratka pre Residual Network.

3.2 Architektúry, ktoré využívajú transformery

V tejto podkapitole si predstavíme architektúry založené na transformeroch.

3.2.1 ViT

ViT [DBK+21] je skratka pre Visual Tranformer a jeho architektúru môžeme vidieť na obrázku č. 3.1.

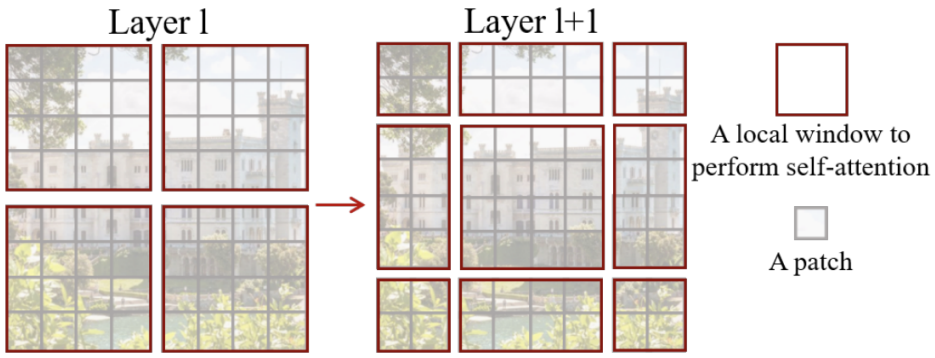


Obr. 3.1: ViT [DBK+21].

3.2.2 Swin Transformer

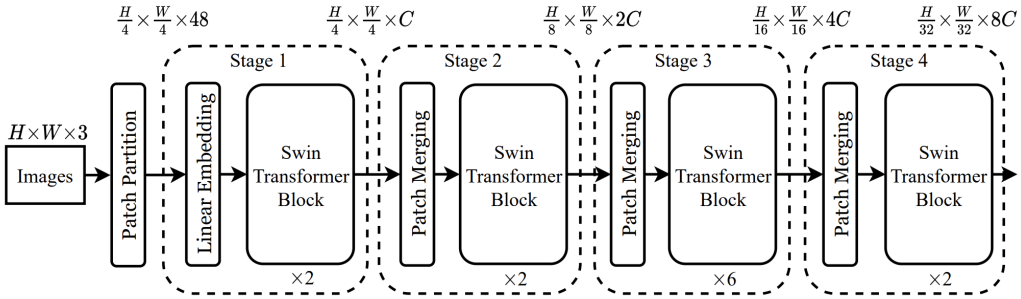
Swin Transformer [LLC+21] je skratka pre Shifted window transformer.

Príklad prístupu shifted window môžeme vidieť na obrázku č. 3.3:



Obr. 3.2: Príklad prístupu shifted window [LLC+21].

Architektúru Swin transformer môžeme vidieť na obrázku č. 3.3:



Obr. 3.3: Architektúra Swin transformera [LLC+21].

3.3 Bag of Tricks and A Strong Baseline for Deep Person Re-identification

V publikácii s názvom Bag of Tricks and A Strong Baseline for Deep Person Re-identification [LGL+19] sa kolektív autorov zaoberal tvorbou baseline

frameworku pre úlohu reidentifikácie ôsob. Venujú sa aj popísaniu a vyhodnoteniu rôznych trikov používaných v aktuálnych prístupov pre riešenie tejto problematiky. V tejto publikácií používali datasety s názvami Market1501 a DukeMTMC-reID. Ako základ pri trénovaní bol použitý ResNet 50 s predtrénovanými parametrami na ImageNet-e a pozmenenou dimenziou plne prepojenej vrsty na N, kde N je počet identít v trénovacích dátach. Ako ďaľšie základné kroky pri trénovaní takýchto modelov použili zmenu rozmerov obrázkov na rozmer 256×128, ktorým následnej pridali nulový padding 10 pixelov a takto vytvorený obrázok ešte náhodne orezali na 256×128 obdlžníkový obrázok. Ďalej bol každý obrázok otočený v horizontálnom smere s pravdepodobnosťou 0.5 a použili aj mnoho iných základných prístupov, ktorým sa už nebude podrobnejšie venovať.

3.3.1 Triky využité pri trénovaní

V tejto časti práce si popíšeme 6 trikov, ktoré podľa autorov tejto publikácie výrazne prispeli k vylepšeniu výsledkov ich baseline frameworku pre reidentifikáciu osôb.

Warmup Learning Rate

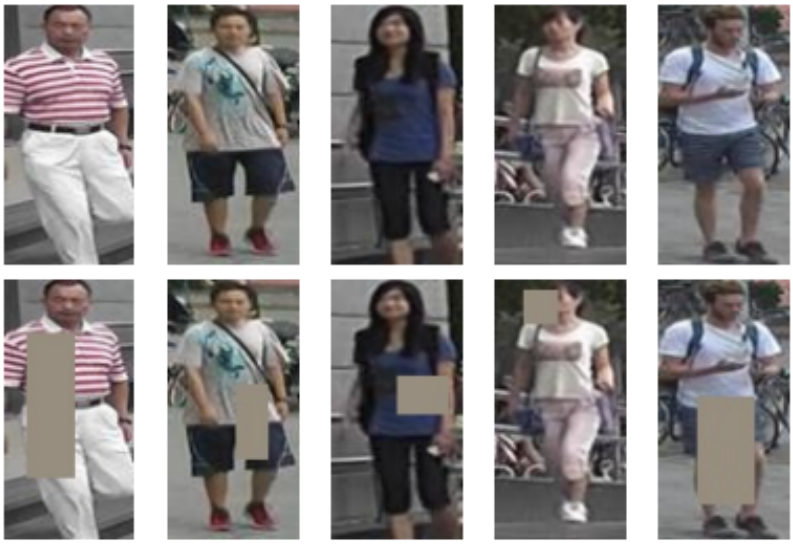
Narozdiel od tradičných baseline prístupov, ktoré sa trénuju konštantnou rýchlosťou učenia použili autori Warmup Learning Rate, ktorý vylepšuje výkon celého re-id modelu. Hodnotu veľkosti kroku učenia môžeme vidieť na

vzorci (3.1) kde lr(t) označuje learning rate počas epochy t.

$$lr(t) = \begin{cases} 3.5 \times 10^{-5} \times \frac{t}{10} & \text{ak } t < 10 \\ 3.5 \times 10^{-4} & \text{ak } 10 < t <= 40 \\ 3.5 \times 10^{-5} & \text{ak } 40 < t <= 70 \\ 3.5 \times 10^{-6} & \text{ak } 70 < t <= 120 \end{cases} \tag{3.1}$$

Random Erasing Augmentation

Random Erasing Augmentation (skrátene REA) je prístup, ktorý rieši problém rôznych oklúzií objektov. REA fugnuje na princípe, že si vyberie náhodný štvorcový región v obraze a vymaže jeho pixely náhodnými hodnotami. Príklad REA môžeme vidieť na obrázku 3.4



Obr. 3.4: Príklad REA, prvý riadok sú pôvodné obrázky a druhý riadok sú obrázky po aplikácii REA [LGL+19].

Label Smoothing

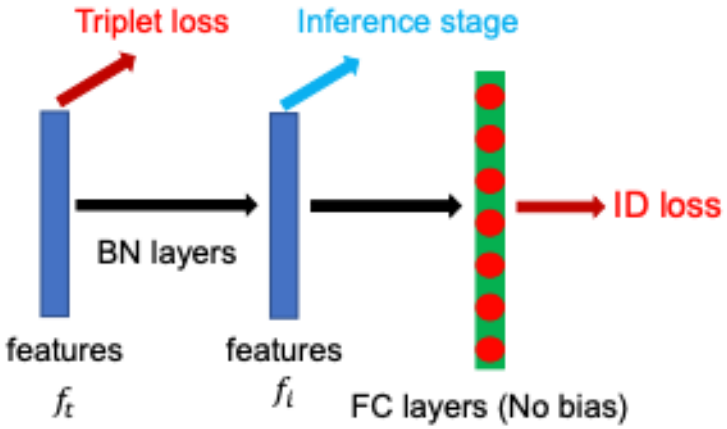
TODO: Label Smoothing je trik, ktorý sa veľmi často využíva pri klasifikačných úloh aby sme predišli javu overfitting (preučenie).

Last Stride

Last Stride je trik, ktorý v poslednej vrstve ResNet-u, ktorý je použitý ako základ trénovania, zmenia poslednú vrstvu, (stride o veľkosti 2) na stride o veľkosti 1, čím autori získali na výstupe väčšiu mapu príznakov (16×8) čo v konečnom dôsledku zmení iba nepatrne dobu výpočtu, nezmení počet trénovacích parametrov, ale zaručí výrazne vylepšenie výsledkov.

BNNeck

TODO: BNNeck prídava oproti štandardnému prístupu vrstvu normalizácie dávky po získaní príznakov. BNNeck môžeme vidieť na obrázku 3.5



Obr. 3.5: BNNeck [LGL+19].

Center Loss

TODO: Definíciu Center Loss môžeme vidieť na rovnici (3.2)

$$\mathcal{L}_{\mathcal{C}} = \frac{1}{2}\sum_{j=0}^{B}\| f_{tj} - c_{yj}\|_2^2 \tag{3.2}$$

Trénovanie prebehlo pomocou 120 epôch a vďaka popísaným prístupom dosiahol tento baseline framework 94.5% rank-1 a 85.9% mAP na datasete Market1501 a 86.4% rank-1 a 76.4% mAP na datasete DukeMTMC-reID.

3.4 The Devil is in the Details: Self-Supervised Attention for Vehicle Re-Identification

V publikácii The Devil is in the Details: Self-Supervised Attention for Vehicle Re-Identification [KPcCC20] sa zaoberali reidentifikáciou vozidiel.

3.5 TransReID: Transformer-based Object Re-Identification

V publikácii TransReID: Transformer-based Object Re-Identification [HLW+21] sa zaoberali reidentifikáciou vozidiel pomocou ViT.

Kapitola 4

Výskum

V tejto kapitole sa budeme venovať navhrhú nášho riešenia pre reidentifikáciu vozidiel. Na začiatku kapitoly sa budeme venovať vytvoreniu architektúry a komponentov z ktorých sa skladá. Následne si popíšeme voľbu hyperparametrov, priebeh trénovania, trénovacie triky a na záver kapitoly si ukážeme ako majú jednotlivé trénovacie triky vplyv na výsledok reidentifikácie.

4.1 Návrh architektúry

Pri návrhu riešenia architektúry sme vychadzali z poznatkov, ktoré sme nadubli pri trénovaní metódy TransReID prezentovanej v publikácii od He a kol. [HLW+21], ktorá využívala ViT a dosahovala 82.3% mAP a trikom z publikácie od Luo a kol. [LGL+19], ktorá sa venovala reidentifikácii osôb. Ako základ našej architektúry sme sa rozhodli využiť Swin Transformer.

4.1.1 Extrakcia príznakov

4.1.2 Optimalizátor

4.1.3 Cenová funkcia

4.2 Trénovacie triky

4.2.1 Krok učenia

4.3 Výsledky

4.4 Vplyv trénovacích trikov na výsledky

Kapitola 5

Implementácia

Kapitola 6

Vyhodnotenie

Záver

Záver

Zoznam obrázkov

Literatúra

[Cho17] François Chollet. *Deep Learning with Python*. Manning, November 2017.

[DBK+21] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale, 2021.

[GBC16] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`.

[HLW+21] Shuting He, Hao Luo, Pichao Wang, Fan Wang, Hao Li, and Wei Jiang. Transreid: Transformer-based object re-identification, 2021.

[HZRS15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015.

[KPcCC20] Pirazh Khorramshahi, Neehar Peri, Jun cheng Chen, and Rama Chellappa. The devil is in the details: Self-supervised attention for vehicle re-identification, 2020.

[KSDFF13] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *4th International IEEE Workshop on 3D Representation and Recognition (3dRR-13)*, Sydney, Australia, 2013.

[KU19] Sultan Daud Khan and Habib Ullah. A survey of advances in vision-based vehicle re-identification. *Computer Vision and Image Understanding*, 182:50–63, May 2019.

[LGL+19] Hao Luo, Youzhi Gu, Xingyu Liao, Shenqi Lai, and Wei Jiang. Bag of tricks and a strong baseline for deep person re-identification, 2019.

[LLC+21] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows, 2021.

[LLMM16] Xinchen Liu, Wu Liu, Tao Mei, and Huadong Ma. A deep learning-based approach to progressive vehicle re-identification for urban surveillance. In *ECCV (2)*, pages 869–884, 2016.

[LLMM18] Xinchen Liu, Wu Liu, Tao Mei, and Huadong Ma. Provid: Progressive and multimodal vehicle reidentification for large-scale urban surveillance. *IEEE Transactions on Multimedia*, 20(3):645–658, 2018.

[mat] Convolutional neural network. `https://www.mathworks.com/discovery/convolutional-neural-network-matlab.html`. Navštívené: 23. apríl 2021.

[NWA+21] Milind Naphade, Shuo Wang, David C. Anastasiu, Zheng Tang,

Ming-Ching Chang, Xiaodong Yang, Yue Yao, Liang Zheng, Pranamesh Chakraborty, Anuj Sharma, Qi Feng, Vitaly Ablavsky, and Stan Sclaroff. The 5th ai city challenge. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2021.

[SY19] A Shustanov and P Yakimov. Modification of single-purpose cnn for creating multi-purpose cnn. *Journal of Physics: Conference Series*, 1368:052036, 11 2019.

[SZ15] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition, 2015.

[VSP+17] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2017.

[YSS19] Muhamad Yani, M.T. Budhi Irawan S, Si., and M.T. Casi Setiningsih S.T. Application of transfer learning using convolutional neural network method for early detection of terry's nail. *Journal of Physics: Conference Series*, 1201(1):012052, may 2019.