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

ADAPTACE NEURONOVÝCH SÍTÍ PRO IDENTIFIKACI OSOB

MODEL ADAPTATION IN PERSON IDENTIFICATION

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE **Bc. JAN STRATIL**
AUTHOR

VEDOUCÍ PRÁCE **Ing. MICHAL HRADIŠ, PhD.**
SUPERVISOR

BRNO 2019

Vysoké učení technické v Brně
Fakulta informačních technologií

Ústav počítačové grafiky a multimédií (UPGM) Akademický rok 2018/2019

Zadání diplomové práce

Student: **Stratil Jan, Bc.**
Program: Informační technologie Obor: Počítačová grafika a multimédia
Název: **Adaptace neuronových sítí pro identifikaci osob**
Model Adaptation in Person Identification
Kategorie: Zpracování obrazu

Zadání:

1. Prostudujte základy metody pro identifikaci osob podle obličeje a metody adaptace neuronových sítí.
2. Vytvořte si přehled o současných metodách pro adaptaci neuronových sítí v rozpoznávání osob a o možnostech generování a úprav záběrů obličejů.
3. Vyberte a případně upravte konkrétní metody a aplikujte je na úlohu rozpoznání tváří se zaměřením na variabilitu prostředí, póz, nebo výrazů.
4. Obstarejte si databázi vhodnou pro experimenty.
5. Implementujte navrženou metodu a proveďte experimenty nad datovou sadou.
6. Porovnejte dosažené výsledky a diskutujte možnosti budoucího vývoje.
7. Vytvořte stručné video prezentující vaši práci, její cíle a výsledky.

Literatura:

- Taigman et al.: DeepFace: Closing the Gap to Human-Level Performance in Face Verification. CVPR 2014.
- Parkhi et al.: Deep face recognition. Proceedings of the British Machine Vision 1.3 (2015): 6.
- Yang, Jiaolong, et al.: Neural aggregation network for video face recognition. arXiv preprint arXiv:1603.05474 (2016).
- Ebrahimi Kahou, Samira, et al.: Recurrent neural networks for emotion recognition in video. Proceedings of the 2015 ACM on International Conference on Multimodal Interaction. ACM, 2015.

Při obhajobě semestrální části projektu je požadováno:

- Body 1 až 4.

Podrobné závazné pokyny pro vypracování práce viz http://www.fit.vutbr.cz/info/szz/

Vedoucí práce: **Hradiš Michal, Ing., Ph.D.**
Vedoucí ústavu: Černocký Jan, doc. Dr. Ing.
Datum zadání: 1. listopadu 2018
Datum odevzdání: 22. května 2019
Datum schválení: 1. listopadu 2018

Abstrakt

Tato práce se zabývá rozpoznáváním tváří pomocí konvolučních neuronových sítí a jejich problémy v dnešní době, jako jsou variabilita póz, osvětlení a výrazů. Shrnuje dosavadní přístupy, architektury a nejnovější chybové funkce. Dále se věnuje metodám pro rotaci obličeje pomocí *GAN* sítí. V rámci práce jsou navrženy a natrénovány 3 neuronové sítí pro rozpoznávání tváří. Nejlepší z nich dosáhla přesnosti 99.38% na datasetu *LFW* a 88.08% na datasetu *CPLFW*. Dále je navržena síť pro rotaci obličeje *PCGAN*, která může být použita pro účely frontalizace obličeje či augmentace dat. Síť je vyhodnocena na datasetu *Multi-PIE* a pomocí frontalizace zvyšuje úspěšnost identifikace.

Abstract

This thesis deals with facial recognition using convolutional neural networks and with their current problems, which are pose, lighting and expression variance. It summarizes existing approaches, architectures and most recent loss functions. Further it deals with methods for rotating faces using *GAN* networks. In this thesis 3 neural networks are designed and trained for facial recognition. The best of them achieves 99.38% accuracy on *LFW* dataset and 88.08% accuracy on *CPLFW* dataset. Next face rotation network *PCGAN* is designed, which can be used for face frontalization or data augmentation purposes. This network is evaluated on *Multi-PIE* dataset and using the face frontalization it increases identification accuracy.

Klíčová slova

neuronové sítě, konvoluční neuronové sítě, rozpoznávání tváří, rotace tváří, syntéza tváří, GAN sítě, augmentace, frontalizace

Keywords

neural networks, convolutional neural networks, facial recognition, face rotation, face synthesis, GAN networks, augmentation, frontalization

Citace

STRATIL, Jan. *Adaptace neuronových sítí pro identifikaci osob*. Brno, 2019. Diplomová práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Michal Hradiš, PhD.

Adaptace neuronových sítí pro identifikaci osob

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně pod vedením pana Ing. Michala Hradiše, PhD. Další informace a data mi poskytl Ing. Marián Beszédeš, Ph.D.. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
Jan Stratil
22. května 2019

Poděkování

Tímto bych chtěl velmi poděkovat mému vedoucímu práce Ing. Michalovi Hradišovi, PhD., za skvělé vedení, ochotu, cenné připomínky a rady, které mi poskytl při řešení této práce. Dále bych chtěl poděkovat firmě Innovatrics za poskytnutá data a anotace k datasetům, konkrétně Ing. Mariánu Beszédešovi, Ph.D. za jeho ochotu. Tato práce vznikla za podpory projektů CERIT Scientific Cloud (LM2015085) a CESNET (LM2015042) financovaných z programu MŠMT Projekty velkých infrastruktur pro VaVaI.

Obsah

Kapitola 1

Úvod

Neuronové sítě mají v oblasti zpracování obrazu stále větší využití. S narůstajícím výpočetním výkonem, nových a lepších architektur, počtem a velikostí datasetů je stále snazší aplikovat tyto metody na problémy, které v minulosti nebylo možné uspokojivě řešit strojově a automatizovaně. Rozpoznávání tváří je jedna z oblastí zpracování obrazu, ve které došlo v poslední době k obrovskému posunu [46]. V biometrických systémech je velice podstatnou složkou, protože že šance, že obličeje 2 různých osob jsou identické je velmi nízká. Z toho důvodu se mohou systémy pro rozpoznávání tváří využívat v mnoha sektorech a zajišťovat tak bezpečnost.

V minulosti, ještě před zpopularizováním neuronových sítí, se používaly metody s ručně vytvořenými příznaky, jako je například metoda *LBP* [1]. Problémem těchto metod bylo, že neměly dostatečné rozlišovací schopnosti a neblížily se ani úspěšnostem, které měl člověk. Ukázalo se, že využitím neuronových sítí lze dosáhnout lepších výsledků.

Největší průlom byl zaznamenán v roce 2014, kdy byla publikována práce výzkumníků ze společnosti Facebook [46], kteří na datasetu *LFW* [20] dosáhli úspěšnosti srovnatelné s úspěšností, která byla dosažena na základě rozhodování lidí. Od této doby bylo zveřejněno mnoho dalších prací na toto téma a spolu s nimi i nové a větší trénovací datasety, které se pro tuto úlohu používají.

I přes významný pokrok se výsledky v experimentálním prostředí podstatně liší od výsledků v reálném prostředí [48]. Problémy dělají špatně nasvícené obličeje, nízká kvalita snímků, extrémní natočení obličejů, variabilita výrazů, zakryté části obličejů, a jiné. V poslední době vzniká mnoho prací, kde je snahou se s těmito problémy nějakým způsobem vypořádat [9].

Řešením variability natočeníq obličeje může být například 3D zarovnání obličeje [46], díky čemuž síť nemusí být tolik invariantní vůči rotaci. Jinou možností je rozšířit trénovací dataset syntetickými daty, které pokrývají výše uvedené problematické situace [31]. Tím se síť dokáže lépe vyrovnat s těmito případy.

Je možné také využít hlubokých neuronových sítí, které se přímo naučí obličej frontalizovat (či obecně syntetizovat) [59]. S příchodem *GAN* sítí [11], které se dokáží naučit produkovat snímky odpovídající dané distribuci trénovacích dat, se ukázalo, že jsou pro tyto problémy vhodné a dokáží doplnit chybějící neviditelné části obličeje [19, 21, 47].

Tato práce se zabývá zhodnocením dosavadních přístupů pro rozpoznávání tváří pomocí konvolučních neuronových sítí se zaměřením na zlepšení výše popsaných problémů. Konkrétně se práce zaměřuje na problematiku variability póz. Práce se skládá ze 2 částí. První část je věnována kompletnímu shrnutí přístupu k rozpoznávání tváří s využitím neuronových sítí a dále pak navrhnutím a natrénováním konkrétní sítě. Cílem je natrénovat síť

tak, aby produkovala co nejvíce diskriminativní několika dimenzionální příznakový vektor, který je invariantní vůči uvedeným problémům. Druhá část se věnuje metodám pro úpravu záběrů obličejů, konkrétně změnám póz obličejů. Cílem je vytvořit síť, která rotuje obličej z libovolné pózy do libovolné pózy. Důraz je kladen na to, aby výstup ze sítě mohl být dále využit pro potřeby rozpoznávání. Obecně se taková síť může použít pro normalizaci (frontalizaci) obličeje při předzpracování dat, nebo pro rozšíření datové sady.

Práce je rozdělena do několika kapitol. V kapitole 2 je věnován prostor přístupům a metodám pro rozpoznávání tváří pomocí neuronových sítí. Je zde naznačen kompletní postup pro rozpoznávání tváří spolu s vývojem architektur sítí a jejich trénování. Jsou zde popsány metody, které aktuálně dosahují nejlepších výsledků (tzv. *state-of-the-art* metody). Kapitola 3 se věnuje shrnutí přístupů pro syntézu obličejů a zhodnocení jejich výsledků. Další kapitola 4 obsahuje stručný výčet použitých či v dnešní době nejvíce využívaných datasetů. Dále následuje kapitola 5, která nastiňuje návrh pro obě části práce. V kapitole 6 jsou popsány experimenty, které byly provedeny a celkové vyhodnocení.

Kapitola 2

Neuronové sítě pro rozpoznávání tváří

Rozpoznávání tváří se skládá z několika klíčových kroků [48]. Prvním z kroků je správná detekce obličeje ze snímku, kdy se získává oblast ve které se obličej nachází (tzv. *bounding box*).

Dalším krokem je zarovnání tohoto obličeje do kanonického tvaru, kdy cílem je, aby části obličejů byly po zarovnání, pokud možno na stejných lokacích pro různé snímky. K tomu je nutné získat význačné body obličeje, pomocí kterých se obličeje zarovnávají. Cílem zarovnání je usnadnit práci modelu pro extrakci příznakového vektoru.

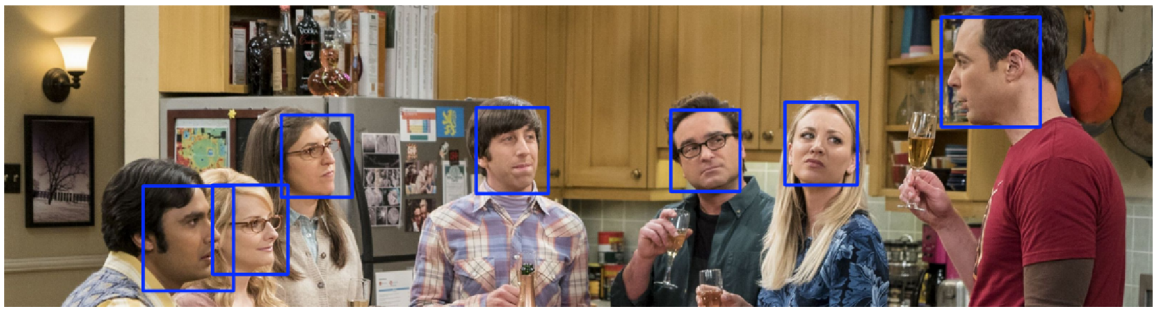
Pomocí modelu pro extrakci příznakového vektoru je poté z tohoto snímku obličeje extrahován kompaktní diskriminativní příznakový vektor reprezentující identitu ze snímku. Tyto příznakové vektory by měly mít tu vlastnost, že snímky s obličeji náležící stejné identitě by měly patřit do podobného vektorového prostoru. Vektory patřící stejným osobám by si tedy měly být velmi podobné a naopak vektory nepatřící stejným osobám co nejméně podobné a tím splňovat mezitřídní separabilitu a vnitrotřídní kompaktnost.

Vektory lze poté vzájemně porovnávat a tím provádět 2 základní úlohy při rozpoznávání tváří, kterými jsou **verifikace** a **identifikace**. Úlohou verifikace je rozhodnout, zda-li 2 obličeje náleží stejné osobě či nikoliv. Jedná se tedy o porovnávání 1:1. Úloha identifikace spočívá v rozhodování, zda-li se identita ze snímku nachází v existující databázi identit a pokud ano, tak o jakou identitu se jedná. Jedná se tedy o porovnávání 1:N.

V této kapitole jsou postupně popsány všechny klíčové kroky při rozpoznávání tváří. Zároveň jsou zde popsány metody trénování konvolučních neuronových sítí pro získání výše uvedeného příznakového vektoru, které jsou dále nazývány jako sítě pro extrakci příznakového vektoru.

2.1 Detekce obličeje

Prvním krokem v rozpoznávání tváří je správná lokalizace obličeje. Cílem detekce je získat ze vstupních snímků ty oblasti, ve kterých se nachází obličej (ukázka detekce je na obrázku 2.1). V minulosti se využívaly přístupy, které nepoužívaly neuronové sítě. Příkladem může být například detektor obličeje od pánů Viola a Jones [49], což je kaskádový detektor, který pracuje s integrálním obrazem a Haarovými příznaky. Tyto detektory byly rychlé a poměrně přesné pro frontální snímky, nicméně s větším natočením obličeje přesnost detekce rapidně klesá. S nástupem neuronových sítí se začaly používat detektory na nich založené. Přesnost



Obrázek 2.1: Ukázka detekce více obličejů v jednom snímku.

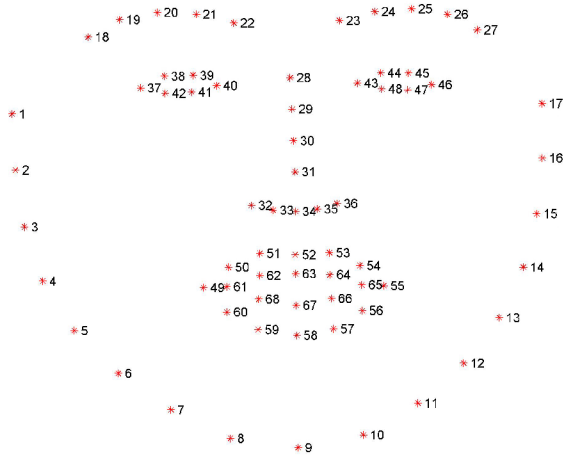
více natočených obličejů se zvýšila, zároveň se snížila míra špatných detekcí. Obecně došlo ke zefektivnění detekce a proto je dnes možné detekovat obličeje v reálném čase. Dále jsou popsané dvě vybrané metody pro detekci obličeje, které byly využity pro detekci obličejů v této práci.

Multi-task Cascaded Convolutional Network
Multi-task Cascaded Convolutional Network Neboli *MTCNN* [60] je neuronová síť pro detekci obličejů v obraze. Skládá se ze 3 sítí, kde první nejmenší síť (tzv. *P-Net*) vybere několik kandidátních oblastí, kde je určitá šance, že se nacházejí obličeje. Všechny kandidátní oblasti jsou poté vstupem do další větší sítě (*R-Net*), která redukuje takové oblasti, kde se obličeje nenachází. Zbylé oblasti jsou vstupem do poslední největší sítě (*O-Net*), která opět redukuje oblasti, které neobsahují tváře a zároveň pro oblasti, kde se tváře nacházejí získá pozice 5 význačných bodů (oči, nos, koutky úst) a souřadnice tzv. *bounding box* (obdélníková oblast, kde se obličej nachází). Autoři uvádějí, že díky tomu, že byla síť trénována víceúčelově (detekce oblasti i lokalizace význačných bodů) bylo dosaženo lepších výsledků. I přesto, že tato práce vznikla v roce 2013, je *MTCNN* síť dodnes hojně využívána díky dobrému poměru rychlosti a přesnosti.

Single Shot Scale-invariant Face Detector
Zkráceně S^3FD [61] je další síť pro detekci obličejů v obraze. V této práci se inspirují z *Faster R-CNN* [36] a z *SSD* [27] detektorů. Přínosem této práce je zlepšení detekce malých obličejů a redukce falešně detekovaných obličejů. Detekce dokáže probíhat v reálném čase na grafické kartě. Úspěšnost detekce je vyšší než v případě *MTCNN* detektoru, nicméně vzhledem k tomu, že je síť založena na robustní architektuře *VGG16* [43] je detekce oproti *MTCNN* pomalejší.

2.2 Zarovnání obličeje

Po získání oblasti, kde se nachází obličej, je dalším krokem získání klíčových bodů v obličeji, které budou využity pro transformaci obličeje do kanonického tvaru. Model pracující se zarovnanými obličeji nemusí být díky zarovnání příliš prostorově a měřítkem invariantní, což dále vede k lepším výsledkům. Pro samotné zarovnání je v drtivé většině případů potřeba znát pozice význačných bodů v obličeji.



Obrázek 2.2: Model obličeje s 68 význačnými body v obličeji (převzato z https://ibug.doc.ic.ac.uk/resources/facial-point-annotations/)

Detekce význačných bodů v obličeji

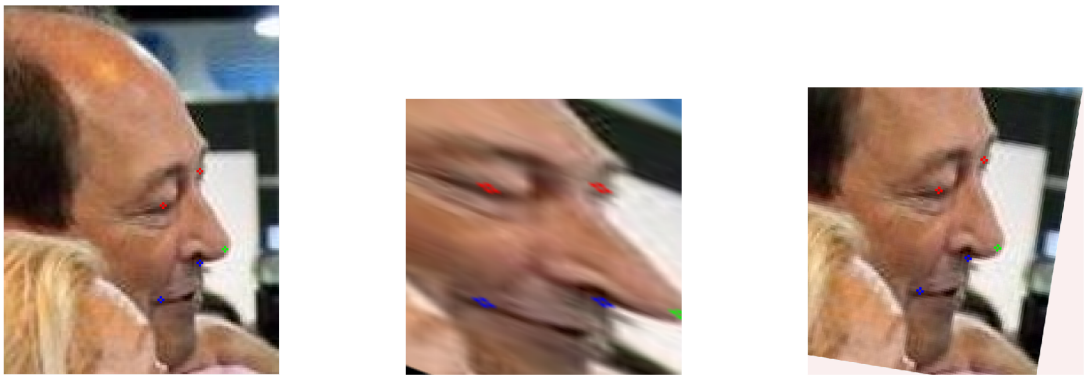
Pozice význačných bodů v obličeji se mohou získávat současně s detekcí obličeje (jako tomu je v případě *MTCNN* [60]) nebo dodatečně z detekované oblasti, kterou získal detektor.

Možným přístupem pro získání lokací je využít regrese teplotních map jako v práci pana Bulata a Tzimoropoulose [4]. V této práci se snaží z již detekovaných oblastí obličeje nalézt 68 význačných bodů v obličeji (model obličeje s těmito body je znázorněn na obrázku 2.2). Pro tyto účely využívají síť, která byla dříve využita pro odhad pózy lidského těla, nazývanou *HourGlass* [33]. 68 význačných bodů je reprezentováno pomocí teplotních map vykreslených gaussovými funkcemi, kde střed určuje lokaci daného bodů. Pro účely trénování je vytvořeno 68 map, které se síť pomocí regrese učí odhadnout. Podobným přístupem dokázali získat i 3D souřadnice bodů v obličeji.

Jinou možností je získat význačné body z 3D masky obličeje. V práci pana Fenga et al. [10] natrénovali poměrně jednoduchou síť typu enkóder – dekóder. Vstupem do této sítě je snímek o maximálních rozměrech 256 pixelů. Výstupem je poziční mapa, jejíž hodnoty pixelů udávají pozice bodů v kostce o rozměrech $256 \times 256 \times 256$. Každý pixel na poziční mapě odpovídá vždy stejnému bodu v obličeji. Síť je díky své jednoduchosti rychlá a i přes svou velikost dosahuje poměrně dobrých výsledků. Pomocí bodů z pozičních map lze také rekonstruovat 3D masku obličeje původního snímku.

2D zarovnání

Pro 2D zarovnání jsou použity získané význačné body v obličeji. S využitím zdrojových a cílových význačných bodů v obličeji jsou vypočteny transformace, které převádí obličeje do kanonického tvaru. Využívat se mohou například afinní [34] nebo podobnostní transformace [29, 53, 8]. Nutno zmínit, že afinní transformace může při extrémně natočeném obličeji způsobit deformaci obličeje. Podobnostní transformace takovou vlastností netrpí. Na obrázku 2.3 jsou porovnány tyto 2 transformace. U afinní transformace je ukázka deformace obličeje. Napravo je ukázka podobnostní transformace, u které k deformaci nedochází. Aktuální *state-of-the-art* metody [8] využívají právě podobnostní transformaci pro 2D zarovnání.



(a) Vstupní nezarovnaný snímek s vykreslenými 5 body v obličeji (oči, nos, koutky úst).

(b) Snímek zarovnaný pomoc afinní transformace s využitím očí a středu úst.

(c) Snímek zarovnaný pomocí podobnostní transformace s využitím očí, nosu a koutků úst.

Obrázek 2.3: Ukázka 2D zarovnání pomocí afinní a podobnostní transformace.

3D zarovnání
3D zarovnání se úzce pojí se syntetickými úpravami obličeje. Proto je 3D zarovnání popsáno v sekci 3, která se věnuje syntéze obličeje.

2.3 Architektury sítí

Zlepšování architektur sítí je jednou z nejdůležitějších částí pro zlepšování výsledků. V průběhu několika let se díky hlubší znalosti učení neuronových sítí a jejich fungování architektury síti transformovaly v takové, které jsou rychlejší, lépe se učí a dosahují lepších výsledků na aplikovaných úlohách. Stejně tak to platí i v oblasti rozpoznávání tváří.

Jedna z možností, jak nahlížet na problematiku rozpoznávání tváří je považovat ji za klasifikační problém [48]. Každou identitu lze reprezentovat jako třídu při klasifikaci. Není tedy náhodou, že se v oblasti rozpoznávání tváří využívají stejné či podobné architektury jako při klasifikaci objektů. Všechny dále zmíněné architektury byly zpopularizovány díky úspěchům v soutěži *ILSVRC*[1]. Výzkumná skupina ze Stanfordu, která za touto soutěží stojí, každoročně vyhlašovala úlohy, ve kterých mohli výzkumníci odeslat svoje řešení a porovnat výsledky svých metod s metodami ostatních výzkumníků. Jednou z úloh byla právě klasifikace objektů.

V roce 2012 byla publikována architektura neuronové sítě *AlexNet* [26], která v rámci této soutěže v daném roce dosáhla nejlepších výsledků. Síť obsahuje 60 milionů parametrů a 650 000 neuronů. Složena je z 5 konvolučních vrstev proložené max-pooling vrstvami. Pro zrychlení trénování využili nesaturované aktivační funkce zvané *ReLU* [32]. Tato síť, resp. její derivace, je využita například v práci pana Sankaranarayanana et al. [40, 39].

Síť *VGG* [43] má pojmenování podle výzkumné skupiny z Oxfordu (*Visual Geometry Group*). Tato síť obsadila v roce 2014 ve stejné soutěži (*ILSVRC*) 2. místo v klasifikaci. Zde je zkoumáno jaký vliv má využití hlubších sítí, které využívají 3×3 konvoluční filtry. Vyhodnocovány byly sítě od 11 vrstvých až po sítě, které mají 19 vrstev. *VGG* síť je využita pro rozpoznávání tváří například v práci *Deep Face* [34]. Síť je poměrně velká a doba inference je v porovnání s dnes využívanými sítěmi relativně dlouhá.

[1] http://image-net.org/challenges/LSVRC/

Na 1. místě v klasifikaci na *ILSVRC* v tomtéž roce 2014 se umístila síť *GoogLeNet* [45]. Hlavním přínosem této práce je vytvoření nového *Inception* modulu. Tento blok je rozdělen do 4 větví, kde se každá větev zaměřuje na něco jiného. První větev provádí pouze 1×1 konvoluci a tím pouze snižuje dimenzionalitu. Druhá a třetí větev nejprve sníží dimenzionalitu vstupu a poté provádí konvoluce s konvolučními jádry o velikosti 3×3 a 5×5. Díky menšímu počtu kanálu se výrazně sníží výpočetní náročnost těchto konvolucí. Poslední větev provádí 3×3 max-pooling s krokem 2 a následuje opět snížení dimenzionality. Nakonec se výsledky z jednotlivých vrstev konkatenují a dále se může provádět jakákoliv operace (následovat může klidně další *Inception* modul). Tato architektura je využita například v práci *NAN* [57].

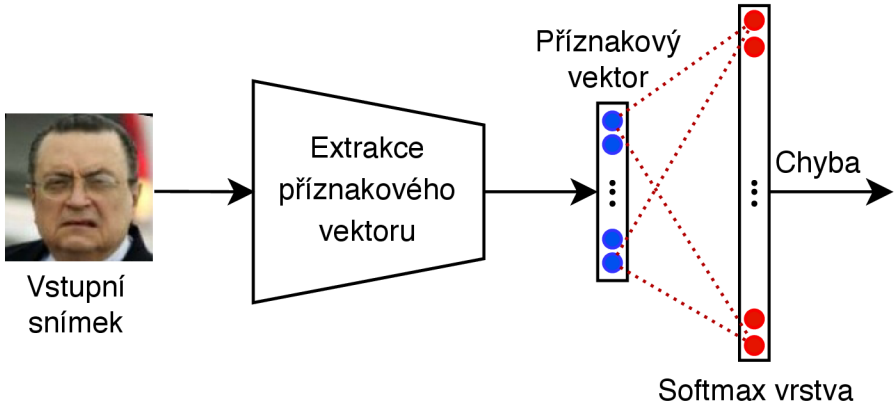
O rok později zvítězila síť s názvem *ResNet* [17]. Autoři využili jako základní strukturu sítě síť *VGG*, ale vytvořili propojení, které by mělo řešit problém degradace s hlubšími sítěmi. Jedná se o problém, kdy síť dosáhne bodu, kdy už nelze zlepšit přesnost a dalším trénováním se tato síť rychle degraduje. Základní myšlenkou je předpoklad, že když se naučí část hluboké sítě pouze produkovat identické výstupy jako vrstva předcházející tuto hlubší část, chyba bude nejhůře stejná jako bez přidání této hluboké části. Problém ale byl ten, že se hluboké vrstvy nedokázaly naučit produkovat stejný výstup. Z toho důvodu navrhnuli autoři propojení, které pouze přeskakuje několik vrstev a poté je sečteno s výstupem těchto vrstev. Argumentují tím, že je jednodušší, aby se vrstvy, které jsou tímto propojením přeskočeny, naučily produkovat nulový výstup těchto vrstev, než aby se naučily produkovat identický výstup. Ukázalo se, že díky těmto propojením lze natrénovat velmi hluboké sítě a nijak se nezvyšuje výpočetní náročnost sítí. Základními bloky jsou *BasicBlock* a *BottleNeck*. *BasicBlock* je blok složený z 2 konvolučních vrstev, které provádí 3×3 konvoluci. *BottleNeck* nahrazuje tyto dvě vrstvy třemi vrstvami. Nejprve je provedena redukce dimenzionality pomocí 1×1 na $1/4$ původního počtu kanálu, poté je provedena konvoluce 3×3 a poté se pomocí 1×1 konvoluce vrátí počet kanálů na původní hodnotu.

Na toto navazuje poslední zde zmíněný typ architektury, kterým je síť *SENet* nebo někdy také *SeResNet* [18], která zvítězila v *ILSVRC-2017*. Autoři rozšiřují *ResNet* síť o *Squueze-and-Excitation* blok, jehož účelem je ohodnotit každý kanál příznakových map podle toho jak moc přínosný je. Každý kanál je vynásoben výslednou hodnotou a až takto upravené příznakové mapy jsou sečtené s reziduální částí. Navýšení výpočetní náročnosti je o 0.26% a snížení top5 chyby je o 1% na vyhodnocovacím datasetu soutěže *ILSVRC-2017*.

Sítě *ResNet* a *SeResNet* jsou dnes nejvíc využívány a existuje několik jejich derivací. V práci *ArcFace* [8] využívají síť, která má místo vstupní velikost 224×224 velikost $2\times$ menší. Z této sítě je vypuštěn krok o velikost 2 v první konvoluci. Zároveň upravili způsob finálního výpočtu příznakového vektoru. Po výstupu z posledního reziduálního bloku je provedena *Batch normalizace* [22], dropout, plně propojená vrstva a opět *Batch normalizace*, což je finální výstup ze sítě.

2.4 Způsoby trénování a chybové funkce

Hned po architektuře sítě je další důležitou části způsobem, jakým se sítě trénují. Klíčová je zde chybová funkce, pomocí které se síť učí a upravují se hodnoty parametrů. Ukázalo se, že použitím sofistikovanější chybové funkce lze významně urychlit trénování a zároveň zlepšit přesnost výsledné sítě. Příkladem je třeba *AM-Softmax* [51] chybová funkce, pomocí které byla natrénovaná síť, jejíž výsledky jsou srovnatelné se sítí, která byla trénována pomocí *TripletLoss* [41] chybové funkce, přičemž pro její trénování byl využit dataset řádově $200\times$ menší.



Obrázek 2.4: Trénovací struktura s využitím softmax funkce. Ze vstupního snímku se extrahuje příznakový vektor, ten je poté vstupem do plně propojené vrstvy a pomocí softmax chybové vrstvy se počítá chyba.

Softmax

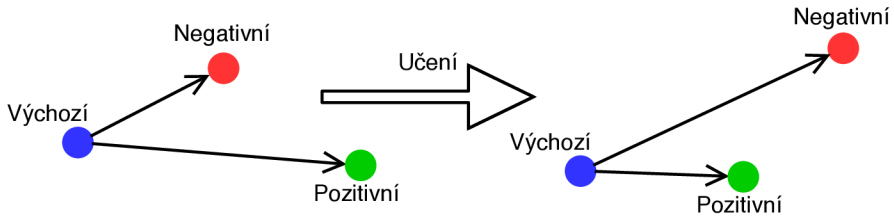
Jak je výše zmíněno, na rozpoznávání lze nahlížet jako na klasifikační problém. V pracích DeepID [44] a DeepFace [46] využili tento přístup pro trénování sítě pro extrakci příznakového vektoru. Základním principem je zde neuronová síť, jejíž výstupem je několikadimenzionální příznakový vektor, který reprezentuje obličej. Při trénování je za tuto poslední vrstvu přidána jedna plně propojená vrstva, kde počet neurononů odpovídá počtu identit, které má síť rozpoznat. Schéma je na obrázku 2.4. Výstup z této poslední vrstvy je vstupem do *Softmax* [24] aktivační funkce a spolu s *Cross-entropy* [24] chybovou funkcí je síť trénována. Tyto dvě funkce dohromady tvoří předpis

$$L_i = \sum -log\left(\frac{e^{f_{y_i}}}{\sum_j e^{f_j}}\right), \tag{2.1}$$

kde i je index trénovacího vzorku, y_i udává index odpovídající správné identitě, f_{y_i} je výstupní hodnota softmax funkce na indexu y_i, f_j je výstupní hodnota softmax funkce na indexu j. Cílem je tedy to, aby hodnota na indexu odpovídající cílové identitě byla co nejvyšší. Tato vrstva, nebo lépe řečeno blok, je využit pouze při trénování. Po natrénování se tento blok odstraní a pracuje se pouze s příznakovým vektorem, který je dále použit pro další porovnávání s ostatními příznakovými vektory. Tento vektor je často normalizován, například podle jeho L_2 normy [34].

Contrastive Loss

Ukázalo se, že *Softmax* chybová funkce není dostačující pro natrénování uspokojivě diskriminativních příznakových vektorů. Důvodem je, že *Softmax* chybová funkce nedokáže dostatečně generalizovat i pro subjekty, které nejsou v trénovací sadě. Předpis chybové funkce dobře řeší mezitřídní separabilitu, ale nemusí dostatečně redukovat vnitrotřídní variabilitu [48]. Proto se dále začalo používat metrické učení, jehož cílem je, aby vektory, které náleží stejné identitě měly mezi sebou co nejmenší vzdálenost a naopak vektory, které jsou rozdílné měly vzdálenost co největší. Proto byly využity chybové funkce, které mají tyto metrické vlastnosti přímo zakomponované ve vlastním předpisu. *Contrastive Loss* [16] je jedna z využívaných chybových funkcí [57] a její předpis je



Obrázek 2.5: Grafické znázornění principu učení pomocí triplet loss. Cíl je snižovat vzdálenost výchozího vzorku s pozitivním a zvyšovat vzdálenost výchozího a negativního vzorku.

$$L_{i,j} = (1 - y_{i,j})max(0, m - \|f_i - f_j\|_2^2) + y_{i_j}\|f_i - f_j\|_2^2 \tag{2.2}$$

kde i a j jsou indexy snímků vstupního páru, f_i a f_j jsou příznakové vektory, $y_{i,j}$ je rovno 1 v případě, že pár (i, j) náleží stejné identitě a 0 v opačném případě. m je konstanta, která definuje velikost hranice pro trénování. Vstupem pro tuto funkci jsou tedy vždy dvojice stejných či rozdílných příznakových vektorů.

Triplet Loss

Podobně na stejných metrických vlastnostech funguje chybová funkce s názvem *Triplet Loss* [41], která se snaží přiblížit vektory patřící stejné identitě a oddálit vektory patřící rozdílným identitám (jak je znázorněno na obrázku 2.5). Narozdíl od *Contrastive Loss* využívá trojic, kde vždy 2 vzorky patří stejné identitě a 1 vzorek patří identitě jiné. *Triplet Loss* má předpis

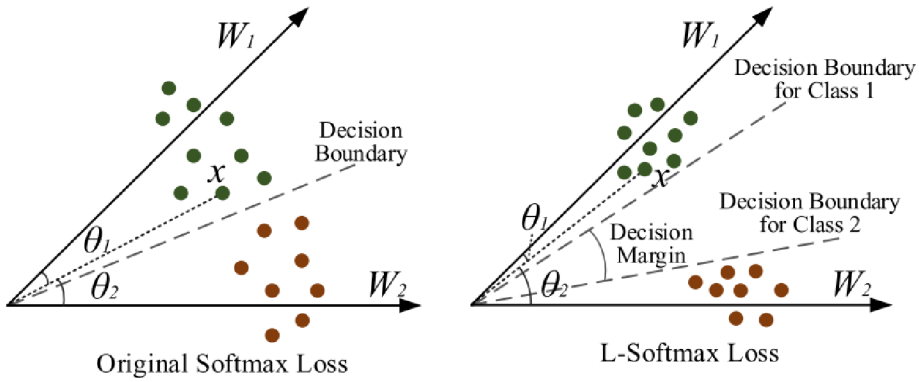
$$L = \|f_a - f_p\|_2^2 + \alpha < \|f_a - f_n\|_2^2 \tag{2.3}$$

kde f_a je vektor, vůči kterému se porovnávají vektory f_p a f_n, f_p je vektor reprezentující stejnou identitu jako vektor f_a a f_n je vektor, který reprezentuje jinou identitu než vektor f_a. α je konstanta definující rozhodující hranici pro trénování. Podobně jako u *Contrastive Loss* je zde problém s vhodným nastavením této hodnoty. Pomocí této chybové funkce je trénováno například v práci *FaceNet* [41]. Často se využívá předtrénování nejprve pomocí *Softmax* chybové funkce a až poté pomocí *Triplet* chybové funkce, stejně jako v jejich práci.

Pro zlepšení dosažených výsledků u *Contrastive* a *Triplet Loss* se při trénování využívá tzv. *Hard mining* [48], při kterém dochází k výběru takových trénovacích dvojic (resp. trojic), které jsou obtížné. Obtížnými dvojicemi (resp. trojicemi) jsou myšleny například vizuálně podobné obličeje, které náleží rozdílným identitám, nebo naopak vizuálně velmi odlišné obličeje, které ale náleží stejné identitě. Tento přístup cíleně podporuje zvyšování vnitrotřídní kompaktnosti a zvyšování mezitřídní separovatelnosti.

Center Loss

Ve snaze o natrénování diskriminativních příznakových vektorů vznikla nová chybovoá funkce jménem *Center Loss* [54]. Základním myšlenkou je stanovení středových vektorů, která definují danou třídu a chyba se počítá podle vzdálenosti od středového vektoru třídy, do které vzorek spadá. Středové vektory by se ideálně měly počítat jako průměrné vektory napříč všemi vektory z dané třídy, což je ale časově náročné. Proto autoři počítají průměrný vektor vždy v rámci trénovací mini-batche. Výslednou chybovou funkci kombinují společně s klasickou *Softmax* chybovou funkcí a výsledný předpis funkce je



Obrázek 2.6: Grafické znázornění principu L-Softmax chybové funkce na příkladu binární klasifikace. Porovnány jsou rozhodovací hranice L-Softmax a Softmax chybových funkcí. Převzato z [28]

$$\mathcal{L} = \mathcal{L}_S + \lambda \mathcal{L}_C = -\sum_{i=1}^{m} log \frac{e^{W_{y_i}^T x_i + b_{y_i}}}{\sum_{j=1}^{n} e^{W_j^T x_i + b_j}} + \frac{\lambda}{2} \sum_{i=1}^{m} \|x_i - c_{y_i}\|_2^2 \tag{2.4}$$

kde $\mathcal{L}_S$ je *Softmax* chybová funkce, která je popsána výše, λ je konstanta, která udává váhu $\mathcal{L}_C$ *Center Loss* chybové funkce, i je index trénovacího vzorku, m je počet trénovacích vzorků, y_i je index třídy, j je index složky vektoru, n je velikost vektoru, který odpovídá počtu tříd. Pokud je $\lambda = 0$, pak se jedná o pouhou *Softmax* chybovou funkci. V experimentech zjistili, že v jejich případě nejlepších výsledků bylo dosaženo s $\lambda = 0.003$.

L-Softmax Loss

Od roku 2017 začaly vznikat chybové funkce, které jsou založené na úhlové (resp. cosinové) vzdálenosti. První průlomovou chybovou funkcí je *L-Softmax Loss* [28]. Zde autoři navázali na chybové funkce, které se snaží vynutit co největší vnitrotřídní kompaktnost a mezitřídní separabilitu. Softmax aktivační vrstvu vždy při trénování předchází plně propojená vrstva. Autoři tedy přepsali podobu softmax chybové funkce a zakomponovali do ní i plně propojenou vrstvu. Ukázali, že se chybová funkce dá zapsat pomoci funkce *cos* jako

$$L_i = -log\Big(\frac{e^{\|W_{y_i}\|\|x_i\| cos(\theta_{y_i})}}{\sum_j e^{\|W_j\|\|x_i\| cos(\theta_j)}}\Big) \tag{2.5}$$

V případě binární klasifikace se pro vzorek x třídy 1 snaží docílit toho, aby platilo $W_1^T x > W_2^T x$, kde skalární součiny se dají přepsat na $\|W_1\|\|x\| \, cos(\theta_1) > \|W_2\|\|x\| \, cos(\theta_2)$. Zde si autoři řekli, že chtějí ztížit kritérium pro klasifikaci a provedli změnu kritéria na $\|W_1\|\|x\| \, cos(m\theta_1) > \|W_2\|\|x\| \, cos(\theta_2)$, kde $m \in \mathcal{N} \wedge m > 1$ a $\theta_1 \in \langle 0, \frac{\pi}{m} \rangle$. Konstanta m tedy určuje úhlovou oblast, která rozděluje hranice pro klasifikaci do tříd, jak je znázorněno na obrázku 2.6. Pro speciální případ $m = 1$ se jedná o klasickou *Softmax* chybovou funkci.

S ohledem na jednodušší dopředný a zpětný průchod, zpětnou propagaci chyby a monotónnost chybové funkce je předpis upraven na

$$\mathcal{L}_i = -log\Big(\frac{e^{\|W_{y_i}\|\|x_i\| \psi(\theta_{y_i})}}{e^{\|W_{y_i}\|\|x_i\| \psi(\theta_{y_i})} + \sum_{j \neq y_i} e^{\|W_j\|\|x_i\| cos(\theta_j)}}\Big) \tag{2.6}$$

$$\psi(\theta) = (-1)^k cos(m\theta) - 2k, \theta \in [\frac{k\pi}{m}, \frac{(k+1)\pi}{m}] \tag{2.7}$$

kde $k \in [0, m-1] \wedge k \in \mathcal{N}$, $m \in \mathcal{N}$ je konstanta definující úhlovou oblast, i je index trénovacího vzorku, y_i je index třídy daného vzorku, x_i je příznakový vektor a W_{y_i} je vektor vah neuronu y_i z plně propojené vrstvy. Při velkém počtu tříd je konvergence obtížnější než u *Softmax* chybové funkce a z toho důvodu je využita kombinace *Softmax* funkce s *L-Softmax* podobně jako tomu je u *Center Loss*.

A-Softmax Loss

Na *L-Softmax* navázali autoři *SphereFace* [29] s *A-Softmax* chybovou funkcí, kteří navíc přidávají normalizaci vah plně propojené vrstvy tak, že $\|W\|_2 = 1$. Tím dosáhli toho, že se vektory budou nacházet na hyperkouli a výsledná chyba záleží pouze na úhlu mezi x_i a W_{y_i}. S hodnotou parametru $m = 4$ bylo docíleno lepších výsledků jako v *L-Softmax*.

AM-Softmax, Large Margin Cosine Loss

U *L-Softmax* a *A-Softmax* je přidána rozhodovací oblast pro rozhodování do chybové funkce pomocí násobení úhlu θ. V práci *AM-Softmax* [51] a *CosFace* [53] přicházejí s obměnou, kdy vkládají rozhodovací oblast do funkce místo násobením pouhým odečtením konstanty m $(cos(\theta) - m)$. Autoři *AM-Softmax* tento přístup porovnávají s *L-Softmax* a *A-Softmax*, kde je jejich *AM-Softmax* mnohem jednodušší, intuitivnější a také vede k jednoduššímu výpočtu gradientu při trénování. Oproti *A-Softmax* je zde využita normalizace jak vah ($\|W_{y_i}\| = 1$), tak příznakových vektorů ($\|x_i\| = 1$). Ukázalo se totiž, že norma příznakového vektoru definuje jeho kvalitu [35]. Pokud je L_2 norma vyšší pak snímky obsahují většinou kvalitní obličeje (frontální, dobře zarovnané, nerozmazané, apod.). Pokud je norma nižší, obličeje na snímcích jsou rozmazné, natočené a obecně nekvalitní. Díky této normalizaci se zlepšují výsledky v nekontrolovaném prostředí.

Problémem u normalizace příznakových vektorů je ten, že vede ke stagnaci chyby, proto za účelem zajištění konvergence při trénování použili, stejně jako v práci *NormFace* [52], parametr s, kterým normovaný vektor násobí. Hodnota parametru s je nastavena fixně na $s = 30$. Výsledný předpis této chybové funkce je

$$\mathcal{L} = -\frac{1}{N}\sum_{i=1}^{N} log\frac{e^{s(cos(\theta_{y_i})-m)}}{e^{s(cos(\theta_{y_i})-m)} + \sum_{j=1,j\neq y_i}^{c} e^{s(cos(\theta_j))}} \tag{2.8}$$

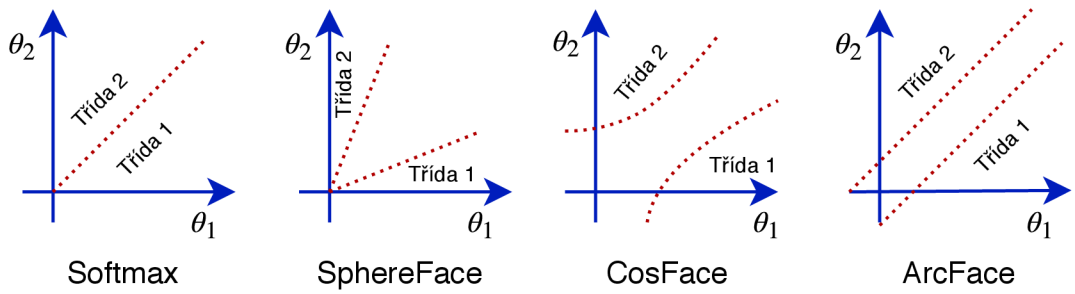
kde s je výše popsaný parametr, m je konstanta použitá pro definici meze při rozhodování a N je počet trénovacích vzorků. Ostatní parametry jsou stejné jako v rovnicích 2.6.

ArcFace

Nejnovější chybovou funkcí, která taktéž využívá úhlovou vzdálenost je funkce *ArcFace* [8]. Zde je rozhodovací oblast vložena přičtením parametru m k úhlu θ (tzn. $cos(\theta+m)$. Předpis této chybové funkce je tedy

$$L = -\frac{1}{N}\sum_{i=1}^{N} log\frac{e^{s\left(cos(\theta_{y_i}+m)\right)}}{e^{s\left(cos(\theta_{y_i}+m)\right)} + \sum_{j=1,j\neq y_i}^{n} e^{scos\theta_j}}, \tag{2.9}$$

kde všechny parametry jsou stejné jako v rovnici 2.6, ale m je úhlová oblast přidána přičtením k úhlu. Autoři experimentálně zjistili, že pro jejich konfigurace byla nejvhodnější



Obrázek 2.7: Porovnání chybových funkcí Softmax [24], SphereFace [29], CosFace [53, 51], ArcFace [8] a jejich rozhodovacích oblastí při binární klasifikaci.

hodnota $m = 0.5$ a $s = 64$. Autoři argumentují, že jejich navržená úhlová oblast má nejlepší geometrickou vlastnost. Na obrázku 2.7 jsou vykresleny geometrické porovnání chybových funkcí při binární klasifikaci.

Taktéž je zde definována obecná chybová funkce, která do sebe agreguje většinu úhlových chybových funkcí popsaných v této sekci. Její předpis je

$$L = -\frac{1}{N}\sum_{i=1}^{N} log\frac{e^{s\left(cos(m_1\theta_{y_i}+m_2)-m_3\right)}}{e^{s\left(cos(m_1\theta_{y_i}+m_2)-m_3\right)} + \sum_{j=1,j\neq y_i}^{n} e^{scos\theta_j}} \tag{2.10}$$

kde m_1 je parametr z *A-Softmax* [29], m_2 je nový parametr z této práce a m_3 je parametr z *AM-Softmax* [51] a *CosFace* [53]. Ostatní parametry jsou stejné jako v rovnici 2.8. Pro *ArcFace* je $m_2 = 0.5$.

Pokud jsou vhodně nastavené parametry m_x, kde $x \in \{1, 2, 3\}$, pak tato chybová funkce může interpretovat *Softmax*, *A-Softmax*, *AM-Softmax*, *CosFace* a samozřejmě i *ArcFace*.

Kapitola 3

Syntéza obličejů

Jak je v úvodu naznačeno, aktuálním problémem rozpoznávání tváří v reálném prostředí je velká variabilita póz, osvětlení, výrazů, věku, zakrytí částí obličeje a mnoho dalších problémů. Obecně existují 2 způsoby jak se s tímto problémem vyrovnávat. Jedním způsobem je získání a vytvoření co nejrozsáhlejších trénovacích dat, které budou dostatečně pokrývat výše uvedené problémy. Druhým způsobem, který řeší variabilitu póz a částečně zakryté obličeje, je normalizace [21] těchto snímků do frontální podoby tak, aby obličej směřoval k pomyslnému fotoaparátu. V tomto případě ale dochází k vysoké ztrátovosti informací kvůli natočení obličeje.

Tvorba rozsáhlých datových sad pro trénování je nejen časově, ale i finančně náročná. Data není potřeba jenom získat, ale i správně anotovat. Jednou z možností je využít automatizovaných procesů pro získání dat, ale problémem tohoto přístupu je velké množství špatně anotovaných dat, což vede k horším výsledkům [50]. Příkladem může být dataset *MS-Celeb-1M* [15], kde se chybovost dat či jejich anotací blíží 60%. Využití syntetických dat je proto jednou z možností jak se těmto problémům vyhnout.

S příchodem *GAN* [11] sítí, které se využívají například pro transformaci stylu nebo super resoluci [23], vznikají práce, které provádějí syntézu obličejů. Základní princip trénování *GAN* sítí je neustálý souboj mezi diskriminátorem a generátorem. Diskriminátor má datovou sadu a snaží se rozpoznávat synteticky vygenerované snímky generátorem od pravých snímků z datové sady. Generátor se učí generovat snímky takové, aby byla co největší šance, že je diskriminátor označí jako pravé. Díky tomu je možné generátor použít na doplnění částí obličeje, které nejsou kvůli natočení viditelné.

V této kapitole jsou popsány některé přístupy jak pro frontalizaci obličeje, tak pro obecné úpravy záběru obličejů. Práce se hlavně věnuje *end-to-end* syntetickým úpravám obličeje pomocí *GAN* sítí.

3.1 Syntéza založená na 3D modelech obličeje

Jedním ze starších příkladů využití syntézy obličeje může být 3D zarovnání v práci pana Taigmana et al. [46]. V této práci v rámci předzpracování dat, které byly poté vstupem do neuronové sítě, prováděli 3D zarovnání obličeje. Nejprve pomocí jednoho *SVR* (Support Vector Regressor) detekovali 6 význačných bodů v obličeji, pomocí kterých provedli 2D fronzalizaci obličeje. Dále v tomto zarovnaném a oříznutém snímku obličeje detekovali pomocí druhého SVR 67 bodů. Zprůměrováním několika 3D skenů obličeje vytvořili generický 3D model, na kterém manuálně umístili 67 bodů. Tím získali korespondenci bodů deteko-

vaných a bodů manuálně vyznačených. Díky tomu poté bylo možné transformovat obličej do frontální polohy (a obecně do libovolné polohy). Zakryté části vyřešili využitím symetrických bodů z druhé strany obličeje. Nejen díky této frontalizaci byli schopní dosáhnout v té době *state-of-the-art* výsledků.

Téměř stejného přístupu využili pan Masi et al. [31], kteří se pomocí této metody rozhodli rozšířit trénovací sadu. Jako základní dataset pro augmentaci obličejů využili *CASIA-WebFace Dataset* [58], který obsahuje přibližně 0.5 milionu snímků přibližně 10 tisíců osob. Narozdíl od výše uvedeného přístupu nehledali korespondence detekovaných bodů pouze s body na jedné masce, ale využili 10 různých 3D masek. Pomocí těchto masek dokázali měnit tvar obličeje. Zároveň obličeje rotovali do 5 různých úhlů ($\{0°, \pm 40°, \pm 75°\}$). Taktéž ještě měnili výrazy úpravami 3D masek tak, aby masky měly otevřenou pusu, zavřenou pusu, nebo se usmívali. Touto augmentací dat dokázali zlepšit výsledky na datasetu *LFW* [20] o 2.7%.

Při rotaci obličejů se ztrácí informace o neviditelných částech. Existuje několik způsobů jak se s touto ztrátou vyrovnat. V práci *UV-GAN* [7] převádějí snímky obličejů na UV mapy a chybějící části se snaží doplnit pomocí *GAN* sítí. Nejprve se provádí detekce obličeje a zarovnání. Poté se pomocí *3DMM* [3] odhadne 3D maska obličeje. Z této masky obličeje se získá *z-buffering* algoritmem informace o tom, jaké části obličeje jsou viditelné a jaké nikoliv. Z masky se vytvoří UV mapa, ve které se doplní části, které nejsou viditelné, pomocí Gaussova šumu. Takto upravená mapa je poté vstupem do generátoru, který se snaží doplnit chybějící části tak, aby byla zachována identita a fotorealističnost obličeje.

Síť je poté trénována kombinací několika chybových funkcí včetně adversariálního trénování. Generátor má architekturu typu *U-Net* [37]. 2 diskriminátory se starají o globální a lokální strukturu obličeje. Pro zachování identity je využita natrénovaná síť *ResNet-27* na datasetu *CasiaWEBFace* [58].

Tuto síť poté použili pro augmentaci dat při trénování a díky tomu dokázali zlepšit přesnost na *CFP* datasetu [42] o 6%, protože se síť lépe naučila extrahovat vektory, které jsou více invariantní vůči póze.

3.2 End-to-end syntéza obličeje

Jak je výše uvedeno, *GAN* sítě [11] mají tu vlastnost, že se dokáží naučit produkovat taková výstupní data, která se podobají distribuci dat z trénovací sady. Díky této vlastnosti jsou často používány jako prostředek pro doplnění chybějících částí obrazu. V této sekci jsou popsané vybrané metody, které provádějí *end-to-end* syntézu s využitím *GAN* sítí.

TP-GAN

V práci pana Huanga et al. [21] navrhnuli síť *TP-GAN*, která provádí frontalizaci obličeje z libovolně rotovaného obličeje. Frontalizaci provádí dvoucestný generátor, který je složen ze 2 sítí typu enkóder-dekóder. Jedna síť se stará o lokální oblasti očí, nosu a pusy a snaží se tyto části obličeje frontalizovat samostatně. Tato síť bude dále referována jako lokální síť. Jednotlivé frontalizované části jsou poté složeny dohromady na fixní pozice. Druhá síť se snaží frontalizovat celý obličej a dále bude referována jako globální síť. Složené lokální části a výstup z globální sítě jsou poté konkatenovány a pomocí jedné konvoluční vrstvy je vytvořen finální výstup generátoru.

Diskriminátor má architekturu jako jednoduchou klasifikační síť. Diskriminátor produkuje 2×2 pravděpodobnostní mapy namísto jedné skalární hodnoty. Díky tomu se může diskriminátor zaměřit na jednotlivé oblasti samostatně.

Cílem trénování je optimalizovat *min-max* problém, který je definovaný jako

$$\min_{\theta_G} \max_{\theta_D} E_{I^F \sim P(I^F)} log D_{\theta_D}(I^F) + E_{I^P \sim P(I^P)} log(1 - D_{\theta_D}(G_{\theta_D}(I^P))), \tag{3.1}$$

kde D_{θ_D} a G_{θ_G} jsou postupně diskriminátor a generátor a θ_D a θ_G jsou jejich parametry, I^F je frontální snímek, I^P je profilový snímek.

Pro trénování bylo využito několik chybových funkcí, které jsou nakonec váhově sečteny. První a nejzákladnější je L_1 chybová funkce napříč pixely s předpisem

$$L_{pixel} = \frac{1}{W \times H} \sum_{x=1}^{W} \sum_{y=1}^{H} \left| I_{x,y}^{pred} - I_{x,y}^{gt} \right|, \tag{3.2}$$

kde W a H jsou rozměry výstupních (resp. vstupních snímků), $I_{x,y}^{pred}$ je hodnota pixelu na pozici x, y z generovaného snímku a $I_{x,y}^{gt}$ je hodnota na pozici x, y cílového snímku. Tato chybová funkce je zároveň použita na podvzorkované průběžné výsledky s rozměry 64×64 a 32×32. Autoři tvrdí, že díky aplikaci této chybové funkce na více rozměrných výstupech dochází k rychlejšímu trénování, ale výsledky jsou díky tomu příliš vyhlazené.

Další chybová funkce využívá vlastnosti, že lidské obličeje jsou skoro symetrické. Díky tomu je možné rekonstruovat části, které nejsou kvůli natočení obličeje viditelné. Symetrická chybová funkce má předpis

$$L_{sym} = \frac{1}{W/2 \times H} \sum_{x=1}^{W/2} \sum_{y=1}^{H} \left| I_{x,y}^{pred} - I_{W-(x-1),y}^{pred} \right|, \tag{3.3}$$

kde W a H jsou rozměry výstupních (resp. vstupních snímků), $I_{x,y}^{pred}$ je hodnota pixelu na pozici x, y z generovaného snímku a $I_{W-(x-1),y}^{pred}$ je hodnota na pixelu symetricky odpovídajícímu na pozici x, y.

Při syntéze obličeje pro rozpoznávání tváří je kritické, aby se touto syntézou nezměnila identita. Z toho důvodu autoři využili chybovou funkci, která zachovává identitu. Zároveň na ní lze pohlížet i jako na *Perceptual* chybovou funkcci [23]. Chybová funkce je počítána z posledních 2 vrstev sítě *LightCNN* [55], kterou autoři dotrénovali na originálních snímcích *Multi-PIE* datasetu [14] a její předpis je

$$L_{ip} = \sum_{i=1}^{2} \frac{1}{W_i \times H_i} \sum_{x=1}^{W_i} \sum_{y=1}^{H_i} \left| F(I^P)_{x,y}^{i} - F(G(I^{pred}))_{x,y}^{i} \right|, \tag{3.4}$$

kde i je index vrstvy, W_i a H_i jsou rozměry této aktivační mapy vrstvy i, $F(x)^i$ je výstupní hodnota aktivační mapy vrtvy i, G je generátor a I^P a I^{pred} jsou profilový a syntetizovaný obličej.

Dále je využita adversariální chybová funkce pro rozhodování, jestli se jedná o reálné frontální snímky nebo snímky syntetické. Funkce má předpis

$$L_{adv} = \frac{1}{N} \sum_{n=1}^{N} -log(D_{\theta_D}(G_{\theta_G}(I_n^P))), \tag{3.5}$$

Metoda	±90°	±75°	±60°	±45°	±30°	±15°
LightCNN [55]	5.51 %	24.18 %	62.09 %	92.13 %	97.38 %	98.59 %
TP-GAN [21]	**64.64** %	**77.43** %	**87.72** %	**95.38** %	**98.06** %	**98.68** %

Tabulka 3.1: Výsledky identifikace na datasetu *Multi-PIE* podle protokolu *Setting 2* (protokol popsán v sekci 4.2). Výsledky jsou porovnány s *LightCNN* sítí.

Obrázek 3.1: Ukázka syntézy sítě *TP-GAN* na testovací sadě datasetu *Multi-PIE*. Nalevo je zdrojový snímek, uprostřed je cílový snímek a napravo je syntetický snímek.

kde N je počet vzorků, G_{θ_G} je generátor, D_{θ_D} je diskriminátor, I_n^P je profilový obrázek

Jako poslední je využita *Total variation regularization* chybová funkce [23], která má za cíl redukovat šum. Předpis je

$$L_{tv} = \sum_{i=1}^{W-1} \sum_{j=1}^{H-1} \left| I_{i,j}^{pred} - I_{i+1,j}^{pred} \right| + \left| I_{i,j}^{pred} - I_{i,j+1}^{pred} \right|, \tag{3.6}$$

kde $I_{i,j}^{pred}$ je hodnota na pozicích i, j predikovaného snímku, W a H je šířka a výška predikovaného snímku.

Dohromady je výsledná funkce dána předpisem

$$L_{syn} = L_{pixel} + \lambda_1 L_{sym} + \lambda_2 L_{adv} + \lambda_3 L_{ip} + \lambda_4 L_{tv}, \tag{3.7}$$

kde λ_1 - λ_4 jsou hyperparametry, které autoři empiricky nastavili na $\lambda_1 = 0.3$, $\lambda_2 = 10^{-3}$, $\lambda_3 = 3 \times 10^{-3}$, $\lambda_4 = 10^{-4}$.

Metodu vyhodnotili na datasetu *Multi-PIE* [14] podle protokolu *Setting 2* [59]. Tento protokol je vysvětlen v sekci 6.2. Výsledky lze vidět v tabulce 3.1. Ukázku syntetických snímků a originálních snímků lze vidět na obrázku 3.1.

Problém tohoto přístupu je, že trénování takové sítě je velmi časově náročné, protože se musí trénovat 4 různé lokální sítě pro frontalizaci oblastí očí, nosu a pusy a dále ještě 1 globální síť pro zachování globální informace. Dalším problémem je omezení pouze pro provádění frontalizace a nemožnost využití této trénovací architektury pro trénování syntézy obličeje do libovolné pózy.

CAPG

Na toto navazují autoři se sítí zvanou *CAPG* [19], kteří navrhnuli architekturu, která umožňuje rotovat obličej z libovolného do libovolného natočení. Informaci o zdrojovém a cílovém natočení se předává pomocí teplotních map, vykreslených pomocí lokací očí, nosu a koutků úst. Pro cílovou teplotní mapu použili zprůměrované lokace očí, nosu a koutků úst.

Oproti autorům *TP-GAN* využili pouze jednu síť pro rotaci, která odpovídá architektuře globální sítě z *TP-GAN*. Společně se vstupním obličejem do této sítě konkatenují teplotní

Metoda	±90°	±75°	±60°	±45°	±30°	±15°
LightCNN [55]	5.51 %	24.18 %	62.09 %	92.13 %	97.38 %	98.59 %
CAPG-GAN [19]	**66.05 %**	**83.05 %**	**90.63 %**	**97.33 %**	**99.56 %**	**99.82 %**

Tabulka 3.2: Výsledky identifikace na datasetu *Multi-PIE* podle protokolu *Setting 2* (protokol popsán v sekci 4.2). Výsledky jsou porovnány s *LightCNN* sítí.

Obrázek 3.2: Ukázka syntézy sítě *CPGAN* na testovací sadě datasetu *Multi-PIE*. Nalevo je zdrojový snímek, uprostřed je cílový snímek a napravo je syntetický snímek.

mapu s vykreslenými zdrojovými a cílovými lokacemi bodů. Jako diskriminátor využívají 2 různé sítě. Jedna síť diskriminátoru dostává na vstup zdrojový a cílový (resp. vygenerovaný) snímek. Druhá síť dostává na vstup cílový (resp. vygenerovaný) snímek a teplotní mapu cílových bodů. Stejně jako v *TP-GAN* jsou u diskriminátorů výstupem pravděpodobnostní mapy, ale zde zvolili velikost těchto map 6×6.

Pro trénování je využita adversariální chybová funkce a L_1 chybová funkce stejně jako v *TP-GAN*. Dále využili stejnou, taktéž dotrénovanou, síť *LightCNN* [55] na datasetu *Multi-PIE* pro zachování identity s chybovou funkcí

$$L_{ip} = \left\| F_p(I^T) - F_p(G(I^S, P^S, P^T)) \right\|_F^2 + \left\| F_{fc}(I^T) - F_{fc}(G(I^S, P^S, P^T)) \right\|_2^2, \tag{3.8}$$

kde I^T je cílový snímek, I^S je zdrojový snímek, P^S a P^T jsou teplotní mapy zdrojových a cílových bodů udávající zdrojovou a cílovou pózu, F_p je výstup poslední pooling vrsty a F_{fc} je výstup poslední plně propojené vrstvy sítě *LightCNN*, $\|\|_F$ a $\|\|_2$ je Frobennova norma a L_2 norma.

Dohromady tyto chybové funkce jsou váhově sečteny pomocí váhových hyperparametrů a chybová funkce je

$$L = \lambda_1 L_{pix} + \lambda_2 L_{adv}^{ii} + \lambda_3 L_{adv}^{pe} + \lambda_4 L_{ip} + \lambda_5 L_{tv}, \tag{3.9}$$

kde $\alpha_1 - \alpha_5$ jsou hyperparametry, L_{pix} je L_1 chybová funkce a L_{adv}^{ii} a L_{adv}^{pe} jsou adversariální chybové funkce, které jsou stejné jako v *TP-GAN*, L_{ip} je chybová funkce pro zachování identity a L_{tv} je chybová funkce pro redukci šumu.

Stejně jako u předchozí metody autoři vyhodnocovali úspěšnost syntézy mimo jiné pomocí protokolu *Setting 2* na datasetu *Multi-PIE* [14]. Výsledky jsou v tabulce 3.2 a ukázka syntetizovaných snímků na obrázku 3.2.

Kapitola 4

Datasety

V této kapitole je věnován prostor pro výčet datasetů, které je možné využít pro trénování sítí pro rozpoznávání tváří, pro vyhodnocení těchto sítí a pro syntézu obličejů. U každého datasetu je stručný popis a statistiky.

4.1 Trénovací datasety

CASIA-WebFace
CASIA-WebFace dataset [58] obsahuje **494 414** obrázků celkem **10 575** osob. Každá osoba má minimálně **2** snímky, maximálně **804** a průměrně **47** snímků. Bohužel je u tohoto datasetu problém s větším počtem špatně anotovaných dat, tudíž je vhodné před využíváním tohoto datasetu tyto data odstranit. V době psaní této práce není již dataset dostupný ke stažení z originálních webových stránek.

VGGFace2
Dataset *VGGFace2* [6] navazuje na dataset *VGGFace* [34] a je jedním z aktuálně největších veřejně dostupných datasetů, který neobsahuje příliš šumu (chybných detekcí, špatných anotací, apod.). Obsahuje **3.31** milionu obrázků celkem **9 131** subjektů. V datasetu každému subjektu náleží minimálně **80** snímků a maximálně **843**. Průměrný počet snímků na osobu je **362.6**. Snímky byly staženy z vyhledávače obrázků z Googlu a obsahují vysokou varianci v natočení obličeje, v osvětlení, ve věku a rase subjektů. Součástí datasetu jsou anotace obsahující pozice očí, nosu a koutků úst. Tento dataset je částečně manuálně vyčištěn a autoři udávají, že dataset obsahuje maximálně **4**% šumu. Oproti svému předcházejícímu datasetu je větší, obsahuje méně šumu a modely natrénované na tomto datasetu dosahují vyšších přesností na testovací sadě.

MS-Celeb-1M
Dalším z rozsáhlých datasetů je dataset *MS-Celeb-1M* [15], který je zároveň jedním ze standardních datsetů pro vyhodnocování. *MS-Celeb-1M* obsahuje **10** milionů obrázků celkem **100** tisíc subjektů. Průměrně každému subjektu náleží **105** snímků. Tento dataset není manuálně čištěný a z toho důvodu obsahuje přibližně 60% šumu – tzn. vyčištěný dataset obsahuje přibližně **3.8** milionu snímků náležící **85** tisícům osobám.

IMDb-Face
V roce 2018 vznikl dataset *IMDb-Face* [50] a jeho výhodou je, že obsahuje **1.7** milionu

snímků pro **59** tisíc osob, které byly manuálně vyčištěné z původních **2** milionů snímků – to znamená, že je zde minimální počet chybných anotací, detekcí, apod.. Průměrně každá osobě má **29** snímků. Jak název vypovídá, tyto snímky byly získány z IMDb[1] stránek. Tento dataset obsahuje v porovnání s *VGGFace2* datasetem více snímků, ve kterých jsou obličeje natočené v extrémních úhlech, ale má zase menší průměrný počet snímků na osobu (**29**).

MultiPIE
MultiPIE [14] je jediným větším aktuálně veřejně dostupným datasetem, který obsahuje snímky získané v jednu chvíli z několika různých úhlů. Nachází se zde přes **750 000** snímků snímané z **15** různých úhlů, **19** různých světelných podmínek a rozdílných výrazech v obličeji. Snímky náleží celkem **337** subjektům. Snímky byly nasnímané v celkem 4 sezeních, kde se více než 100 osob zúčastnilo více sezení. Sezení byla provedeny s časovým odstupem, díky tomu jsou zde větší variance u stejných identit.

4.2 Datasety pro vyhodnocení

MultiPIE
Protokol nazvaný *Setting 2* byl vytvořen v práci pana Yima et al. [59]. Autoři vytvořili tento protokol za účelem vyhodnocení identifikace po provedení frontalizace obličeje. Trénování a testování probíhá na snímcích všech sezení, kde mají osoby neutrální výraz. Trénovací sadě pak náleží 200 prvních identit datasetu a testovací zbylých 137 identit. Těchto 137 identit má celkem 72 000 snímků. Dále definují galerii snímků, vůči které se budou snímky z testovací sady porovnávat a bude se pomocí nich vyhodnocovat úspěšnost identifikace. Galerie snímků obsahuje 137 frontálních snímků nascvícené pomocí světla definovaného jako ID7, což je světlo, které míří přímo na obličej.

Labeled Faces in the Wild
Tento dataset [20] je standardním vyhodnocovacím protokolem, který je v oblasti rozpoznávání tváří od roku 2007. Obsahuje několik druhů verifikačních protokolů, každý s jinými podmínkami (například ohledně využití anotací, dat mimo dataset, apod.). Všechny protokoly využívají *10-fold cross-validation*. V datasetu se nachází **13 233** snímků celkem **5 749** osob, kde celkem **1 680** má více jak jeden snímek. Obličeje ve snímcích z datasetu jsou většinou frontální a přesto, že jsou snímky získané za reálných podmínek, nejsou snímky příliš obtížné (co se natočení, osvětlení, zakrytí a jiných vlastností týče).

Cross Pose Labeled Faces in the Wild
Dataset *CPLFW* [62] navazuje na dataset *LFW*, který upravuje dvojice tak, aby vždy jedna z dvojic měla vyšší úhel natočení obličeje. Tento dataset obsahuje náročnější snímky (kvůli póze) a oproti původnímu *LFW* datasetu se přesnost na tomto datasetu snížila o $8\% - 20\%$. Jak z názvu vyplývá, tento dataset se zaměřuje na obtížnější pózy. Dataset obsahuje 6000 dvojic, kde 3 000 dvojic jsou snímky, na kterých se nachází stejné identitou a 3 000 kde jsou rozdílné identity.

[1] https://www.imdb.com/

Kapitola 5

Návrh řešení a implementace

Práce je rozdělena na 2 části, první část se zabývá návrhem a natrénováním neuronové sítě pro rozpoznávání tváří. Cílem je ji natrénovat tak, aby dokázala extrahovat příznakový vektor, který je co nejvíce diskriminativní a co nejvíce reprezentoval identitu ze vstupního snímku.

Druhou částí je navržení a natrénování neuronové sítě, která generuje syntetické obličeje rotací z libovolné pózy do libovolné pózy druhé. Hlavní důraz je kladen na to, aby výsledné obličeje vypadaly fotorealisticky a byla zachována identita původního obličeje. A díky tomu aby byla možnost použít tyto syntetizované snímky pro rozpoznávání.

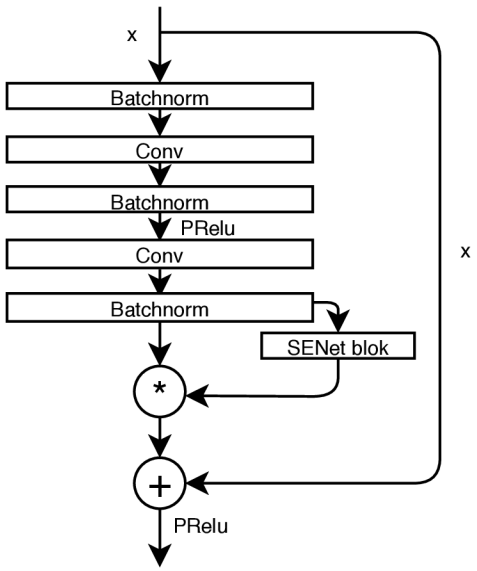
5.1 Síť pro rozpoznávání tváří

Jak je v sekci 2.3 shrnuto, tzv. páteřních sítí pro extrakci příznakového vektoru existuje několik. Pro experimentování jsem zvolil sítě *SeResNet-IR18*, *SeResNet-IR34*, *SeResNet-IR50*. Z názvu těchto sítí vyplývá, že se jedná o *Squueze-and-excitation* rozšíření Reizudálních sítí. Dále tyto sítě mají rozšířený základní reziduální blok *IRBasicBlock*, který je popsaný v sekci 2.3. Tento blok navíc obsahuje aktivační vrstvu jako poslední vrstvu při výstupu z tohoto bloku, schéma bloku je na obrázku 5.1.

Pro trénování byl zvolen dataset *VGGFace2* [6], který údajně obsahuje maximálně 4% chybných dat a zároveň obsahuje velké množství snímků na osobu. Z toho důvodu je vhodný pro použití. Vstupní snímky byly zarovnány pomocí podobnostní transformace (popsané v sekci 2.2) s využitím lokací bodů očí, nosu a koutků úst. Tyto body byly zarovnány na pozice $[38, 51]$, $[73, 51]$, $[56, 71]$, $[41, 92]$, $[70, 92]$, které odpovídají postupně levému oku, pravému oku, nosu, levému koutku a pravému koutku při velikosti snímku 112×112. Ukázka zarovnání je na obrázku 2.3. Vstupem do sítí jsou tedy barevné, zarovnané snímky obličejů o velikosti 112×112. Výstupem je 512 dimenzionální příznakový vektor.

Úspěšnosti sítí jsou vyhodnoceny na datasetech *LFW* [20] a *CPLFW* [62]. Z toho důvodu bylo nutné odstranit z trénovacího datasetu všechny identity, které se nachází v některém z datasetů pro vyhodnocení. Celkem bylo odstraněno 593 identit a počet identit a snímků byl redukován na postupně 8 538 a 3 062 209. Pro získání podobnosti (resp. skóre) 2 vektorů je použita cosinová podobnost.

Jako chybové funkce jsem zvolil *AM-Softmax* [51] a *ArcFace* [8], obě jsou popsané v sekci 2.4. Při trénování je přidána další plně propojená vrstva (která normalizuje své váhy a vstupní vektor podle L_2 normy) s 8 538 neurony a dále se trénuje jako klasifikační problém s využitím uvedených chybových funkcí.

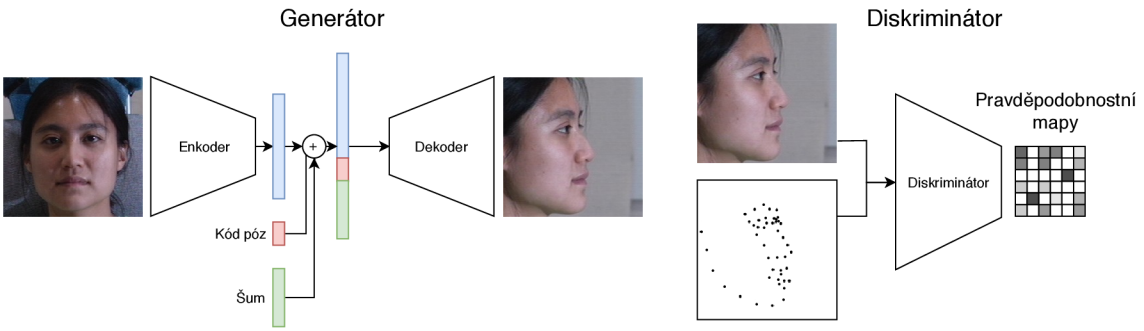


Obrázek 5.1: Architektura bloku IRBasicBlock

5.2 Síť pro sytézu obličejů

Další částí je síť pro syntézu obličejů. Cílem je, aby se síť dokázala naučit rotovat (resp. syntetizovat) obličej z libovolné pózy do libovolné pózy druhé a zároveň byla zachována identita a bylo možné výstupy této sítě použít pro účely rozpoznávání tváří. Pro tuto úlohu jsem navrhnul architekturu podobnou těm v *CAPG* [19] a *TP-GAN* [21] a pojmenoval ji *PCGAN*. Architektura je znázorněna na obrázku 5.2.

Základem této architektury je **generátor**, který provádí rotaci, **diskriminátor**, který se učí správně klasifikovat reálné a syntetické snímky a **síť pro reprezentaci identity z obličeje**, která je použita pro zachování identity při trénování. Trénování probíhá pomocí několika chybových funkcí, které jsou postupně popsány níže.



Obrázek 5.2: Základní architektura navržené sítě *PCGAN*. Vstupem pro generátor je zdrojový snímek a kód pózy, výstupem je syntetický snímek, který se snaží diskriminátor správně, za pomoci vykreslených význačných bodů, v obličeji klasifikovat za falešný nebo pravý (pomocí pravděpodobnostních map).

Úhly		zdrojový kód póz												cílový kód póz											
		-90	-75	-60	-45	-30	-15	0	15	30	45	60	75	90	-90	-75	-60	-45	-30	-15	0	15	30	45	60	75	90
90 °	0 °	0	0	0	0	0	0	0	0	0	0	0	0	**1**	0	0	0	0	0	0	**1**	0	0	0	0	0	0
0 °	-45 °	0	0	0	0	0	0	**1**	0	0	0	0	0	0	0	0	0	**1**	0	0	0	0	0	0	0	0	0
0 °	10 °	0	0	0	0	0	0	**1**	0	0	0	0	0	0	0	0	0	0	0	0	**0.33**	**0.67**	0	0	0	0	0
33 °	-28 °	0	0	0	0	0	0	0	0	**0.8**	**0.2**	0	0	0	0	0	0	0	**0.87**	**0.13**	0	0	0	0	0	0	0

Tabulka 5.1: Ukázka formátu kódů póz, který vstupuje spolu se vstupním snímkem do generátoru pro syntézu obličeje.

Řízení rotace

Pro usnadnění práce generátoru mu je poskytována informace jakou rotaci má provádět. Rotace je řízena pomocí vektoru definující zdrojový a cílový kód pózy, který vstupuje společně se vstupním snímkem do generátoru. Vzhledem k tomu, že byl pro trénování použit dataset *Multi-PIE* [14] a v tomto datasetu jsou nasnímány obličeje v 13 různých pózách, je tento kód vektorem o velikosti 26 reálných číslech. Prvních 13 hodnot udává zdrojovou pózu a dalších 13 hodnot cílovou pózu. Indexy odpovídají úhlům, které jsou seřazeny od $-90°$ do $90°$ s krokem $15°$ a nastavením hodnoty 1 na 2 indexy nějakého z úhlů je definována zdrojová a cílová póza obličeje. Tato reprezentace je zvolena do budoucna, aby bylo možné pomocí interpolace provádět natočení z libovolných úhlů do libovolných úhlů. Zdrojové a cílové úhly lze tedy definovat nastavením hodnoty 1 na pozici bázového úhlu nebo za pomocí váženého součtu dvou sousedních bázových úhlů na základě rovnice

$$a = k_1 b_i + k_2 b_{i+1}, \tag{5.1}$$

kde a je výsledný úhel, $k_1, k_2 \in \langle 0, 1 \rangle$ jsou koeficienty, které budou na pozicích i, $i+1$ a platí, že $k_1 = 1 - k_2$. Dále b_i a b_{i+1} jsou bázové úhly na pozicích i a $i+1$, přičemž $i \in \langle 0, 11 \rangle$. Ukázky některých vybraných kódů jsou v tabulce 5.1. Tento kód je konkatenován k vektoru, který je výstupem z enkóderové části generátoru, společně s gaussovských šumem, jak je dále popsáno

Generátor

Shrnutí celé architektury generátoru, včetně specifikace parametrů a vstupů, je v tabulce 5.2. Generátor se skládá z enkóderové části redukující vstupní snímek až na 512 dimenzionální vektor a dekóderové části. Enkóderová část je složena z 5 reziduálních základních bloků (*BasicBlock*) popsaných v sekci 2.3. Tyto vrstvy mají konvoluční filtry o velikosti postupně $7, 5, 3, 3, 3$ a aktivační funkci jsem zvolil *LeakyRElu* [30]. Všechny vrstvy, kromě první, snižují prostorovou velikost. Postupně je velikost snížena ze vstupní velikosti 128×128 na velikost 8×8. Se snížením prostorové velikosti se zvyšuje počet kanálů. Poté následuje plně propojená vrstva s 512 neurony jejíž výstup pokračuje do *Maxout* vrstvy [12], která produkuje výsledný 256 dimenzionální vektor. Jak je výše naznačeno, k tomuto vektoru se konkatenuje kód póz a také 100 dimenzionální vektor standardního gaussovského šumu, stejně jako v článcích *TP-GAN* [21] a *CAPG* [19].

Takto konkatenovaný 382 dimenzionální vektor je vstupem do dekodéru, kde vstupuje do další plně propojené vrstvy, která má 4096 neuronů. Dekóder je složen ze 3 částí. Prvním krokem je převedení vektoru o velikosti 4096 na mapy $8 \times 8 \times 64$. Tyto mapy jsou poté vstupem do první dekonvoluční části. První dekonvoluce má jádro 4×4 a provádí se s krokem 4. Poté následují 2 dekonvoluce, každá s jádrem 2×2 a krokem 2. Po dekonvoluci následuje vždy reziduální blok. Druhou částí je posloupnost 4 dekonvolucí s jádrem 2×2 a kroken 2, která ke svým vstupům konkatenuje výstupy z bloků enkóderu z odpovídající úrovně. V posledních 2 dekonvolucích ke vstupu navíc konkatenuje podvzorkované vstupní

Název vrstvy	Vstup	Filtry/Krok/Typ	Vstupní velikost	Výstupní velikost
conv0_0	I^S	$7 \times 7/1$/K	$128 \times 128 \times 3$	$128 \times 128 \times 64$
conv0_1	conv0_0	$5 \times 5/2$/K	$128 \times 128 \times 64$	$64 \times 64 \times 64$
conv0_2	conv0_1	$3 \times 3/2$/K	$64 \times 64 \times 64$	$32 \times 32 \times 128$
conv0_3	conv0_2	$3 \times 3/2$/K	$32 \times 32 \times 128$	$16 \times 16 \times 256$
conv0_4	conv0_3	$3 \times 3/2$/K	$16 \times 16 \times 256$	$8 \times 8 \times 512$
fc1	conv0_4	-/-/F	33768	512
maxout	fc1	-/2/M	512	256
fc2	maxout, pose_code, noise	-/-/F	382	4096
dc0_1	fc2	$4 \times 4/4$/D	$8 \times 8 \times 64$	$32 \times 32 \times 32$
dc0_2	dc0_1	$2 \times 2/2$/D	$32 \times 32 \times 32$	$64 \times 64 \times 16$
dc0_3	dc0_2	$2 \times 2/2$/D	$64 \times 64 \times 16$	$128 \times 128 \times 8$
dc1_1	fc2, conv0_4	$2 \times 2/2$/D	$8 \times 8 \times 576$	$16 \times 16 \times 512$
dc1_2	dc1_1, conv0_3	$2 \times 2/2$/D	$16 \times 16 \times 768$	$32 \times 32 \times 256$
dc1_3	dc1_2, conv0_2, I^S_{32}, dc0_1	$2 \times 2/2$/D	$32 \times 32 \times 419$	$64 \times 64 \times 128$
dc1_4	dc1_3, conv0_1, I^S_{64}, dc0_2	$2 \times 2/2$/D	$64 \times 64 \times 211$	$128 \times 128 \times 64$
conv1_0	dc1_2	$3 \times 3/1$/K	$32 \times 32 \times 256$	$32 \times 32 \times 3$
conv1_1	dc1_3	$3 \times 3/1$/K	$64 \times 64 \times 128$	$64 \times 64 \times 3$
conv1_2	dc1_4, conv0_0, I^S, dc0_3	$5 \times 5/1$/K	$128 \times 128 \times 139$	$128 \times 128 \times 64$
conv1_3	conv1_2	$3 \times 3/1$/K	$128 \times 128 \times 64$	$128 \times 128 \times 32$
conv1_4	conv1_3	$3 \times 3/1$/K	$128 \times 128 \times 32$	$128 \times 128 \times 3$

Tabulka 5.2: Architektura generátorové sítě pro syntézu obličejů.

Název vrstvy	Vstup	Filtry/Krok	Vstupní velikost	Výstupní velikost
conv1	I^D, I^L	$5 \times 5/1$	$128 \times 128 \times 4$	$128 \times 128 \times 64$
conv2	conv1	$3 \times 3/2$	$128 \times 128 \times 64$	$64 \times 64 \times 128$
conv3	conv2	$3 \times 3/2$	$64 \times 64 \times 128$	$32 \times 32 \times 256$
conv4	conv3	$3 \times 3/2$	$32 \times 32 \times 256$	$16 \times 16 \times 512$
conv5	conv4	$3 \times 3/2$	$16 \times 16 \times 512$	$8 \times 8 \times 512$
conv6	conv5	$3 \times 3/1$	$8 \times 8 \times 512$	$6 \times 6 \times 1$

Tabulka 5.3: Architektura diskriminátoru

snímky o rozměrech 32×32 a 64×64 a výstupy posledních 2 bloků z první části dekodéru. V poslední části se již neprovádí dekonvoluce, ale pomocí reziduálního bloku se vytváří 2 podvzorkované výsledné snímky (opět s rozměry 32×32 a 64×64). Výsledný snímek vzniká z konkatenovaných výstupů 2 posledních dekonvolučních vrstev a vstupního snímku, které jsou vstupem do 3 reziduálních bloků. Výstupem z generátoru jsou 3 snímky o rozměrech 32×32, 64×64 a 128×128.

Diskriminátor

Základní architektura diskriminátoru je popsána i s parametry v tabulce 5.3. Vstupem do diskriminátoru je vygenerovaný snímek či reálný cílový snímek konkatenovaný s vykreslenými význačnými 68 body obličeje. Bloky se skládají z konvoluční vrstvy, instanční normalizační vrstvy a LeakyRElu. Bloky `conv1`, `conv2`, `conv3` obsahují 2 zřetězené bloky za sebou, ostatní obsahují pouze jeden blok. Poslední blok má jako aktivaci *Sigmoidu*, a výstupem je 6×6 pravděpodobnostní mapa.

Data

Pro trénování byl použit *Multi-PIE* dataset, který je popsán v sekci 4.1. Bohužel tento dataset neobsahuje ani souřadnice obličeje ani význačné body v obličeji. Z toho důvodu

jsem provedl anotaci tohoto datasetu pomocí *FAN* [5]. Jak je popsáno výše, dataaset byl snímán v několika sezeních z několika pohledů a s různým osvětlením. Podstatné zde je, že při změně osvětlení byl obličej stále na stejném místě a docházelo k zanedbatelným pohybům (přivřené oči kvůli změně osvětlení v průběhu snímání, apod.). Některé snímky jsou kvůli špatnému osvětlení velice tmavé a pro detektor je velice obtížné správně lokalizovat body v obličeji a někdy dokonce i detekovat obličej. Díky výše uvedené vlastnosti ale lze detekovat obličej postupně od nejlepších úhlů osvětlení až po nejhorší, dokud se nepovede obličej detekovat. Pokud se podaří obličej detekovat a v něm i význačné body, lze tyto body použít i pro zbývající snímky s jiným osvětlením, které ještě nebyly zpracovány. Tímto způsobem byl zpracován celý dataset. Nutno podotknout, že u extrémních výrazů, kdy měly subjekty extrémně otevřenou pusu, se často nepodařilo body detekovat správně (snímky s těmito výrazy proto nebyly použity pro trénování na celém datasetu).

Pro syntézu z libovolné polohy do libovolné polohy existuje celkem 156 kombinací zdrojových a cílových póz. Proto jsou kombinace zredukovány z libovolné nefrontální pózy do frontální pózy a z frontální pózy do libovolné jiné pózy. Pokud za sebe poskládáme tyto dvě transformace, obličej může být rotován do libovolné pózy s mezikrokem frontalizace. K trénování je tedy potřeba pouze 24 kombinací z původních 156 kombinací. Dvojice snímků se generují tak, aby snímky pocházely ze stejného sezení a bylo i stejné osvětlení. Pozice kamer, ze kterých byly snímky nasnímané, se volí podle zadané zdrojové a cílové pózy.

Zdrojové a cílové snímky o velikosti 128×128 jsou zarovnány pomocí afinní transformace, kde zdrojové body jsou střed očí a střed úst. Cílové body jsou na souřadnicích $[63, 58]$ střed očí a $[63, 105]$ střed úst. Tento přístup je zvolen z toho důvodu, že je vhodné mít obličej vždy pod stejným vertikálním úhlem (v tomto případě pomyslně kolmý podle přímky definované uvedenými dvěma body). Díky tomu síť nebude muset být příliš invariantní vzhledem k této rotaci. A síť může očekávat tyto oblasti vždy na stejném (či podobném místě).

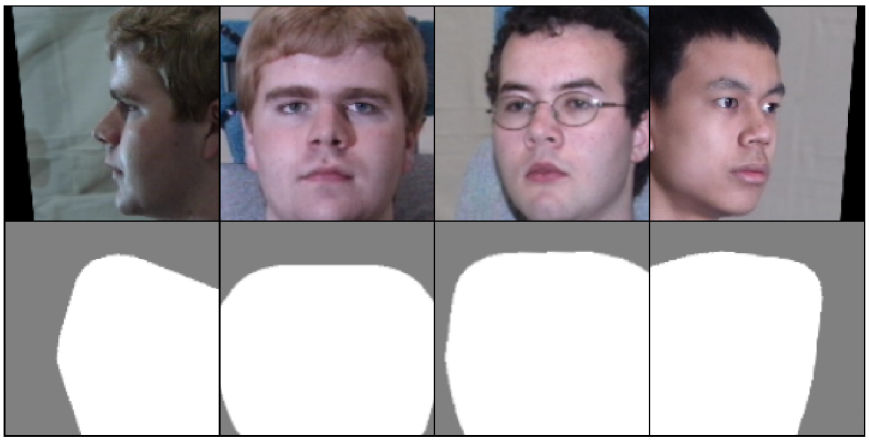
Chybové funkce

Stejně jako v uvedených pracích [19, 21] jsem použil kombinaci několika chybových funkcí. Nejzákladnější chybovou funkcí je pixelová chyba v podobě váhované L_1 chybové funkce. V situaci, kdy je obličej natočený pod úhlem větším jako 45° je skoro polovina snímku složena pouze z nepodstatného pozadí. Z toho důvodu jsem využil 68 význačných bodů, které jsem získal z anotací snímků, k získání konvexní obálky těchto bodů. Konvexní obálka je poté ještě rozšířena o 20% šířky obličeje z toho důvodu, aby byly konvexní obálkou pokryty všechny podstatné oblasti i s ohledem na nepřesnou detekci bodů a jiné objekty náležící k obličeji. Pomocí této konvexní obálky je vytvořena maska stejných rozměrů jako snímek s obličejem. Hodnoty, které jsou svou lokací uvnitř této obálky mají hodnotu 1 a ostatní body hodnotu 0.5. Tato maska je následně normalizována tak, aby její průměrná hodnota byla 1, což je prováděno podle rovnice

$$M_n = \frac{M}{M_{avg}}, \tag{5.2}$$

kde M je původní maska, M_{avg} je průměrná hodnota napříč celou maskou a M_n je normalizovaná maska. Ukázka vytvoření masky z příslušných bodů v obličeji je na obrázku 5.3. Tato normalizovaná maska je poté použita v L_1 pixelové chybě, která má předpis

$$L_{pixel} = \frac{1}{W \times H} \sum_{x=1}^{W} \sum_{y=1}^{H} \left| I_{x,y}^{pred} - I_{x,y}^{gt} \right| \times M_{x,y}, \tag{5.3}$$



Obrázek 5.3: Ukázka masek obličejů pro váhovanou L_1 chybovou funkci.

kde W a H jsou šířka a výška vstupních a výstupních snímků, x, y jsou indexy udávající pozici pixelu, $I^{pred}_{x,y}$ je predikovaná hodnota pixelu, $I^{gt}_{x,y}$ je cílová hodnota pixelu a $M_{x,y}$ je váha pixelu definovaný normalizovanou maskou. Chybová funkce je taktéž aplikována na podvzorkované mezivýsledky sítě o rozměrech 64×64 a 32×32, kdy cílový snímek je taktéž stejně podvzorkovaný a stejně tak i maska. Díky této vícerozměrné chybové funkci se síť trénuje rychleji, ale snímky jsou kvůli tomu příliš vyhlazené.

Pro zachování identity (což je kriticky důležité, jak je výše zmíněno) je použita funkce pro zachování identity, na kterou se dá hledět i jako na *perceptuální* chybovou funkci [23]. Pro reprezentaci identity ze snímku obličeje, jsem vybral síť *LightCNN* [55] stejně jako v předchozích pracích [21, 19]. Použita je konkrétně verze *LightCNN-29-V2*. Chybová funkce má předpis

$$L_{ip} = \left\| F_p(I^T) - F_p(G(I^S, P)) \right\|_F + \left\| F_{fc}(I^T) - F_{fc}(G(I^S, P)) \right\|_2, \tag{5.4}$$

kde I^T je cílový snímek, I^S je zdrojový snímek, P je výše popsaný kód póz, F_p je výstup do poslední pooling vrsty a F_{fc} je výstup předposlední plně propojené vrstvy sítě *LightCNN*, $\|\|_F$ a $\|\|_2$ je Frobennova a L_2 norma.

Dále je použita adversariální chybová funkce [11], jejíž cílem je přiblížit generované snímky distribuci datové sady, kterou má k dispozici diskriminátor. Tato funkce je použita za účelem doplnění částí, které nejsou na zdrojovém snímku viditelné a celkově pro generování snímků, které jsou více realistické. Generátor a diskriminátor se snaží optimalizovat následující min-max problém

$$\min_{\theta_G} \max_{\theta_D} E_{I^T \sim P(I^T)} log D_{\theta_D}(I^T, L^T) + E_{I^G \sim P(I^G)} log(1 - D_{\theta_D}(G_{\theta_D}(I^S, P), L^T)), \tag{5.5}$$

kde I^T je cílový snímek, I^G je vygenerovaný snímek generátorem, I^S je zdrojový snímek, L^T jsou vykreslený význačné body obličeje v cílovém snímku, P je kód póz, D_{θ_D} je diskriminátor a G_{θ_G} je generátor.

Z toho plyne, že chybová funkce pro generátor je

$$L_{adv} = -log(D_{\theta_D}(G_{\theta_D}(I^S, P), L^T)), \tag{5.6}$$

kde I^T je cílový snímek, I^G je vygenerovaný snímek generátorem, I^S je zdrojový snímek, L^T jsou vykreslené význačné body obličeje v cílovém snímku, P je kód póz, D_{θ_D} je diskriminátor a G_{θ_G} je generátor.

Poslední použitou chybovou funkcí je *Total variation regularization* [23] chybová funkce, která se snaží zredukovat šum ve výstupních snímcích. Její předpis je

$$L_{tv} = \sum_{i=1}^{W-1} \sum_{j=1}^{H-1} \left| I_{i,j}^{pred} - I_{i+1,j}^{pred} \right| + \left| I_{i,j}^{pred} - I_{i,j+1}^{pred} \right|, \tag{5.7}$$

Celková chybová funkce je váhový součet jednotlivých funkcí, což dává předpis

$$\min_{\theta_G} \max_{\theta_D} L = \lambda_1 L_{pixel} + \lambda_2 L_{adv} + \lambda_3 L_{ip} + \lambda_4 L_{tv}, \tag{5.8}$$

kde L_{pixel}, L_{adv}, L_{ip}, L_{tv} jsou výče popsané chybové funkce a $\lambda_1 - \lambda_4$ jsou hyperparametry, které jsou popsané dále v sekci 6.2.

Podobně jako v popsaných pracích [21, 19] byla síť *LightCNN* dotrénována na originálních snímcích datasetu *Multi-PIE* [14].

5.3 Implementace

Práce je implementována v jazyce *Python*[1] verze 3.6.7. Pro práci s neuronovými sítěmi je využit framework *PyTorch*[2]. Trénováno bylo na strojích výpočetního centra *MetaCentrum*[3], konkrétně byly experimenty prováděny na grafické kartě Nvidia GeForce GTX 1080TI (11GB).

[1] https://www.python.org/
[2] https://pytorch.org/
[3] https://metavo.metacentrum.cz/cs/

Kapitola 6

Experimenty a vyhodnocení

V rámci této práce bylo experimentováno s několika sítěmi a chybovými funkcemi pro rozpoznávání tváří. Výsledkem jsou poznatky o parametrech, které se osvědčily při trénování těchto sítí.

Druhou rozsáhlejší částí tvoří experimenty se sítěmi pro syntézu obličejů. Zde jsou zmíněny základní problémy tohoto trénování, zhodnocení přínosu jednotlivých chybových funkcí a shrnutí variací architektur. Závěrem jsou zde zhodnoceny kvalitativní a kvantitativní výsledky, kterých bylo dosaženo.

6.1 Síť pro rozpoznávání tváří

V rámci experimentů byly trénovány 3 různě velké sítě pro rozpoznávání tváří. Sítě byly založeny na *SEResnet* [18] s modifikacemi způsobu získání příznakového vektoru a strukturou základního bloku (což je popsáno v sekci 5.1). Konkrétně byly trénovány sítě *SENet-18E-IR*, *SENet-34E-IR*, *SENet-50E-IR*. Pro jejich trénování byly využity chybové funkce *Am-Softmax* [51] a *ArcFace* [8].

Dataset *VGGFace2* [6] použitý pro trénování, vyčištěný od identit, které jsou obsaženy v datasetech pro vyhodnocení, byl rozdělen v poměru 19 : 1 na trénovací a validační sadu. Pro trénování bylo využito 2 752 155 snímků a pro validaci 310 054 snímků.

Vstupní snímky byly normalizovány do rozsahu $\langle -1, 1 \rangle$. Trénovací data byly augmentovány pomocí náhodných úprav jasu, kontrastu, saturace a odstínu. Dále náhodné rotace v rozsahu $\langle -10°, 10° \rangle$, náhodné translace v rozsahu $\langle -5\%, 5\% \rangle$ v obou směrech. Náhodně se měnilo měřítko v rozsahu $\langle 0.9, 1.1 \rangle$ v relativním poměru a náhodné zrcadlové otočení v poměru 1 : 1.

Trénovací parametry byly použité podle původních článků [51, 8]. Optimalizátor byl zvolen *SGD* s počáteční *learning rate* = 0.1. *momentum* = 0.9 a *weight decay* = $5e^{-4}$. V původních článcích používali trénovací mini-batche o velikosti 512. Já jsem kvůli paměťovým omezením použil mini-batche o velikost 256 pro síť *SENet-18E-IR* a 128 pro síť *SENet-18E-IR*. *Learning rate* byla snižována po 100, 180 a 260 tisících iteracích 10×.

Nejprve bylo experimentováno s chybovou funkcí *AM-Softmax* [51]. Parametry byly ponechány jako v originálním článku a to na $m = 0.32$ a $s = 30$. U nejmenší sítě proběhlo trénování v pořádku a níže v tabulce 6.1 jsou vypsány výsledky. Nicméně u 2 větších sítí docházelo z počátku ke stagnaci. Po snížení *learning rate* byla stagnace odstraněna. Později bylo zjištěno, že by se chybová funkce měla měnit v závislosti na velikosti mini-batchi. Proto byla počáteční *learning rate* snížena u 2 větších sítí na 0.01 [13].

Síť/Metoda	LFW	CPLFW
Člověk-individuálně	97.27%	81.21%
Člověk-fůze	99.85%	85.24%
Center Loss [54]	98.75%	77.48%
SphereFace [29]	99.27%	81.40%
MS1MV2, R100/ArcFace [8]	99.87%	92.08%
SENet-18E-IR/AM-Softmax	99.07%	81.81%
SENet-34E-IR/AM-Softmax	99.15%	82.58%
SENet-50E-IR/AM-Softmax	99.27%	83.21%
SENet-18E-IR/ArcFace($m = 0.3, s = 30$)	99.27%	87.15%
SENet-34E-IR/ArcFace($m = 0.3, s = 30$)	99.33%	87.73%
SENet-50E-IR/ArcFace($m = 0.3, s = 30$)	**99.38%**	**88.08%**

Tabulka 6.1: Porovnání výsledků natrénovaných sítí s dalšími metodami.

Dále bylo experimentováno s *ArcFace* [8] chybovou funkcí. Stejně tak zde byly použity originální hodnoty parametrů $m = 0.5$ a $s = 64$. Problém nastal po 150 tisících iteracích, kdy se přestala snižovat chyba. Však ani po snížení *learning rate* se chyba nesnižovala. Po několika experimentech bylo zjištěno, že ideální hodnoty parametrů jsou $m = 0.3$ a $s = 30$.

Porovnání výsledků všech sítí na uvedených datasetech jsou v tabulce 6.1. Nejlepších výsledků dosáhla síť, která byla trénována pomocí *ArcFace*. Zajímavé je pozorovat, jak se zlepšila přesnost na datasetu *CPLFW* [62] pouze použitím jiné chybové funkce. Zároveň bylo na tomto datasetu dosaženo lepších výsledků, než kterých dosahují lidé.

6.2 Syntéza obličejů

Trénování *GAN* sítí je obecně obtížný problém [38], při kterém může dojít k několika problémům. Sítě jsou citlivé na hyperparametry a průběh trénování obecně není příliš stabilní. Z toho důvodu bylo experimentováno inkrementálně jak s chybovými funkcemi, tak s komplikovanější architekturou, aby se projevily přínosy jednotlivých modifikací.

Přínosy složitějších architektur a více chybových funkcí
Základní architekturou byla architektura podobná *U-Net* [37] architektuře. Trénováno bylo pouze pomocí pixelové chyby několik tisíc iterací. Na snímku 6.1 jsou znázorněny výsledky 2 sítí, kdy jedna využívá a druhá nevyužívá propojení vstupních snímků do koncové části dekodéru. Jak je ze snímku zřejmé, využití původního snímku dává síti možnost využít přímo lokální informace obličeje a použít je do výstupu sítě. Je to logické, protože při natočení obličeje do malého úhlu je plocha neviditelných oblastí minimální. Proto při frontalizaci dojde ke transformaci obličeje více do středu. Analogicky při provádění rotace z frontální polohy například do 90° se využije pouze jedna polovina obličeje a její lokální informace.

Aplikací L_1 chybové funkce na originální velikost a k tomu na 2 podvzorkované snímky se zásadně urychlí trénování a síť se lépe naučí globální strukturu obličeje. Podobně je na tom použití chybové funkce pro zachování identity. S jejím použitím se dokáže lépe síť naučit využívat lokálních informací ze zdrojových snímků. Zároveň po vyhodnocení podle protokolu *Setting 2* na datasetu *Multi-PIE* [14] bylo dosaženo lepších výsledků při identifikaci než pouze s L_1 chybovou funkcí.

Obrázek 6.1: Ukázka urychlení trénování pomocí propojení vstupních snímků do dekodérové části. První řádek je vstupní snímek (natočení hlavy 30°). Druhý řádek je výstup ze sítě po 10 tisících iterací, která tato propojení nemá. Třetí řádek je výstup ze sítě po 10 tisících iterací, která tato propojení má. Poslední řádek jsou cílové snímky.

Obrázek 6.2: Ukázka jak se síť učí využívat lokálních informací v obličeji a tím více zachovává typické rysy identit. První sloupec je zdrojový obličej a poslední sloupec je cílový sloupec. 2-4 sloupec jsou snímky vždy po 10 tisících iterací trénování. Řádky mají zdrojový obličej natočený postupně do 15°, 45°, 75°, 90°.

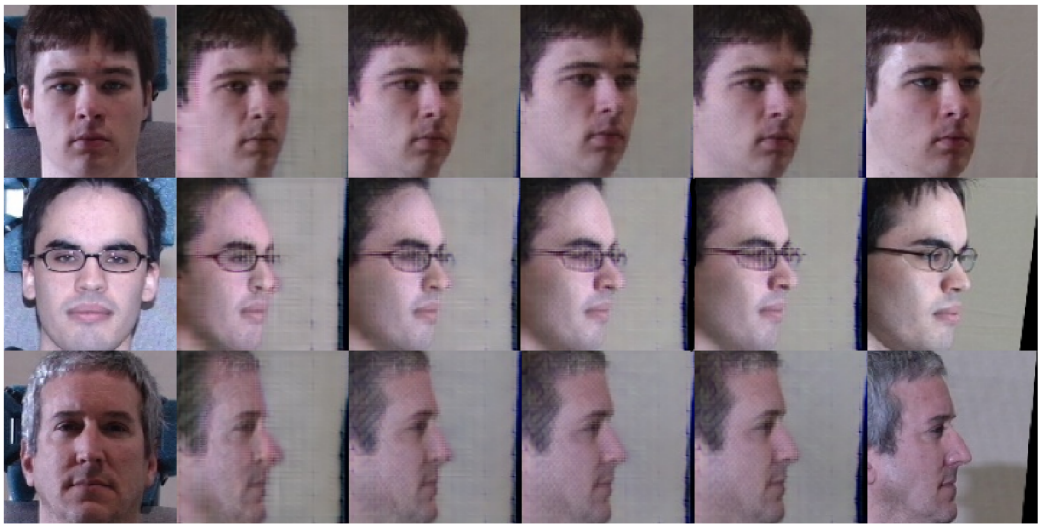
Problém nicméně nastává při více natočených obličejích, kdy síť produkuje poměrně rozmazané snímky. Rysy, které jsou ale viditelné jsou ve snímcích zachovány. Toto je vidět v obrázku 6.2, kde je u mírně natočených snímků (první 2 řádky) vidět, jak se síť naučila využít lokálních informací ze vstupních snímků. U snímků s více natočenými obličeji dochází k uvedenému rozmazání. Taktéž je na tomto obrázku vidět, že se síť hned v počátku nejdříve naučí v podstatě oříznout vstupní frontální obličej podle zadané cílové pózy a poté dochází k transformaci částí na svá místa.

Zatím byly na vyobrazeny pouze případy, kdy se prováděly frontalizace obličeje. V případě rotace z frontální polohy do profilové polohy je vždy využita polovina obličeje. Na obrázku 6.3 je ukázána rotace několika obličejů z frontální polohy.

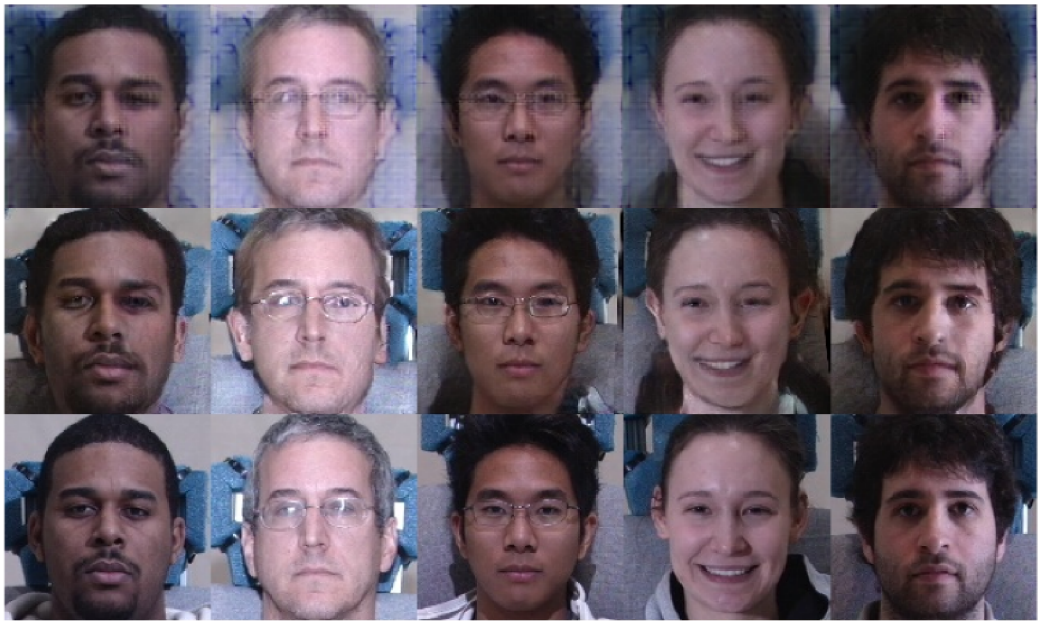
Fotorealističnost dokáže vyřešit adversariální chyba. Jak je výše popsáno byl navrhnut generátor a k němu diskriminátor. Do diskriminátoru jsou jako trénovací vzorky reálné snímky spolu s jejich vykreslenými 68 význačnými body v obličeji. Díky tomu se diskriminátor dokáže naučit nejenom rozlišovat reálné obličeje od nereálných, ale zároveň i to, že je obličej natočený do odpovídajícího úhlu. Po aplikaci této funkce se vhodně doplňují části obličeje, které v původní póze nejsou viditelné a také se odstraňuje přílišné vyhlazení obličejů, což je vidět na snímku 6.4. Taktéž se doplňuje pozadí obličeje, kde je zřejmé, že při frontalizaci se generátor naučil generovat opěradla židle, na kterých subjekty v datasetu sedí.

Parametry trénování

Hledání optimálních parametrů je časově a tím i výpočetně náročný problém. Proto zde



Obrázek 6.3: Ukázka rotace z frontální pózy do profilu. První a poslední sloupce zdrojové snímky a cílové snímky. Mezi nimi zleva doprava výstupy sítě po 10 tisících iterací.



Obrázek 6.4: Ukázka efektu adversariální chybové funkce. První řádek je výstup ze sítě po 20 tisících iterací s použitím pouze pixelové chybové funkce a funkce pro zachování identity. Druhý řádek je výstup po přidání adversariální chybové funkce po 5 tisících iteracích. Třetí řádek jsou cílové snímky.

zmíním konkrétní konfigurace trénování, které se mi osvědčily. Jako optimalizátor byl využit *Adam* [25], který si umí nejlépe poradit s hyperparametry, jako v tomto případě. *Learning rate* byla nastavena na 0.0003 a byla skokově snižována, *weight decay* byl nastaven na $1e^{-4}$ a *beta* parametry byly nastaveny na $(0.5, 0.999)$. Velikost *Mini-batchí* byla 32, ideální by byla vyšší, nicméně se zde naráží na paměťové limity. Vstupní a výstupní snímky byly normalizovány tak, aby rozsah jejich hodnot byl v intervalu $\langle -1, 1 \rangle$ a rozměry těchto snímků byly 128×128. Experimentováno bylo taktéž s výstupním typem diskriminátoru, ukázalo se ale, že pravděpodobnostní mapy fungují v tomto konkrétním případě lépe než skalární výstup. Nejspíš díky tomu, že se mapy mohou soustředit na konkrétní oblasti.

Co se týče postupu trénování, jak je naznačeno výše, nejprve se síť trénovala pouze za pomoci L_1 chybové funkce a funkce pro zachování identity. Bylo trénováno 100 tisíc iterací s *learning rate* $3e^{-4}$, dalších 100 tisíc iterací s $1e^{-4}$ a finálních 60 tisíc iterací s $lr = 1e^{-5}$. Ostatní parametry byly stejné jako výše. Celková chybová funkce měla předpis

$$L_{pix,ip} = \lambda_1 L_{pix} + \lambda_2 L_{ip}, \tag{6.1}$$

kde parametry mají hodnoty $\lambda_1 = 15$ a $\lambda_2 = 0.08$. Dále byly přidány adversariální a total variation regularization chybové funkce. Dohromady tvoří předpis

$$L = \lambda_1 L_{pix} + \lambda_2 L_{ip} + \lambda_3 L_{adv} + \lambda_4 L_{tv}, \tag{6.2}$$

kde L_x jsou již popsané chybové funkce a $\lambda_3 = 0.06$ a $\lambda_4 = 1e^{-5}$ jsou hyperparametry určující váhy chybových funkcí. Nutno poznamenat, že chybové funkce nevyjadřují váhu nebo poměr jednotlivých přínosů chybových funkcí, ale závisí na hodnotách, kterých chybové funkce nabývají.

U adversariální chybové funkce se osvědčilo využití metody *One sided label smoothing* [38]. Ta spočívá v tom, že se u hodnot, které určují o jaký snímek se jedná (generovaný/falešný nebo cílový/z datové sady) nahrazuje 1 za hodnoty blízké 1 při trénování diskriminátoru s reálnými snímky. Cílem je, aby si diskriminátor nebyl příliš jistý a tím se více stabilizovalo trénování.

Vyhodnocení vizuálních výsledků

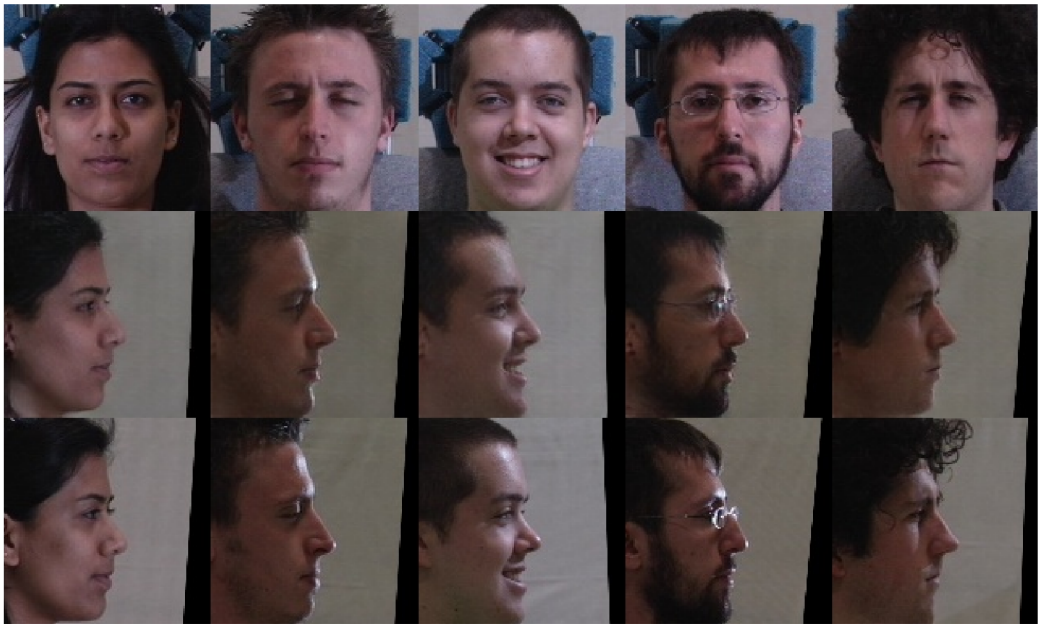
Síť se naučila zachovávat detaily v obličejích jako jsou různé textury v obličeji, vousy, brýle, tvar vlasů, apod.. Na obrázku 6.5 je to na několika příkladech vyznačeno. Taktéž je vidět, že při frontalizaci síť dokáže doplnit informace z neviditelné části obličeje tak, že je poměrně symetrická k té druhé části.

Zároveň díky adversariálnímu trénování se síť dokázala naučit přidávat i odlesky a osvětlení, které jsou způsobené různými úhly osvětlení při snímání datasetu (To lze vidět na obrázku 6.6). Dokonce někdy zachovává původní směr osvětlení nebo někdy přidává náhodné odlesky. Vzhledem k tomu, že dataset obsahuje subjekty zachycené na stejných sedadlech, síť se naučila doplňovat i toto pozadí, což ale může být pro použití mimo data tohoto formátu problémem pro použití.

Vyhodnocení použitelnosti pro rozpoznávání tváří

Pro zjištění, jestli je možné použít generátor pro účely rozpoznávání pomocí generování (angl. *Recognition via generation*) byla síť vyhodnocena pomocí *Setting 2* [59] protokolu datasetu *Multi-PIE* [14]. Cílem je vyhodnotit přínos frontalizace oproti využití původních snímků pro rozpoznávání. Pro extrakci příznakového vektoru byla využita síť *LightCNN_29_V2*, ale pouze natrénovaná na datasetu *MS-Celeb-1M*, tzn. **nebyla** dotréno-

Obrázek 6.5: Ukázka zachování detailů ze zdrojových snímků jako jsou vousy, detail vlasů, pihy, akné, brýle, tvar obočí. První řádek obsahuje zdrojové snímky, druhý řádek syntetické snímky a třetí řádek cílové snímky.



Obrázek 6.6: Ukázka zachování správného osvětlení a odlesků a zároveň ukázka rotace obličeje z frontální pózy do profilové pózy. První řádek je vstupní snímek, druhý řádek je výstup produkovaný sítí a třetí řádek je cílový snímek.

Metoda	±90°	±75°	±60°	±45°	±30°	±15°	Průměr
LightCNN [55]	5.51 %	24.18 %	62.09 %	92.13 %	97.38 %	98.59 %	63.31 %
TP-GAN [21]	64.64 %	77.43 %	87.72 %	95.38 %	98.06 %	98.68 %	86.99 %
CAPG-GAN [19]	**66.05 %**	**83.05 %**	**90.63 %**	**97.33 %**	**99.56 %**	**99.82 %**	**89.41 %**
LightCNN_29_V2 [55]	37.78 %	**83.25 %**	**96.73 %**	**99.49 %**	**99.88 %**	**99.98 %**	86.19 %
PCGAN	**58.51 %**	75.65 %	90.18 %	97.54 %	99.66 %	99.97 %	**86.91 %**

Tabulka 6.2: Identifikace na Multi-PIE

vána na datasetu *Multi-PIE*. Poté se postupovalo podle protokolu *Setting 2* [59], popsaný v sekci 4.2

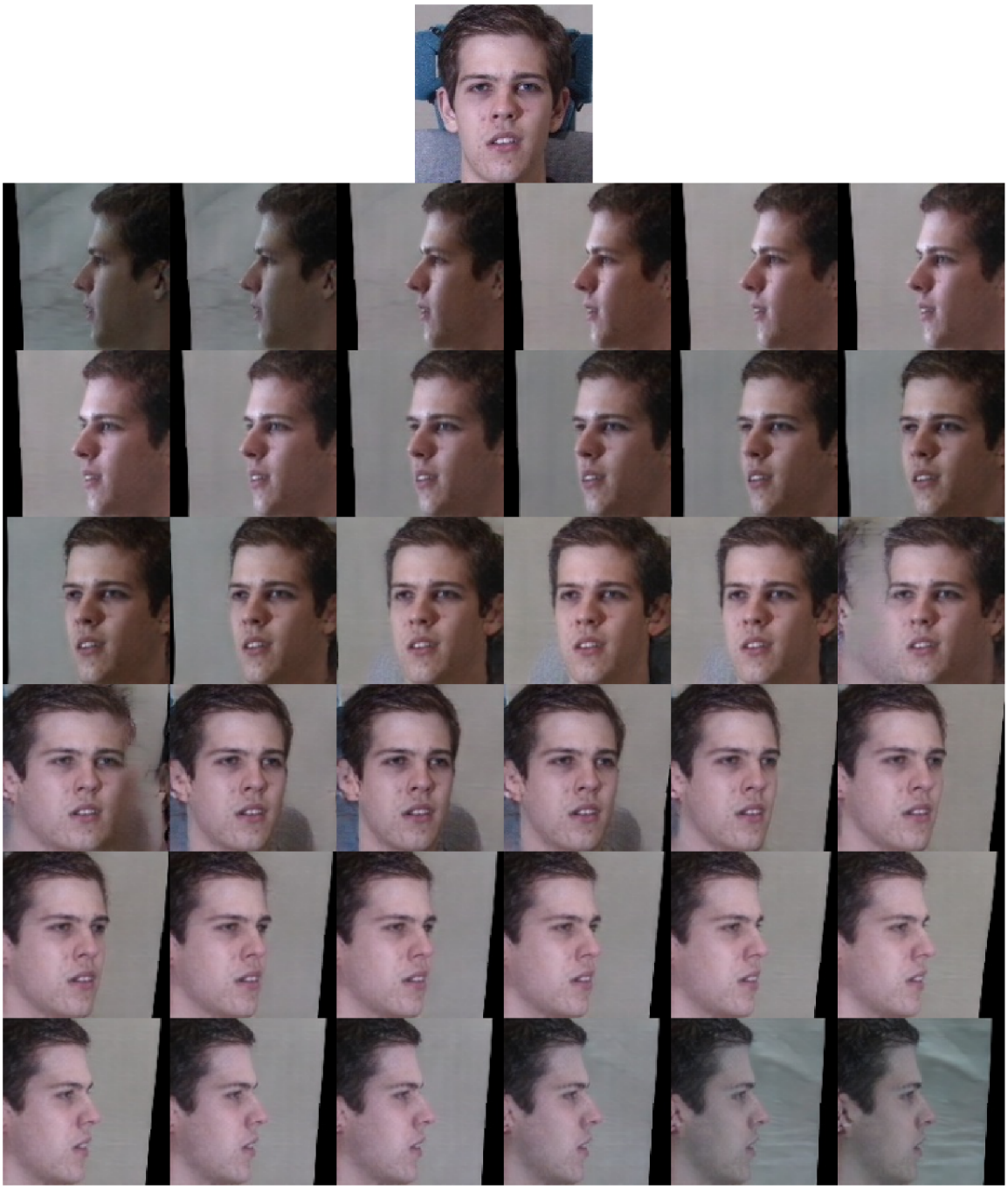
Jak je v tabulce 6.2 patrné, úspěšnost identifikace po frontalizaci se výrazně zlepšila u snímků, kde byl obličej natočený do ±90°. U ostatních snímků byla úspěšnost identifikace lehce horší, než při použití originálních snímků. Po zprůměrování výsledků napříč všemi úhly vychází, že výsledná úspěšnost identifikace je nepatrně lepší, než při použití původních snímků. Důvodem, že nedošlo ke zlepšení jako v předchozích pracích, může být, že nebyly nalezeny nejoptimálnější parametry pro trénování. Zároveň byla síť *LightCNN* trénována na *MS-Celeb-1M* s jiným zarovnáním než je použito pro syntézu.

Díky využití kódu pózy pomocí vektoru lze jednoduše interpolovat hodnoty, které jsou mimo základní úhly natočení. Lze tak provádět rotaci z jakéhokoliv úhlu do jakéhokoliv jiného. Kromě rozsahu $\langle -10°, 10° \rangle$ (s výjimkou 0°) tato interpolace funguje obstojně. Na obrázku 6.7 je vykreslena ukázka různě natočeného obličeje po 5°. Díky tomu, že rotace z frontálních poloh produkuje téměř identické snímky jako cílové snímky (což lze vidět na obrázku 6.6), má síť nutné předpoklady pro použití pro rozšíření dat.

Vyhodnocení použitelnosti na snímcích mimo *Multi-PIE* dataset

Jak je ukázáno na snímku 6.6, rotace obličeje z frontální polohy pomocí interpolace funguje velice dobře na datech, které byly snímány ve stejném formátu jako trénovací data. Proto jsem vyzkoušel, jak se síť bude chovat, pokud vstupní snímek obličeje bude mimo dataset *Multi-PIE*. Vybral jsem snímek *Colina Farrella* z datasetu *LFW*, a provedl stejnou rotaci obličeje. Výsledek je vidět na obrázku 6.8, ze kterého plyne, že navrhnutá síť má potenciál pro použití pro augmentaci dat.

Pro další kvalitativní vyhodnocení úspěšnosti frontalizace na snímcích byly použity ukázky z předchozích prací, ve kterých autoři frontalizovali 4 osoby z datasetu *LFW* a porovnávali s výsledky ostatních. Z obrázku 6.9 lze tvrdit, že zachování identity u sítě *PCGAN* funguje lépe než u metody *CPGAN*. U prvních 2 snímků se viditelně mění identita snímku. Zároveň dochází k přílišnému vyhlazení a nejsou použity oblasti ze zdrojového obličeje.



Obrázek 6.7: Využití interpolace kódu póz pro rotaci obličeje z testovací sady *Multi-PIE* datasetu. Horní obličej je vstupní snímek. Další snímky jsou syntetizované obličeje s natočením od $\langle -90, 90 \rangle$ s krokem 5°.

Obrázek 6.8: Využití interpolace kódu póz pro rotaci obličeje z datasetu *LFW*. Horní obličej je vstupní snímek. Další snímky jsou syntetizované obličeje s natočením od $\langle -90, 90 \rangle$ s krokem 5°.

(a) *LFW* (b) *PCGAN* (c) *CPGAN* [19] (d) *TP-GAN* [21] (e) *HPEN* [56]

Obrázek 6.9: Kvalitativní vyhodnocení na osobách z datasetu *LFW* a porovnání s jinými metodami.

Kapitola 7

Závěr

V této práci je zhodnocen kompletní postup rozpoznávání tváří, včetně toho co předchází samotnému rozponávání, jaké se využívají architektury a jakým způsobem se sítě trénují. Jsou zde zmíněny a popsány aktuálně nejnovější a nejlepší přístupy, díky kterým se dosahuje *state-of-the-art* výsledkům. Taktéž jsou zde zmíněny aktuální problémy, které se snaží výzkumníci odstranit.

Dále jsou zde zmíněny možnosti syntetických úprav záběrů obličeje. Podrobněji jsou popsány metody využívající *GAN* sítí.

V rámci práce byly navrženy a natrénovány 3 sítě pro rozpoznávání tváří, které byly trénovány pomocí 2 různých chybových funkcí. Výsledkem jsou poznatky o přesných parametrech trénování pro tuto konkrétní konfiguraci. Největší síť *SEResNet-50E-IR* dosáhla přesnosti **99.38%** na datasetu *LFW* a **88.08%** na datasetu *CPLFW* [62], čímž dosáhla lepších výsledků než dosáhli lidé. Prakticky bylo dokázáno, že s lepší chybovou funkcí lze dosáhnout lepších výsledků na datasetech, které mají vyšší variabilitu póz.

Další částí byl návrh sítě *PCGAN* pro rotaci obličejů a její trénování. Výstupem této části jsou poznatky o parametrech trénování, o přínosech jednotlivých chybových funkcí a architektury. Síť dokázala zlepšit úspěšnost identifikace ač pouze o pár desítek %. Díky způsobu vložení informace o pózách do sítě je možné generovat záběry pod libovolným úhlem. Taktéž se ukázalo, že síť není omezená funkčností pouze na datasetu, na kterém se trénovala, ale dokáže produkovat dobré výsledky i na jiných datasetech. Výsledky rotace z frontální pózy jsou velmi pozitivní a síť je tedy potenciálně možné použít pro augmentaci trénovací sady.

V budoucnu by bylo vhodné se více zaměřit na zlepšení přesnosti identifikace. Jednou z možností by mohlo být využití části vektoru produkovaného z enkodérové části generátoru pro trénování rozpoznávání. Zde by se mohla využít některá nová chybová funkce. Tím by byl kladen větší důraz na zachování identifikace a mohlo by to vést k lepším výsledkům.

Zajímavé by také bylo rozšířit touto sítí datovou sadu pomocí této sítě a vyhodnotit, jestli síť pro rozpoznávání tváří, která se bude na těchto datech trénovat, dosáhne lepších výsledků.

Taktéž by bylo vhodné vyzkoušet jinou větší síť pro extrakci příznakového vektoru, i přesto, že tím dojde ke zpomalení trénování. Případně vyzkoušet jinou adversariální chybovou funkci, např. *Wasserstein* [2].

Literatura

[1] Ahonen, T.; Hadid, A.; Pietikainen, M.: Face Description with Local Binary Patterns: Application to Face Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, ročník 28, č. 12, Dec 2006: s. 2037–2041, ISSN 0162-8828, doi:10.1109/TPAMI.2006.244.

[2] Arjovsky, M.; Chintala, S.; Bottou, L.: Wasserstein GAN. 2017.

[3] Blanz, V.; Vetter, T.; aj.: A morphable model for the synthesis of 3D faces. In *Siggraph*, ročník 99, 1999, s. 187–194.

[4] Bulat, A.; Tzimiropoulos, G.: How Far are We from Solving the 2D & 3D Face Alignment Problem? (and a Dataset of 230,000 3D Facial Landmarks). *Proceedings of the IEEE International Conference on Computer Vision*, ročník 2017-Octob, 2017: s. 1021–1030, ISSN 15505499, doi:10.1109/ICCV.2017.116.

[5] Bulat, A.; Tzimiropoulos, G.: Super-FAN: Integrated facial landmark localization and super-resolution of real-world low resolution faces in arbitrary poses with GANs. , č. c, 2017, doi:10.1109/CVPR.2018.00019.

[6] Cao, Q.; Shen, L.; Xie, W.; aj.: VGGFace2: A dataset for recognising faces across pose and age. *Proceedings - 13th IEEE International Conference on Automatic Face and Gesture Recognition, FG 2018*, 2018: s. 67–74, doi:10.1109/FG.2018.00020.

[7] Deng, J.; Cheng, S.; Xue, N.; aj.: UV-GAN: Adversarial Facial UV Map Completion for Pose-invariant Face Recognition. *CoRR*, ročník abs/1712.04695, 2017.

[8] Deng, J.; Guo, J.; Xue, N.; aj.: ArcFace: Additive Angular Margin Loss for Deep Face Recognition. *CoRR*, 2018, ISSN 1365-2702, doi:10.1111/j.1365-2362.2004.01432.x.

[9] Ding, C.; Xu, C.; Tao, D.: Multi-Task Pose-Invariant Face Recognition. *IEEE Transactions on Image Processing*, ročník 24, č. 3, 2015: s. 980–993, ISSN 10577149, doi:10.1109/TIP.2015.2390959.

[10] Feng, Y.; Wu, F.; Shao, X.; aj.: Joint 3D Face Reconstruction and Dense Alignment with Position Map Regression Network. *CoRR*, ročník abs/1803.07835, 2018.

[11] Goodfellow, I. J.; Pouget-Abadie, J.; Mirza, M.; aj.: Generative Adversarial Networks. *Corrosion*, jun 2014.

[12] Goodfellow, I. J.; Warde-Farley, D.; Mirza, M.; aj.: Maxout Networks. 2013.

[13] Goyal, P.; Dollár, P.; Girshick, R.; aj.: Accurate, Large Minibatch SGD: Training ImageNet in 1 Hour. 2017.

[14] Gross, R.; Matthews, I.; Cohn, J.; aj.: Multi-PIE. *Image Vision Comput.*, ročník 28, č. 5, Květen 2010: s. 807–813, ISSN 0262-8856, doi:10.1016/j.imavis.2009.08.002.

[15] Guo, Y.; Zhang, L.; Hu, Y.; aj.: MS-Celeb-1M: A Dataset and Benchmark for Large Scale Face Recognition. 2016.

[16] Hadsell, R.; Chopra, S.; LeCun, Y.: Dimensionality reduction by learning an invariant mapping. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, ročník 2, 2006: s. 1735–1742, ISSN 10636919, doi:10.1109/CVPR.2006.100.

[17] He, K.; Zhang, X.; Ren, S.; aj.: Deep Residual Learning for Image Recognition. *CoRR*, ročník abs/1512.03385, 2015.

[18] Hu, J.; Shen, L.; Sun, G.: Squeeze-and-Excitation Networks. *CoRR*, ročník abs/1709.01507, 2017.

[19] Hu, Y.; Wu, X.; Yu, B.; aj.: Pose-Guided Photorealistic Face Rotation. *Cvpr*, 2018: s. 8398–8406, doi:10.1109/CVPR.2018.00876.

[20] Huang, G. B.; Ramesh, M.; Berg, T.; aj.: Labeled Faces in the Wild: A Database for Studying Face Recognition in Unconstrained Environments. Technická Zpráva 07-49, University of Massachusetts, Amherst, October 2007.

[21] Huang, R.; Zhang, S.; Li, T.; aj.: Beyond Face Rotation: Global and Local Perception GAN for Photorealistic and Identity Preserving Frontal View Synthesis. *Proceedings of the IEEE International Conference on Computer Vision*, ročník 2017-October, 2017: s. 2458–2467, ISSN 15505499, doi:10.1109/ICCV.2017.267.

[22] Ioffe, S.; Szegedy, C.: Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. *CoRR*, ročník abs/1502.03167, 2015.

[23] Johnson, J.; Alahi, A.; Li, F.: Perceptual Losses for Real-Time Style Transfer and Super-Resolution. *CoRR*, ročník abs/1603.08155, 2016.

[24] Karpathy, A.: Convolutional Neural Networks for Visual Recognition. [online], [cit. 2019-01-10].
URL http://cs231n.github.io/linear-classify/

[25] Kingma, D. P.; Ba, J.: Adam: A Method for Stochastic Optimization. 2014: s. 1–15.

[26] Krizhevsky, A.; Sutskever, I.; Hinton, G. E.: Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, 2012, s. 1097–1105.

[27] Liu, W.; Anguelov, D.; Erhan, D.; aj.: SSD: Single Shot MultiBox Detector. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, ročník 9905 LNCS, dec 2015: s. 21–37, ISSN 16113349, doi:10.1007/978-3-319-46448-0_2.

[28] Liu, W.; Wen, Y.; Yu, Z.; aj.: Large-Margin Softmax Loss for Convolutional Neural Networks. 2016: s. 507–516.

[29] Liu, W.; Wen, Y.; Yu, Z.; aj.: SphereFace: Deep Hypersphere Embedding for Face Recognition. *CoRR*, ročník abs/1704.08063, 2017.

[30] Maas, A. L.; Hannun, A. Y.; Ng, A. Y.: Rectifier nonlinearities improve neural network acoustic models. *Icml '13*, ročník 28, 2013: str. 6.

[31] Masi, I.; Tran, A. T.; Leksut, J. T.; aj.: Do We Really Need to Collect Millions of Faces for Effective Face Recognition? *CoRR*, ročník abs/1603.0, č. 1, 2016, ISSN 0302-9743, doi:10.1007/978-3-319-46448-0.

[32] Nair, V.; Hinton, G. E.: Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, 2010, s. 807–814.

[33] Newell, A.; Yang, K.; Deng, J.: Stacked Hourglass Networks for Human Pose Estimation. *CoRR*, ročník abs/1603.06937, 2016.

[34] Parkhi, O. M.; Vedaldi, A.; Zisserman, A.; aj.: Deep face recognition. In *BMVC*, ročník 1, 2015, str. 6.

[35] Ranjan, R.; Castillo, C. D.; Chellappa, R.: L2-constrained Softmax Loss for Discriminative Face Verification. *CoRR*, ročník abs/1703.09507, 2017.

[36] Ren, S.; He, K.; Girshick, R.; aj.: Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, ročník 39, č. 6, 2017: s. 1137–1149, ISSN 01628828, doi:10.1109/TPAMI.2016.2577031.

[37] Ronneberger, O.; Fischer, P.; Brox, T.: U-Net: Convolutional Networks for Biomedical Image Segmentation. *CoRR*, ročník abs/1505.04597, 2015.

[38] Salimans, T.; Goodfellow, I. J.; Zaremba, W.; aj.: Improved Techniques for Training GANs. *CoRR*, ročník abs/1606.03498, 2016.

[39] Sankaranarayanan, S.; Alavi, A.; Castillo, C. D.; aj.: Triplet probabilistic embedding for face verification and clustering. *2016 IEEE 8th International Conference on Biometrics Theory, Applications and Systems, BTAS 2016*, 2016, ISSN 1878-3562, doi:10.1109/BTAS.2016.7791205.

[40] Sankaranarayanan, S.; Alavi, A.; Chellappa, R.: Triplet Similarity Embedding for Face Verification. *CoRR*, ročník abs/1602.03418, 2016.

[41] Schroff, F.; Kalenichenko, D.; Philbin, J.: FaceNet: A unified embedding for face recognition and clustering. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, ročník 07-12-June, 2015: s. 815–823, ISSN 10636919, doi:10.1109/CVPR.2015.7298682.

[42] Sengupta, S.; Chen, J. C.; Castillo, C.; aj.: Frontal to profile face verification in the wild. *2016 IEEE Winter Conference on Applications of Computer Vision, WACV 2016*, 2016, ISSN 1095-9572, doi:10.1109/WACV.2016.7477558.

[43] Simonyan, K.; Zisserman, A.: Very Deep Convolutional Networks for Large-Scale Image Recognition. *International Conference on Learning Representations (ICRL)*, 2015: s. 1–14, ISSN 09505849, doi:10.1016/j.infsof.2008.09.005.

[44] Sun, Y.; Wang, X.; Tang, X.: Deep learning face representation from predicting 10,000 classes. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2014: s. 1891–1898, ISSN 10636919, doi:10.1109/CVPR.2014.244.

[45] Szegedy, C.; Liu, W.; Jia, Y.; aj.: Going deeper with convolutions. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, ročník 07-12-June, 2015: s. 1–9, ISSN 10636919, doi:10.1109/CVPR.2015.7298594.

[46] Taigman, Y.; Yang, M.; Ranzato, M.; aj.: DeepFace: Closing the gap to human-level performance in face verification. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2014: s. 1701–1708, ISSN 10636919, doi:10.1109/CVPR.2014.220.

[47] Tran, L.; Yin, X.; Liu, X.: Disentangled representation learning GAN for pose-invariant face recognition. *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, ročník 2017-Janua, 2017: s. 1283–1292, doi:10.1109/CVPR.2017.141.

[48] Trigueros, D. S.; Meng, L.; Hartnett, M.: Face Recognition: From Traditional to Deep Learning Methods. 2018.

[49] Viola, P.; Jones, M.: Rapid object detection using a boosted cascade of simple features. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, ročník 1, IEEE, 2001, s. I–I.

[50] Wang, F.; Chen, L.; Li, C.; aj.: The devil of face recognition is in the noise. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, ročník 11213 LNCS, 2018: s. 780–795, ISSN 16113349, doi:10.1007/978-3-030-01240-3_47.

[51] Wang, F.; Cheng, J.; Liu, W.; aj.: Additive Margin Softmax for Face Verification. *IEEE Signal Processing Letters*, ročník 25, č. 7, 2018: s. 926–930, ISSN 10709908, doi:10.1109/LSP.2018.2822810.

[52] Wang, F.; Xiang, X.; Cheng, J.; aj.: NormFace: L_2 Hypersphere Embedding for Face Verification. *CoRR*, ročník abs/1704.06369, 2017.

[53] Wang, H.; Wang, Y.; Zhou, Z.; aj.: CosFace: Large Margin Cosine Loss for Deep Face Recognition. *CoRR*, ročník abs/1801.09414, 2018.

[54] Wen, Y.; Zhang, K.; Li, Z.; aj.: A discriminative feature learning approach for deep face recognition. In *European Conference on Computer Vision*, Springer, 2016, s. 499–515.

[55] Wu, X.; He, R.; Sun, Z.: A Lightened CNN for Deep Face Representation. *CoRR*, ročník abs/1511.02683, 2015.

[56] Xiangyu Zhu; Lei, Z.; Junjie Yan; aj.: High-fidelity Pose and Expression Normalization for face recognition in the wild. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015, ISSN 1063-6919, s. 787–796, doi:10.1109/CVPR.2015.7298679.

[57] Yang, J.; Ren, P.; Chen, D.; aj.: Neural Aggregation Network for Video Face Recognition. *Arxiv*, 2016.

[58] Yi, D.; Lei, Z.; Liao, S.; aj.: Learning Face Representation from Scratch. *CoRR*, ročník abs/1411.7923, 2014.

[59] Yim, J.; Jung, H.; Yoo, B.; aj.: Rotating your face using multi-task deep neural network. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015: s. 676–684.

[60] Zhang, K.; Zhang, Z.; Li, Z.; aj.: Joint Face Detection and Alignment Using Multitask Cascaded Convolutional Networks. *IEEE Signal Processing Letters*, ročník 23, č. 10, 2016: s. 1499–1503, ISSN 10709908, doi:10.1109/LSP.2016.2603342.

[61] Zhang, S.; Zhu, X.; Lei, Z.; aj.: S3FD: Single Shot Scale-Invariant Face Detector. *Proceedings of the IEEE International Conference on Computer Vision*, ročník 2017-Octob, 2017: s. 192–201, ISSN 15505499, doi:10.1109/ICCV.2017.30.

[62] Zheng, T.; Deng, W.: Cross-pose LFW: A database for studying cross-pose face recognition in unconstrained environments. Technická Zpráva 18-01, Beijing University of Posts and Telecommunications, February 2018.

Příloha A

Obsah přiloženého paměťového média

data	obsahuje zarovnané snímky pro demonstraci
doc	obsahuje zdrojové dokumenty k práci a samotnou práci v PDF
models	obsahuje natrénované sítě
src	obsahuje zdrojové kódy
video	obsahuje video, které bylo součástí zadání
README.md	obsahuje podrobnější popis jednotlivých souborů na přiloženém médiu