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

VISION TRANSFORMERY PRO ROZPOZNÁVÁNÍ TVÁŘÍ

VISION TRANSFORMERS FOR FACIAL RECOGNITION

SEMESTRÁLNÍ PROJEKT
TERM PROJECT

AUTOR PRÁCE — **ŠIMON STRÝČEK**
AUTHOR

VEDOUCÍ PRÁCE — **Ing. JAKUB ŠPAŇHEL, Ph.D.**
SUPERVISOR

BRNO 2023

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

Zadání diplomové práce

154524

Ústav:	Ústav počítačové grafiky a multimédií (UPGM)
Student:	**Strýček Šimon, Bc.**
Program:	Informační technologie a umělá inteligence
Specializace:	Počítačové vidění
Název:	**Vision transformery pro rozpoznávání tváří**
Kategorie:	Zpracování obrazu
Akademický rok:	2023/24

Zadání:

1. Prostudujte základy zpracování obrazu. Zaměřte se zejména na problematiku neuronových sítí.
2. Prostudujte dostupné materiály na rozpoznávání tváří a vision transformers.
3. Zorientujte se v současných metodách na rozpoznávání tváří a architekturách vision transformers.
4. Vyberte vhodnou metodu a navrhněte systém pro rozpoznávání tváří s použitím vision transformers.
5. Experimentujte s vaší implementací a případně navrhněte vlastní modifikace metod.
6. Porovnejte dosažené výsledky s existujícími řešeními využívající vision transformers tak s čistě konvolučními neuronovými sítěmi. Diskutujte možnosti budoucího vývoje.
7. Vytvořte stručný plakát a video prezentující vaši bakalářskou práci, její cíle a výsledky.

Literatura:

- DENG, Jiankang, et al. Arcface: Additive angular margin loss for deep face recognition. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2019. p. 4690-4699.
- DOSOVITSKIY, Alexey, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.
- LIU, Ze, et al. Swin transformer: Hierarchical vision transformer using shifted windows. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021. p. 10012-10022.
- ZHONG, Yaoyao; DENG, Weihong. Face transformer for recognition. *arXiv preprint arXiv:2103.14803*, 2021.
- Dále dle pokynů vedoucího

Při obhajobě semestrální části projektu je požadováno:

- Splnění prvních tří bodů zadání
- Značně rozpracovaný čtvrtý bod zadání

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

Vedoucí práce:	**Špaňhel Jakub, Ing.**
Konzultant:	Ing. Jan Stratil
Vedoucí ústavu:	Černocký Jan, prof. Dr. Ing.
Datum zadání:	1.11.2023
Termín pro odevzdání:	17.5.2024
Datum schválení:	9.11.2023

Abstrakt

Tato práce se zabývá aplikací architektur neuronových sítí na bázi vision transformer (ViT) v oblasti rozpoznávání tváří. Práce se soustředí na průzkum existujících moderních ViT architektur. To zahrnuje experimenty s existujícími implementacemi, alternativními druhy dat a hledání optimálních parametrů pro trénink. Cílem této práce je prokázat potenciál vision transformerů konkurovat již dlouho dominujícím konvolučním neuronovým sítím právě v tomto oboru. Výstupem je analýza provedených experimentů, demonstrace kladů a záporů moderních architektur ViT a nalezení optimálních podmínek pro jejich využití v úlohách rozpoznávání tváří.

Abstract

This thesis focuses on applying vision transformer-based neural networks to face recognition related tasks. It focuses on exploring modern vision transformer (ViT) architectures, experimenting with alternative data, and finding the suitable parameters to train ViTs to compete with the already established dominance of convolutional neural networks in face recognition. The goal of this work was to show the suitability of vision-transformers for face recognition. The output of this work contains results of various experiments, demonstrations of benefits and drawbacks of some of the modern and popular ViTs, the definition of an optimal setup when wanting to employ vision transformers for facial recognition, and interesting observations from working with vision transformers.

Klíčová slova

rozpoznávání tváří, vision transformer, zpracování obrazu, neuronové sítě

Keywords

face recognition, vision transformer, image processing, neural networks

Citace

STRÝČEK, Šimon. *Vision transformery pro rozpoznávání tváří*. Brno, 2023. Semestrální projekt. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Jakub Špaňhel, Ph.D.

Vision transformery pro rozpoznávání tváří

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně pod vedením pana Ing. Jakuba Špaňhela Další informace mi poskytl také odborný konzultant, Ing. Jan Stratil. Uvedl jsem všechny literární prameny, publikace a další zdroje, ze kterých jsem čerpal.

.....
Šimon Strýček
16. května 2024

Poděkování

Rád bych poděkoval vedoucímu práce, panu Ing. Špaňhelovi, za vstřícný a aktivní přístup. Dále bych rád poděkoval za poskytnutí výpočetních prostředků díky kterým byla tato práce možná.

Obsah

Kapitola 1

Úvod

Se systémy pro rozpoznávání tváří se dnes můžeme setkat v širokém spektru oborů. Jejich uplatnění je možné najít jak v méně kritických aplikacích jako například v agregaci fotografií ve fotogalerii mobilního zařízení, tak i v bankovnictví, kde na správném fungování závisí někdy i celé jmění lidí. Ať se jedná o jakékoli využití, požadavky na přesnost a efektivitu řešení těchto problémů stále rostou. Dosavadní implementace pro stabilní systémy rozpoznávající obličeje zakládají na využití neuronových sítí. Konkrétně se jedná o konvoluční neuronové sítě, rozšířené právě v oblasti zpracování obrazu. Tato řešení už dlouhodobě dominují ve svém oboru a již několik let neznaly konkurenci. V posledních letech se ovšem objevil nový hráč v oblasti zpracování obrazu, konkrétně architektura *vision transformer*.

Od jejího prvního představení se naskytl nový vhled na zpracování tohoto typu dat neuronovými sítěmi. Doposud dominantním metodám postaveným na bázi konvolucí začal konkurovat tento koncept přejatý z oboru zpracování přirozeného jazyka. Novější publikace jako například model Swin transformer cílí na omezení hlavních nevýhod této implementace a staví tuto architekturu před již dlouho používané alternativy. Dnešní variace vision transformeru již prokazatelně předčily svého předchůdce hned v několika různých úlohách jako např. klasifikace obrazu, detekce objektů či sémantická segmentace. Přes tento úspěch stále přibývá nových druhů implementací této architektury s ještě lepšími výsledky a samotná architektura stále roste na popularitě. Klíčovou vlastností v porovnání s konvolučními sítěmi je jejich schopnost zachytávání informačních souvislostí na dlouhé vzdálenosti ve vstupní sekvenci dat. Tato výhoda prokazatelně pomáhá neuronové síti v porozumění globálním vzorům v obraze, a tak umožňuje dosahovat lepších výsledků na některých typech problémů.

Tato práce se tedy zabývá problematikou využití architektury vision transformer v kontextu rozpoznávání tváří. Jelikož tento způsob již úspěšně posunul latku ve vícero oblastech, je motivací prozkoumat benefity tohoto řešení i v tomto oboru. Hlavním cílem tedy bylo zprovoznění a odladění vhodné implementace vision transformeru pro takovouto úlohu a zjištění, zdali jsou její hlavní přednosti využitelné i zde. Řešení zahrnuje průzkum existujících implementací, výběr nejvhodnější z nich a provedení následných experimentů s cílem zlepšení výsledků na takovémto druhu úlohy. V následující kapitole lze nalézt podrobnější vhled do problematiky jak dané architektury, tak i samotného tématu rozpoznávání tváří. V dalších kapitolách budou poté představeny postupy pro výběr vhodné varianty implementace vision transformeru, podoba a výsledky průběžných testů i samotný výběr. Dále budou představeny provedené experimenty a jejich výsledky. Výstupem této práce je analýza těchto experimentů, definice ideálních podmínek pro využití vision transformerů a demonstrace zajímavých postřehů při jejich aplikaci na téma rozpoznávání obličejů.

Kapitola 2

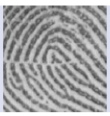
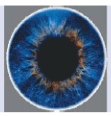
Biometrie a metody rozpoznávání obličeje

Využití biometrických dat je jednou z bezkonkurenčních metod pro robustní identifikaci osob. S jejich využitím se váže pestrá škála dnes využívaných implementací, přičemž s mnohými z nich se setkáváme každodenně. Mezi nejčastější případy může patřit například identifikace osob v galeriích mobilních zařízení, různé zabezpečovací systémy či zabezpečení mobilních zařízení. V dřívější době patřily mezi klíčové biometrické údaje spíše skeny otisků prstů či celých rukou. V dnešní době postupují do popředí naopak spíše metody využívající lidských obličejů. Ačkoli jsou lidé schopni rozpoznávat různé tváře bez většího úsilí, co se týče automatizovaného postupu, implementace není tak přímočará. To je rovněž jedním z důvodů proč se tyto metody staly populárními až v nedávné době. Tato kapitola se zabývá právě tímto tématem a nastiňuje implementační detaily takovýchto systémů.

2.1 Úloha rozpoznávání obličejů

Jak již bylo zmíněno, pro identifikaci osob se běžně používá celá řada různých biometrických údajů. Jak můžeme vidět také na obrázku 2.1, metody založené na rozpoznávání obličejů nepatří mezi nejpřesnější. Přesto má využití takovýchto biometrických údajů řadu jiných důležitých výhod. Jednou z významných předností je dostupnost tohoto druhu dat. Mnoho veřejných datových sad pro tuto úlohu pochází z fotografií známých osobností, které jsou ve velkém množství dostupné z různých internetových zdrojů. [14, 8, 2] Další podstatná výhoda tohoto přístupu pramení ze způsobu získávání dat uživatele k identifikaci. Pořizování fotografií či jiných záznamů obličejů na rozdíl od např. zpracovávání otisků prstů nevyžaduje fyzický kontakt, což činí tyto metody velmi intuitivní. Ačkoli tedy existují postupy, jež jsou obecně považovány za spolehlivější způsoby identifikace, lidské obličeje obsahují rovněž dostatečně unikátní parametry pro stabilní ověřování totožnosti. [15]

Obecný princip využívání biometrických dat spočívá v identifikaci jednotlivých klíčových oblastí pozorované části lidského těla. U rozpoznávání obličejů je princip totožný, jediným rozdílem je podoba vstupních dat. Kombinace specifik těchto klíčových bodů se poté používá k rozeznání rozdílů mezi záznamy různých lidí. Samotná podstata úlohy ve výsledku tedy spočívá v porovnávání těchto extrahovaných parametrů obličeje s databází známých záznamů.

BIOMETRIC	FINGERPRINT	FACE	HAND GEOMETRY	IRIS	VOICE
					
Barriers to universality	Worn ridges; hand or finger impairment	None	Hand impairment	Visual impairment	Speech impairment
Distinctiveness	High	Low	Medium	High	Low
Permanence	High	Medium	Medium	High	Low
Collectibility	Medium	High	High	Medium	Medium
Performance	High	Low	Medium	High	Low
Acceptability	Medium	High	Medium	Low	High
Potential for circumvention	Low	High	Medium	Low	High

Obrázek 2.1: Srovnání výhod využití jednotlivých biometrických údajů k identifikaci osob. Převzato z webu https://www.innovatrics.com/glossary/biometrics/.

2.2 Definice úlohy

Pojem rozpoznávání tváří může zastřešovat hned řadu rozdílných druhů úloh s různými podmínkami pro jejich evaluaci. Náročnost těchto úloh se může zásadně lišit, a proto je potřebné dobře definovat druh úlohy, kterým se tato práce zabývá.

Tato práce je převážně zaměřena na problém verifikace obličejů. To znamená, že hodnocení dosažených výsledků bude prováděno srovnáváním fotografií 1:1, kde se jedná o obrázek buďto totožného jedince či nikoli. Cílem evaluačních metrik poté bude srovnávání výstupů systému s binárními anotacemi, znázorňující zdali se jedná o dvě fotografie stejného jedince, nebo se jedná o dvě rozdílné osoby. Tento způsob hodnocení je jedním z nejčastějších a existuje řada dedikovaných veřejně dostupných datových sad, určených ke srovnávání systémů rozpoznávání tváří.

Další část specifikace úlohy zahrnuje pojmy *open-set* a *closed-set*, které definují vlastnost otevřenosti úlohy. Tento pojem představuje otevřenost množiny vstupních dat v ohledu na počet kategorií. V případě *closed-set* úloh je tento počet u vstupních dat fixní. Příkladem může být zabezpečení mobilních telefonů, kde se dá předpokládat, že vstupní data budou rozdělena pouze do dvou kategorií, a to v závislosti na tom zdali se jedná o obličej majitele zařízení či nikoli. Dalším příkladem může následně být identifikace zaměstnanců s neměnným počtem známých tváří. Tzv. *open-set* úlohy na druhou stranu řeší komplexnější problém. Počet tříd je předem neznámý, a tedy systémy pro rozpoznávání tváří musí v tomto případě umět rozlišit dříve neviděné osoby. Tento fakt vyžaduje odlišný způsob náhledu na zpracování. Používané systémy musí být v tomto případě schopné dostatečně dobře identifikovat klíčové vlastnosti obličeje tak, aby bylo za pomoci nich možné unikátně rozeznat rozdíly mezi jakýmikoli tvářemi, které jsou systému představeny. V následujících kapitolách bude na pojem „rozpoznávání tváří" nahlíženo jako na úlohu typu *open-set*.

2.3 Existující řešení úlohy rozpoznávání tváří

Dnes nejrozšířenější způsoby řešení této úlohy zakládají na využití konvolučních neuronových sítí. Ačkoli existují starší metody s mnoho podobnými implementačními prvky, metody založené na neuronových sítí významně dominují ve většině těchto aplikací. Takováto řešení dokáží s dostatečným množstvím trénovacích dat hravě předčit své předchůdce, což je také jedním z důvodů významného vzrůstu popularity využití právě obličejů v oblasti biometriky. Neuronové sítě v takovémto případě slouží k extrakci klíčových parametrů obličeje. Ačkoli tento postup není pravidlem, výstupem pro každou fotografii bývá nejčastěji vektor hodnot, který by měl být v ideálním případě unikátní pro každou osobu a zároveň by měl být invariantní vůči různým podobám jednoho konkrétního obličeje. Pro obličej totožného jedince by tedy výstupní vektor extrakčního algoritmu měl být stejný bez ohledu na pozici obličeje v obrázku, výraz člověka, věkový či časový rozestup mezi pořízenými záznamy a bez ohledu na další proměnné z prostředí reálného světa. Tyto výstupní vektory je následně možné porovnávat a dle vzájemné rozdílnosti lze zjišťovat příslušnost fotografie obličeje k dané osobě. Extrahované vektory se v těchto řešeních často porovnávají pomocí metrik vzdálenosti v prostoru výstupního kódování. Lze zde využít například eukleidovskou vzdálenost nebo v některých případech i kosinovou podobnost. Tato vzdálenost ve výsledku udává úroveň podobnosti daných obličejů. Mezi reprezentanty systémů pro rozpoznávání tváří s využitím konvolučních neuronových sítí poté můžou být například starší architektury *FaceNet* nebo *DeepFace*.

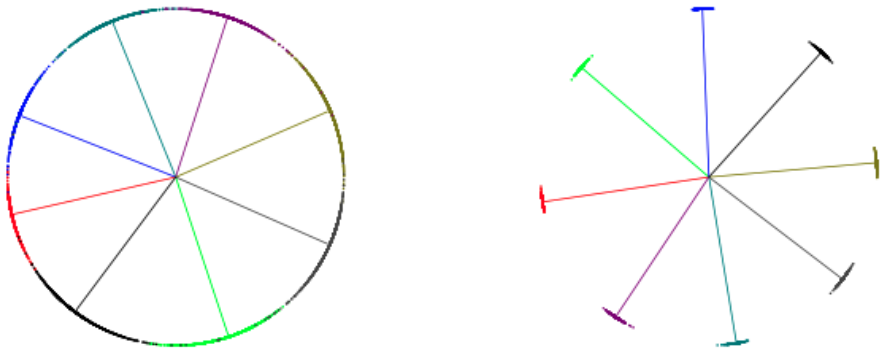
Jednou z populárních a zároveň veřejně dostupných implementací pro účely rozpoznávání tváří je metoda představená v článku *FaceNet: A Unified Embedding for Face Recognition and Clustering* [18]. Tento princip zakládá na využití hluboké neuronové sítě k transformaci vstupního obrázku do prostoru klíčových bodů tváře. Prostor těchto klíčových bodů má eukleidovskou charakteristiku a zakódované tváře lze v tomto prostoru srovnávat pomocí eukleidovské vzdálenosti. Trénování takovéto neuronové sítě se provádí za pomocí trojic obličejů. Trojice vždy obsahuje jednu vzorovou tvář, tvář shodného člověka z jiné fotografie a fotografii jiné osoby. Neuronová síť se pak učí predikovat body v prostoru klíčových charakteristik obličejů tak, aby fotografie shodných osob měly v tomto prostoru malou vzdálenost, zatímco fotografie neshodných tváří ji měly velkou.

Na druhou stranu, provedení z článku *DeepFace: Closing the Gap to Human-Level Performance in Face Verification* [21] vsází na tradičnější přístup k řešení této problematiky. Princip fungování navazuje na dřívější postupy, kde rozpoznávání tváří tvořilo vícero různých fází zpracování. Na rozdíl od metody představené v architektuře FaceNet, rozděluje implementace DeepFace problém na čtyři podúlohy: detekci obličeje, zarovnání obličeje do standardních rozměrů a proporcí, reprezentace obličeje dle klíčových bodů a nakonec klasifikace obličeje dle detekovaných charakteristik. Toto rozdělení pramení z dřívějších způsobů řešení a tato metoda pouze přidává prvky neuronových sítí do již známých postupů.

Jednou z dnes nejefektivnějších implementací je ta z článku *AdaFace: Quality Adaptive Margin for Face Recognition* [11]. Její úspěch tkví ve využití hluboké neuronové sítě ResNet, sloužící k extrakci klíčových parametrů tváře. Učení je prováděno pomocí speciálně upravené varianty *Softmax loss* funkce, podobné funkci *ArcFace loss* (která je popsána v kapitole 2.4). Kombinace této efektivní funkce, hluboké konvoluční sítě a kvalitní datové sady vede k velmi dobrým validačním výsledkům této implementace.

2.4 ArcFace loss funkce

Funkce *ArcFace loss* je jednou z velmi používaných loss funkcí pro účely trénování systémů na rozpoznávání tváří. [4] Obzvláště vhodná je pro účely rozpoznávání obličejů na tzv. *open-set* úlohách (viz. kapitola 2.2). Implementace vychází z dříve známé myšlenky funkce *SoftMax loss*. ArcFace loss uvažuje v prostoru hyperkoule a místo Eukleidovských vzdáleností hodnotí vzdálenosti mezi body v prostoru pomocí geodetik. Promítání výstupních vektorů do hyperkoule má své prokazatelné výhody v podobě lepší generalizace modelů, stability a citlivosti na drobné odchylky v obrazu jako např. jeho osvětlení. Velkým benefitem je zde normalizace do prostoru hyperkoule s daným poloměrem, která odstraňuje problémy s velkými vzdálenostmi mezi body v tomto prostoru. Klíčovou implementační vlastností této funkce je představení úhlového rozestupu m mezi třídami, který vynucuje větší vzájemnou separaci jednotlivých tříd. Demonstraci zmíněného rozestupu je možné pozorovat na obrázku 2.2. S její pomocí je možné model naučit lépe separovat predikce dat, a umožnit tak přesnější rozhodování o příslušnosti vstupu k dané třídě.



Obrázek 2.2: Obrázek ukazuje rozdíl v distribuci tříd v 2D prostoru s použitím SoftMax loss funkce (vlevo) a s využitím ArcFace loss funkce (vpravo). Lze zde pozorovat separaci tříd v prostoru dle fixního rozestupu m. Pro následnou identifikaci obličeje je poté jednodušší rozpoznat ke které třídě daný vzorek patří. Obrázek byl převzat z článku představující implementaci ArcFace. [4]

Funkce ArcFace loss určuje chybu dle úhlu mezi výstupním vektorem x_i a vzorovou hodnotou. Princip získávání této hodnoty spočívá ve vypočtení kosinové podobnosti se vzorovými reprezentanty tříd a následným převodem těchto hodnot za pomocí funkce arccos. Finální výpočet hodnoty ArcFace loss má poté následující podobu:

$$ArcFace(x) = -\frac{1}{N}\sum_{i=1}^{N} log\frac{e^{s\cdot(cos(\theta_{y_i}+m))}}{e^{s\cdot(cos(\theta_{y_i}+m))} + \sum_{j=1,j\neq y_i}^{n} e^{s\cdot cos\theta_j}}$$

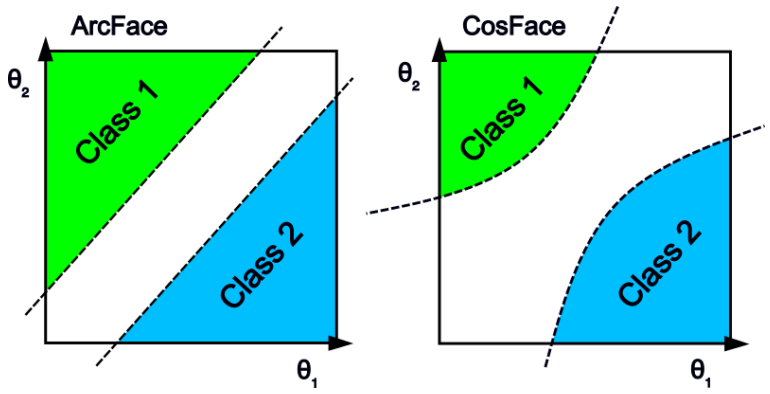
Hodnota θ_j je úhel mezi parametrem W_i a výstupním vektorem neuronové sítě x_i, příslušící třídě y_i. Poloměr hyperkoule je s a m je volený konstantní rozestup mezi jednotlivými třídami. Řádky matice vah W obsahují vzorová kódování jednotlivých tříd (v tomto případě obličejů/lidí). Jedná se o optimalizovatelnou vrstvu této loss funkce. Tato implementace vychází z faktu, že datové sady pro rozpoznávání obličejů neobsahují anotace v podobě výstupních vektorů, nýbrž pouze anotace jmen či případně index třídy/obličeje (viz. kapitola 2). Výhodou využití optimalizovatelných anotací je, že má model větší volnost při učení

reprezentace rysů obličeje. Skutečné hodnoty jeho výstupu pro danou fotografii v praktickém využití nehrají žádnou roli mimo jejich vlastností ve vztahu k výstupu pro jiné fotografie. Volba učitelných anotací je v tomto případě vhodným řešením.

2.5 CosFace loss funkce

Alternativou k hodnotící funkci ArcFace loss je její jednodušší varianta *CosFace loss* (Cosine Margin Loss), představená v článku [23]. Přestože dříve zmiňovaná ArcFace loss funkce vychází z konceptů tohoto předchůdce a snaží se je vylepšit, CosFace loss není z pohledu dosažených výsledků oproti jejímu nástupci výrazně pozadu. Z testů v článku [4] vyplývá, že se autorům této novější metody podařilo zlepšit výsledky pouze v řádu zlomků procent v metrice *Accuracy* na hodnotící datové sadě *LFW*. Znamená to tedy, že obě tyto metody představují dobrou volbu při učení neuronových sítí k robustnímu rozpoznávání obličejů.

Výrazným rozdílem mezi těmito dvěmi funkcemi je způsob vynucování rozestupu mezi shluky jednotlivých tříd. Zatímco ArcFace loss vynucuje třídní rozestupy až po transformaci výstupních vektorů modelu přímo ve sférickém prostoru hyperkoule, implementace CosFace loss využívá tento rozestup nad netransformovanými vektory. Tento rozdíl v rozestupech tříd je možné nejlépe pozorovat při zobrazení rozhodovacích oblastí v závislosti na úhlu mezi výstupním a vzorovým vektorem tříd. V zjednodušené reprezentaci na obrázku 2.3 můžeme pozorovat rovnoměrné rozestupy, kterých autoři ArcFace loss chtěli docílit. Účel rovnoměrného odsazení je lepší distribuce predikovaných vektorů v tomto prostoru a tím i zlepšená generalizace.



Obrázek 2.3: Obrázek znázorňuje rozdíly v podobě třídních rozestupů mezi funkcemi ArcFace loss a CosFace loss. Jednotlivé osy značí kosinovou podobnost mezi vzorovými vektory a výstupními vektory neuronové sítě.

Ačkoli se tedy zdá být dříve popsaná ArcFace loss výhodnější volbou, výběr kteréhokoli z těchto přístupů má dostatečný potenciál k natrénování robustního systému pro rozpoznávání tváří. Využití CosFace loss tedy nemusí být nutně vyloučeno, a jak se také ukázalo v následujících experimentech, v některých případech může být upřednostnění této volby loss funkce výrazně výhodnější.

2.6 Anotace datových sad pro rozpoznávání tváří

Při učení neuronových sítí v takovéto konfiguraci je důležité dbát na podobu výstupního vektoru. Tento vektor by měl unikátně identifikovat každou osobu. V praxi ovšem nebývají k dispozici anotace fotografií lidských obličejů v této zakódované podobě. Anotace v tomto případě mívají podobu přiřazení jmen či případně jiných identifikátorů daným obličejům. Pro účely učení modelů neuronových sítí je ovšem anotace v podobě těchto vektorů potřebná. Řešením tohoto problému často bývá zahrnutí tvorby této informace do mechanismu trénování. Řešením může být například využití existujících konceptů pro trénování neuronových sítí. Jednou z používaných metod je vložení učitelné vrstvy neuronové sítě, kde je možné jednotlivé vektory reprezentovat jako řádky matice vah. Tyto vektory je tedy doslova možné optimalizovat spolu s parametry neuronové sítě v průběhu trénování. Výběr hyperparametrů je ovšem v tomto případě klíčový. Druhou možností je vygenerování anotací pro danou datovou sadu buďto ručně, anebo s použitím předtrénovaného modelu neuronové sítě. V případě předem natrénovaných modelů lze využít například FaceNet nebo VGG-Face. Tato možnost ovšem zanáší limitaci do trénovacího procesu. Trénovaný model je v tomto případě omezen schopností extrakce druhého modelu, pomocí kterého byly vytvořeny tyto anotace. Zahazujeme zde tedy potenciál trénovaného modelu vytvářet si optimálnější reprezentaci obličejů.

Během trénování je snaha přimět model predikovat pro jednotlivé obrázky co nejpodobnější vektory vzhledem k těmto dynamickým anotacím. Přesto je během inference potřeba uvažovat nad těmito vzorovými vektory jako nad centroidy shluků různých variant totožných tváří. Obzvláště v reálném provozu je potřeba myslet na fakt, že i při fotografiích stejné osoby bude přítomna různorodost vzhledem k podmínkám za kterých byly fotografie pořízeny. Při navrhování loss funkcí, používaných při trénování, se tedy často zavádí mechanismy, které zaručují lepší separabilitu jednotlivých tříd.

Kapitola 3

Vision transformery pro rozpoznávání obličeje

Adaptace vision transformerů v oblasti rozpoznávání obličejů je poměrně přímočará. Ačkoli existujících implementací zaměřených konkrétně na téma rozpoznávání tváří není takové množství jako u jiných úloh, postačující jsou téměř jakékoli přístupy určené ke zpracování obrazu. Tato kapitola se tedy zabývá obecným shrnutím principu fungování architektury vision transformer a představuje několik moderních implementací tohoto druhu využitelných pro účely této práce. V následujících experimentech budou uvažovány právě tyto vybrané implementace.

3.1 Architektura vision transformer

Typ Architektury neuronových sítí, vision transformer, je vedle konvolučních sítí jednou z používaných architektur zaměřených na zpracování obrazu. Princip vychází z konceptu transformeru používaného primárně pro účely zpracování přirozeného jazyka. Jednou z předních výhod této architektury je její schopnost zachytávat souvislosti v dlouhých sekvencích dat. V porovnání s konvolučními sítěmi, které zpracovávají lokální oblasti, má tato architektura předpoklady k předčení svého konkurenta v oblasti zpracování obrazu. Tento předpoklad je rovněž motivací této práce, jejíž cílem je právě využití vision transformerů v oblasti rozpoznávání obličejů. Přes své výhody vision transformery dědí od svého předchůdce z oblasti zpracování přirozeného jazyka také své nevýhody. Neuronové sítě na bázi transformerů často mívají tendence k tomu být oproti ostatním architekturám výpočetně náročnější. Pro kompletní natrénování vyžadují velké množství dat a pozvolnější trénink. Na druhou stranu, stejně jako je tomu u zpracování přirozeného jazyka, je tato metoda velmi vhodná pro tzv. *fine-tuning*, kdy je model předem naučen na obecné datové sadě a je dále dotrénován pro konkrétní úlohy. Tato práce na tomto přístupu také z velké části zakládá.

3.1.1 Transformer vs. vision transformer

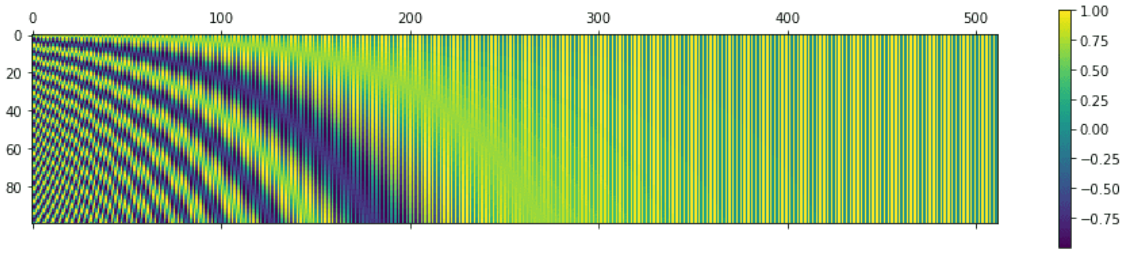
Vision transformery se od běžných transformerů liší hned v několika ohledech. Změny je možné vidět například v podobě vstupních dat, které v případě vision transformerů sestávají z obrázků (namísto psaného textu v případě klasických transformerů). Tento fakt vytváří nové komplikace, a také vyžaduje rozdílný pohled na předzpracování vstupních dat. Architektura obecných transformerů sestává ze dvou částí, a to části *encoder* a *deco-*

der. V kontextu vision transformerů pro se ovšem počítá s pouze jednou z nich, konkrétně s částí *encoder*. Zatímco *encoder* v originální implementaci slouží k zakódování vstupní informace pro účely transformace a dekódování částí *decoder*, v oblasti rozpoznávání obličejů je tato zakódovaná podoba dat vhodná bez nutnosti dalšího zpracování.

3.1.2 Předzpracování dat

Jak již bylo dříve zmíněno, vision transformery vychází z konceptu známého z oboru zpracování přirozeného jazyka. Tento fakt se mimo jiné promítá i do samotného zpracování obrazových dat v této architektuře neuronových sítí. Na obrazová data se zde obecně nahlíží jako na sekvenci tokenů, které je potřeba na základě obrazových dat vygenerovat. V tomto ohledu se různé varianty architektury vision transformer často liší. Ve většině případů princip spočívá v rozdělení obrázku na primitiva, nejčastěji podoblasti o velikosti v řádech pixelů. Z těchto podčástí se následně tvoří tokeny. Rozdíly v architekturách je možné často pozorovat ve způsobu transformace těchto primitiv na vstupní tokeny. Nejčastěji se setkáme s využitím lineární transformace, najdou se ale případy, kdy se ke generování tokenů využívá konvolučních vrstev. [29]

Nedílnou součástí přepracování dat je také zakódování poziční informace do vstupní sekvence. Obecný koncept transformerů je totiž pozičně invariantní. Bez této informace o pozici tokenu by tedy nezáleželo na jejich pořadí ve vstupních sekvencích a výstup modelu by byl totožný pro každou permutaci vstupních dat. Tento fakt by mohl vést ke snížení schopnosti učení neuronové sítě, a také k nestabilitě přesnosti výstupu. Zakódování informace o pozici tokenů ve vstupních sekvencích se v různých implementacích liší. Mezi nejčastěji používané způsoby patří metoda *sin-cos embedding* (demonstrováno na obrázku 3.1), která spočívá ve využití sinových a kosinových funkcí.



Obrázek 3.1: Na obrázku je možné vidět vizualizaci pozičního kódování za pomocí sinových a kosinových funkcí. Ve skutečnosti se jedná o matici s rozměrem LxE, kde L značí počet tokenů a E stanovenou délku výstupního vektoru. Pozice pro daný token se získává výběrem příslušného řádku dle indexu pozice tokenu ve vstupní sekvenci. Jedná se o často používaný způsob kódování informace o pozici.

3.1.3 Self-attention vrstva

Hlavním stavebním blokem obecných transformerů jsou tzv. *self attention* vrstvy. Tyto vrstvy umožňují modelovat vztahy na dlouhé vzdálenosti mezi daty vstupních sekvencí. Tento attention mechanismus je schopný pro každý token ve vstupní sekvenci vybrat relevantní informace ze všech těchto tokenů. Přestože je princip attention mechanismu dobře optimalizovaný, fakt že je potřeba pro každý token sekvence počítat relevanci všech ostatních tokenů představuje jednu z hlavních limitací tohoto přístupu. Počítání self attention

může tedy mít vysokou paměťovou a výpočetní náročnost s rostoucí délkou vstupu. Pro zpracování dlouhých sekvencí se proto využívají různé přístupy, které rozdílnými způsoby limitují počet zohledňovaných položek při zpracování aktuálního tokenu. [28] [5]

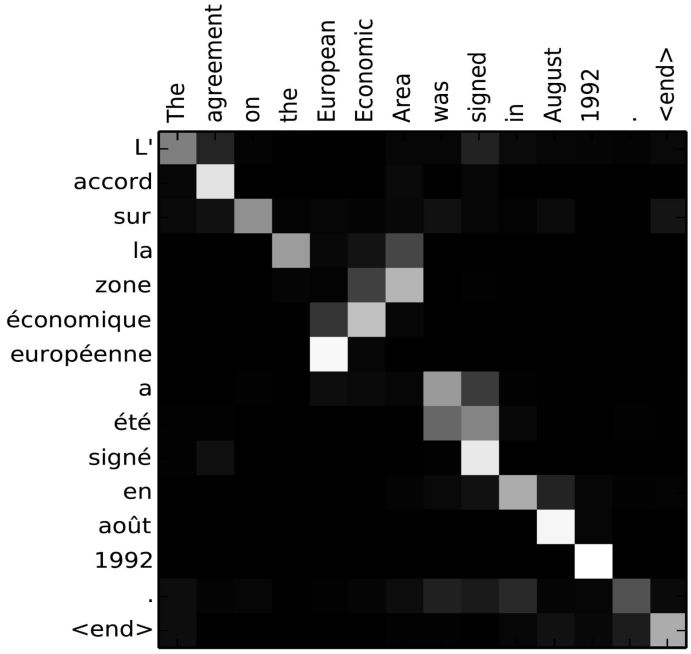
Obecná self attention vrstva funguje na principu výpočtu tří matic Q, K a V (*Query*, *Key* a *Value*). Tyto matice jsou definovány jako:

$$Q = XW_q \; ; \; K \; = XW_k \; ; \; V \; = XW_v$$

Výstup attention vrstvy je poté definován následovně:

$$A(Q, K, V) = softmax(QK^T)V$$

Matice W_q, W_k a W_v představují učitelné parametry této vrstvy neuronové sítě. X je vstupní sekvence tokenů reprezentovaná jako matice $l \times e$, kde l je délka sekvence a e je délka kódování tokenu. Jednotlivé řádky matice Q představují zakódovaný dotaz vztahující se k právě zpracovávanému tokenu. Dotazem je zde myšlen vektor, jež po vynásobení s odpovídajícími položkami matice K produkuje vektor relevancí jednotlivých tokenů sekvence. Matici K si lze představit jako sérii identifikátorů všech tokenů vstupu (sloupce) korespondující s hodnotami tokenů v matici V. Jeden řádek výstupního produktu je poté míra relevance všech tokenů sekvence k jednomu danému tokenu, jak je ukázáno na obrázku 3.2.



Obrázek 3.2: Demonstrace produktu násobení matic Q a K v attention vrstvě určené ke zpracování přirozeného jazyka. Relevance jednotlivých tokenů je demonstrována intenzitou zabarvení grafu. Zde se jedná o demonstraci obecné attention vrstvy, self attention vrstva by v tomto případě na obou osách obsahovala totožná data (dle vstupní sekvence). Obrázek byl převzat z článku [1].

V odbodných textech se často vyskytuje pouze název self attention, kterým se ovšem nemyslí tato „klasická" implementace, ale její varianta s názvem *multihead self attention*. Ideou této varianty je počítání celého principu vícenásobně s využitím vícero různých W_q,

W_k a W_v matic. Tyto matice jsou v takovémto případě nezávisle učeny. Jedné sadě této matic spolu s jejím průchodem se říká attention hlava. Motivací za využitím této duplicity v podobě několika hlav attention vrstvy je schopnost naučit se vyhledávat vícero různých datových souvislostí současně. Každá hlava se v průběhu tréninku většinou naučí rozpoznávat v datech jiné návaznosti, jejichž diverzita následně slouží k lepší extrakci informací. Takovéto výpočty se, pokud možno, ve většině případů provádí paralelně.

3.1.4 Problém komplexity a jeho řešení

Ačkoli architektura vision transformer za jistých podmínek předčí svého konkurenta - konvoluční neuronové sítě (viz. [6]), z pohledu schopnosti zpracování obrazu má stále své slabé stránky, které se mnozí snaží adresovat. Hlavní problém pramení z principu právě té nejdůležitější součásti, a to z mechanismu samotných self attention vrstev. Na rozdíl od úlohy zpracování přirozeného jazyka se úloha zpracování obrazu liší množstvím vstupních dat, to obecně bývá větší. Originální podoba self attention vrstvy, představená v článku *Attention Is All You Need* [22], má paměťovou složitost $\mathcal{O}(n^2)$ kvůli faktu, že je potřeba modelovat vzájemné závislosti všech tokenů sekvence. Spolu v kombinaci s množstvím vstupních dat poté mohou vznikat problémy s výpočetní náročností. Z těchto důvodů bývají často vision transformery výpočetně náročnější než v případě využití konvoluce (porovnáváme-li modely dle hloubky sítě). Tento známý fakt se ovšem snaží řešit hned několik různých implementací. [13, 28, 31, 5]

3.2 Moderní varianty vision transformerů

Tato část popisuje průzkum výchozích implementací, ze kterých tato práce vychází. Jedná se o výběr moderních provedení vision transformerů představených v posledních ročnících konferencí *Conference on Computer Vision and Pattern Recognition* (CVPR) a *International Conference on Computer Vision* (ICCV). Mezi vhodné kandidáty nebylo potřeba řadit pouze modely, zabývající se čistě jen rozpoznáváním tváří. Pro účely této práce byly vyhovující téměř jakékoli modely týkající se zpracování obrazu s architekturou na bázi vision transformeru.

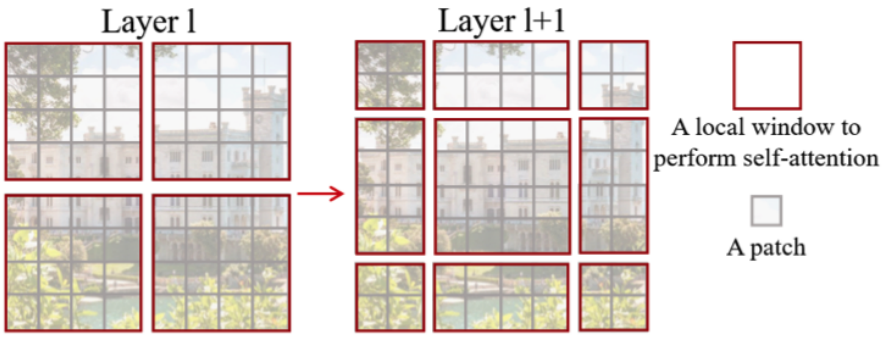
3.2.1 Swin Transformer

Model s názvem Swin Transformer z článku [13] bývá často prezentovaný jako ideální reprezentant, když přijde na architekturu vision transformer. Pro tento model existuje již řada drobných variací a pokusů. Toto provedení vychází z obecného faktu o náročnosti vision transformerů a snaží se tento problém řešit. Architektura představuje vícero změn, přičemž nejdůležitější ideou je omezení množství tokenů, nad kterými je počítán self attention mechanismus. Jedná se o nejzávažnější problém vision transformerů, který se pokouší řešit mnoho článků včetně tohoto.

Původní implementace vision transformeru využívá pro generování tokenů z obrázku jednoduchý mechanismus rozdělování fotografie na bloky o velikosti 16x16 pixelů. Jediné další zpracování, které zde následuje je aplikace lineární transformace a zploštění této 16x16 matice do jednorozměrného vektoru. Zmiňovaný rozměr zde přitom hraje zásadní roli a konstanta 16 zde byla zvolena dle své optimality. Větší rozměry těchto bloků můžou znemožňovat modelu správně rozpoznávat drobné detaily v obraze, které můžou být klíčové v dané úloze. Na druhou stanu, volba této hodnoty rozhoduje také o paměťové náročnosti, která

se s rostoucím rozlišením obrázků výrazně zvětšuje. Pro příklad uvažujme o volbě varianty s okny o velikosti 4x4 pixelů. Při rozlišení obrázku 256x256 bychom se dostali na celkový počet 4096 tokenů, ale v případě obrázku s rozlišením 1920x1080 by už počet tokenů činil téměř 130 000 tokenů. Tento fakt se pokouší implementace Swin transformer řešit využitím vícero různých vylepšení.

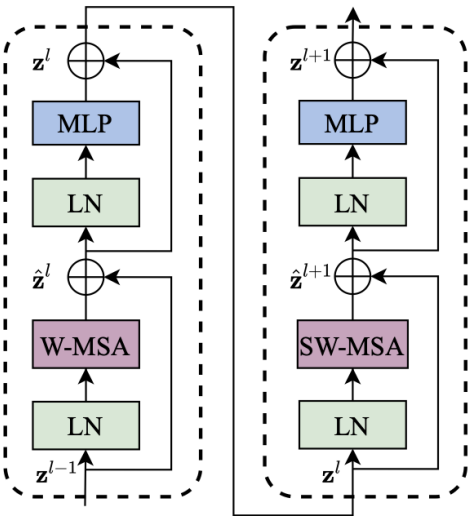
V první řadě, při generování jednotlivých tokenů, architektura rozdělí obraz na bloky o rozměru 4x4 pixelů. (Všimněme si, že díky dalším změnám v architektuře bylo možné tento rozměr zmenšit se zachováním rozumné paměťové náročnosti.) Tyto bloky jsou dále transformovány za pomocí konvoluční vrstvy do vektorů o délce C. Hodnota C značí kapacitu sítě a je volena jako hyperparametr. Rozlišení se při průchodu sítí, podobně jako je tomu u konvolučních implementací jako je *ResNet*, redukuje slučováním. Zde se ale nejedná o konvoluční vrstvy, nýbrž o vrstvy obsahující attention. Pro snížení komplexity těchto vrstev se v implemntaci Swin transformeru limituje počet tokenů, nad kterými je tento výpočet prováděn. Vstupní sekvence tokenů (předzpracovaný obraz) je rozdělena do N oken, které jsou nezávisle zpracovávány attention vrstvou. Rozdělení do oken je možné vidět v obrázku 3.3.



Obrázek 3.3: Ukázka principu *shifted windows* představeného v článku [13]. Jedná se o řešení problému s propagací informací mezi jednotlivými okny zpracovávanými self attention vrstvou. Posunutí okna oproti pozici oken v předchozí vrstvě zapříčiní jejich překryv, což umožňuje propagovat informace i mimo jejich hranice.

Tento postup ovšem zanáší separaci informací mezi okny a brání attention vrstvě v modelování vzájemných datových závislostí mezi nimi. Takovýto fakt by bez dodatečné propagace úplně zastínil výhody pramenící z principu attention. Autoři se tedy snaží tento problém vyřešit přidáním dodatečné vrstvy, kde se tento výpočet provádí nad okny, které jsou oproti předchozímu rozložení posunuty přesně o polovinu rozměru jednoho takového okna (viz obrázek 3.3). Jeden kompletní Swin modul, z nichž se tento model skládá, je tedy složen z těchto dvou průchodů s různým rozložením oken. Podoba tohoto bloku je vyobrazena v diagramu na obrázku 3.4. Díky této změně se při následném průchodu tato okna překrývají a umožňují vzájemnou propagaci podstatných informací skrze celý prostor vstupního obrázku.

V průběhu pokusného trénování byla tato architektura poměrně stabilní a mezi všemi testovanými architekturami patřila mezi ty optimálnější z ohledu využití paměti. Délkou trénování se Swin transformer rovněž umístil v horních příčkách seznamu. Jak také představují výsledky z kapitoly 4.2, tento model nasadil vysokou latku v ohledu přesnosti predikcí na testovacích datech.



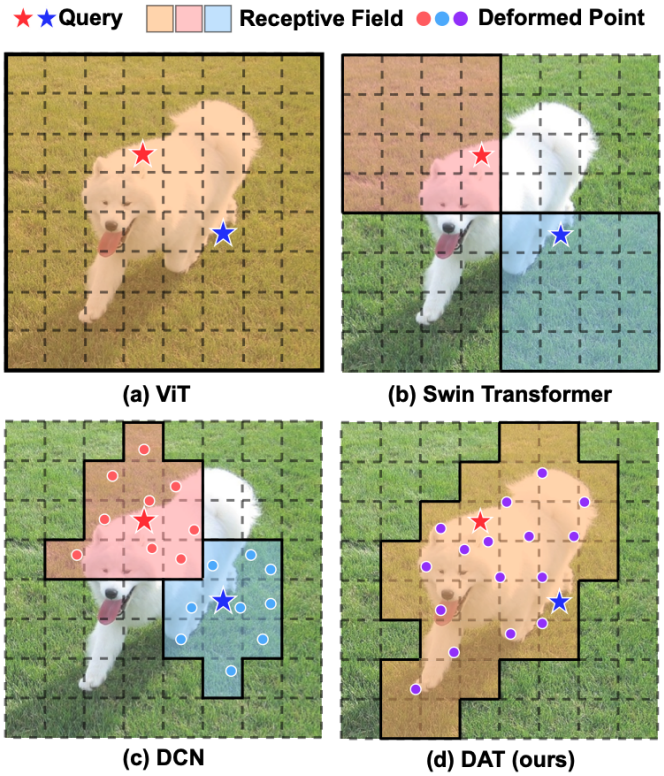
Obrázek 3.4: Na obrázku je možné si všimnout diagramu Swin bloku, ze kterých je architektura Swin transformer složena. Blok se skládá ze dvou průchodů téměř totožného postupu. W-MSA je vrstva *multi-head self attention* počítaná nad jednotlivými okny. Vrstva SW-MSA obsahuje totožný výpočet, akorát s posunutým rozložením oken. Obrázek je převzat z článku [13].

3.2.2 Vision Transformer with Deformable Attention

Autoři architektury DAT (Vision Transformer with Deformable Attention) [28] se pokouší o řešení totožného problému jako Swin transformer. Snaží se v této implementaci eliminovat problém komplexity omezením části obrazu, nad kterým je potřeba počítat self attention. Autoři DAT tvrdí, že přístup používaný ve Swin transformeru zanedbává podobu dat a může bránit modelu správně zachytávat datové souvislosti. V článku je navržen postup limitující attention vrstvu dynamicky dle kontextu obrázku do podoby deformovaných bloků. Tento princip má za účel místo ručního výběru tokenů, které se mají v attention vrstvě uvažovat, umožnit neuronové síti vybrat si, které části obrazu jsou dle kontextu podstatné pro zpracování pomocí attention vrstvy.

Princip představený v tomto článku zakládá na již známém konceptu z architektury konvoluční neuronové sítě DCN. [3] Princip spočívá v rovnoměrném navzorkování bodů dle nedeformované mřížky, kterou by tvořily jednotlivé aplikace konvolučního filtru. Tyto body následně představují lokace aplikací konvoluce ve zdrojovém obraze. Design architektury zahrnuje učitelnou část, která se stará o generování predikcí posuvu těchto rovnoměrně rozložených bodů v obraze. Pro každý zapracovávaný token je tedy predikován posuv těchto bodů, který slouží k deformaci zpracovávaného pole a posuvu aplikace konvolučních filtrů. Implementace DAT přizpůsobuje tento mechanismus architektuře vision transformer a zároveň přináší řadu optimalizací. V tomto řešení metoda rovněž využívá konceptu rozdělování kontrolních bodů do mřížky a následnou predikci jejich posuvu. Zde se ale jedná o manipulaci s tokeny a o jejich výběr pro účely výpočtů v attention vrstvě. Dále autoři počítají s faktem, představeným v dřívějších pracích, kde bylo vypozorováno, že se globální attention vrstvy většinou soustředí na totožné oblasti ve vstupních sekvencích tokenů pro každý dotaz Q_i. Implementační změny tedy zahrnují aproximaci v podobě predikce jediné průměrné deformace celého zorného pole attention vrstvy, která je následně použita při

zpracování všech tokenů fotografie. Tento fakt je možné pozorovat na obrázku 3.5, který byl přejat z článku představující tuto architekturu neuronové sítě. [28]



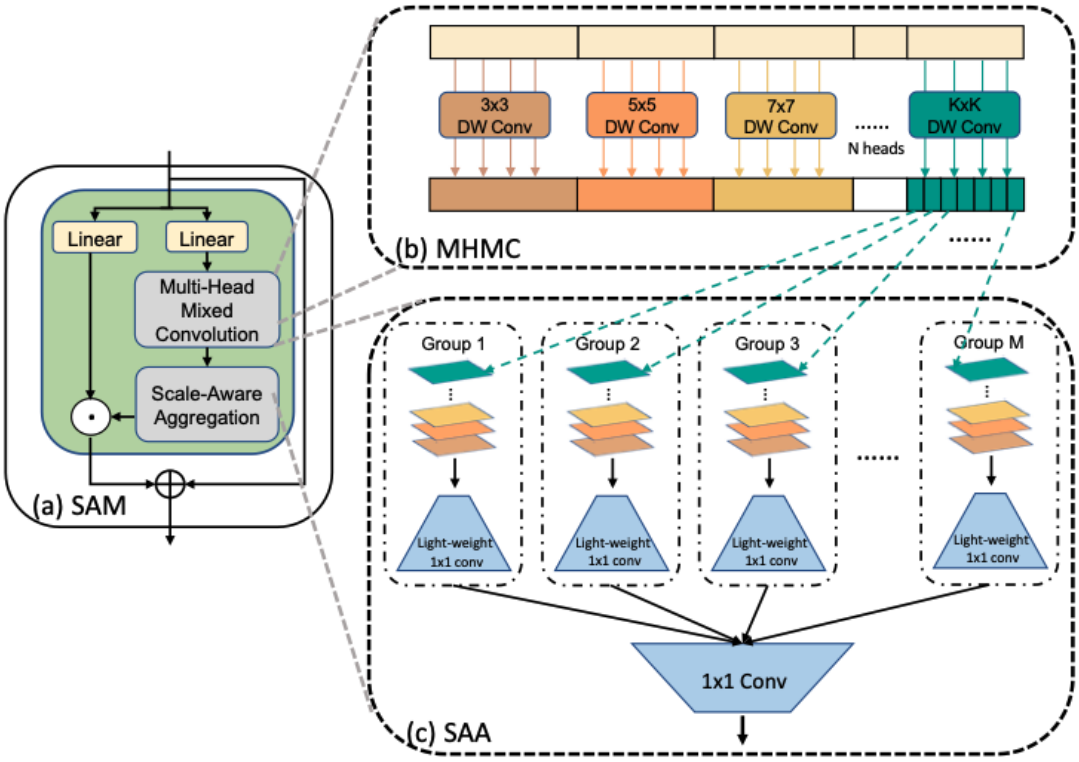
Obrázek 3.5: Demonstrace deformovatelných receptivních polí architektury DAT v porovnání s ostatními řešeními. Zabarvená pole znamenají části obrazu, nad kterými je počítán attention pro aktuálně zpracovávaný token. Ten je v obrázku vyznačen hvězdou. Pro lepší demonstraci jsou zobrazeny dva zpracovávané tokeny.

Experimenty s touto architekturou poukázaly na nejdůležitější výhodu tohoto řešení. Model měl nejmenší paměťovou náročnost v porovnání s ostatními vybranými adepty. Co se týká času nutného ke zpracování vstupu byla tato metoda zároveň nejrychlejší. Přes své výhody v efektivitě práce se zdroji tento model scházel na schopnostech učení. V průběhu trénování byl model nestabilní a pro dosažení korektního optima bylo zapotřebí využití dynamického plánovače pro řízení úrovně rychlosti učení. Ve výsledných testech nad vybranými modely nakonec tato architektura obsazovala spíše spodní místa v žebříčku nejpřesnějších predikcí.

3.2.3 Scale-Aware Modulation Meet Transformer

Implementace z článku *Scale-Aware Modulation Meet Transformer* [12] na druhou stranu představuje hybridní neuronovou síť imitující prvky z oblasti vision transformerů konvolučními vrstvami. Autoři se snaží najít kompromis mezi schopností attention vrstev modelovat globální závislosti v datech a zároveň výpočetní efektivitou konvolučních sítí. Autoři zde představují nový typ konvoluční vrstvy založený na zachytávání informací z obrazu v různých úrovních granularity paralelně. Zároveň ale stále využívají klasických multi-head self attention bloků.

Nově představený modul *Scale-Aware Modulation*, vyobrazený také na obrázku 3.6, zakládá na extrakci klíčových vlastností v různých úrovních detailu s pomocí konvolucí. Toho je zde docíleno několika paralelními konvolučními vrstvami s různým rozměrem konvolučního jádra. Tuto část nazývají *Multi-Head Mixed Convolution* (MHMC) a jejím výstupem je M různých aktivací o fixním počtu vrstev N. Tento počet N je u každé z konvolučních vrstev totožný. Následující částí tohoto modulu je vrstva *Scale-Aware Aggregation* (SAA), která slučuje výstupy z MHMC. Slučování se provádí vždy nad jednou z N vrstev výstupů všech konvolucí v MHMC. Slučování se provádí s pomocí konvoluční vrstvy o velikosti jádra 1x1. Těchto N sloučených výstupů se následně opět slučuje do jediné aktivace rovněž pomocí 1x1 konvoluce. Motivací této implementace je rozšíření zorné plochy oproti jednoduché konvoluční vrstvě se zachováním úrovně detailu. Přestože tato architektura představuje tento nový typ bloku, který má za účel rozšíření zorného pole, stále sází na přednosti attention vrstev. Autoři ve své implementaci kombinují právě tyto nové *Scale-Aware Modulation* bloky spolu s vrstvami Multi-Head self attention.



Obrázek 3.6: Obrázek znázorňuje diagram části nově navržené vrstvy napodobující multi-head self attention. Princip spočívá v extrakci informací pomocí série konvolučních hlav s různou granularitou (vrchní část obrázku označená jako MHMC). Část označená jako SAA se pak stará o zkombinování informací získaných ze všech hlav pomocí 1x1 konvolucí. Obrázek je přejat z článku [12].

Architektura SMT prokázala velmi dobrou přesnost predikce. V žebříčku testovaných modelů obsazovala převážně horní příčky. Časovou a paměťovou náročností nebyla nijak výrazná, dobou výpočtu se architektura umístila přibližně ve středu seznamu.

3.2.4 FLatten Transformer

Článek *FLatten Transformer: Vision Transformer using Focused Linear Attention* [9] se na druhou stranu pokouší řešit problém komplexity přímo u jejího zdroje, a to modifikací self attention vrstvy. Autoři se zde pokouší o optimalizaci s využitím aproximovaného výpočtu, který byl představen v článku *Transformers are RNNs: Fast Autoregressive Transformers with Linear Attention* [10].

Implementace linear attention se snaží o zmírnění komplexity aproximací vzorce běžné self attention vrstvy a optimalizací jejího výpočtu. Autoři zde vsází na fakt, že délka zapracovávaných sekvencí N je ve většině případů výrazně větší v porovnání s délkou reprezentace tokenů (D). Aproximace spočívá v úpravě vzorce pro klasickou self attention vrstvu, formovaném následovně (princip self attention je detailněji popsán v kapitole 3.1.3):

$$A_x = softmax(\frac{Q \cdot K^T}{\sqrt{D}}) \cdot V$$

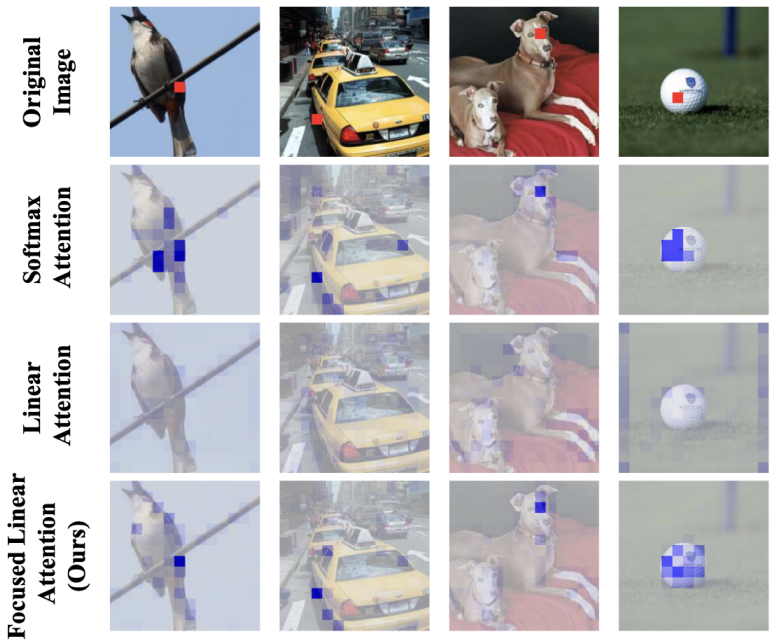
Upravený vzorec pro linear attention má následující podobu:

$$V_i' = \frac{\phi(Q_i)^T \sum_{j=1}^{N} \phi(K_j) V_j^T}{\phi(Q_i)^T \sum_{j=1}^{N} \phi(K_j)}$$

Všechny zmíněné proměnné odpovídají původnímu vzorci s rozdílem v aplikaci operátoru $\phi(\cdot)$. Autoři v tomto případě používají aproximaci jádrem z teorie *vector state machines*. Jádro $\phi(x)$ má zde podobu $\phi(x) = elu(x) + 1$. Redukce následně spočívá v záměně pořadí výpočtů nad maticemi Q, K a V tak, že výsledná komplexita výpočtu činí z $\mathcal{O}(N^2D)$ (běžná self attention vrstva se SoftMax funkcí) pouhých $\mathcal{O}(Nd^2)$. Mechanismus lineární attention vrstvy ovšem přináší své limitace, a ty se autoři článku pokouší řešit vlastním návrhem tzv. *focued linear attention*.

Focused linear attention se soustředí na eliminaci dopadu změn představených v *linear attention*. Dle experimentů provedených autory architektury FLatten Transformer tvoří tento postup artefakty v podobě rovnoměrnějšího rozložení pozornosti této vrstvy vůči vstupní sekvenci tokenů. Výstupní podobnosti mezi odpovídajícími záznamy v maticích Q a K mají v běžné softmax attention vrstvě spíše výraznější charakteristiku. Je zde viditelná separace těch opravdu relevantních informací dle vstupní sekvence tokenů. Oproti tomu mají attention mapy v případě linear attention mechanismu rovnoměrnější rozložení a působí v porovnání s výstupem originální attention vrstvy rozmazaně, což je možné pozorovat na obrázku 3.7. Odtud také autoři čerpají název této modifikované varianty linear attention. Fakt, že attention mapa má takovouto podobu má za následek zhoršenou přesnost výstupů takovéto vrstvy.

Mimo zmiňované úpravy v attention blocích byla testovaná varianta tohoto modelu totožná s implementací Swin, což zapříčinilo dosažení velmi podobných výsledků. Paměťová a časová náročnost trénovaní byla tedy srovnatelná a z pohledu přesnosti se architektura v provedených testech držela vždy těsně za svou nelinearizovanou variantou. Po odladění všech hyperparametrů má tento model jistě potenciál k lepším výsledkům.



Obrázek 3.7: Obrázek porovnává rozdíly v podobě attention map mezi linear attention, softmax attention a focused linear attention vrstvami. Intenzita zabarvení (modrá) tokenů v obraze značí větší podobnost ve vztahu k aktuálně zpracovávanému toknu (označeno červenou). Obrázek byl převzat z článku [10].

3.2.5 Vision Transformer with Bi-Level Routing Attention

Architektura *BiFormer* [31] se pokouší rovněž limitovat nutnost výpočtu self attention mechanismu mezi všemi páry tokenů. Implementace BiFormer zakládá (podobně jako tomu je u metody DAT) na dynamickém výběru vhodných částí obrazu. Oproti deformování zpracovávaného pole zde autoři naopak provádí filtrování vstupních tokenů. Způsob filtrování se zde provádí za pomocí agregace částí obrazu do regionů (podobné konceptu oken ve Swin transformeru). BiFormer provádí aproximované výpočty self attention principu nad celými regiony místo nad jednotlivými tokeny. Z těchto regionů vybírá důležité součásti obrazu, podobným způsobem jako klasická self attention vrstva. Zde ovšem vybírá vždy k nejvýhodnějších oblastí, které jsou následně zpracovávány běžnou attention vrstvou na úrovni jednotlivých tokenů. Autoři tohoto článku poukazují na dříve představené mechanismy deformující zorné pole attention vrstvy s tvrzením, že v případě běžného vision transformeru nejsou pro všechny dotazy Q_i podstatné stejné oblasti obrazu.

Výběr vhodných oblastí k výpočtu attention vrstvy zde tedy není prováděn, jako tomu je u architektury DAT, jednou pro celou vstupní fotografii. Obraz je zde rozdělen na dříve zmiňované regiony, za pomoci kterých je poté prováděn výběr vhodných tokenů ke zpracování. Selekce se provádí na základě sémantické podobnosti těchto regionů. Pro získání této podobnosti se nejprve provede extrakce statistik z matic Q a K za pomocí průměrování dle pokrytí těchto regionů. Výsledkem jsou matice Q_r a K_r. Princip selekce následně spočívá v konstrukci matice sousednosti a výběru důležitých regionů. Nejprve se tedy spočítá matice sousednosti, která představuje jednotlivé tokeny jako uzly grafu. V implementaci je matice sousednosti získávána vynásobením matic Q_r a K_r. Tento graf se po spočtení prořezává dle

k nejvýhodnějších přechodů, kde k se volí jako hyperparametr sítě. Prořezáním grafu získáváme k nedůležitějších regionů v obraze, které budou mít pro účely self attention vrstvy největší význam.

Ačkoli toto řešení přináší řadu dalších optimalizací, následné experimenty ukázaly, že architektura nenakládá s prostředky ideálním způsobem. Přestože pro řešení komplexity využívá zmiňovaný princip víceúrovňového výpočtu self attention, tato úprava zjevně není dostačující. Paměťová náročnost je zde v porovnání s ostatními adepty řádově větší a čas výpočtu je rovněž delší. Ani přes kapacitu výchozí implementace tohoto modelu a přes svou časovou náročnost se bohužel v průběhu experimentů neukázaly podstatné výhody v přesnosti rozpoznávání obličejů. V průběhu trénování sice měl tento model výrazný náskok v dosažení >80 % F1 skóre na testovací datové sadě v pouhé 4. epoše trénování, následující progres ovšem poukazoval spíše na problémy s přetrénováním.

Kapitola 4

Experimenty a vyhodnocení

Jednou z nejdůležitějších částí této práce je provedení experimentů s konkrétními vybranými architekturami. Cílem těchto experimentů bylo nalezení nejvhodnější varianty implementace vision transformeru a její následné odladění pro získání co nejefektivnějšího systému pro rozpoznávání obličejů. Prvním experimentem bylo otestování vybraných architektur, dále následoval průzkum dopadu podoby trénovacích dat na přesnost predikcí a nakonec bylo provedeno odladění hyperparametrů. Experimenty založené na změně trénovací datové sady zahrnovaly využití různých populárních veřejně dostupných sad či využití tzv. *cross-age* fotografií. Z tématu ladění hyperparametrů byl nejdůležitější položkou výběr trénovací loss funkce. Správna volba v tomto uspořádání hrála klíčovou roli a její správná volba rozhodovala o konvergenci modelů v průběhu trénování. V poslední řadě byly provedeny speciální experimenty s některými z architektur, využívající jejich specifika. Konkrétně se jednalo o experimenty s architekturou CLIP spolu se zahrnutím dodatečných vstupních informací do procesu tréninku. Dále byly provedeny například experimenty zkoumající škálovatelnost modelu FLatten transformer či využití předtrénovaných modelů na specifických datových sadách. Implementace trénovacího a validačního kódu včetně implementace experimentů byla provedena s využitím převážně Python knihoven *PyTorch* a *PyTorch Lightning*. V případě implementací jednotlivých architektur byl využit vždy originální zdrojový kód autorů.

4.1 Metriky pro vyhodnocení kvality výstupu

V průběhu celé práce bylo potřeba vyhodnocení kvality výstupů jednotlivých implementací. Ať se jednalo o úvodní otestování všech vybraných variant, nebo o vyhodnocení výsledků experimentů, bylo potřeba zvolit robustní hodnotící metody. Hodnocení zde bylo provedeno výběrem dvojic fotografí v testovací datové sadě a následným porovnáváním výstupního vektoru neuronové sítě. Pro účely úvodních testů a průběžných validací byla vybrána populární sada s názvem *Labeled Faces in the Wild* (LFW) [20], obsahující přesně 5749 různých osobností na přibližně 13 000 fotografiích. Pro finální vyhodnocení byla využita často používaná evaluační datová sada IJB-C [16] spolu s evaluačním kódem z repozitáře *Insightface*[1]. Výstupní vektory pro dané dvojice byly porovnávány vždy za pomocí kosinové podobnosti, což vychází z konceptů využitých loss funkcí. Mezní hodnota pro rozhodování, zdali obličeje dle výstupního vektoru patřily totožné osobě, byla určována automaticky za pomocí stanovených předpokladů o modelu či za pomocí bodu EER (viz. kapitola 4.1.1). Výběr konkrétních hodnot bude popsán v následujícím textu.

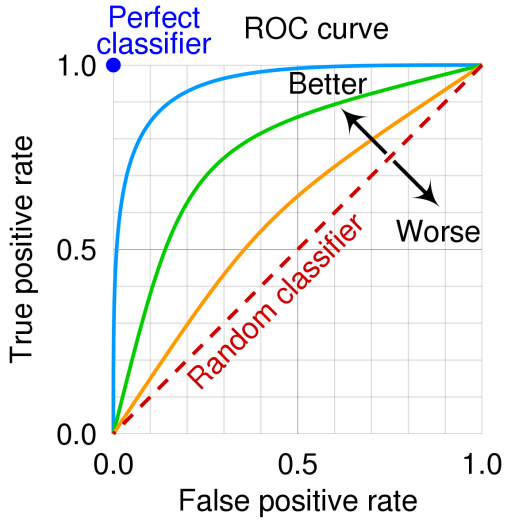
[1] Repozitář Insightface: https://github.com/deepinsight/insightface

4.1.1 ROC křivka

První z použitých metrik byla ROC křivka (Receiver operating characteristic curve), zobrazující vztah dvou různých metrik *True Positive Rate* (TPR) a *False Positive Rate* (FPR). [27] Ukázka obecné podoby ROC křivky je znázorněna na obrázku 4.1. Metrika FPR odpovídá chybě prvního typu a obě tyto veličiny vychází ze statistiky matice záměn, přičemž jsou počítány následovně:

$$TPR = \frac{TP}{TP + FN} \; ; \; FPR = \frac{FP}{FP + TN}$$

Hodnota TP (*True Positives*) značí počet správně predikovaných shodných dvojic, FN (*False Negatives*) je poté počet nesprávně predikovaných dvojic (predikováno jako neshodné), které ve skutečnosti byly shodné. FP (*False Positives*) znamená počet špatně predikovaných shod a TN (*True Negatives*) počet správně zamítnutých dvojic, které nebyly shodné. Obě metriky tedy udávají poměr dané predikce k celkovému počtu pozitivních či negativních shod v datové sadě. Pomocí této metriky lze nejenže porovnávat kvalitu výstupu natrénovaných modelů, ale také odhadovat hraniční hodnotu pro rozhodování, zdali jsou obličeje na dvou různých fotografiích stejné či nikoli.

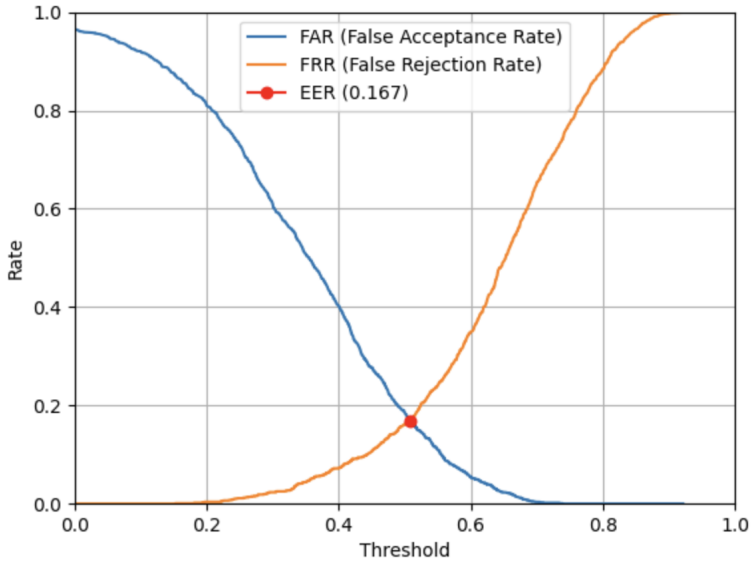


Obrázek 4.1: Na zjednodušené ukázce lze vidět znázorněnou obecnou ROC křivku. Diagonální čára představuje predikci pomocí náhodného výběru. Snaha je proto o to, aby křivka co nejlépe kopírovala levý a horní okraj grafu. Graf byl převzat z webu `www.medium.com` [19]

Při rozhodování o shodě obličejů na dvou různých obrázcích za pomocí výstupních vektorů neuronové sítě může být náročné stanovit správnou hranici podobnosti. Toto téma je ještě závažnější v případě vektorů s rostoucí dimensionalitou, kde se může projevovat tzv. prokletí dimensionality. [25] Tento jev způsobuje řídkost rozložení mnohodimenzionálních dat a může tak způsobit, že i vektory příslušící stejné třídě (obličeji) budou mít nízké hodnoty kosinové podobnosti. Přestože vysoká dimensionalita výstupu neuronové sítě tímto může působit kontraproduktivně, cílem většinou je, aby se model během trénování správně naučil identifikovat velké množství klíčových bodů obličeje. Za účelem výběru správné mezní hodnoty bylo pro záměry testování využito právě ROC křivky. Samotná křivka totiž promítá tuto hranici do vztahu TPR a FPR veličin. Stačí tedy pouze zvolit požadované kritérium

pro jednu z těchto metrik a vybrat mezní hodnotu dle nejlepšího výsledku druhé metriky. Pro účely úvodního testování byla využita maximální úroveň TF 5 %. Znamená to tedy, že se se zvolenou mezní hodnotou připouští maximální chybovost 5 % v případě, kdy obličeje nepatří stejné osobě. K porovnání state-of-the-art řešení se v praxi využívají hodnoty 0,1 % a nižší. Často se ke srovnávání využívá rovněž metrik značených jako $TPR@FPR = 1e-n$, které znázorňují úroveň TPR s připuštěnou chybou FPR o dané hodnotě $1e-n$. Hodnoty se zde obvykle pohybují od $1 \cdot 10^{-2}$ až po $1 \cdot 10^{-6}$.

Často používanou metodou pro volbu optimální mezní hodnoty je hledání tzv. *Equal Error Rate (EER)* bodu. Nalezení tohoto bodu je možné promítnutím veličin FPR a FNR v závislosti na zvolené mezní hodnotě a nalezením jejich průsečíku, jak je také demonstrováno na obrázku 4.2. Hodnota FNR je v tomto případě chyba druhého druhu a lze ji získat inverzí statistiky TPR jako $FNR = 1 - TPR$. S pomocí bodu EER lze mimo rozhodovací hranici získat také odhad o efektivitě systému pro rozpoznávání biometrických údajů. Ačkoli už samotná podoba této optimální rozhodující hranice může napovídat o stabilitě systému, úroveň chyby, na které se potkávají tyto dvě křivky, poskytuje velmi dobrý náhled na přesnost predikcí.



Obrázek 4.2: Demonstrace nalezení bodu rovnosti chyb EER. Metriky FAR (False Acceptance Rate) a FRR (False Rejectance Rate) jsou zaměnitelné s hodnotami FPR a FNR.

Další vhodnou metrikou pro přímé srovnání kvality predikcí modelu, založené na výsledcích dostupných při počítání ROC statistiky, je AUC (*Area Under the Curve*) skóre. Jedná se o hodnotu znázorňující plochu pod ROC křivkou. Ačkoli tato metrika ve většině případů vypovídá o kvalitě predikce jednotlivých modelů, její samostatné použití není dostatečným kritériem pro usouzení závěru. V extrémních případech se může stát, že tato metrika bude dosahovat vysokých hodnot i za předpokladu velké chybovosti v predikci shodných či neshodných párů. Ačkoli se hodnoty této metriky mohou pohybovat v rozmezí 0–1, hodnota 0,5 dle významu ROC křivky značí skóre náhodného klasifikátoru. V praxi s rostoucí přesností modelu toto skóre rychle dosáhne hodnot blízkým maximálnímu ohodnocení 1. Porovnávání modelů za pomocí pouze tohoto skóre je tedy obtížné kvůli zmenšujícím se rozdílům s blížící se hranicí ideálního klasifikátoru.

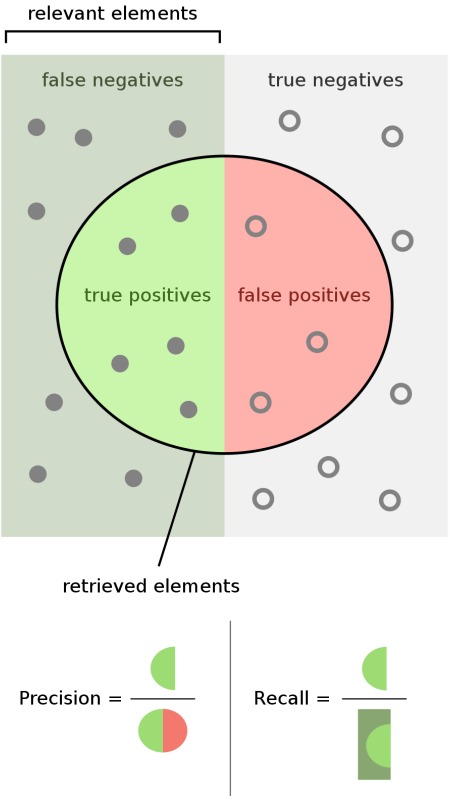
4.1.2 F1 a Accuracy skóre

Dříve zmíněné metriky sice poskytují důležité informace pro evaluaci, samy o sobě příliš nevypovídají o celkové přesnosti predikcí. Z těchto důvodů byly do analýzy zařazeny hodnotící funkce F1 a Accuracy, které tyto dříve zmíněné metriky doplňují. F1 skóre rovněž vychází ze statistik matice záměn. Pro ni je ovšem tentokrát zapotřebí stanovení pevného mezního bodu pro porovnávání fotografií. Zde se může uplatnit buďto dříve zmiňovaný výpočet EER, případně je možné jinými způsoby najít a určit vhodnou rozhodovací mez. Vzorec pro výpočet F1 skóre je následující:

$$F1 = 2 \cdot \frac{precision \cdot recall}{precision + recall}, \text{ kde jsou } precision \text{ a } recall \text{ počítány následovně:}$$

$$precision = \frac{TP}{TP + FP} \; ; \; recall = \frac{TP}{TP + FN}$$

Zde veličiny TP a FP mají stejný význam jako v případě ROC statistiky. Přibývá zde pouze hodnota FN (False Negative), představující počet špatně predikovaných (zavrhnutých) shodných dvojic. Výpočet hodnot precision a recall je demonstrován na obázku 4.3. Tato metrika je obzvláště náchylná na nevyváženost počtu shodných a neshodných dvojic, a proto je v případě nevyvážené sady potřeba nadvzorkování či podvzorkování jedné ze skupin. F1 skóre dosahuje hodnot 0–1, přičemž 1 značí nulovou chybovost a 0 zase nulovou přesnost.



Obrázek 4.3: Obrázek znázorňuje význam veličin *precision* a *recall*, používaných v hodnotící funkci F1. Obrázek byl přejat z Wikipedie. [26]

Méně náchylným, ale zároveň méně vypovídajícím způsobem je hodnocení metrikou *Accuracy*. V tomto případě se nevyužívá informace o typu predikce, nýbrž pouze informace o její správnosti. Metrika Accuracy se určuje jako poměr správných predikcí k celkovému počtu vzorků v testovací sadě. Metoda Accuracy byla zvolena za účelem jednoduchého srovnávání výsledků v žebříčku existujících řešení. Právě tato metrika, ačkoli ne příliš detailní, je jednou z nejpoužívanějších a bývá často součástí analýzy jednotlivých implementací.

4.2 Prvotní zhodnocení výchozích implementací

Před provedením pokročilejších experimentů bylo potřeba nejprve zjistit, se kterými z výchozích implementací je vhodné pokračovat. Za tímto účelem byl proveden první z experimentů v podobě pokusného natrénování a otestování všech modelů ve výchozí podobě. Všechny modely byly trénovány za podobných podmínek, přičemž hodnotícími kritérii byla rychlost trénování, paměťová náročnost modelu, výsledná přesnost predikcí dle hodnotící funkce a stabilita během trénování. Testovací trénování bylo prováděno na podmnožině datové sady *VGG-Face2* obsahující pouze obrázky o velikosti větší než 200x200 px.[2] Původní trénovací set sestával z 3,3M obrázků s 9280 různými obličeji. Vyfiltrovaná podmnožina datové sady *VGG-Face2* poté činila necelých 1,5M obrázků o stejném počtu obličejů. Tato redukce byla provedena za účelem minimalizace vlivu velikosti obrázku na kvalitu výstupu modelu. Evaluace byla prováděna na testovací datové sadě LFW obsahující opět pouze zarovnané obličeje. Testování zahrnovalo ve všech případech stavy všech trénovacích epoch. Analyzován tedy nebyl pouze nejlepší dosažený výsledek, nýbrž celý proces trénování. Pro každou z trénovacích epoch byl vždy uvažován pouze jeden stav, a to vždy finální stav po průchodu celou datovou sadou. Jako referenční model architektury konvolučních neuronových sítí, určený k porovnání s vybranými vision transformery, byl zvolen model ResNet-50.

4.2.1 Zvolené hyperparametry

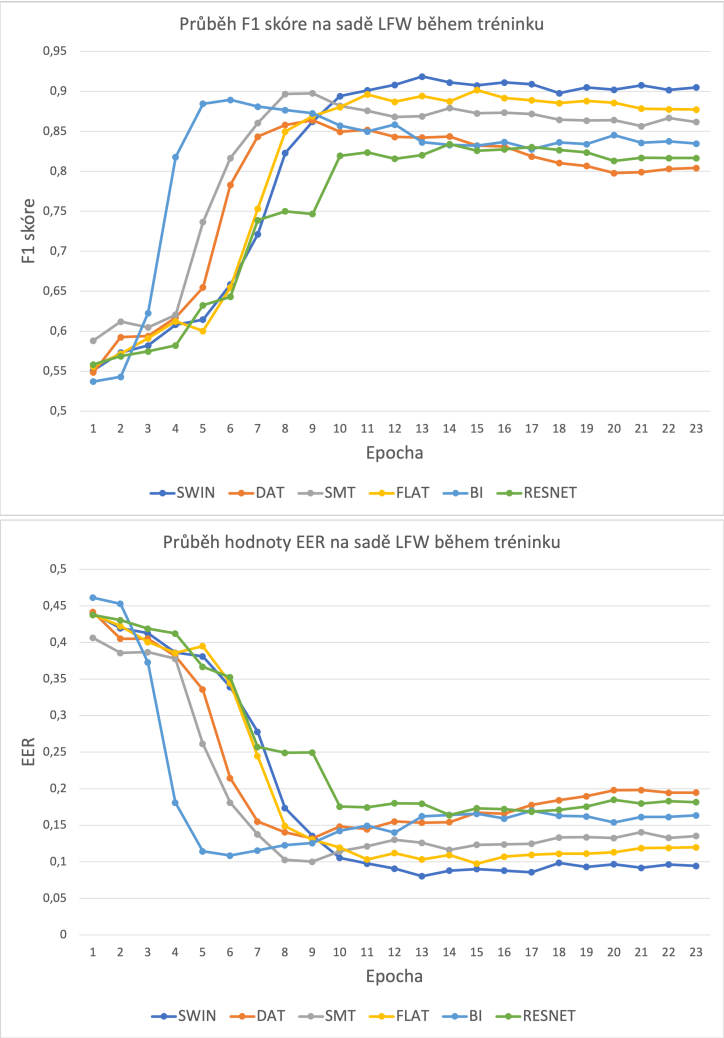
Pro správné zhodnocení a porovnání všech využitých modelů bylo pro trénování a evaluaci stanoveno rovné prostředí. Všechny testované architektury byly trénovány za stejných podmínek a následná evaluace byla prováděna pomocí stejných metod. Trénink byl prováděn na sadě 1,5M fotografií v celkem 23 epochách. Délka výstupního vektoru byla stanovena na hodnotu 512. Pro účely trénování byla využita *ArcFace loss* funkce popsaná v sekci 2.4. Trénování bylo prováděno na grafických kartách *NVIDIA RTX A5000*. Pro jednoduchost přibližné výměry času počítání bylo provedeno testovací trénování na pouze jedné grafické katě. Využitým optimalizátorem pro trénování samotné sítě byl ve všech případech optimalizátor *Adam*, dostupný z Python knihovny *PyTorch*. Trénování ve všech případech zahrnovalo také optimalizaci vzorových vektorů jednotlivých tříd/obličejů pro výpočet ArcFace loss funkce. Pro tyto účely byl vybrán optimalizační algoritmus *SGD*. Testování bylo prováděno na datové sadě určené ke srovnávání systémů pro rozpoznávání obličejů LFW (*Labeled Faces in Wild*)[3].

[2] Datová sada VGG-Face2 je k dispozici ke stažení z: https://academictorrents.com/details/535113b8395832f09121bc53ac85d7bc8ef6fa5b

[3] Datová sada *LWF* je dostupná z: http://vis-www.cs.umass.edu/lfw/

4.2.2 Výsledky analýzy

Na obrázku 4.4 je možné vidět průběh přesnosti jednotlivých modelů během trénování. Zde si je možné všimnout, že architektura BiFormer (BI), představená v kapitole 3.2.5, dosáhla svého optima rekordně již v 5. epoše trénování. Architektura Swin transformer (SWIN) a její linearizovaná varianta FLatten transformer (FLAT) vyžadovaly pro dosažení optima obecně větší počet epoch oproti ostatním. Držely se ovšem v horních příčkách z pohledu maximální dosažené přesnosti. Architektura DAT, z kapitoly 3.2.2, se ukázala být nejméně stabilní a zároveň měla nejhorší výslednou přesnost. Toho si je možné všimnout také v tabulce 4.1, kde se umístila ve všech srovnáních na posledním místě. Model SMT (kapitola 3.2.3) pak představuje kompromis mezi rychlou konvergencí a dostatečnou úrovní přesnosti.



Obrázek 4.4: Grafy na obrázku zobrazují změny v přesnosti jednotlivých modelů v průběhu trénování na zarovnaných datech. Je zde možné si všimnout výrazného posunu z pohledu přesnosti u všech vision transformerů oproti referenční konvoluční sítě ResNet-50.

Za účelem přehlednějšího srovnání výsledků testování byla brána v potaz jednak maximální dosažená přesnost, ale i průměrná přesnost během všech epoch trénování. V tabulce 4.1 je možné vidět výrazný náskok v případě architektury Swin transformer z ohledu maxi-

mální dosažené přesnosti. Přestože se jedná o referenční implementaci, v průběhu testování se ukázalo, že tuto variantu vision transformeru není ani v oboru rozpoznávání tváří jednoduché předčit. V tabulce si je dále možné všimnout, že lineární varianta Swin transformeru (FLatten transformer) se vždy drží těsně za touto architekturou. Dominance modelů BiFormer a SMT z pohledu průměrných statistik pak spíše poukazuje na jejich včasnou konvergenci. Výchozí implementace BiFormer transformeru má zde ovšem výhodu ve své kapacitě, která je v porovnání s ostatními modely výrazně větší. Přes svůj prvotní náskok se ovšem BiFormeru nepodařilo výrazně konkurovat ostatním architekturám a v průběhu dalšího trénování docházelo spíše k výraznému poklesu v přesnosti z důvodu přetrénování. Na druhou stranu, model SMT dokonvergoval k optimálnímu stavu rovněž poměrně rychle a svou přesnost si v průběhu tréninku obstojně udržel i přes svou kompaktní velikost.

	MAX F1	AVG F1		MAX AUC	AVG AUC
1	SWIN (0,920)	BI (0,759)		SWIN (0,969)	SMT (0,884)
2	FLAT (0,905)	SMT (0,736)		SMT (0,957)	BI (0,879)
3	BI (0,899)	Swin (0,696)		FLAT (0,955)	SWIN (0,868)
4	SMT (0,894)	FLAT (0,690)		BI (0,950)	FLAT (0,861)
5	DAT (0,846)	DAT (0,676)		DAT (0,930)	DAT (0,847)
6	RESNET (0,782)	RESNET (0,587)		RESNET (0,908)	RESNET (0,818)

Tabulka 4.1: Žebříček testovaných modelů řazený dle výsledků jednotlivých metrik. Položky *MAX* znázorňují maximální dosaženou hodnotu odpovídající metriky a položky *AVG* odpovídají průměru dané metriky skrze všechny epochy trénování.

Z pohledu efektivity výpočtu byl proveden experiment porovnávající délky trénování každého z modelů. Jak je možné vidět v tabulce 4.2, zatímco model DAT disponoval bezkonkurenčně malou paměťovou a časovou náročností trénování, ostatní architektury se držely okolo hranice 5 hodin na jednu epochu. Na druhou stranu výchozí implementace modelu BiFormer byla daleko náročnější na trénování. Průběh jediné epochy v případě této implementace trval průměrně více než 16 hodin. Při vyhodnocení délky trénování je v tomto případě potřebné se rovněž soustředit na kapacitu modelu. BiFormer měl totiž oproti ostatním podstatně větší počet učitelných parametrů.

Pořadí	Architektura	ø čas trénování epochy [h]	Počet parametrů	GFLOPs
1	RESNET	2,07	25,6 M	4,09
2	DAT	3,56	29 M	4,6
3	Swin	3,97	28 M	4,5
4	FLAT	5,94	29 M	4,5
5	SMT	6,37	32 M	7,7
6	BI	16,57	56 M	9,8

Tabulka 4.2: Tabulka znázorňuje žebříček vybraných architektur podle průměrné doby trénování jedné epochy dle parametrů definovaných v kapitole 4.2.1.

Dle provedených testů se ukázalo, že princip selektivních výpočtů self attention vrstev, představený v obou architekturách DAT i BiFormer, má pro úlohu rozpoznávání tváří na zarovnaných obličejích spíše negativní přínos. Z dřívě provedených testů na nezarovnaných tvářích se ovšem ukázal jejich potenciál, a to zlepšením úrovně přesnosti natrénovaných modelů. Přesto ani s větší kapacitou implementace BiFormer se nepodařilo dosáhnout srov-

natelných výsledků jako u architektur na bázi Swin transformeru. Provedení architektury SMT na druhou stranu prokazatelně usnadňuje konvergenci modelu. Přístup založený na hybridní kooperaci konvolučních vrstev s prvky vision transformerů se nezdá být k zahození. Přesto osvědčené metody postavené na populární implementaci Swin transformeru stále dosahují lepších výsledků. A to ať se jedná o originální architekturu, tak i její linearizovanou variantu FLatten transformer. Proto budou v následujících experimentech uvažovány pouze tyto dvě zmíněné implementace (Swin a FLatten transformery).

4.3 Experimenty s dalšími architekturami

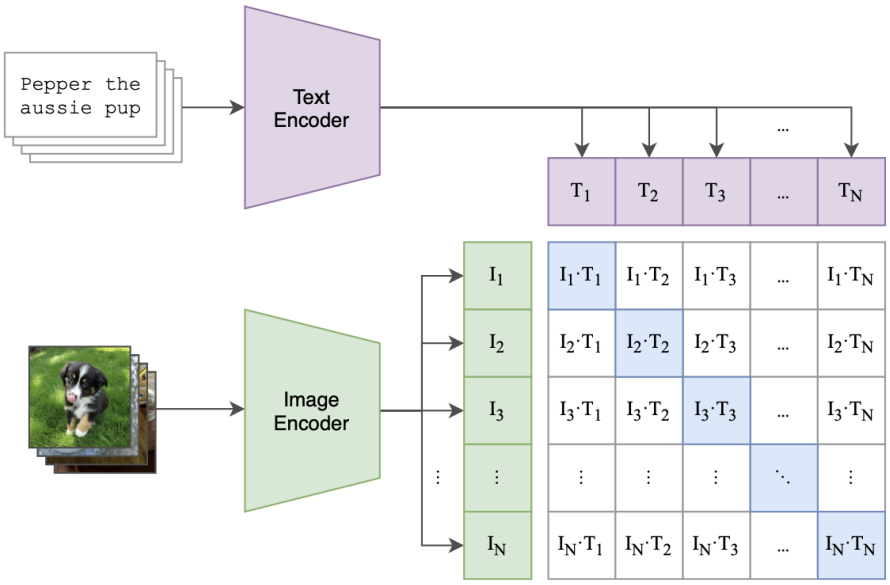
Za účelem zjednodušení následujících experimentů bylo zahrnuto rozšíření seznamu testovaných architektur na první místo při provádění následujících pokusů. Seznam byl konkrétně rozšířen o dvě další architektury. Mezi nově uvažované architektury patřila architektura CLIP[17] a architektura CMT[7]. Jejich přidání zahrnovalo jejich otestování způsobem popsaném v kapitole 4.2. Architektura CMT byla vybrána pro její způsob implementace. Ta je založená na konvolučních vrstvách a pokouší se imitovat principy chování attention vrstev a podstatu architektur vision transformer. Motivace pro druhou architekturu byl rovněž potenciál ve způsobu jejího provedení. Zároveň byla vybrána na základě výsledků testů předtrénované sítě, které představili její autoři. Výrazně úspěšná byla bohužel pouze architektura CLIP, a proto mimo úvodních experimentů dále nebyla architektura CMT uvažována. V následujícím textu budou popsány detaily implementace CLIP a výsledky tohoto experimentu.

4.3.1 Architektura CLIP

Architektura CLIP je multimodální varianta neuronové sítě kombinující vizuální a jazykový vstup. Tento model sestává ze dvou částí, a to obrazové a textové. Textová část má typické prvky neuronové sítě určené ke zpracování přirozeného jazyka a jedná se o běžnou sestavu známou z tohoto oboru. Obrazová část v této implementaci může sestávat buďto z konvoluční sítě či vision transformeru. V případě obou transformerů (vizuální i textový) se jedná pouze o část *encoder*, sloužící k redukci dimenzionality vstupu a extrakci podstatných vlastností. V případě vision transformeru se zde opět nejedná o příliš specifickou variantu, jde o standardní implementaci vycházející z článku [6]. Charakteristická vlastnost architektury CLIP, popsaná v článku [17], je způsob kombinace těchto textových a obrazových dat. Výstup celé takto kombinované sítě obsahující dva separátní modely je spočten na základě podobnosti jejich výstupních hodnot, jak je také znázorněno na obrázku 4.5.

Klíčová změna v implementaci CLIP oproti podobným architekturám je ve způsobu předtrénování. V případě CLIP je namísto separátního tréninku každé z těchto podsítí využit přístup založený na hodnocení podobností predikovaných výstupů vizuálního i jazykového transformeru. Předmětem hodnocení je, aby byly v matici podobností maximalizovány hodnoty na diagonále (na obrázku 4.5 znázorněno modře) a zároveň minimalizovány hodnoty mimo diagonálu. Výhodou této architektury není výrazná změna principu fungování její obrazové části, v případě varianty s vision transformerem tato implementace nepřináší mnoho nových věcí. Potenciál v architektuře CLIP je na druhou stranu dostupnost modelů předtrénovaných na velkém množství dat. V této práci byly prováděny experimenty s předtrénovanou variantou *OpenCLIP* od organizace *OpenAI* a variantou *LAION*[4].

[4]Odkaz na stránky organizace LAION: https://laion.ai



Obrázek 4.5: Diagram demonstrující princip architektury OpenCLIP. Obrázek je převzat z článku představující architekturu OpenCLIP. [17]

Jako všechny dříve zvolené implementace, i tato byla nejprve podrobena prvotním testům. Pro tyto účely byl extrahován pouze zmiňovaný vision transformer. Detailnější experimenty s celou architekturou CLIP budou popsány v kapitole 4.6.1. Prvotní experimenty byly prováděny s předtrénonvanou variantou *OpenCLIP*. Parametry prováděných experimentů byly zvoleny stejně jako u dřívějších pokusů, jak je také popsáno v kapitole 4.2.

4.3.2 Výsledky testů

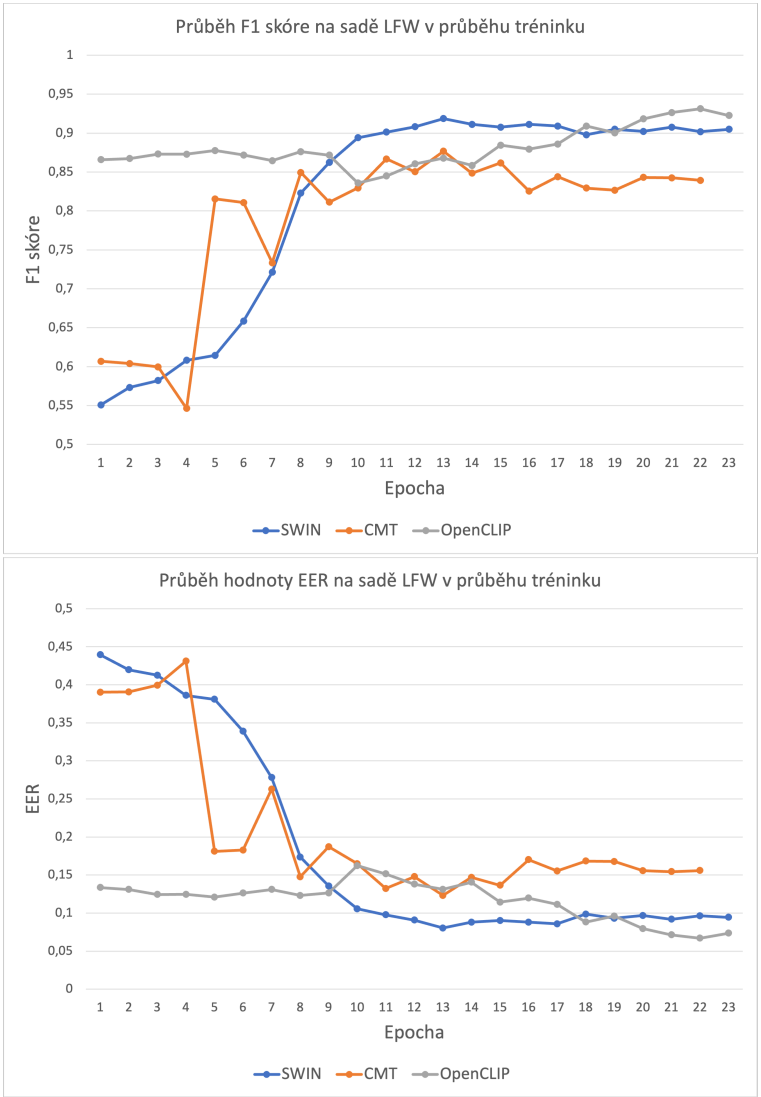
Jak ukazuje tabulka 4.3 a graf 4.6, předtrénovaná varianta modelu OpenCLIP dosáhla nejlepších výsledků. S maximální hodnotou F1 skóre $0,931$ předčila výsledky dříve zvolené architektury SWIN transformer, dosahující maxima pouze $0,918$. Ačkoli je transformer OpenCLIP počtem trénovatelných parametrů (celkem 139M) více než dvojnásobný, oproti doposud největší síti BiFormer je velmi efektivní. Dle autorů model OpenCLIP dosahuje necelých 15G FLOPs (celý model) a pokusné trénování ukázalo bezkonkurenční rychlost trénování oproti dřívějším variantám. Mimo své přednosti z pohledu efektivity implementace CLIP nabízí prostor pro experimenty s kombinací textových a obrazových dat a tudíž bude hlavním předmětem následujících pokusů.

Architektura	MAX F1	Počet parametrů	GFLOPs
OpenCLIP - visual	0,931	87.8 M	8,8
Swin	0,920	28 M	4,5
CMT	0,877	11 M	1,0

Tabulka 4.3: Tabulka s parametry každé z architektur a úrovní přesnosti dosažené při pokusném trénování. Hodnota MAX F1 je maximální dosažená úroveň metriky F1.

Na druhou stranu, architektura CMT neukázala žádný významný potenciál. Na výsledcích testů je možné pozorovat, že alternativní metody napodobující attention mechanismus

za pomocí konvolučních vrstev nedokážou plně nahradit attention vrstvy. Z těchto úvodních experimentů vyplývá, že v případě využití hybridních sítí je lepší zvolit architektury, které stále zachovávají alespoň menší množství attention vrstev. Takovýmto případem je například model SMT, který v úvodním testu ukázal výsledky srovnatelné s modelem Swin. Výsledky experimentu je možné vidět na následujících grafu 4.6 a tabulce 4.3.



Obrázek 4.6: Grafy zobrazující průběh metrik F1 a EER modelů CLIP a CMT spolu s referenčním modelem Swin Transformer.

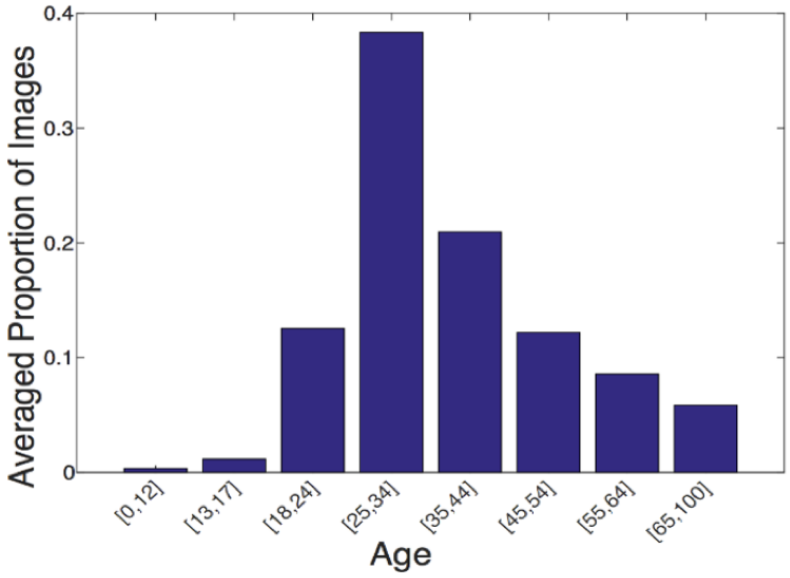
4.4 Využití různých datových sad

Druhou a zároveň nejpodstatnější skupinou experimentů byla analýza volby trénovacích datových sad. Jak bylo dříve popsáno, výchozí datovou sadou byla zvolena sada *VGG-Face2*. Volba trénovacích dat, jak je také ukázáno v následujících experimentech, měla nezanedbatelný dopad na přesnost predikcí modelu. Některé z vybraných trénovacích dat obsahovaly mimo jiné také alternativní anotace, jako např. pohlaví, barva vlasů, apod. Využití těchto

specifických anotací bude podrobněji popsáno v kapitole 4.6.1, která se zabývá tzv. *multitask* učením. Na druhou stranu, tato kapitola se soustředí primárně na výsledky experimentů s jednotlivými datovými sadami bez využití těchto alternativních anotací. Jejich srovnání bylo prováděno s pomocí předtrénovaného vision transformeru, extrahovaného z architektury OpenCLIP. Důvod výběru této architektury byla její rychlá konvergence a tudíž i menší časová náročnost experimentů.

4.4.1 Výchozí datová sada VGG-Face2

Datová sada VGG-Face2 je jedna z velmi používaných veřejně dostupných sad určených k trénování na úlohách rozpoznávání tváří. Sada obsahuje 3,31 miliónů fotografií celkem 9131 lidí. Na každého člověka připadá průměrně 362,6 obrázků. Průměrné rozlišení fotografií je 137x180 px. Svou velikostí a kvalitou se řadí mezi kvalitnější sady. Z pohledu distribuce věkových kategorií má tato sada spíše log-normální rozdělení, což je možné vidět na grafu distribuce z obrázku 4.7. Nejpočetnější kategorií jsou zde osoby věku 25-34 let. Součástí této sady jsou mimo obrazových dat k dispozici také anotace o pohlaví osob. Díky popularitě VGG-Face2 existuje celá řada modifikací a vylepšení. Jednou z nich může být například dostupnost alternativních anotací jako barva vlasů či označení, zda osoba na fotografii nosí brýle.



Obrázek 4.7: Rozložení věkových kategorií fotografií datové sady VGG-Face2. Převzato z článku [2].

V provedených testech se modely, trénované s pomocí VGG-Face2, umístily na přijatelných hodnotách přes $0,9$ Accuracy na sadě LFW. Maximální dosažená úroveň Accuracy byla $0,934$ s vision transformerem CLIP a $0,943$ s využitím multitask učení spolu s textovou částí transformeru CLIP (viz. 4.6.1). V porovnání s výsledky dosaženými využitím datové sady MS1Mv3 je kvalita VGG-Face2 znatelně horší. Využití této varianty je přesto vhodné, obzvláště v počátečních fázích. Dosažená úroveň přesnosti je pro účely srovnání dostačující vzhledem k velikosti této sady, která je 2x menší než dříve zmiňovaná MS1Mv3.

4.4.2 Uměle vylepšená data

Jedním z dalších provedených experimentů bylo využití uměle vylepšených dat. Konkrétně se jednalo o použití datové sady VGG-Face2-HQ [5]. Tato datová sada obsahuje fotografie z VGG-Face2 vylepšené pomocí modelu GFPGAN [24]. Jedná se o fotografie s větším rozlišením, a to konkrétně s rozměry 500x500 px. Mimo takovéto vylepšení je sada totožná se svou originální variantou. Využití těchto dat bylo cíleno převážně na architektury Swin transformer a FLatten transformer. Ačkoli mají větší a detailnější obrázky potenciál k lepším výsledkům, tato vylepšená data obsahují řadu vad a nepřesností, což taky prokázaly samotné výsledky vlastních experimentů, kde průměrné hodnoty validačních metrik ukazovaly na přibližně 10% pokles. Jednu z přítomných vad je možné vidět na obrázku 4.8. V tomto případě se jednalo o běhy trénování s menšími počty epoch, a to z důvodu časové náročnosti zpracování obrázků s velkým rozlišením. Tyto experimenty byly primárně prováděny za účelem průzkumu škálovatelnosti architektury FLatten transformer, který je detailněji popsán v kapitole 4.6.2.

Obrázek 4.8: Ukázka srovnání fotografií tváří z datové sady VGG-Face2 a VGG-Face2-HQ. Horní fotografie pochází z originální sady zatímco spodní dvě jsou odpovídající fotografie, uměle „vylepšené"pomocí GFPGAN. Na pravé straně je možné vidět jednu z vadných fotografií vzniklých špatnou detekcí obličeje (detekována byla tvář v pozadí).

4.4.3 Cross-age datové sady

Při výběru vhodné datové sady byl brán ohled mimo jiné i na fotografie totožných lidí s vetším časovým rozestupem. Tato diverzita v datech má předpoklad k natrénování robustnějšího systému. Věková variabilita přidává do dat různorodost délky vlasů, přítomnost brýlí a podobné aspekty, které mohou při trénování přimět neuronovou síť se soustředit na klíčovější parametry obličejů pro jejich dobré rozpoznávání.

Pro účely tohoto experimentu byla využita datová sada *Cross-Age Celebrity Dataset* (CACD). Tato sada obsahuje fotografie daných osob, pořízené s rozestupy v řádech až desítek let. Tato sada se skládá z 163446 fotografií s celkem 2000 různými identitami.

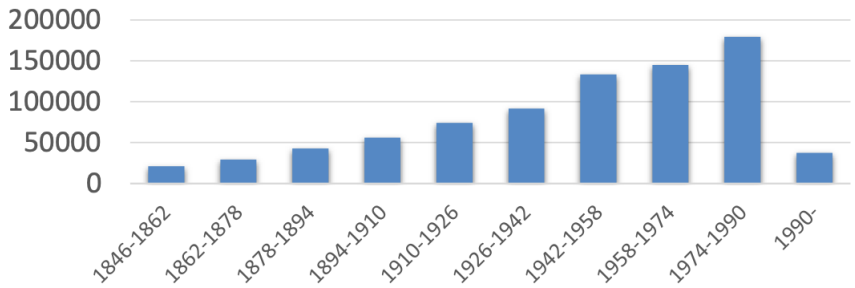
[5] Detailnější informace a odkaz na stažení je možné nalézt zde: https://github.com/NNNNAI/VGGFace2-HQ

Trénování probíhalo způsobem tzv. *finetuning*, kde bylo využito modelu předtrénovaného z předchozích experimentů.

Výsledky experimentu s datovou sadou CACD neukázaly žádný významný posun v ohledu na výsledné evaluace. Na testech s datovou sadou LFW toto řešení dosahovalo průměrně horších výsledků než výchozí bod, ze kterého bylo trénování prováděno. Tato využitá datová sada je výrazně menší než ty, na kterých bylo prováděno předtrénování. Kvalita přiřazení obličejů jednotlivým entitám této sady byla v některých případech pochybná.

4.4.4 Datová sada MS1Mv3

Datová sada MS1Mv3 je v porovnání z doposud zmíněnými trénovacími sadami největší s celkovým počtem fotografií okolo 5,3 milionu a počtem identit okolo 92 tisíc. Jedná se o jednu z větších a nejkvalitnějších veřejně dostupných trénovacích sad. Dle výsledků experimentů z článku *ArcFace* [4] dosahují architektury natrénované na datové sadě MS1Mv3 o více než 9% větší úspěšnosti v porovnání s modely trénovanými na sadě VGG-Face2. Vyváženost dat je zde rovněž znatelně lepší, jak je mimo jiné možné vidět na obrázku 4.9.



(c) Date of Birth

Obrázek 4.9: Ukázka rozdělení dat narození osob z fotografií v datové sadě MS1M. Převzato z článku [8].

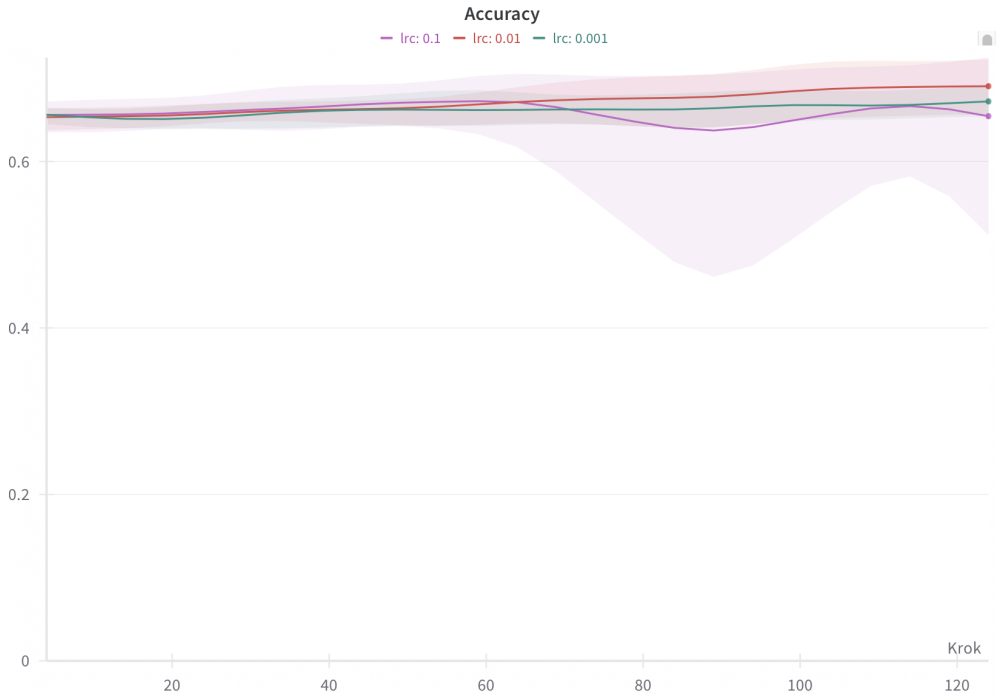
Výsledky trénování na této sadě byly významně lepší než na dříve využité sadě VGG-Face2, a to i v případě využití jejich alternativních anotací pro multitask učení. Maximální dosažená hodnota Accuracy na validační sadě LFW činila rekordních $0,9898$, jak je také ukázano v kapitole 4.7.1. V rámci všech ostatních experimentů využívajících CosFace loss s touto sadou bylo možné dosáhnout hodnot přes $0,98$ Accuracy. Kombinace této sady spolu s CosFace loss funkcí představovala nejlepší výsledek z pohledu validace na datové sadě LFW.

4.5 Ladění hyperparametrů

Již v průběhu prvotních experimentálních tréninků bylo znatelné, že daná konfigurace je v případě využití vision transformerů velice citlivá na zvolené hyperparametry. Mezi hyperparametry v tomto případě patří hodnoty *learning rate*, volba poměrů jednotlivých úloh při *multitask* učení, či velikost výstupního vektoru. Nejprve byla dle experimentů empiricky zvolena dostatečně optimální kombinace pro pokračování s prvotními pokusy a zároveň byl nalezen obor rozumných hodnot jednotlivých hyperparametrů. V pozdějších fázích byl proveden kombinovaný průzkum všech rozumných kombinací za pomocí krátkých experimentů,

přičemž nejvýhodnější varianta byla zvolena dle následného otestování ve větším měřítku. Ačkoli výběr hyperparametrů znatelně ovlivňoval výsledky trénovacích běhů, důkladnější studie optimálních parametrů by byla znatelně časově náročná.

Automatizované testování probíhalo pokusným trénováním na malé podmnožině originální sady dat. Z důvodu vysokého počtu kombinací a samotné náročnosti těchto architektur bylo testování prováděno systematicky na této zmenšené sadě. Pro získání lépe demonstrujících výsledků a analýzu vlivu hyperparametrů na širší rozsah trénování bylo testování prováděno s modely jak předtrénovanými z dřívějších experimentů, tak i s modely nepředtrénovanými. Pro získání lepšího statistického vzorku byl každý běh zopakován vždy alespoň třikrát na náhodné podmnožině trénovacích dat. Počet trénovacích epoch byl zvolen na hodnotu 20. Velikost takovéto zmenšené datové sady činila 100 000 vzorků. Ukázka jednoho z výstupů testu je zobrazena na obrázku 4.10.



Obrázek 4.10: Obrázky znázorňují průběh metriky Accuracy během pokusného trénování při ladění hyperparametrů. Jedná se konkrétně o seskupení výsledků podle hodnoty learning rate pro vrstvu predikcí vzorových vektorů. Popisky *lrc* představují hodnotu tohoto parametru, přičemž křivka odpovídající hodnotě $0,01$ vykazuje nejlepší výsledky. Z důvodu velkého množství jsou zobrazeny jsou pouze vybrané volby tohoto parametru.

Mezi testované hyperparametry a zvolené hodnoty patřily:

- Hodnota *learning rate* modelu

  Obor hodnot: $(1 \cdot 10^{-5}), \ldots, (1 \cdot 10^{-8})$

- Hodnota *learning rate* vzorových vektorů tříd

  Obor hodnot: $(1 \cdot 10^{-1}), (1 \cdot 10^{-2}), \ldots, (1 \cdot 10^{-5})$

- Délka výstupního vektoru

  Obor hodnot: 512, 1024

- Poměr hodnot *ArcFace loss* a hodnoty loss funkce klasifikátoru

  Obor hodnot: $0,6, \ldots, 0,9$

- Volba plánovače *learning rate scheduler*

- Volba parametrů ve fázi *warmup*

Provedené testy ukázaly převážně na důležitost poměru learning rate modelu a vrstvy vzorových vektorů, která je součástí implementace loss funkce. V úvodních experimentech byla využita na základě empirických experimentů hodnota learning rate $1{\cdot}10^{(-6)}$ pro páteřní vision transformery. Tento fakt se také potvrdil na těchto experimentech. Největší vliv na konvergenci měla následná volba této hodnoty pro vrstvu vzorových vektorů. Jak je možné pozorovat na grafech z obrázku 4.10, nejoptimálnější volbou této hodnoty byla varianta $1 \cdot 10^{(-2)}$. Důležitá byla rovněž konfigurace trénování ve fázi *warmup*. Nalezená optimální konfigurace je shrnuta v tabulce 4.4.

Parametr	Fáze tréninku	Hodnota parametru
Backbone learning rate	mid	$(1 \cdot 10^{-5})$–$(1 \cdot 10^{-6})$
Loss learning rate	mid	$(1 \cdot 10^{-1})$–$(1 \cdot 10^{-2})$
Backbone learning rate	warmup	$1 \cdot 10^{-5}$
Loss learning rate	warmup	$1 \cdot 10^{-1}$
Délka výstupního vektoru	–	512
Poměr loss funkcí	–	$0,85/0,15$
Plánovač learning rate	–	CosineAnnealing

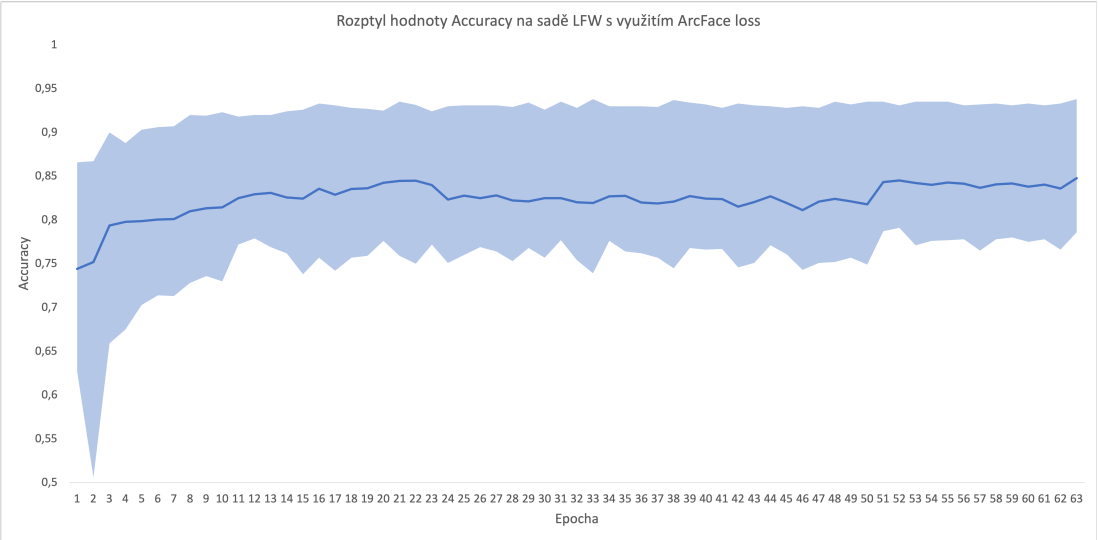
Tabulka 4.4: Tabulka nejoptimálnější kombinace hyperparametrů nalezená experimenty. Fáze tréninku *warmup* a *mid* představují počátek tréninku a střední fázi.

4.5.1 Využití CosFace loss funkce

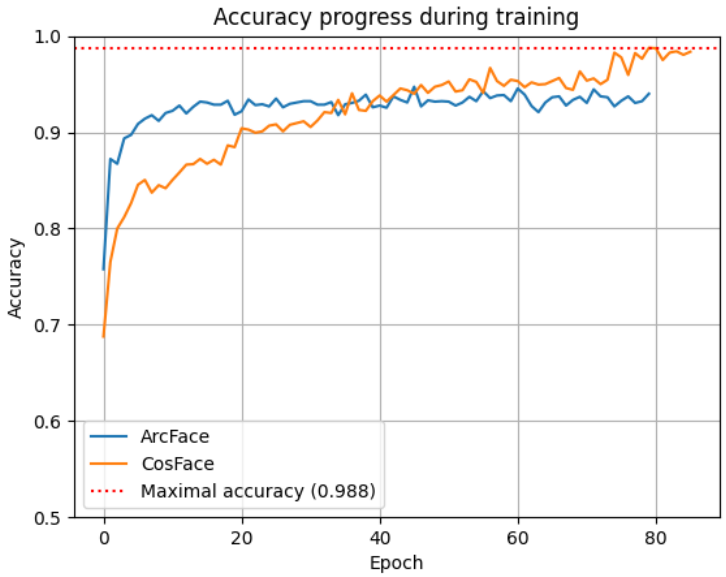
Ve většině provedených experimentů byla při trénování rozpoznávání tváří využita ArcFace loss funkce, konkrétně její implementace z Python balíčku `pytorch_metric_learning`. Tato volba se zdála být nejoptimálnější, a to hned z několika důvodů. Důvodem byla popularita ArcFace loss v implementacích rozpoznávání tváří s využitím konvolučních sítí a také výsledky prezentované autory. Přes tyto výhody se v rámci této práce nepodařilo natrénovat dostatečně robustní systém pro rozpoznávání tváří, který by využíval právě tuto funkci spolu s architekturou stylu vision transformer. S využitím ArcFace loss docházelo k problémům s konvergencí trénování a celkovou schopností učení kvalitních predikcí. Žádný z vision transformerů trénovaný pomocí této loss funkce nepřesáhnul hodnoty větší než $0,95$ Accuracy skóre na validační datové sadě LFW, což je možné pozorovat také na obrázku 4.11.

Problém s konvergencí bylo možné vyřešit s použitím CosFace loss funkce jako náhrady za dříve používanou ArcFace. Jak je možné vidět na srovnání z obrázku 4.12, tato loss funkce vedla sice k pomalejší, ale stabilnější konvergenci. Takto trénované modely ve výsledném srovnání dosáhly znatelně lepších výsledků ve všech testovaných metrikách. Zajímavým poznatkem byly výsledky validací modelů natrénovaných s pomocí ArcFace na sadě LFW

z pohledu průměrné kosinové podobnosti predikovaných vektorů. Průměrná podobnost na validační sadě činila $0,954$, což poukazuje na nerovnoměrnost rozložení predikcí v prostoru výstupních vektorů. Při inspekci naučených vzorových vektorů byla tato podobnost mezi jednotlivými třídami trénovacích dat na druhou menší než v případě modelů trénovaných s funkcí CosFace.

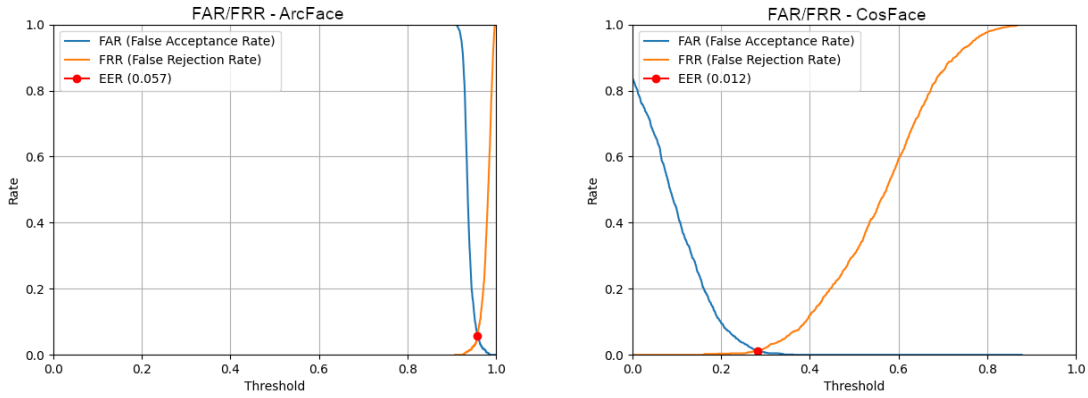


Obrázek 4.11: Obrázek znázorňuje rozptyl hodnot validační metriky Accuracy všech provedených experimentů s využitím ArcFace loss funkce. Na obrázku si je možné všimnout, že maximum nepřekračuje hodnotu $0,95$.



Obrázek 4.12: Graf na obrázku ukazuje rozdíl progresu přesnosti predikcí modelu trénovaného s pomocí ArcFace a CosFace loss funkcí. V případě ArcFace je zde možné vidět znatelnou stagnaci při učení.

Problémy s konvergencí byly mimo jiné znatelné taky při detailnějším pohledu na metriku EER, konkrétně při pohledu na rozhodovací hranici, kde se bod EER nachází. Z grafů na obrázku 4.13 je možné vidět, že přechod křivek FAR a FRR je v případě ArcFace loss velmi strmý a bod EER leží na příliš vysoko. Tento fakt naznačuje, že takovýto trénink opravdu vedl k nedostatečné separaci tříd v prostoru výstupních vektorů. Z důvodů problémů s dosažením rozumné míry přesnosti byla metoda *ArcFace* testována na větším množství experimentů s využitím různých kombinací hyperparametrů. Na obrázku 4.11 je ale možné vidět, že žádný z experimentů nebyl dostatečně úspěšný.



Obrázek 4.13: Na obrázcích je možné vidět srovnání křivek FAR a TAR modelů trénovaných s pomocí ArcFace loss funkce a CosFace loss funkce. Graf příslušící metodě ArcFace poukazuje na špatnou kvalitu predikcí sítě.

4.6 Experimenty s architekturami

Po odladění optimálních hyperparametrů následovaly experimenty se samotnými architekturami. Záměrem těchto experimentů bylo využití potenciálu jak jejich implementací, tak i dostupných alternativních trénovacích dat. Cílem těchto experimentů nebylo výrazně změnit podobu architektury, nýbrž spíše podpořit schopnost učení kvalitní predikce. Experimenty byly omezeny pouze na architektury *CLIP*, *FLatten* a *SWIN tranformer*, a to z důvodu jejich potenciálu proudícího ze specifik konkrétních implementací.

4.6.1 Multitask learning

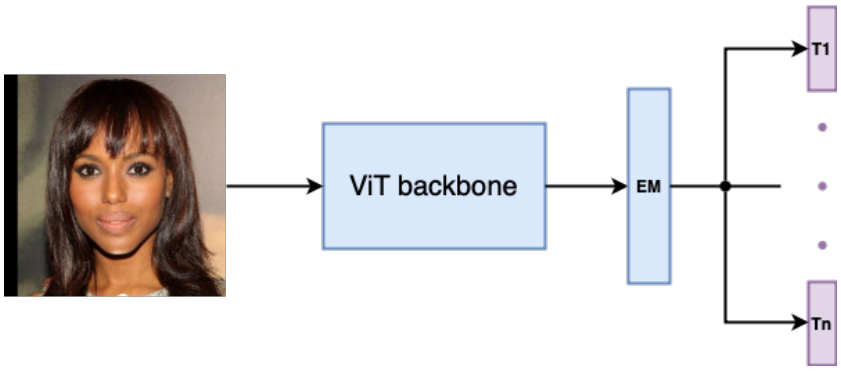
Trénovací datová sada *VGG-Face2* obsahuje mimo obrázků a anotací jejích příslušnosti dané osobě také informace o pohlaví osob na fotografiích. Mimo to, pro tuto sadu již existují také alternativní anotace vytvořené za pomocí předtrénované sítě *MTCNN*[30], běžně používané k předzpracování obrazových dat se zaměřením na lidské tváře.

Dostupnost takovýchto informací nabízí možnost jejich využití k podpoření tréninku sítě určené k rozpoznávání obličejů. Podpoření jejího učení je možné s využitím tzv. *multitask* učení, kde je model učen predikovat nejen výstupní kódování, ale rovněž rozpoznávat tyto charakteristiky lidí. Potenciál tohoto přístupu tkví ve faktu, že některé z těchto klíčových informací mohou hrát důležitou roli v unikátnosti obličejů. Přestože zmiňovaná datová sada obsahuje širší množství alternativních anotací, pro účely tohoto experimentu nebyly využity všechny tyto anotace, nýbrž jen jejich podmnožina. Při jejich výběru byly brány v potaz

aspekty jako kvalita anotací a také míra relevance těchto informací k tématu rozpoznávání obličejů. Mezi využité anotace patřily:

- Pohlaví (pouze binární označení)
- Barva vlasů (kategoricky, např: hnědé, blond, šedé, ...)
- Délka vlasů (dlouhé/krátké/žádné)
- Informace zda osoba nosí brýle (sluneční/dioptrické/žádné)
- Informace zda osoba nosí pokrývku hlavy (binární označení)
- Informace zda osoba má vousy (binární označení)
- Informace zda má osoba otevřená ústa (binární označení)

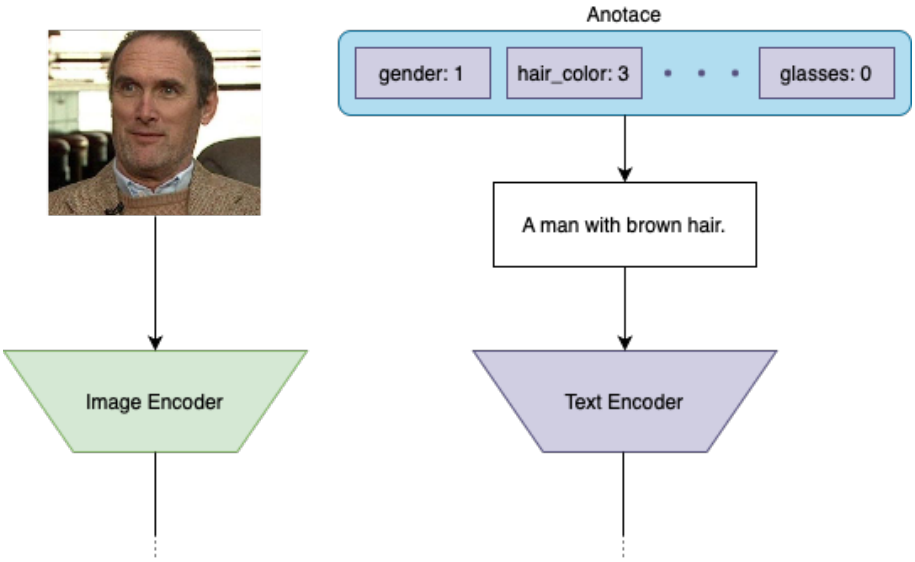
Změny v architektuře v případě těchto experimentů zahrnovaly přidání lineárních vrstev pro každou ze zpracovávaných úloh. Schéma této implementace je vyobrazeno na obrázku 4.14. Každá úloha zde obsahuje dedikovanou loss funkci. V případě binárních anotací byla zvolena funkce `BinaryCrossEntropyLoss`, v případech s více třídami (např. barva vlasů) se jednalo o `CrossEntropyLoss`. Tyto úlohy nebyly během tréninku učeny současně. Trénování probíhalo náhodným výběrem jedné z úloh a zpracováním jedné dávky dat. Konzistence vybrané úlohy v případě využití paralelismu byla realizována výběrem fixní inicializační konstanty náhodného generátoru. V každém takovém kroku byla využita dedikovaná funkce loss spolu s ArcFace/CosFace loss. Výběr úlohy byl prováděn způsobem rulety dle důležitosti úlohy. Tato důležitost byla stanovena ručně dle vypozorované kvality anotací a předpokládaného potenciálu. Příkladem může být úloha rozpoznávání pohlaví osoby, které byla přidružena největší důležitost. Důvodem byla dostupnost ručních anotací v datové sadě VGG-Face2 a zároveň vysoký potenciál této informace při rozeznávání osob.



Obrázek 4.14: Diagram implementace multitask učení. Vrstva označená jako *EM* značí vrstvu pro generování výstupního vektoru, jak bylo popsáno v dřívějších experimentech. Vrstvy *T1* až *Tn* představují vrstvy dedikované k predikci dle dané alternativní anotaci.

Mimo tuto změnu v posledních vrstvách sítě byl prováděn také experiment zaměřený na specifika architektury *CLIP*, popsané dříve v kapitole 4.3.1. Konkrétně se jednalo o využití kombinace textových a obrazových dat způsobem ukázaným na obrázku 4.15. Díky dostupným anotacím bylo možné vytvořit jednoduché textové popisy lidí, které následně sloužily

jako vstup jazykové části modelu CLIP. Do trénování byla mimo CosFace loss funkci zahrnuta také kontrastní loss funkce, kterou využívali autoři k předtrénování modelu. Účelem bylo přimět model predikovat kódování obrazu, která odpovídají zakódovaným textovým popisům. Tomu poté napomáhaly dodatečné úlohy predikce alternativních anotací, které měly za úkol přimět model predikovat tyto kódování v podobě, která explicitněji vyjadřují tyto vlastnosti tváří. Poměr zpětné vazby obou z loss funkcí byl stanoven na 1:4. Takovýto poměr byl zvolen s myšlenkou, že požadované chování modelu stále nejlépe vyjadřuje CosFace loss funkce, a tudíž si její poměr při zpětné propagaci musí stále zachovat svou vysokou prioritu.



Obrázek 4.15: Demonstrace generování textových dat a jejich využití v architektuře CLIP.

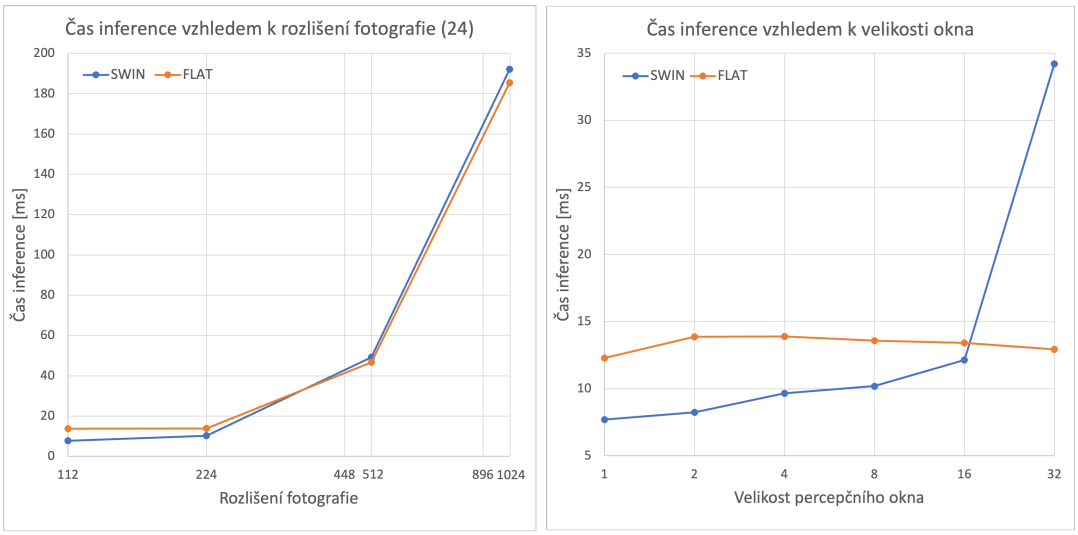
Přes tuto architekturální výhodu a doplnění o učení způsobem *multitask learning* se s touto konfigurací nepodařilo v rozumné trénovací době (necelých 200 epoch) dostat kompetetivní výskedky v porovnání s jinými provedenými experimenty. Důvodem může být horší kvalita dat sady VGG-Face2 anebo také fakt, že alternativní anotace nebyly vytvořeny ručně. Ať se jedná o jakýkoli z těchto důvodů, tento experiment přesto prokázal posun oproti konfiguraci bez využití alternativních dat. Výsledná úroveň metrik Accuracy a F1 dosáhla úrovně $0,943$ a metrika EER $0,058$. Při pohledu na trénovací křivku je ale zjevné, že s využitím delšího tréninku by bylo možné tyto hranice dále překonat.

4.6.2 Škálovatelnost FLatten transformeru

Hlavním potenciálem architektury *FLatten transformer*, detailněji popsané v kapitole 3.2.4, je její údajná lepší škálovatelnost s ohledem na počet tokenů, zpracovávaných attention vrstvami. U běžných transformerů, které počítají globální attention nad celým vstupním obrazem je proto klíčovým parametrem rozměr výřezů v obraze. Jeho rozměr v tomto případě rozhoduje o úrovni detailu v obraze, které dokáže takovýto model rozeznat. Určuje ale zároveň jeho výpočetní složitost, která v tomto ohledu roste kvadraticky. Ideální reprezentace by u vision transformerů představovala reprezentaci každého pixelu jediným tokenem, což by ovšem v praxi vyžadovalo velmi malé rozlišení fotografií, aby byl proces rozumně spočítatelný. Implementace FLatten transformer vychází z architektury Swin transformer,

a tudíž nezpracovává tento globální attention mechanismus nad celým vstupním obrazem. Škálovatelnost zde tedy závisí spíše na velikosti okna z terminologie implementace Swin. I v tomto případě ovšem snížení velikosti okna vede ke kvadratickému nárůstu výpočetní složitosti architektury. Tato složitost, jak bylo popsáno v kapitole 3.1.4, proudí z principu implementace self attention vrstvy, která je základním stavebním kamenem všech vision transformerů. Architektura *FLatten transformer* se v tomto případě snaží o snížení dopadu zpracovávané sekvence na výpočetní náročnost těchto vrstev. Tato vlastnost tedy představuje možnost zvětšení percepčního okna neuronové sítě na bázi Swin transformeru s menším dopadem na její náročnost.

Využití škálovatelnosti bylo testováno jednak zvětšením percepčního okna, a zároveň pokusy s rozměrem vstupní fotografie. Škálovatelnost byla testována pokusnou inferencí vždy na sérii 100 fotografií. Před každým testem bylo provedeno 10 dopředných průchodů pro stabilizaci času výpočtu. Z naměřené doby výpočtu byla odhadnuta průměrná doba zpracování jedné fotografie. Výsledky analýzy poukazují na fakt, že podoba rozlišení hraje roli pouze v případech s nízkou délkou vektoru představujícího token. Na levém grafu obrázku 4.16 je možné vidět, že délka inference byla nižší až v případě fotografie s rozměrem 1024 px. Co se týče závislosti délky inference na velikosti percepčního okna, je zde viditelná režie a k demonstraci benefitů této sítě dochází až při velkých rozměrech tohoto „zorného pole". Vlastnosti této architektury jsou tedy využitelné pouze v případě forografií s velkým rozlišením (>512) a tudíž je tato implementace pro běžně dostupné trénovací sady nevhodná.



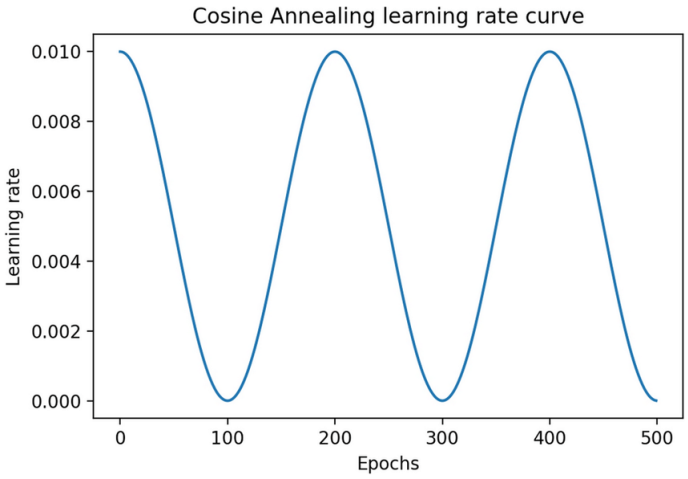
Obrázek 4.16: Na obrázku je možné vidět výsledky testů škálovatelnosti FLatten transformeru. Levý graf ukazuje škálovatelnost vzhledem k rozlišení vstupních dat s volbou velikosti vnitřní reprezentace tokenu o délce 24. Vpravo je možné vidět závislost času inference na velikosti percepčního okna transformeru.

4.7 Dlouho trvající trénování

Z důvodu kombinace omezených výpočetních prostředků a výpočetní náročnosti vision transformerů nebylo možné všechny experimenty provést v plnohodnotném měřítku. Z provedených testů bylo zjevné, že volba pozvolnějšího, ovšem podstatně delšího tréninku byla

pro achitektury vision transformer pro tyto účely výhodnější. Z doposud popsaných experimentů bylo zároveň zjevné, že ačkoli se v pozdějších fázích tréninku progres trénování výrazně zpomalil, prodloužení trénování mělo stále potenciál k lepším výsledkům. Z tohoto důvodu byly provedeny pokusy s pozvolným trénováním za účelem nalezení maximální úrovně přesnosti predikcí modelu za daných podmínek.

Tento pokus byl prováděn s nejnadějnějšími konfiguracemi – tudíž s předtrénovanými modely *Laion CLIP* a *Swin transformer*. Po srovnání výsledků těchto dvou architektur byl proveden jeden finální trénink s architekturou, dosahující lepších výsledků (CLIP). Tento finální trénink byl proveden s jeho hlubší variantou. Pro natrénování byla využita nejvýhodnější konfigurace, popsaná také v kapitole 4.2.1, konkrétně zobrazená v tabulce 4.4. Toto trénování bylo rozděleno do tří fází s postupnou regulací hodnoty *learning rate*. Ve všech třech fázích byl využit plánovač learning rate – *CosineAnnealingLR* (viz. obrázek 4.17). Úvodní a střední fáze tréninku byla provedena dle zmíněné konfigurace. Poslední fáze tréninku zahrnovala snížení všech hodnot learning rate 10x a byla provedena až po viditelném zpomalení progresu trénování.



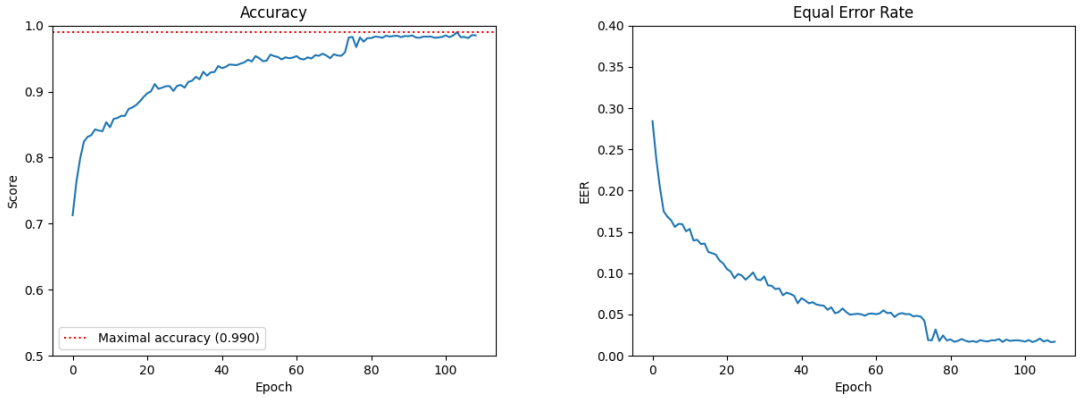
Obrázek 4.17: Ukázka průběhu plánovače *CosineAnnealingLR*.

S využitím dříve zmíněného procesu bylo možné dosažení výsledků, které jsou srovnatelné s dnešními state-of-the-art řešeními. Přes délku trvání takovéhoto trénování je zde ovšem stále prostor pro zlepšení. Jak je možné vidět také na obrázku 4.19, kde i po více než 100 epochách tréninku je stále možné pozorovat, že nedochází k úplnému zastavení progresu trénování. Tudíž s dalším laděním by bylo možné dosáhnout lepších výsledků. V následujícím textu budou demonstrovány výsledky tohoto finálního experimentu.

4.7.1 Model CLIP

Dřívější experimenty s architekturou CLIP ukázaly na výjimečnou kvalitu předtrénovaných modelů této implementace. Úrovně validačních metrik v případě využití předtrénované varianty LAION již na počátku tréninku startovaly na solidních hodnotách. Dosažení $> 0,9$ Accuracy na datové sadě LFW bylo možné v rekordním počtu epoch. Dlouhý a pozvolný trénink měl v tomto případě úspěch. Tomu ovšem musela předcházet fáze *warmup* o délce alespoň deseti epoch s násobně vyšší hodnotou learning rate. Toto pozvolné trénování bylo prováděno s menší variantou s velikostí okna 32x32 a díky úspěchu také s větší variantou a detailnějším rozměrem okna 16x16.

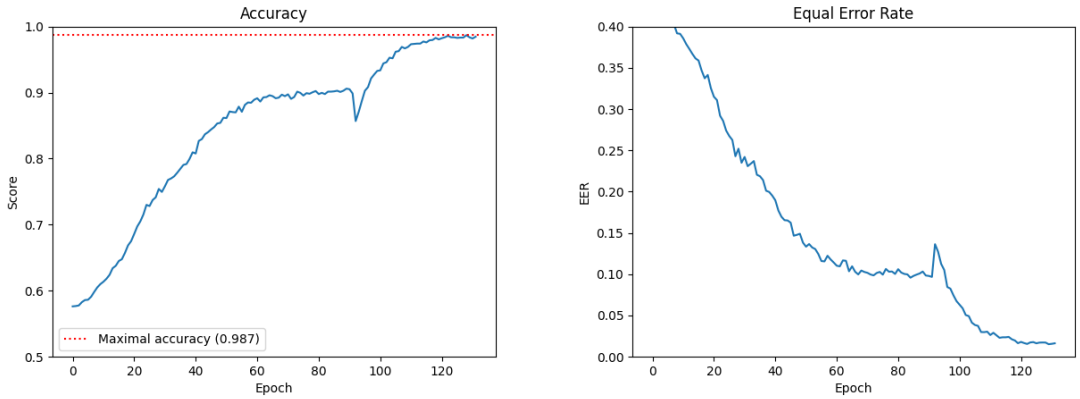
Výsledky metrik architektury s větším oknem dosahovaly až k hodnotě 0,9898, jak je také viditelné na grafech z obrázku 4.18. Minimální dosažená hodnota EER byla 0,015. V tomto případě se jednalo o variantu CLIP s velikostí okna 32x32 pixelů. Délka tréninku se pohybovala v řádech dní. V následném experimentu s hlubším modelem CLIP bylo následně možné dosáhnout úrovně srovnatelné s dnešními state-of-the-art modely. Maximální hodnoty se zde pohybovaly nad hranicí 0,99 Accuracy s maximální hodnotou ,9941. Detailnější výsledky jsou uvedeny v kapitole 4.8.



Obrázek 4.18: Průběh metrik Accuracy (vlevo) a EER (vpravo) během prodlouženého trénování vision části transformeru CLIP.

4.7.2 Model Swin Transformer

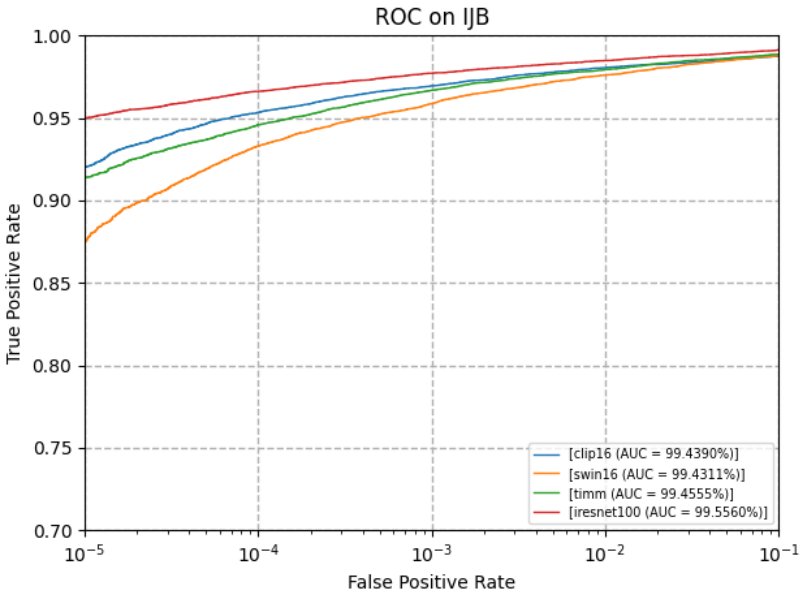
Výsledky prodlouženého trénování výrazně demonstrovaly rozdíly předtrénované varianty v porovnání s LAION CLIP modelem. Výchozí přesnost je na znatelně nižší úrovni a konvergence je viditelně pomalejší. Přesto se však s touto architekturou stále podařilo dosáhnout solidních výsledků, a to i za předpokladu výrazně nižší kvality předtrénovaného modelu.



Obrázek 4.19: Průběh metrik Accuracy (vlevo) a EER (vpravo) během prodlouženého trénování architektury Swin Transformer. Z grafu je možné poznat, že ani po více než 100 epochách pozitivní progres u obou metrik neustal. Z grafu je také znatelný přechod na poslední fázi trénování zastavením.

4.8 Shrnutí provedených experimentů

Provedené experimenty ukázaly výrazné výhody předtrénovaných variant modelů typu vision transformer. Zde architektura CLIP ukázala své přednosti v podobě modelů organizace LAION předtrénovaných na velikých datových sadách. Solidních výsledků bylo v případě těchto předtrénovaných modelů možné dosáhnout v rekordním čase, kdy obě testované varianty se správnou konfigurací dosáhly hodnot $> 0,9$ Accuracy na sadě LFW již v prvních třech epochách trénování. Provedené experimenty rovněž prokázaly výhody multitask učení a využití textových anotací pro vytěžení maxima z této implementace transformeru. V tomto případě se jednalo o zlepšení výsledků o více než 1% i přes omezenou kvalitu trénovacích dat. Testy s hyperparametry demonstrovaly důležitost volby mezi funkcemi ArcFace loss a CosFace loss, kde CosFace loss vyšla jako nejlepší varianta při učení vision transformerů na úloze rozpoznávání tváří. Shrnutí všech experimentů naznačuje, že tento typ architektur má potenciál k překonání dodnes dominujících konvolučních neuronových sítí v tomto oboru. Jejich využití ovšem není tak přímočaré a vyžaduje podrobnější odladění hyperparametrů a delší dobu tréninku. Přesto s využitím správné kombinace těchto faktorů, definovaných v rámci této práce, je možné dosáhnout kompetetivních výsledků, a to i v relativně krátkém čase. V grafu 4.5 je možné vidět výsledky dosažené v rámci provedených experimentů spolu se srovnáním s dnešními state-of-the-art implementacemi *InsightFace*[6] (modely *IResNet*) a dřívějšími pokusy o adaptace vision transformerů v oblasti rozpoznávání obličejů (model *TIMM*)[7]. Modely *CLIP-V-32* a *CLIP-V-16* představují vizuální část architektury CLIP s velikostí attention okna 16x16 a 32x32. Podrobné srovnání výsledků verifikace vybraných modelů na datové sadě IJB-C je možné vidět na obrázku 4.20.



Obrázek 4.20: Graf znázorňuje srovnání modelů na datové sadě IJB-C za pomocí metriky TPR@FPR. Modely *swin16* a *clip16* demonstrují výsledky dosažené v této práci.

[6]Předtrénované IResNet modely byly převzaty přímo z *Insightface* repozitáře: https://github.com/deepinsight/insightface.

[7]Předtrénovaný model ViT je dostupný ze stránky *HuggingFace*: https://huggingface.co/gaunernst/vit_tiny_patch8_112.cosface_ms1mv3.

Model	Metoda	Data	LFW Accuracy	IJB-C AUC	IJB-C TPR@FPR
IResNet-100	ArcFace	MS1Mv3	$0,998$	0.9956	$0,978$
IResNet-50	ArcFace	MS1Mv3	$0,998$	0.9956	$0,973$
TIMM	CosFace	MS1Mv3	$0,996$	0.9946	$0,965$
CLIP-V-16	CosFace	MS1Mv3	$0,996$	$0,9944$	$0,970$
CLIP-V-32	CosFace	MS1Mv3	$0,990$	0.9949	$0,933$
SWIN-B	CosFace	MS1Mv3	$0,987$	0.9943	$0,930$

Tabulka 4.5: Tabulka zobrazující srovnání dosažených výsledků s existujícími řešeními na bázi konvolučních sítí i transformeru (*TIMM*). Metrika Accuracy byla měřena na datové sadě LFW. Testy AUC a TPR@FPR byly prováděny na testovací datové sadě IJB-C s hodnotou $FPR = 1 \cdot 10^{-3}$. Varianty modelu *CLIP* a model *SWIN* představují výsledky dosažené v rámci této práce.

Kapitola 5

Závěr

Cílem této práce byl průzkum vhodnosti architektury vision transformer v oboru biometriky (konkrétně rozpoznávání obličejů), který byl proveden experimentováním s existujícími implementacemi. Výsledky experimentů provedených v této práci demonstrovaly, že tato varianta neuronových sítí má potenciál k velmi dobrým výsledkům a dokáže konkurovat řešením na bázi konvolučních sítí. Vision transformery se zároveň ukázaly být v případě použití na úloze rozpoznávání tváří velmi citlivé na trénovací podmínky, které bylo v rámci provedených experimentů nutné pečlivě odladit. I přes tyto komplikace bylo ovšem možné dosáhnout srovnatelných výsledků s existujícími systémy. S využitím nejvhodnější z architektur dle provedeného průzkumu bylo možné dosáhnout výsledků konkurujících dnešním state-of-the-art implementacím s dosaženou úrovní $0,9961$ Accuracy skóre na používané validační sadě LFW. Dále bylo taky možné dosažení velmi dobrých výsledků i s nejmenší dostupnou variantou tohoto vision transformeru, a to konkrétně $0,99$ Accuracy. Tato práce zahrnuje průzkum moderních architektur tohoto typu a jejich otestování spolu s analýzou jejich kladů a záporů. Dále obsahuje popis a analýzu celé řady provedených experimentů nad nejvhodnějšími kandidáty včetně definice ideálních parametrů pro natrénování vysoce efektivního systému pro zpracování lidských tváří. V poslední řadě byla provedena finální evaluace nejúspěšnějších pokusů a jejich srovnání s moderními implementacemi pro robustní rozpoznávání obličejů.

Ačkoli byla v rámci práce provedena celá řada experimentů, trénink vision transformerů je časově náročný a vyžaduje dostatek prostředků a dat. Mezi další experimenty mimo rozsah této práce by mohlo patřit například využití velké datové sady WebFace, která má potenciál k naplnění jednoho z nedostatků transformerů, a to jejich požadavků na množství trénovacích dat. Mezi další možná pokračování by mohl patřit také detailnější průzkum parametrů, zaměřený převážně na konfigurace využitých loss funkcí a parametrů architektur, provedený více do hloubky.

Z osobního pohledu mohu říci, že tato práce splnila má očekávání z oblasti získaných zkušeností. Právě náročnost tohoto druhu architektur neuronových sítí mě navedla k prozkoumání širokého množství konceptů z tématu strojového učení a zpracování obrazových dat.

Literatura

[1] Bahdanau, D., Cho, K. a Bengio, Y. *Neural Machine Translation by Jointly Learning to Align and Translate.* 2016 [cit. 2023-19-12]. Dostupné z: https://arxiv.org/pdf/1409.0473.

[2] Cao, Q., Shen, L., Xie, W., Parkhi, O. M. a Zisserman, A. *VGGFace2: A dataset for recognising faces across pose and age.* 2018 [cit. 2023-19-12]. Dostupné z: https://arxiv.org/pdf/1710.08092.

[3] Dai, J., Qi, H., Xiong, Y., Li, Y., Zhang, G. et al. *Deformable Convolutional Networks.* 2017 [cit. 2023-10-01]. Dostupné z: https://arxiv.org/pdf/1703.06211.

[4] Deng, J., Guo, J., Yang, J., Xue, N., Kotsia, I. et al. ArcFace: Additive Angular Margin Loss for Deep Face Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence.* Institute of Electrical and Electronics Engineers (IEEE). říjen 2022, sv. 44, č. 10, s. 5962–5979, [cit. 2023-10-01]. DOI: 10.1109/tpami.2021.3087709. ISSN 1939-3539. Dostupné z: http://dx.doi.org/10.1109/TPAMI.2021.3087709.

[5] Dong, X., Bao, J., Chen, D., Zhang, W., Yu, N. et al. *CSWin Transformer: A General Vision Transformer Backbone with Cross-Shaped Windows.* 2022 [cit. 2023-19-12]. Dostupné z: https://arxiv.org/pdf/2107.00652.

[6] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X. et al. *An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale.* 2021 [cit. 2023-19-12]. Dostupné z: https://arxiv.org/pdf/2010.11929.

[7] Guo, J., Han, K., Wu, H., Tang, Y., Chen, X. et al. *CMT: Convolutional Neural Networks Meet Vision Transformers.* 2022 [cit. 2024-25-04]. Dostupné z: https://arxiv.org/pdf/2107.06263v3.

[8] Guo, Y., Zhang, L., Hu, Y., He, X. a Gao, J. *MS-Celeb-1M: A Dataset and Benchmark for Large-Scale Face Recognition.* 2016 [cit. 2023-19-12]. Dostupné z: https://arxiv.org/pdf/1607.08221.

[9] Han, D., Pan, X., Han, Y., Song, S. a Huang, G. *FLatten Transformer: Vision Transformer using Focused Linear Attention.* 2023 [cit. 2023-19-12]. Dostupné z: https://arxiv.org/pdf/2308.00442.

[10] Katharopoulos, A., Vyas, A., Pappas, N. a Fleuret, F. *Transformers are RNNs: Fast Autoregressive Transformers with Linear Attention.* 2020 [cit. 2023-10-01]. Dostupné z: https://arxiv.org/pdf/2006.16236.pdf.

[11] Kim, M., Jain, A. K. a Liu, X. *AdaFace: Quality Adaptive Margin for Face Recognition*. 2023 [cit. 2024-08-05]. Dostupné z: https://arxiv.org/pdf/2204.00964.

[12] Lin, W., Wu, Z., Chen, J., Huang, J. a Jin, L. *Scale-Aware Modulation Meet Transformer*. 2023 [cit. 2023-19-12]. Dostupné z: https://arxiv.org/pdf/2307.08579.

[13] Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y. et al. *Swin Transformer: Hierarchical Vision Transformer using Shifted Windows*. 2021 [cit. 2023-19-12]. Dostupné z: https://arxiv.org/pdf/2103.14030.

[14] Liu, Z., Luo, P., Wang, X. a Tang, X. *Deep Learning Face Attributes in the Wild*. 2015 [cit. 2023-19-12]. Dostupné z: https://arxiv.org/pdf/1411.7766.

[15] Mekinec, D. *The benefits of face recognition technology*. Visage Technologies, 2023 [cit. 2024-08-05]. Dostupné z: https://visagetechnologies.com/benefits-of-face-recognition/.

[16] Nada, H., Sindagi, V. A., Zhang, H. a Patel, V. M. *Pushing the Limits of Unconstrained Face Detection: a Challenge Dataset and Baseline Results*. 2018 [cit. 2024-08-05]. Dostupné z: https://arxiv.org/abs/1804.10275.

[17] Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G. et al. *Learning Transferable Visual Models From Natural Language Supervision*. 2021 [cit. 2024-25-04]. Dostupné z: https://arxiv.org/pdf/2103.00020.

[18] Schroff, F., Kalenichenko, D. a Philbin, J. FaceNet: A unified embedding for face recognition and clustering. In: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, červen 2015 [cit. 2023-20-01]. DOI: 10.1109/cvpr.2015.7298682. Dostupné z: http://dx.doi.org/10.1109/CVPR.2015.7298682.

[19] Song, M. *Understanding the ROC-AUC Curve*. 2023 [cit. 2024-08-05]. Dostupné z: https://medium.com/@msong507/understanding-the-roc-auc-curve-cc204f0b3441.

[20] Srivastava, Y., Murali, V. a Dubey, S. R. *A Performance Comparison of Loss Functions for Deep Face Recognition*. 2019 [cit. 2024-08-05]. Dostupné z: https://arxiv.org/pdf/1901.05903.

[21] Taigman, Y., Yang, M., Ranzato, M. a Wolf, L. DeepFace: Closing the Gap to Human-Level Performance in Face Verification. In: *2014 IEEE Conference on Computer Vision and Pattern Recognition*. 2014, s. 1701–1708 [cit. 2023-20-01]. DOI: 10.1109/CVPR.2014.220. Dostupné z: https://ieeexplore.ieee.org/document/6909616.

[22] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L. et al. *Attention Is All You Need*. 2023 [cit. 2023-19-12]. Dostupné z: https://arxiv.org/pdf/1706.03762.

[23] Wang, H., Wang, Y., Zhou, Z., Ji, X., Gong, D. et al. *CosFace: Large Margin Cosine Loss for Deep Face Recognition*. 2018 [cit. 2023-25-04]. Dostupné z: https://arxiv.org/pdf/1801.09414.

[24] Wang, X., Li, Y., Zhang, H. a Shan, Y. *Towards Real-World Blind Face Restoration with Generative Facial Prior*. 2021 [cit. 2024-08-05]. Dostupné z: https://arxiv.org/pdf/2101.04061.

[25] Wikipedia. *Curse of dimensionality* [online]. 2024 [cit. 2023-19-12]. Dostupné z: https://en.wikipedia.org/wiki/Curse_of_dimensionality.

[26] Wikipedia. *F-score* [online]. 2024 [cit. 2023-19-12]. Dostupné z: https://en.wikipedia.org/wiki/F-score.

[27] Wikipedia. *Receiver operating characteristic* [online]. 2024 [cit. 2023-19-12]. Dostupné z: https://en.wikipedia.org/wiki/Receiver_operating_characteristic.

[28] Xia, Z., Pan, X., Song, S., Li, L. E. a Huang, G. *Vision Transformer with Deformable Attention*. 2022 [cit. 2023-19-12]. Dostupné z: https://arxiv.org/pdf/2201.00520.

[29] Yuan, L., Chen, Y., Wang, T., Yu, W., Shi, Y. et al. *Tokens-to-Token ViT: Training Vision Transformers from Scratch on ImageNet*. 2021 [cit. 2023-19-12]. Dostupné z: https://arxiv.org/pdf/2101.11986.

[30] Zhang, K., Zhang, Z., Li, Z. a Qiao, Y. Joint Face Detection and Alignment Using Multitask Cascaded Convolutional Networks. *IEEE Signal Processing Letters*. Institute of Electrical and Electronics Engineers (IEEE). říjen 2016, sv. 23, č. 10, s. 1499–1503, [cit. 2024-25-04]. DOI: 10.1109/lsp.2016.2603342. ISSN 1558-2361. Dostupné z: http://dx.doi.org/10.1109/LSP.2016.2603342.

[31] Zhu, L., Wang, X., Ke, Z., Zhang, W. a Lau, R. *BiFormer: Vision Transformer with Bi-Level Routing Attention*. 2023 [cit. 2023-19-12]. Dostupné z: https://arxiv.org/pdf/2303.08810.

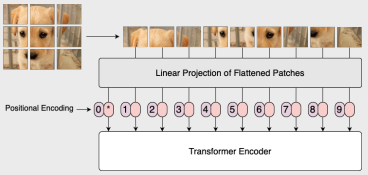
Příloha A

Plakát

VISION TRANSFORMERY PRO ROZPOZNÁVÁNÍ TVÁŘÍ

Šimon Strýček
Supervisor: Ing. Jakub Špaňhel

Vision Transformery

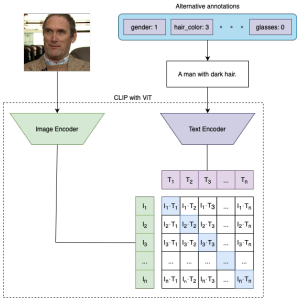


Experimenty

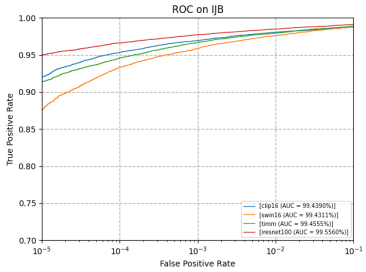
- Testování moderních ViT architektur
- Kombinování různých druhů trénovacích dat
- Testování několika veřejných datových sad
- Hledání optimálních podmínek pro trénink

Trénování s využitím kombinovaných dat

- Textové popisy generovány na základě anotací
- CosFace + contrastive CLIP loss

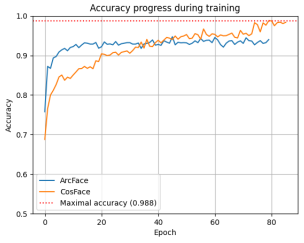


Výsledky evaluace na vybraných modelech



Důležitost volby správné loss funkce

- ArcFace vs. CosFace loss funkce při trénování ViT



Shrnutí dosažených výsledků

	Model	Metoda	Data	LFW Accuracy	IJB-C AUC	IJB-C TPR@FPR=1e-3
Existující předtrénovaná řešení	IResNet-100	ArcFace	MS1Mv3	0,998	0,9956	0,978
	IResNet-50	ArcFace	MS1Mv3	0,998	0,9956	0,973
	TIMM-Face-ViT	CosFace	MS1Mv3	0,996	0,9946	0,965
Modely trénované v rámci této práce	CLIP-V-16	CosFace	MS1Mv3	0,996	0,9944	0,970
	CLIP-V-32	CosFace	MS1Mv3	0,990	0,9949	0,933
	SWIN	CosFace	MS1Mv3	0,987	0,9943	0,930

Excel@FIT 2024 - Vision Transformers for Facial Recognition