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

AUDIOVIZUÁLNÍ ROZPOZNÁVÁNÍ OSOBY

AUDIOVISUAL PERSON RECOGNITION

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE **ONDŘEJ BAHOUNEK**
AUTHOR

VEDOUCÍ PRÁCE **Ing. OLDŘICH PLCHOT, Ph.D.**
SUPERVISOR

BRNO 2024

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

Zadání bakalářské práce

153223

Ústav:	Ústav počítačové grafiky a multimédií (UPGM)
Student:	**Bahounek Ondřej**
Program:	Informační technologie
Název:	**Audiovizuální rozpoznávání osoby**
Kategorie:	Zpracování řeči a přirozeného jazyka
Akademický rok:	2023/24

Zadání:

1. Seznamte se s technikami a architekturami modelů pro zpracování řeči a obrazu. Zaměřte se na modely pro extrakci embedingů založených na neuronových sítích.
2. Seznamte se s technikami pro porovnání embedingů za účelem verifikace identity.
3. Seznamte se s datovými sadami dostupnými ve Speech@FIT, proveďte rešerši dalších dostupných datových sad vhodných k trénování a verifikaci audiovizuálních systémů pro verifikaci osob.
4. Navrhněte a natrénujte biometrický systém pro verifikaci osob, jehož vstupem budou embedingy získané z řeči a videa, případně fotografie.
5. Vyhodnoťte vámi navržený systém a porovnejte jej s předchozími systémy vyvinutými ve Speech@FIT.

Literatura:

- Alam et al., "Analysis of ABC Submission to NIST SRE 2019 CMN and VAST Challenge", Proc. of Odyssey 2020 The Speaker and Language Recognition Workshop
- Alam et al., "Development of ABC systems for the 2021 edition of NIST speaker recognition evaluation", Proc. of Odyssey 2022 The Speaker and Language Recognition Workshop
- other literature provided by the supervisor

Při obhajobě semestrální části projektu je požadováno:
Body 1-3.

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

Vedoucí práce:	**Plchot Oldřich, Ing., Ph.D.**
Vedoucí ústavu:	Černocký Jan, prof. Dr. Ing.
Datum zadání:	1.11.2023
Termín pro odevzdání:	9.5.2024
Datum schválení:	9.11.2023

Abstrakt

Tahle práce se zabývá audiovizuální verifikací osoby ve videu nebo ze snímku obličeje a hlasové nahrávky. Modely využívají fúze hlasových a obličejových embeddingů. Modely přidělují váhy oběma modalitám, podle nichž kladou větší pozornost na jednu z nich. Výsledky modelů se vyznačují dobrou odolností proti poškození jedné z modalit.

Abstract

This work focuses on audiovisual verification of a person in a video or from a facial image and a voice recording. The models use a fusion of voice and face embeddings. The models assign weights to both modalities, allowing them to give more attention to one or the other. The results from these models demonstrate good resistance to the degradation of one of the modalities.

Klíčová slova

audiovizuální verifikace osoby, embeddingy, rozpoznání řečníka, rozpoznání tváře, fúze modalit, fúze embeddingů, WavLM, MHFA, Inception Resnet

Keywords

audivisual person verification, embeddings, speaker recognition, face recognition, multimodal fusion, embedding fusion, WavLM, MHFA, Inception Resnet

Citace

BAHOUNEK, Ondřej. *Audiovizuální rozpoznávání osoby.* Brno, 2024. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Oldřich Plchot, Ph.D.

Audiovizuální rozpoznávání osoby

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing. Oldřicha Plchota, Ph.D.. Uvedl jsem všechny literární prameny, publikace a další zdroje, ze kterých jsem čerpal.

.....
Ondřej Bahounek
6. května 2024

Poděkování

Rád bych poděkoval vedoucímu práce panu Ing. Oldřichu Plchotovi, Ph.D. za jeho podporu, vedení a poskytování četných konzultací v průběhu tvorby mé bakalářské práce.

We acknowledge VSB – Technical University of Ostrava, IT4Innovations National Supercomputing Center, Czech Republic, for awarding this project access to the LUMI supercomputer, owned by the EuroHPC Joint Undertaking, hosted by CSC (Finland) and the LUMI consortium through the Ministry of Education, Youth and Sports of the Czech Republic through the e-INFRA CZ (grant ID: 90254).

Obsah

Kapitola 1

Úvod

V dnešní moderní době se digitální ověření identity stalo jedním z klíčových bezpečnostních prvků. Především v online prostoru se s rostoucím počtem internetových platforem ověření identity stalo zcela zásadní otázkou. Digitální rozpoznání osob má ale i rozsáhlé užití ve skutečném světě, ať už se jedná o udělení přístupu do budovy nebo rozpoznávání osob z kamer využívaných v bezpečnostních složkách státu. Ale klasické metody, jako jsou přístupová hesla, už nemusí být v kritických aplikacích dostatečná, kvůli krádežím přístupových údajů (např. internetové bankovnictví). Taktéž metody vyžadující fyzický předmět jako přístupovou kartu jsou velmi náchylné na odcizení. Metody využívající lidských biometrických vlastností se stanou zcela zásadní pro zabezpečení bezpečnosti a důvěryhodnosti. Zejména v případě fyzického ověření, kdy není biometrický signál posílán vzdáleně a nehrozí nebezpečí spoofingu. (Např. kamery ověřující totožnost na letišti.)

Pro rozpoznání identity se typicky využívá pouze jeden zdroj informace snímek obličeje, otisk prstu, DNA nebo hlas člověka. Avšak kombinace více zdrojů zajišťuje přesnější a robustnější výsledky verifikace. Navíc může fungovat i při ztrátě jednoho z informačních kanálů.

Tahle práce se zabývá verifikací osoby z audio-vizuálních dat, a to konkrétně snímku obličeje a hlasu člověka. Práce si dává za cíl prozkoumat různé možnosti kombinace embeddingů reprezentujícího člověka z audia a z obrázků. A dalším úkolem je vytvořit model, který bude mít dobré verifikační schopnosti i v případě, že jedna z modalit bude poškozená. Výsledky získané z kombinace informací z obrázku a řeči se potom pokusí aplikovat přímo na video obsah. Navržené video-modely dokáží určit kvalitu videa v jeho částech a podle ní posuzovat části videa. Pro verifikaci osoby ve videu byl také navržen model, který určuje kvalitu obou modalit ve videozáznamu v každém dílčím okamžiku a dokáže rozhodnout, které modalitě v daný okamžik věnovat kolik pozornosti a díky tomu vytvořit verifikační systém robustní při poškození vstupních dat.

Kapitola 2

Rozpoznání osoby

Rozpoznávání osoby je biometrická disciplína, která porovnáváním kvantifikovatelných znaků osoby zjišťuje její identitu. Typicky se rozděluje na dva podproblémy:

- Identifikace osoby
- Verifikace osoby

Identifikace je proces, který zjišťuje, které osobě z databáze známých osob patří daný příznak. Naproti tomu **verifikace**, kterou se primárně zabírá tahle práce, zjišťuje, jestli osoba je tím, za koho se vydává. Porovnáním dvou záznamů a vyhodnotí, jestli oba dva patří stejné osobě. Hlavní rozdíl tedy je, že verifikace porovnává pouze jeden záznam proti jednomu a identifikace jeden záznam proti mnoha různým.

Osoby lze rozpoznávat podle řady různých charakteristik: DNA, otisky prstů, oční sítnice atd... Tahle práce se zabývá **audiovizuální** verifikací osob, kde vizuální část představují obličeje lidí získaných z videa a audio představuje mluvená řeč osoby na videu. Každá z těchto modalit sama o sobě poskytuje dostatečnou úroveň přesnosti při verifikaci osob. Avšak synergická kombinace těchto modalit výrazně zlepšuje celkovou přesnost.

Typický systém pro verifikaci osob se skládá z několika částí. Na samotném vstupu je zachycena digitální reprezentace objektu z fyzického světa, například snímek osoby z videa nebo hlas z mikrofonu v příslušném formátu. Systém následně pomocí různých algoritmů vyextrahuje ze záznamu identifikační příznaky typicky do formy mnohodimenzionálního vektoru, který následně porovnává proti záznamu dané osoby v databázi. Také vyhodnotí, jestli se jedná o stejnou osobu nebo nikoliv.

2.1 Výsledky verifikačního systému

Ačkoli výstupem verifikačního systému je pouze binární hodnota ano nebo ne (říkající, jestli se jedná o stejnou osobu), tak interpretování výsledků systému je komplexnější. Verifikační systém může mít hned čtyři typy výstupu - dva pokud dvojici vyhodnotil správně a dva pokud dvojici vyhodnotil chybně.

- **Správně:** Jedná o stejnou osobu
- **Správně:** Nejedná se o stejnou osobu
- **Chybně:** Systém vyhodnotí, že osoby jsou stejné, ačkoliv nejsou. Tuto chybu budeme dále nazývat anglicky **"false alarm"**.

- **Chybně:** Systém vyhodnotí, že se nejedná o stejnou osobu, i když ano. Tuto chybu budeme dále nazývat anglicky **"miss"**.

2.2 Reprezentace osoby

Pro potřeby rozpoznávání se k reprezentaci osoby nejčastěji používá vysocedeminzonální vektor, dále také nazývaný **"embedding"**. V této práci se zabýváme rozpoznání osoby bez kontextu. To znamená, že identita osoby by se neměla vztahovat na nic jiného, než pouze na zkoumanou charakteristiku. V případě reprezentace řečníka to znamená, že embedding by měl nést pouze informace o identitě řečníka, nikoli o významovém obsahu řeči nebo například o emocích, které jdou z hlasu poznat. Podobně u identifikace osoby podle obličeje by reprezentace osoby měla nést stejné výsledky nezávisle na tom, jestli se osoba směje, mračí nebo má třeba pokrývku hlavy.

2.3 Kosinová podobnost

V posledním kroku verifikace osoby se porovnávají dva embeddingy a vyhodnocuje se, jestli náleží stejné osobě. Pro zjištění podobnosti se nejčastěji používá **kosinová podobnost**. Kosinova podobnost počítá kosinus úhlu mezi dvěma nenulovými n-dimenzionálními vektory. Což se dá také reprezentovat jako podíl skalárního součinu vektorů a součinu velikostí vektorů.

$$\text{cosine\_similarity}(A, B) = \cos(\theta) = \frac{A \cdot B}{\|A\| \cdot \|B\|} = \frac{\sum_{i=1}^{n} A_i \cdot B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \cdot \sqrt{\sum_{i=1}^{n} B_i^2}} \tag{2.1}$$

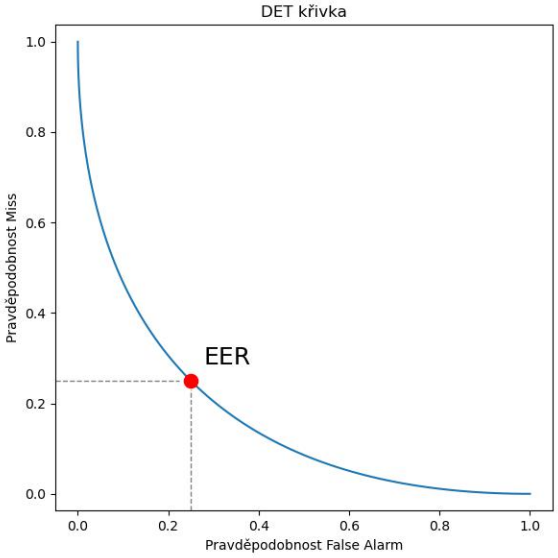
Pro dva stejné vektory kosinová vzdálenost udává hodnotu 1, pro dva ortogonální vektory vrací hodnotu 0. Systém následně vyhodnotí, jestli jsou embeddingy natolik podobné, aby mohl říct, že náleží stejné osobě.

2.4 Metriky systémů

Přesnost systému se vyhodnocuje na datasetech, které mají jasně definovanou "enrollment" osobu, kterou verifikujeme oproti druhému "test" videu, záznamu řeči atd..., kdy v testovacím záznamu se daná osoba nacházet může nebo musí. Zároveň se v test záznamu mohou nacházet jiné osoby. Systém pro každý testem definovaným trial (dvojice enrollment a test záznamů) vrací kosinovu vzdálenost. Systém sám si musí určit hodnotový práh (**threshold**), který rozděluje podobnosti na ty o kterých tvrdí, že osoba je stejná, a na ty, které ne.

2.4.1 DET křivka a EER

Pro vysoký práh se zvyšuje pravděpodobnost false alarm a snižuje pravděpodobnost miss a pro nízký práh naopak. DET (Detection Error Tradeoff) křivka znázorňuje tenhle fakt pro daný systém. Pro vyhodnocování přesnosti verifikačního systému se nejčastěji používá hodnota **EER** (Equal Error Rate). EER je bod na DET křivce, ve kterém je pravděpodobnost obou typů chyb stejná.



Obrázek 2.1: DET křivka a EER bod

2.4.2 DCF

Další běžně používanou metrikou je DCF (Detection Cost Function). DCF bere do úvahy (narozdíl od EER) dva další faktory. Oba se vztahují k případu užití systému. První odráží fakt, že podle typu užití systému chceme minimalizovat miss nebo false alarm chybu. Kdy, například u verifikace osoby při přístupu do bankovního účtu, chceme minimalizovat false alarm, tedy šanci, že se na účet dostane jiná osoba. Naopak při pátrání po hledané osobě z kamer, chceme minimalizovat miss a tím zachytit i menší shody. Tenhle fakt reprezentuje tím, že udává cenu miss $\mathcal{C}_{\text{miss}}$ a false alarm $\mathcal{C}_{\text{fa}}$ chyby. Druhý faktor, který bere v potaz, je předem definovaná pravděpodobnost $\mathcal{P}_{\text{same}}$, že osoba k verifikaci je ta, za kterou se vydává. Například u odemykání telefonu obličejem se očekává, že ve většině pokusů bude ona osoba ta pravá. Na rozdíl u pátrání po osobě se očekává, že šance, že jsme narazili na správnou osobu je malá.

$$\text{DCF} = C_{miss} \cdot p(\text{miss}|\mathcal{T}, \tau) \cdot P_{same} + C_{fa} \cdot p(\text{fa}|\mathcal{T}, \tau) \cdot P_{diff}, \tag{2.2}$$

kde $p(miss|\mathcal{T}, \tau)$ udává šanci miss chyby v datasetu $\mathcal{T}$ pro threshold τ. A pro P_{diff} platí:

$$\mathcal{P}_{\text{diff}} = 1 - \mathcal{P}_{\text{same}}\,. \tag{2.3}$$

K hodnocení systému se následně používá **minDCF**, což je nejmenší DCF spočítané pro všechny různé thresholdy. Threshold pro který platí výsledná minDCF je tedy z hlediska DCF optimální. Výsledná cena se ještě normalizuje, aby vzala do úvahy náročnost datasetu, a to tak, že se vydělí cenou C_{Default}, které je minimální cena mezi odmítnutím nebo přijetím všech trialů.

$$C_{\text{Default}} = min \begin{cases} C_{\text{miss}} \cdot p(P_{same}) \\ C_{\text{fa}} \cdot p(P_{diff}) \end{cases} \tag{2.4}$$
$$C_{\text{Norm}} = C_{\text{Det}}/C_{\text{Default}}$$

Kapitola 3

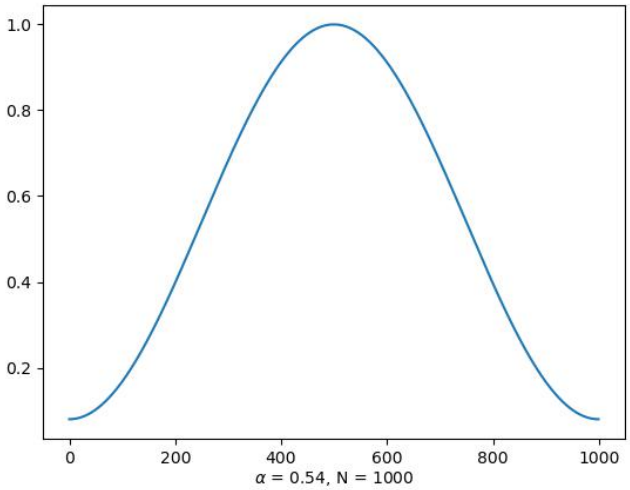
Rozpoznávání řečníka

Jak zmíněno v odstavci 2.2 osobu reprezentujeme pomocí embeddingu. Pro získání embeddingu reprezentujícího řečníka se musí vykonat několik důležitých kroků.

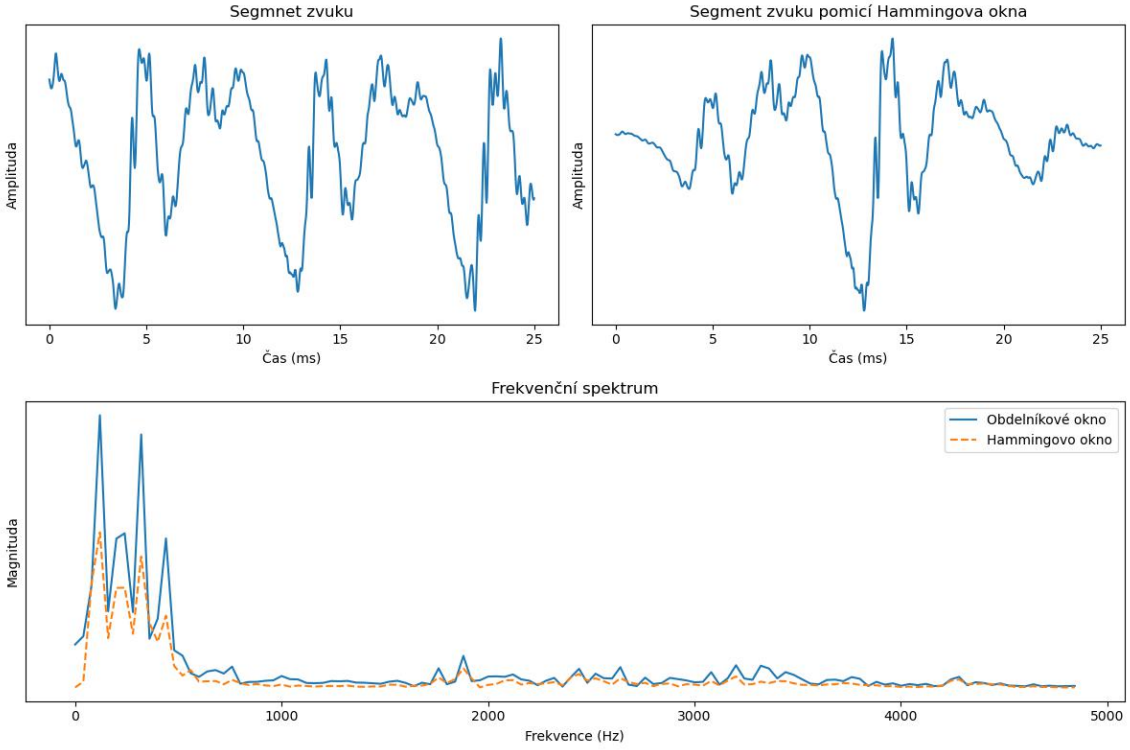
3.1 Rámcování signálu

Protože tradiční metody pro extrakci spektrální reprezentace pracují spolehlivě pouze na stacionárních signálech (tj. signálu, jehož statistické charakteristiky jsou nezměněny vzhledem k času), tak se jako první krok musí zvukový signál rozdělit na krátké překrývající se **rámce**, které se díky jejich krátké délce dají považovat za stacionární. Délka jednoho okna je obyčejně v rozsahu 10-30 ms. Jednotlivé rámce se mohou překrývat z 0-50 % [11]. Místo jednoduchého obdélníkového okna, které kvůli náhlému začátku a konci signálu způsobuje při převodu z časové domény do spektrální "postranní laloky", tj. nežádoucí efekt na okrajích rámce, se používá Hammingovo okno, které má schopnost tuhle skutečnost redukovat a dává lepší výsledek pro spektrální analýzu. Funkce Hammingovo okna:

$$w(n) = \alpha - (1 - \alpha) \cos\left(\frac{2\pi n}{N - 1}\right) \tag{3.1}$$



Obrázek 3.1: Hammingovo okno



Obrázek 3.2: Spektrum řeči

Samozřejmě používání Hammingovy okenní funkce způsobuje ztrátu přesnosti na začátku a konci zvukového signálu v důsledku postupného snižování amplitudy vzorků směrem k okrajům rámce. Právě kvůli tomu rozdělujeme rámce posouváním okna tak, aby se jednotlivé rámce překrývaly (obvykle 10 ms).

3.2 Extrakce příznaků

Extrakce nejlepší parametrické reprezentace akustického signálu je důležitý krok pro dosažení nejlepšího výsledku při rozpoznávání osoby. Efektivita téhle metody je důležitá, protože ovlivňuje chování dalších fází rozpoznávání [13]. Pro analyzování řeči je frekvenční doména vhodnější než časová doména [14]. Existují lepší alternativy než pouhé spektrum pro extrakci příznaků z řečových signálů. Jednou z těchto efektivnějších metod je využití Mel-Frekvenčních Cepstrálních Koeficientů (**MFCC**), které poskytují kompaktní a vysoce informativní reprezentaci řečových signálů.

3.2.1 MFCC

Mel-Frekvenční Cepstrální Koeficienty je dnes standardní metoda pro extrakci příznaků z řečových signálů. Dvě hlavní výhody této metody oproti pouhému spektru spočívají v lepším odpovídání výsledků lidskému vnímání zvuku a v redukci dimenzionality výsledného vektoru, což je výhodné zejména při trénování neuronových sítí.

Převod signálů do MFCC probíhá v několika krocích. Nejprve se signál pomocí vhodné okenní funkce rozdělí na jednotlivé rámce 3.1. Následně se na rámce aplikuje krátkodobá diskrétní Fourierova transformace (short-time DFT) a vezme se její absolutní hodnota. Následně se spektrum vyhladí skrze banku Mel-frekvenčních filtrů. Mel-frekvenční filtry jsou trojúhelníkové filtry rozložené nelineárně na frekvenční stupnici. Filtry jsou rozmístěné pomocí Melovské stupnice, tato stupnice je navržena tak, aby lépe odpovídala lidskému sluchovému vnímání, které je citlivější na nižší frekvence. Počet filtrů použitých v Mel-frekvenčním bankovním filtru určuje přesnost výsledné reprezentace, přičemž vyšší počet filtrů poskytuje jemnější frekvenční rozlišení. Výstup Mel-frekvenčního bankovního filtru je vektor, jehož dimenze odpovídá počtu použitých filtrů, zachycující energetickou distribuci napříč různými frekvenčními pásmy. Lidský sluch je méně citlivý ke změnám energie ve zvukovém signálu u nižších energií než u vyšších. Proto se na mel spektrum aplikuje logaritmus, který lépe imituje lidské vnímání zvuku. Poslední krok je dekorelizace vektoru použitím Diskrétní Kosinové Transformace (DCT) [17].

Pro přidání dynamických informací k statickým cepstrálním příznakům jsou následující vektorové příznaky rozšířeny o jejich aproximace prvního a druhého řádu derivace. Tyto derivace jsou označovány jako delta a dvojitá-delta (nebo zrychlení) koeficienty. První derivace pro vektor příznaků $\mathbf{c}$ ve snímku $\mathbf{k}$ je vypočtena jako lineární kombinace okolních $\pm$N příznakových vektorů [17]:

$$\Delta\mathbf{c}(k) = \sum_{j=-N}^{N} \mathbf{c}(k-j) \tag{3.2}$$

3.2.2 Normalizace průměru a odchylky

Výsledné MFCC jsou velmi náchylné na šum v nahrávce. Proto se používá normalizace průměru a odchylky MFCC přes celou několika sekundovou promluvu, která od každého koeficientu odečte průměr na jeho pozici přes celou nahrávku a následně ho vydělí jeho směrodatnou odchylkou. Tímto postupem je dosaženo snížení vlivu variability kanálu a šumu na výsledné koeficienty MFCC [18].

3.2.3 Detekce hlasové aktivity

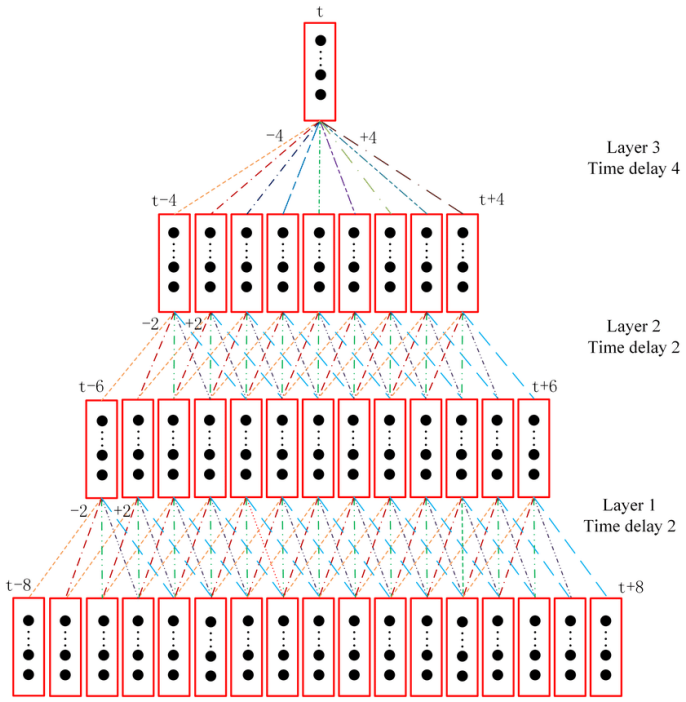
V rozpoznání mluvčího má detekce aktivity hlasu (VAD) zásadní význam při identifikaci segmentů řeči v audio signálu. Jejím hlavním úkolem je identifikovat segmenty řeči v signálu a odstranit období ticha. Pomocí různých algoritmů a metod analyzuje VAD akustické vlastnosti signálu a rozlišuje mezi rámci obsahujícími řeč a těmi obsahujícími ticho. V praxi VAD pomáhá šetřit výpočetní zdroje tím, že filtruje neřečové rámce, což umožňuje použití pouze relevantních hlasových rysů pro identifikaci mluvčích.

3.3 Extrakce embeddingů

Po získání reprezentace řeči se použije trénovatelný extraktor embeddingů pro získání reprezentace řečníka přes jednotlivé rámce. V téhle sekci se seznámíme se dvěma základními architekturami pro extrakci embeddingů založenými na hlubokých konvolučních neuronových sítích.

3.3.1 x-vektor

Jedním z hlavních problémů, se kterým se potýkají extraktory embeddingů, je potřeba přeměnit segmenty řeči s různou délkou na embeddingy s pevnou délkou. Architektura x-vektor [25] založená na **TDNN** (Time Delay Neural Network [29]) se tímhle problémem vypořádává pomocí statistického agregování napříč celým segmentem.



Obrázek 3.3: Znázornění TDNN využité v publikaci [33]

Layer	Layer context	Total context	Input x output
frame1	$[t-2, t+2]$	5	120×512
frame2	$\{t-2, t, t+2\}$	9	1536×512
frame3	$\{t-3, t, t+3\}$	15	1536×512
frame4	$\{t\}$	15	512×512
frame5	$\{t\}$	15	512×1500
stats pooling	$[0, T)$	T	$1500T \times 3000$
segment6	$\{0\}$	T	3000×512
segment7	$\{0\}$	T	512×512
softmax	$\{0\}$	T	$512 \times N$

Tabulka 3.1: Popis vrstev x-vektor extraktoru [25]

Architektura x-vektorového extraktoru se skládá ze dvou hlavních částí: jedna část pracuje na úrovni jednotlivých rámců, zatímco druhá část pracuje na úrovni segmentů. Prvních pět úrovní zpracovává řečové rámce, které jsou reprezentovány 24-dimenzionálními normalizovanými Melovými filtrbankami. Tento proces zahrnuje použití TDNN, což je v podstatě jednoduchá 1D konvoluce, kde se N rámců kolem rámce t spojí dohromady. Vyšší vrstvy extraktoru provádějí podobnou operaci, což umožňuje získání širšího kontextu okolo daného rámce t. Tahle operace se provede pro všechny rámce t. Následně se pro všechny rámcové výstupy provede statistická agregace. A to tak, že se agregují všechny rámcové výstupy, vypočítá se jejich průměr a směrodatná odchylka. Tyhle dvě statistiky jsou konkatenovány a dále podrobeny dvou dalším vrstvám, která již tedy pracují nad celým segmentem řeči. Na závěr se používá softmax vrstva, která určí pravděpodobnost, že hlasový segment náleží jednomu z N řečníků [25]. Pro řečníka i se z vektoru y spočítá jako:

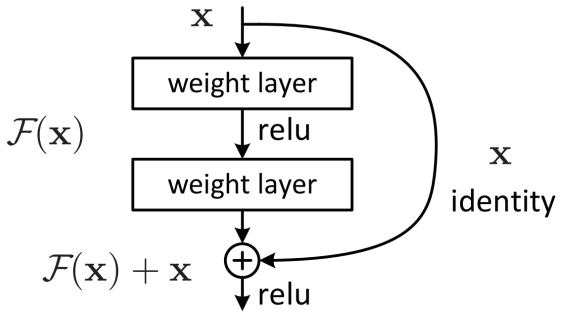
$$S(y)_i = \frac{e^{y_i}}{\sum_{j=1}^{N} e^{y_j}} \tag{3.3}$$

Softmaxová vrstva se po vytrénování už nepoužívá. Výsledný x-vektor, lze vyextrahovat z libovolné vrstvy nad statistickou agregací.

3.3.2 ResNet

Architektura **ResNet** je hluboká konvoluční neuronová síť, původně navržená pro rozpoznání obrazu [8]. Na rozdíl od architektury x-vektor nepoužívá TDNN, nýbrž klasickou 2D konvoluci. Tím způsobem, že pro kus řeči vypočítá FBANK nebo MFCC, a ty následně předloží CNN jako obrázek. Pro jednotlivé kusy řeči se poté provede segmentové zpracování obdobné jako při x-vektoru [30].

Největší předností ResNet architektury je výjimečná hloubka. Tradiční neuronové sítě se často potýkají s problémem mizejícího gradientu, kdy se při zpětné propagaci gradient postupně zmenšuje, což může znemožnit trénování nižších vrstev. Autoři architektury ResNet tento problém řeší zavedením reziduálních spojení. Spojení, které umožňují přeskočit několik propojených vrstev při úpravách parametrů. Tenhle signál je následně sečten se signálem z propojených vrstev. Tímto způsobem je gradient snáze přenášen zpět do sítě a umožňuje efektivnější trénink [8].



Obrázek 3.4: Blok reziduálního spojení [8]

3.4 Předtrénované modely

V posledních letech se velkému úspěchu těší velké **předtrénované řečové modely**. Jedná se o modely využívající metod samoučení, které se naučí reprezentovat akustické vlastnosti hlasového signálu. Modely se učí sami bez učitele na rozsáhlém korpusu neanotovaných dat. Takové modely se pak dají využít jako skvělé extraktory příznaků pro modely s konkrétním úkolem jako například rozpoznání řeči nebo rozpoznání řečníka. Následné modely využívající předtrénované modely se potom učí kratší dobu a stačí jim daleko menší množství anotovaných dat, než kdyby se učili vše od začátku. Jejich výkon lze ještě dále zdokonalit dotrénováním předtrénovaného modelu společně s částí specifickou pro konkrétní úkol [16]. Metoda samoučení je druh trénování bez učitele, kde si model sám vytváří data, podle kterých se učí a slouží tedy jako svůj vlastní učitel.

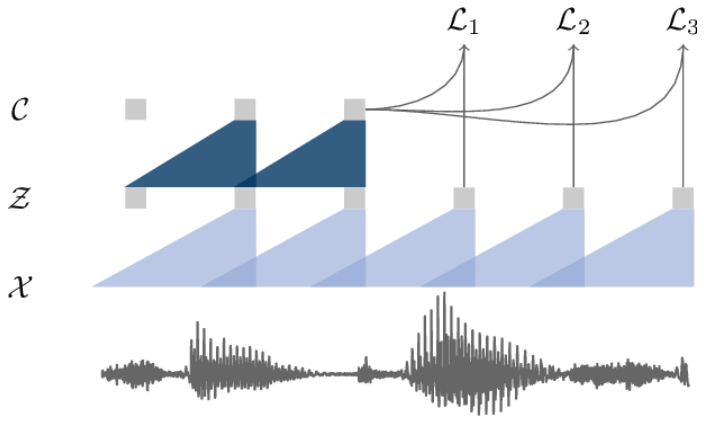
3.4.1 wav2vec

Jedním z prvních modelů využívající samoučení je **wav2vec** (wave to vector) od společnosti Facebook představen v roce 2019. Wav2vec se snaží předpovídat budoucí signál. Učí se pomocí kontrastní ztráty při porovnání předpověděného signálu a falešných signálů.

Wav2vec skutečně nepředpovídá holý signál, ale jeho naučenou reprezentaci. Využívá k tomu dvě konvoluční sítě, jedna $f : X \to Z$, která hlasový signál $x_i \in X$ transformuje do latentního prostoru Z. Výsledkem jsou vektory $z_i \in Z$ s daleko menší frekvencí než původní signál, konkrétně jeden vektor pro 30 ms signálu s posunem 10 ms. Nad vektory z latentního prostoru se nachází druhá konvoluční síť, síť kontextová $g : Z \to C$, která kombinuje několik latentních reprezentací do jednoho kontextového vektoru $c_i = g(z_i \dots z_{i-v})$ s zorným polem v. Model se učí rozlišováním mezi skutečnou latentní reprezentací v kroku k a náhodnými reprezentacemi $\tilde{z}$ z rozložení p_n, minimalizováním ztráty kontrastní ztráty pro každý krok $k = 1, \dots, K$:

$$L_k = -\sum_{i=1}^{T-k} \left(\log \sigma(z_{i+k}^\top h_k(c_i)) + \underset{\tilde{z} \sim p_n}{\lambda \mathbb{E}} [\log \sigma(-\tilde{x}^\top h_k(c_i))] \right), \tag{3.4}$$

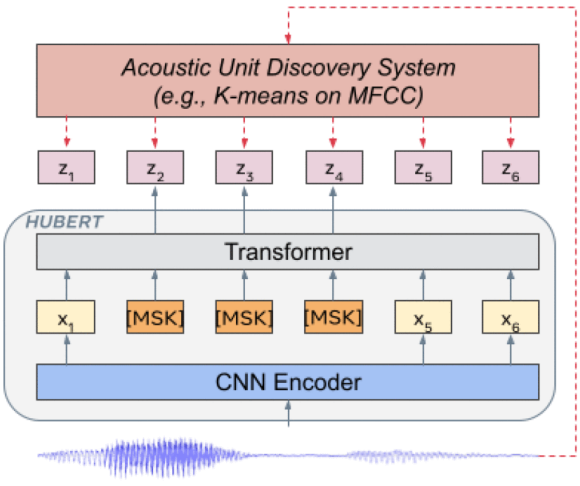
kde σ je sigmoidní funkce definována $\sigma(x) = 1/(1 + \exp(-x)$ a kde $\sigma(z_{i+k}^\top h_k(c_i))$ je pravděpodobnost, že z_{i+k} je pravý vzorek. Pro c_i a krok k se použije afinní transformace $h_k(c_i) = W_k c_i + b_k$. Ve skutečnosti falešné reprezentace jsou deset reprezentací náhodně vybraných ze hlasové sekvence, tedy $p_n(z) = \frac{1}{T}$, kde T je delká sekvence a λ je počet falešných reprezentací [21].



Obrázek 3.5: Znázornění předtrénování z audia X pomocí dvou konvolučních neuronových sítí, které jsou optimalizovány předpovídáním budoucích reprezentací [21].

3.4.2 HuBERT

Další důležitým počinem v předtrénovaných hlasových modelech je architektura **HuBERT** [9] (Hiden-Unit BERT), která vychází z modelu pro reprezentaci přirozeného jazyka BERT [5]. HuBERT i BERT jsou architektury typu transformer popsané v [27], které nahrazují konvoluční a rekurentní neuronové sítě za **attention** mechanismus, konkrétně **self-attention** mechanismus, který modeluje vztah každého slova ve větě s každým jiným slovem v oné větě (v případě BERT). Dosahuje tedy zorného pole přes celou nahrávku/větu již v první attention vrstvě. HuBERT je architektura, která využívá několik bloků transformer **enkodérů**. Používá techniku samoučení, která vybírá části vstupu a vytváří pro ně **skryté jednotky**. Tyto vybrané části jsou následně zamaskovány a spolu s celým vstupem jsou předány blokům transformeru. Výstupní reprezentace je porovnána se skrytými jednotkami, což umožňuje modelu naučit se reprezentovat řeč. HuBERT je tedy obousměrný transformer [9].



Obrázek 3.6: HuBERT předpovídá skryté jednotky rámců y_2, y_3, y_4 získaných pomocí k-means shlukování [9].

Pro tvorbu skrytých jednotek se v první iteraci používá MFCC signálu, v dalších iteracích se používají kontextuální reprezentace části signálu získané z mezivrstev transformeru. Takhle získané embeddingy jsou pomocí shlukovacího algoritmu, jako například k-means, shlukovovány do C tříd. Každé třídě je přiřazen jeden embedding, který je později použit pro porovnávání. Shlukování je tady použito jako efektivní způsob kvantizace rámce pro vytvoření pseudo-značek pro učení. Z druhé strany HuBERT použije CNN pro transformaci signálu do latentního prostoru. Část latentních reprezentací je zamaskována, až 50 %, a všechny latentní reprezentace jsou předány transformer enkodéru s L vrstvami. Transofrmer vytvoří kontextový vektor pro každý latentní embedding. Výsledné kontextové embeddingy projdou jednou lineární vrstvou pro redukci dimenzionality na stejnou délku jako skryté jednotky. Kontextové vektory zamaskovaných částí jsou pak pomocí kosinovské podobnosti porovnány se všemi možnými skrytými jednotkami pro získání předpovědí. Cross-entropy ztráta je použita pro penalizování špatných predikcí.

3.4.3 MHFA

Jeden ze způsobů dotrénování předtrénovaného audio modelu pro verifikaci řečníka je multi-head factorized attentive pooling (**MHFA**) [16]. Jedná se o backend navržený na Fakultě informatiky VUT v Brně. Model je plně založen na mechanismu attention, a na rozdíl od tradičních backendů, které také používají audio transformery jako extraktory příznaků (např. TDNN + statistická poolingová vrstva), se vyznačuje nízkým počtem parametrů. MHFA extrahuje výstup každého transformer bloku, mezi kterými dělá vážený součet napříč všemi bloky. Váhy jsou naučené parametry a jsou separátní pro keys a values. Matice keys a values se tedy spočítají jako:

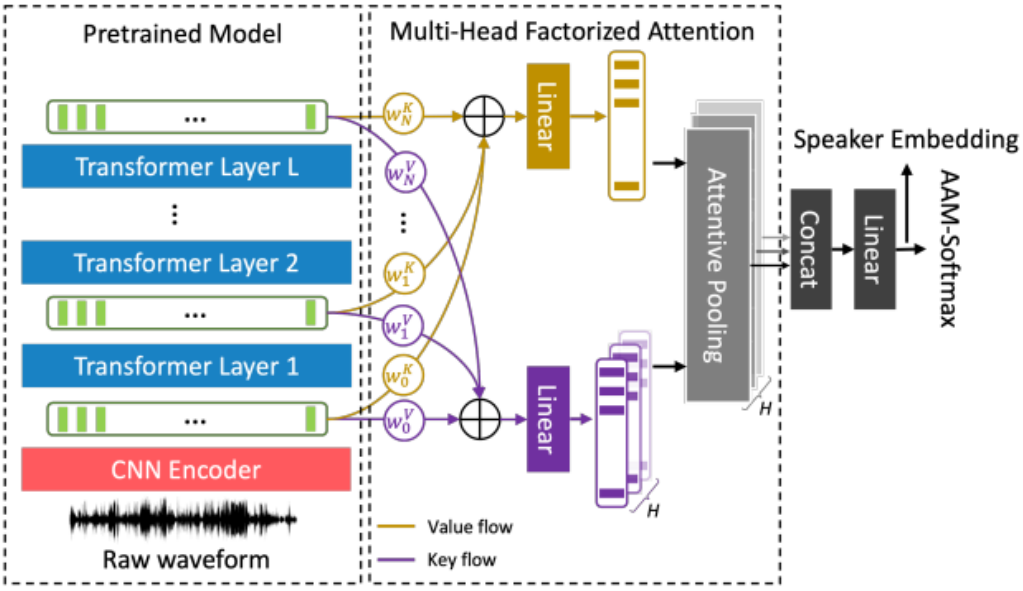
$$\begin{aligned} \mathrm{K} &= \left(\sum_{l=1}^{L} w_l^k \mathbf{Z}_l\right) S^k \\ \mathrm{V} &= \left(\sum_{l=1}^{L} w_l^v \mathbf{Z}_l\right) S^v, \end{aligned} \tag{3.5}$$

kde L je počet vrstev předtrénovaného transformeru, Z jsou výstupy transformer bloků a w jsou naučené váhy. Matice S jsou naučené matice (plně propojené vrstvy neuronů) určené pro redukci dimenzionality. Model se tedy sám naučí rozpoznat, které úrovně transformer modelu mají nejlepší diskriminativní vlastnosti pro verifikaci řečníka. Autoři práce spekulují, že values nesou informace o identitě řečníka a keys obsahují fonetické informace [16]. Jméno multi-head factorized attention znamená, že model naučí k jaké hlavě má daný vstup dávat největší pozornost. Každá hlava by se měla naučit agregovat pouze specifickou množinou fonémů. Attention matice $\mathbf{A}$ se tedy spočítá jako

$$\mathbf{A} = \mathrm{softmax}(\mathbf{KQ}) \tag{3.6}$$

kde Q je query matice naučená mapovat keys k jednotlivým hlavám. MHFA následně agreguje hodnoty přes framy následujícím způsobem:

$$\begin{aligned} \mathbf{c}_h &= \sum_{t=1}^{T} \mathbf{A}_{h,v} \mathbf{V}_t \\ \mathbf{c} &= \mathrm{concat}(\mathbf{c}_1, \cdots, \mathbf{c}_H) \end{aligned} \tag{3.7}$$



Obrázek 3.7: Multi-head factorized attentive pooling [16]

kde H je počet hlav a T je počet framů. Protože u verifikace osoby nezáleží na pořadí a na obsahu řeči, lze hodnoty jednotlivých framů pouze sečíst přes časovou doménu a nezabývat se jejich pořadím. Výsledný vektor c ještě projde jednou plně propojenou neuronovou vrstvu. Z výsledného vektoru se potom klasickým způsobem vypočítá ztráta.

Kapitola 4

Rozpoznání obličeje

Následující kapitola se zaměří na rozpoznávání obličeje z digitálních snímků. I přesto, že samotné rozpoznávání osoby probíhá ve videu, extrakce příznaků se často provádí na jednotlivých snímcích. V rámci rozpoznávání obličeje je zkoumání časové domény často málo efektivní a nepřináší významné zlepšení. Většina volně dostupných nástrojů pro klasifikaci ve videu, které využívají 3D konvoluce přes časovou a prostorovou doménu, je primárně určena pro rozpoznávání aktivit ve videu a není tedy vhodná pro rozpoznávání jednotlivých osob. Dalším problémem rozpoznávání obličeje ve videu je možnost výskytu více obličejů v jednom záběru nebo možnost, kdy zkoumaný obličej ze záběru zcela zmizí. Tyto faktory značně komplikují vytvoření komplexního nástroje pro rozpoznávání osob ve videu. Z těchto důvodů se práce soustředí pouze na rozpoznávání obličeje ze statických snímků a časová doména je řešena způsoby popsanými v popisu architektury 6. Pro verifikaci osoby podle obličeje se typicky musí vykonat čtyři kroky:

- Detekce obličeje
- Zarovnání obličeje
- Extrakce embeddingu
- Verifikace

4.1 Detekce obličeje

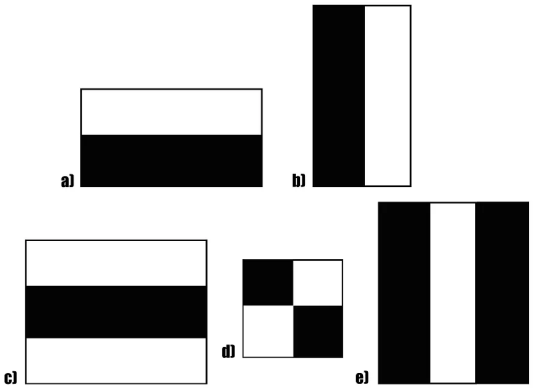
Před samotnou extrakcí obličejových příznaků je vhodné provést detekci obličeje a získat tak **bounding box** (rámeček kolem obličeje). Tímto způsobem se extraktoru předloží pouze snímek obsahující obličej, což usnadní následnou analýzu a zpracování.

Viola-Jones

Jeden z nejvíce osvědčených algoritmů pro detekci obličeje vznikl již v roce 2001 pod názvem **Viola-Jones**, známé tako jako Haar Cascade Features [28]. Tento algoritmus se vyznačoval nejen svou přesností, ale především svou vysokou rychlostí, která mu umožňovala zpracovávat videa téměř v reálném čase.

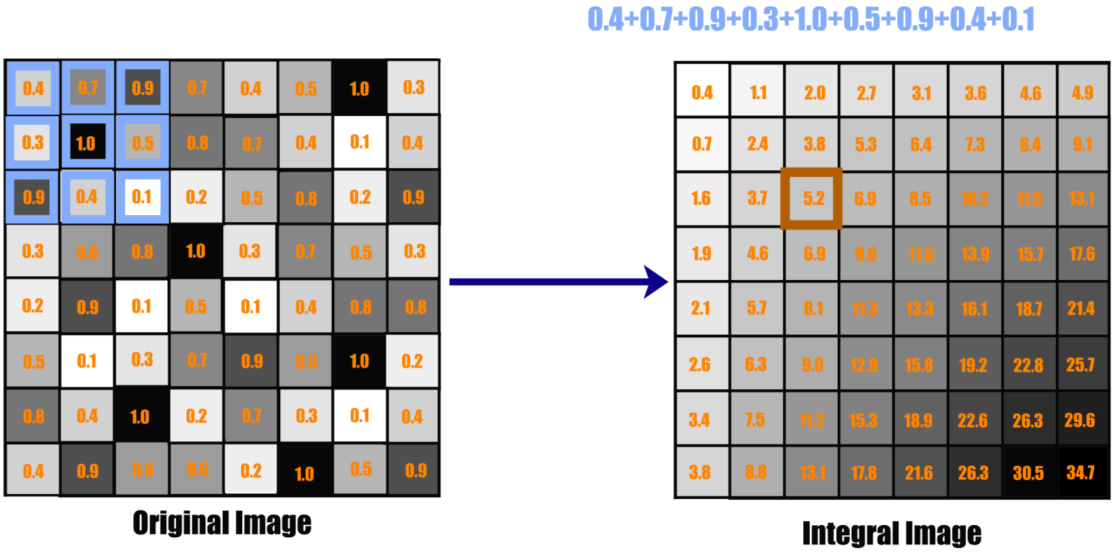
Algoritmus využívá **Haarových příznaků** k identifikaci hran a oblastí s náhlými změnami intenzity pixelů v obrázku. Haarové příznaky se skládají ze dvou částí (bílé a černé pro ilustraci), přičemž v obou částech se vypočítá průměr intenzity pixelů a poté spočítá

jejich rozdíl. Pokud je rozdíl blízko jedné, detekuje se hrana. Autoři práce pomocí metody AdaBoost vybrali ze 180 tisíc příznaků jen 6000 příznaků, ty mají největší potenciál pro detekci obličeje.



Obrázek 4.1: Haarové příznaky [2]

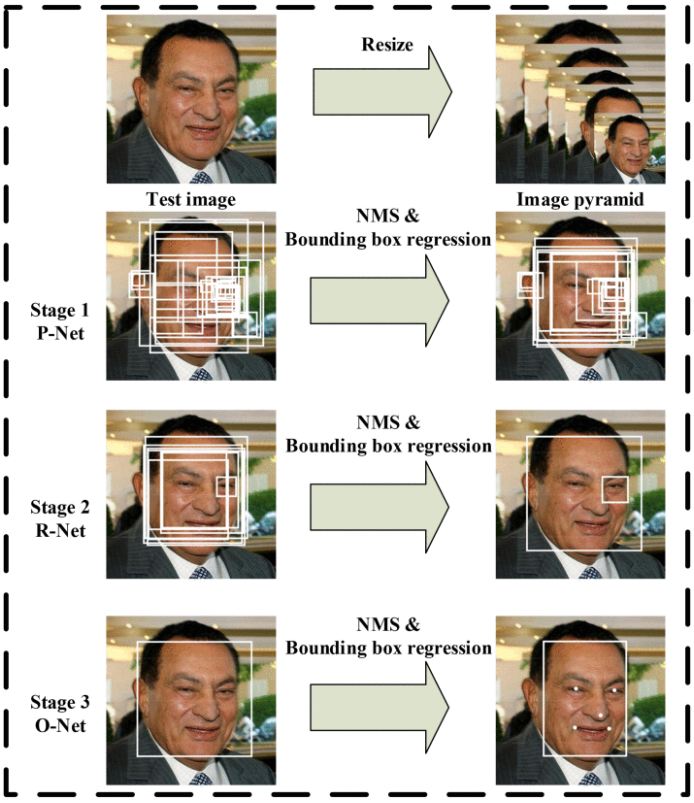
Haarové příznaky jsou postupně posouvány přes obrázek. Nejprve je přes obrázek posouván AdaBoostem vybraný nejlepší (zamítne nejvíc false-alarmů) příznak, a až pokud detekuje hledanou hranu, tak se na nalezené okno postupně testují čím dál víc komplexnější příznaky. Pokud se některý příznak vyhodnotí negativně, přejde se na testování dalšího okna. Tímhle způsobem se neplýtvá výpočty na část obrázku, kde se obličej jasně nenachází. Zásadním faktorem pro rychlost algoritmu je použití integrálního obrazu, který umožňuje výpočet Haarových příznaků s minimálními výpočetními nároky. Integrální obraz je vytvořen sečtením hodnot pixelů od počátečního bodu vlevo nahoru. Díky tomu jsou pro výpočet libovolného Haarova příznaku potřeba pouze čtyři operace sčítání [28] [2].



Obrázek 4.2: Integrální obraz [2]

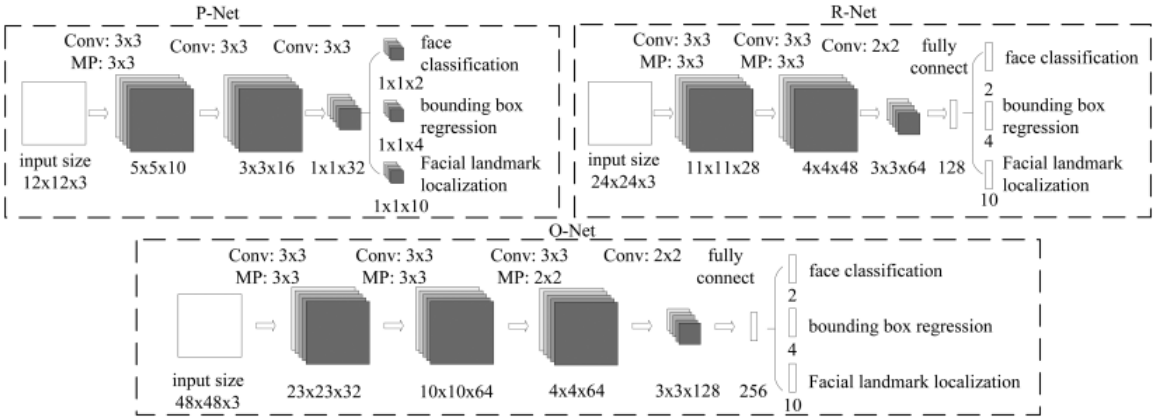
MTCNN

V dnešní době jsou nejpopulárnější detektory obličejů založené na konvolučních sítích na příklad **Multi-task Cascaded Convolutional Networks** (MTCNN). Oproti Viola-Jones je přesnější a zvládá pracovat i nad obličeji v nefrontálních pozicích, avšak jeho zpracování je pomalejší než u zmíněného model. MTCNN je navrženo nejen pro detekci obličeje, ale také pro jeho zarovnání, a to nalezením pěti obličejových orientačních bodů pro oči, nos a ústa. Prvním krokem před detekcí obličeje je změna velikosti snímků na různé velikosti a vytvořením obrázkové pyramidy, která je na vstupu sítě.



Obrázek 4.3: Znázornění postupu lokalizace obličeje pomocí MTCNN [32]

MTCNN se skládá ze tří konvolučních sítích. První **Proposal Network** (P-Net) je plně propojená konvoluční síť, slouží k nalezení kandidátních oken s obličejem. Počet překrývajících se oken je následně zredukován pomocí algoritmu Non-Maximum Suppression. Druhá konvoluční síť **Refine Nework** (R-Net) je hlubší než ta první. R-Net dále snižuje počet kandidátních oblastí a provádí kalibraci pomocí bounding box regrese. Jejím výstupem je informace o přítomnosti obličeje v dané oblasti, 4-dimenzionální vektor s bounding boxem a 10-dimenizionální vektor s pozicemi orientačních bodů. Třetí síť **Output Network** (O-Net) je podobná té druhé, ale je hlubší a zaměřuje se na podrobnější popis obličeje a přesnější lokalizaci charakteristických bodů [7] [32].



Obrázek 4.4: Architektura P-Net, R-Net a O-Net [32]

4.2 Zarovnání obličeje

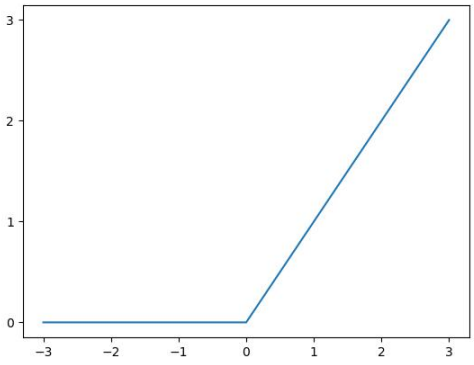
Zarovnání obličeje je klíčovým krokem pro optimalizaci výkonu systémů verifikace obličeje. Cílem je, aby orientační body obličeje jako jsou oči a ústa, byly zarovnány do stejné polohy a úrovně napříč všemi obrázky. Prvním krokem je nalezení orientačních bodů. Existuje mnoho architektur pro lokalizaci orientačních bodů, například kaskádová síť MTCNN. Následně se obličej upravuje pomocí operací jako jsou změna velikosti, rotace a translace, aby byl zarovnán do standardního postavení. Následně se obličej upravuje pomocí operací jako je afinní transformace, aby byl zarovnán do standardní polohy[12].

4.3 Extrakce embeddingu

Stejně jako u extrahování embeddingu pro reprezentaci řečníka, tak i u extrakce obličejového embeddingu, jsou nejpoužívanější metodou konvoluční neuronové sítě. Na rozdíl od řeči, už ale není potřeba samotný vstup nijak transformovat, konvoluce se vykonává přímo nad samotnými obrázky.

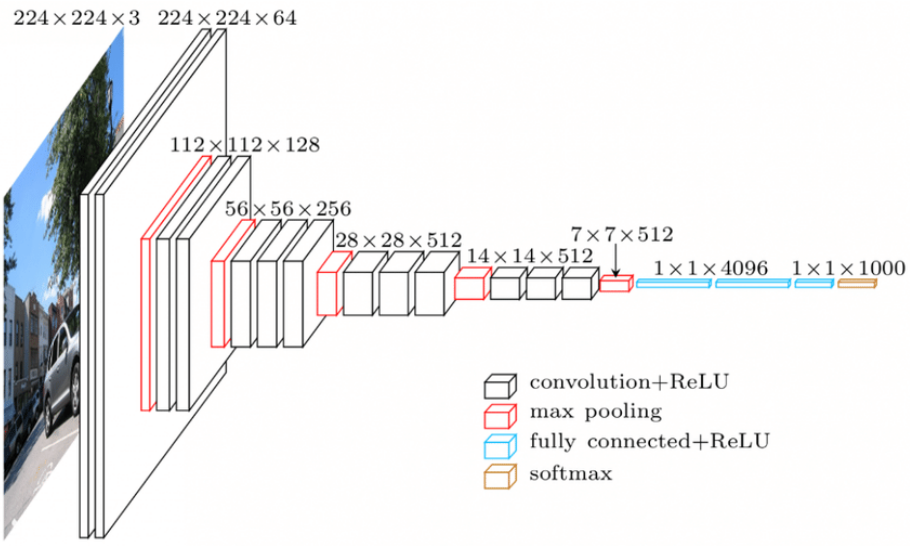
VGG

Za základní model v oblasti rozpoznání objektu, v našem případě obličeje, se dá považovat síť VGG. Síť má na vstupu obrázek jako tři matice, každá pro jeden RGB barevný kanál. V celém modelu se používají pouze 3x3 konvoluční filtry. Pro představení nelinearity se za každou konvoluční vrstvou nachází ReLU aktivační funkce.



Obrázek 4.5: ReLu aktivační funkce podle $f(c) = \begin{cases} x & x > 0 \\ 0 & jinak \end{cases}$

Po dvou nebo třech vrstvách se pomocí 2x2 max-pooling operace s krokem 2 sníží velikost obrázku na polovinu. Počet konvolučních filtrů přibývá s hloubkou sítě. Na konec je výstup zploštěn a předán třem finálním plně propojeným vrstvám. Poslední vrstvou je softmax vrstva pro klasifikaci mezi třídami. Významným rysem této architektury je použití více 3x3 filtrů za sebou namísto velkých konvolučních filtrů jako 11x11 a 5x5 používaných v předcházejících modelech. Například dva 3x3 filtry za sebou mají stejný *receptive field* jako jeden 5x5 filtr, ale mají méně parametrů, a to umožňuje vytvořit hlubší síť [24].



Obrázek 4.6: Architektura VGG-16

Inception

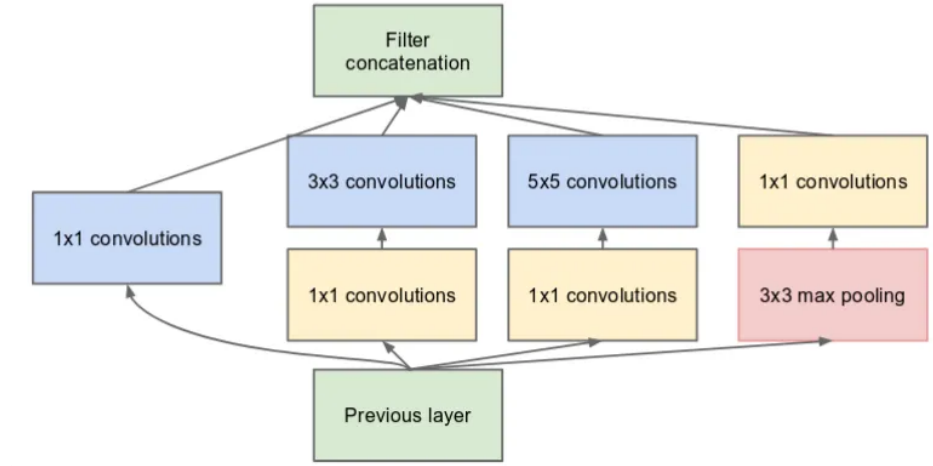
Konvoluční klasifikátory čelí problému velikosti klasifikovaného objektu na obrázku. Pokud předmět zabírá celý obrázek, tak větší konvoluční filtr je vhodnější. Naopak při klasifikaci menšího objektu jsou pro zachycení jemných detailů lepší menší filtry. To mělo za následek rozdílné výsledky napříč různými datasety. Tento problém motivoval vznik architektury

Inception (známé také jako GoogLeNet), která přinesla výsledky i u verifikace obličeje [26].

Místo snahy vytvářet pořád hlubší sítě, se autoři Inception pokusili síť rozšířit do šířky. Pomocí Inception modulu se na jedné úrovni sítě vykoná konvoluce s filtry různých velikostí. Výsledky různých filtrů jsou dále zkonkatenovány a propagovány dále do sítě. Pro zredukování kanálu se před konvolucí většími filtry využívá 1 x 1 konvoluce. Navržená síť byla 22 vrstev hluboká. Aby autoři předcházeli vymření gradientu, použili dva pomocné klasifikátory v mezi vrstvách, aby pomohly diskriminaci u nižších vrstev. Celková hodnota ztrátová funkce se spočítá jako:

$$Ztráta_{celková} = Ztráta_{skutečná} + 0.3 * Ztráta_{pomocná\_1} + 0.3 * Ztráta_{pomocná\_2} \quad (4.1)$$

Vzniklo mnoho verzí Inception architektury, různé verze vyměnily vetší filtry za 3 x 3 filtry, 3 x 3 filtry vyměnili za 3 x 1 a 1 x 3 filtry nebo použily reziduální spojení z ResNet atd... [20]



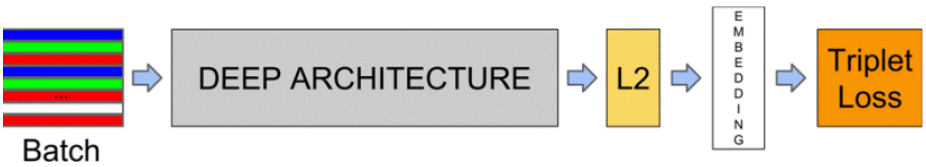
e

Obrázek 4.7: Incpetion modul [26]

FaceNet

FaceNet [22] je systém vyvinutý společností Google v roce 2015, určený pro verifikaci, identifikaci a shlukování obličejů. Systém produkuje obličejové embeddingy, jejichž L2 vzdálenost (euklidovská vzdálenost) mezi sebou přímo koresponduje k podobnosti obličejů.

Metoda využívá hlubokou konvoluční síť. Autoři v práci pracovali se sítěmi Zeiler&Fergus [31] a Inception [26] avšak bez použití softmax funkce. Na místo ní se po plně propojených vrstvách výstup L2 normalizuje a použije se metoda *triplet lost*. Obrovskou výhodou je redukce dimenzionality face embeddingu, kde autoři používají face embedding s 128 dimenzemi, který se dá s dobrou přesností reprezentovat na 128 bytech, oproti architekturám využívající softmax, které obvykle využívají 1024 a více dimenzí [22].



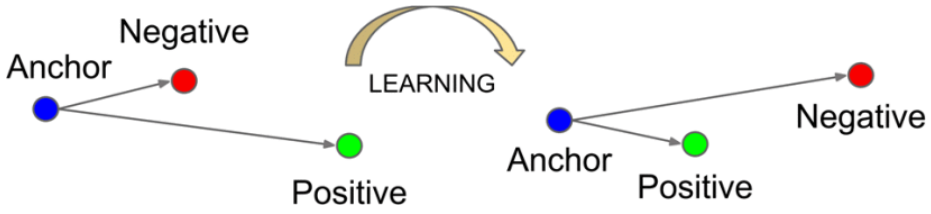
Obrázek 4.8: Architektura systéme FaceNet [22]

Metoda triplet loss využívá trojice obličejů, kotvu x_i^a (anchor), rozdílný snímek stejné osoby x_i^p (positive) a snímek jiné osoby x_i^n (negative). Pozitivní obličej se snaží v euklidovském prostoru přiblížit ke kotvě a negativní oddálit. Pro embedding $f(x_i)$ by tedy mělo platit:

$$\|f(x_i^a) - f(x_i^p)\|_2^2 + \alpha < \|f(x_i^a) - f(x_i^n)\|_2^2, \quad \forall (f(x_i^a), f(x_i^p), f(x_i^n)) \in \mathcal{T} \tag{4.2}$$

kde α je vzdálenost vynucována mezi pozitivním a negativním embeddingem a $\mathcal{T}$ je množina všech trojic snímků. Ztráta, kterou se snaží minimalizovat, se pro N trojic spočítá podle:

$$L = \sum_i^N \left[\|f(x_i^a) - f(x_i^p)\|_2^2 - \|f(x_i^a) - f(x_i^n)\|_2^2 + \alpha\right]_+ \tag{4.3}$$



Obrázek 4.9: Znázornění trénování síťě s využitím funkce triplet loss [22]

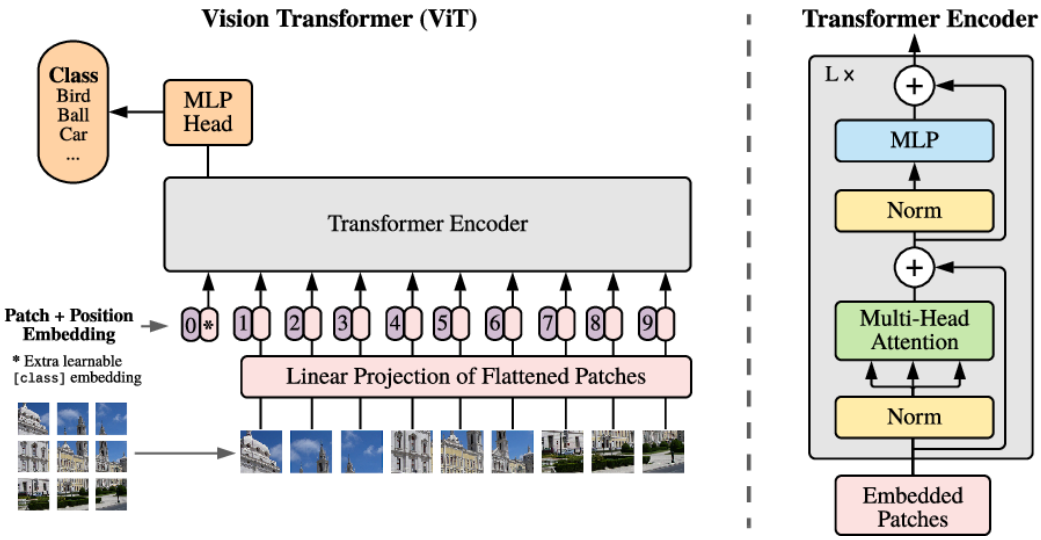
layer	size-in	size-out	kernel	param	FLPS
conv1	$220 \times 220 \times 3$	$110 \times 110 \times 64$	$7 \times 7 \times 3, 2$	9 K	115M
pool1	$110 \times 110 \times 64$	$55 \times 55 \times 64$	$3 \times 3 \times 64, 2$	0	
rnorm1	$55 \times 55 \times 64$	$55 \times 55 \times 64$		0	
conv2a	$55 \times 55 \times 64$	$55 \times 55 \times 64$	$1 \times 1 \times 64, 1$	4 K	13M
conv2	$55 \times 55 \times 64$	$55 \times 55 \times 192$	$3 \times 3 \times 64, 1$	111 K	335M
rnorm2	$55 \times 55 \times 192$	$55 \times 55 \times 192$		0	
pool2	$55 \times 55 \times 192$	$28 \times 28 \times 192$	$3 \times 3 \times 192, 2$	0	
conv3a	$28 \times 28 \times 192$	$28 \times 28 \times 192$	$1 \times 1 \times 192, 1$	37 K	29M
conv3	$28 \times 28 \times 192$	$28 \times 28 \times 384$	$3 \times 3 \times 192, 1$	664 K	521M
pool3	$28 \times 28 \times 384$	$14 \times 14 \times 384$	$3 \times 3 \times 384, 2$	0	
conv4a	$14 \times 14 \times 384$	$14 \times 14 \times 384$	$1 \times 1 \times 384, 1$	148 K	29M
conv4	$14 \times 14 \times 384$	$14 \times 14 \times 256$	$3 \times 3 \times 384, 1$	885 K	173M
conv5a	$14 \times 14 \times 256$	$14 \times 14 \times 256$	$1 \times 1 \times 256, 1$	66 K	13M
conv5	$14 \times 14 \times 256$	$14 \times 14 \times 256$	$3 \times 3 \times 256, 1$	590 K	116M
conv6a	$14 \times 14 \times 256$	$14 \times 14 \times 256$	$1 \times 1 \times 256, 1$	66 K	13M
conv6	$14 \times 14 \times 256$	$14 \times 14 \times 256$	$3 \times 3 \times 256, 1$	590 K	116M
pool4	$14 \times 14 \times 256$	$7 \times 7 \times 256$	$3 \times 3 \times 256, 2$	0	
concat	$7 \times 7 \times 256$	$7 \times 7 \times 256$		0	
fc1	$7 \times 7 \times 256$	$1 \times 32 \times 128$	maxout p=2	103M	103M
fc2	$1 \times 32 \times 128$	$1 \times 32 \times 128$	maxout p=2	34M	34M
fc7128	$1 \times 32 \times 128$	$1 \times 1 \times 128$		524 K	0.5M
L2	$1 \times 1 \times 128$	$1 \times 1 \times 128$		0	
total				140M	1.6 B

Tabulka 4.1: Konvoluční síť založená na Zeiler&Fergus použitá v FaceNet [31]

Autoři zdůrazňují důležitost správného výběru trojic pro optimální trénování. Je nutné vybírat trojice, které porušují podmínku z rovnice 4.2. Autoři dokonce pro kotvu x_i^a vybírají pozitivní snímek jako $argmax_{x_i^p} \|f(x_i^a) - f(x_i^p)\|_2^2$ a negativní jako $argmin_{x_i^n} \|f(x_i^a) - f(x_i^n)\|_2^2$. Vybírají tedy nejrozdílnější pozitivní a nejpodobnější negativní snímek. Vzhledem k náročnosti počítání argmin a armax nad celým datasetem autoři přicházejí se dvěma řešeními. Prvním z nich je vybírat trojice *offline*, každých n kroků pozastavit trénování a vypočítat argmin a armgax z posledního checkpointu. Druhé, které se ukázalo jako lepší, je vybírání náročných trojic *online* při trénování pouze z mini-batch o velikosti 1800 vzorků [22].

Vision transformer

Podobně jako v případě zpracování zvukových dat, tak i v případě vizuálních dat, se v posledních letech začala využívat architektura typu transformer, ktera se zakládá na mechanismu attention. Jedna z prvních a nejvýznamnějších publikací, které se podařilo efektivně adaptovat transformer architekturu na zpracování obrazu, vyšla v roce 2020, což znamená, že jde o relativně novou architekturu. Navržený **vision transformer** je použitý na rozpoznávání objektu na obrázku a lze ho dobře využít i k rozpoznávání obličejů [6]. Na rozdíl od transformerů popsaných v 3.4 se model učí s učitel v plně *supervised* stylu. V následujících letech vision transformer adoptovalo mnoho prací, které ukázaly, že i pro vizuální informaci se dá se skvělým výsledkem použít self-supervised učení, jako například SiT[1] nebo DINO[3], které sebou přináší všechny výhody self-supervised učení.



Obrázek 4.10: Přehled vision transformer architektury [6]

Transformery pracují s tokeny hodnot na vstupu. Jelikož self-attention počítá vztah každého tokenu s každým, nelze kvůli kvadratickému růstu výpočetní náročnosti naivně počítat attention pro každý pixel u většího obrázku. Obrázek je tedy rozdělen do několika menších nepřekrývajících se částí (*patches*). V původním publikaci [6] využívali u obrázků s velikostí 224x224 pixelů patch velikost 16x16 pixelů. Obrázek tedy rozdělili na 196 částí, pro které je již počítání self-attention výhodné. Tyhle části jsou zarovnány a projektovány do nižších dimenzí, pomocí jedné naučitelné plně propojené neuronové vrstvy na embeddingy. K obrázkovým embeddingům je přidán jeden speciální naučitelný **třídní embedding**. Je-

hož obsah je po průchodu transformer enkodéry použit ke klasifikaci. Před předložením obrázkových embdeddingů enkodéru je k nim ještě přičten jejich poziční embedding. Vzhledem k tomu, že self-attention mechanismus nemá zabudovaný způsob, jak vnímat pozici nebo pořadí vstupních embeddingů, je tenhle krok zcela zásadní. Na rozdíl od původních transformerů navržených pro zpracování přirozeného jazyka se u vision transformeru namísto použití pozičního kódování založeného na funkcích sinus a kosinus používá jednoduchý naučitelný poziční embedding, který se přičte obrazovému embeddingu. Embeddingy, obohacené o jejich poziční informaci, jsou následně předány transformer ekodéru s L bloky. Z výstupu posledního bloku je převzat výstupní třídní embedding, který je ještě předložen vícevrstvému perceptronu s jednou skrytou vrstvou a aktivační funkcí. Ztráta je následně spočítana pomocí cross-entropy loss nebo jiné libovolné ztrátové funkce. Vision transformery mají oproti klasickým konvolučním modelům několik výhod. Vision transformer je na rozdíl od CNN architektur schopný zpracovat obrázky různých velikostí, má globální kontext již od první vrstvy, což se může pro některé aplikace hodit a mělo by být jednodušší ho dotrénovat na konkrétní úkol.

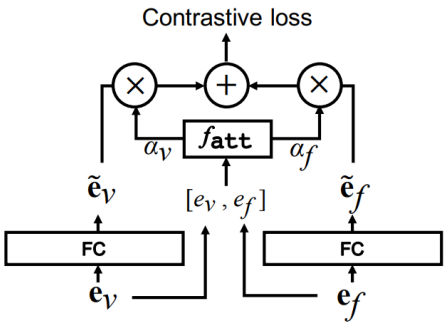
Kapitola 5

Audiovizuální verifikace

Kombinace zvukových a vizuálních informací umožňuje využít synergii mezi těmito dvěma modalitami a získat komplexnější a spolehlivější informace o identitě osoby. Kombinace těchto dvou modalit může také zlepšit schopnost systému identifikovat osoby v různých podmínkách, jako jsou špatné světelné podmínky nebo hlučné prostředí. Zároveň kombinace hlasu a obrazu může začít nabírat na důležitosti z hlediska bezpečnosti. Použití multimodálních dat umožňuje systému poskytovat vyšší úroveň přesnosti ověření identity, což může přispět k prevenci neoprávněného přístupu nebo zneužití identifikačních systémů. Problémem ale je, že špatná kvalita jedné z modalit může poškodit výsledek v případě, že druhá je v pořádku. Dalším důležitým faktem je, že hlas a obličej člověka jsou do jisté míry korelovány. Právě malá korelace mezi charakteristikami hlasu a obličeje, které určují identitu, zaručuje jejich dobrou kombinaci. Například model Speech2Face [15] generuje snímek obličeje podle hlasové nahrávky. To na teoretické úrovni znamená, že znalost obličeje asociovaného s hlasem může mít schopnost zlepšit diskriminativní vlastnosti systému pro rozpoznávání řečníka a obráceně.

Pro verifikaci osoby se jako nejintuitivnější způsob kombinace modalit nabízí fúze na úrovně skóre. To znamená vypočítat podobnost dvou záznamů zvlášť pro audio a video a vypočítat průměr oněch dvou podobností. Tato metoda je ekvivalentní s jednoduchou fúzí na úrovni embeddingů, kdy se audio a video embedding zkombinují obyčejnou konkatenací. Výsledná podobnost se poté počítá pouze mezi dvěma embeddingy reprezentujícími obě modality. Tyto naivní metody značně zvednou přesnost oproti uni-modální verifikaci ve většině případů. Ale selhávají v situacích, kdy je jedna z modalit poškozená nebo dokonce chybí. I běžný případ, kdy je jeden snímek obličeje vyfocen z profilu a druhý z frontální polohy, může celkovou podobnost páru zhoršit oproti využití pouze hlasu pro verifikaci.

Jako první řešení se objevilo využít kombinace embeddingů s využitím attention mechanismu. Konkrétně by se model měl naučit, jaké z modalit by měl dávat kolik pozornosti. Model využívající mechanismu nazývaný také jako **soft-attention** byl poprvé představen v [23]. Model je postavený nad dvěma separátními extraktory embeddingů. Model nejprve namapuje audio embedding e_a a obličejový embedding e_f do sdíleného vektorového prostoru pomocí jedné plně propojené neuronové vrstvy, která je separátní pro každou modalitu na $\tilde{e}_a$ a $\tilde{e}_f$. Paralelně s tím se z původních embeddingů e_a a e_f vypočítá attention hodnota pro obě modality. Oba vektory se skonkatenují a jsou předloženy neuronové vrstvě, která má pouze dvě výstupní hodnoty. Na výsledné hodnoty je použita softmax funkce a tím je získána váha každé z modalit. Embeddingy ve sdíleném prostoru jsou vynásobeny příslušnými váhami. Nakonec jsou vážené embedingy $\tilde{e}_a$ a $\tilde{e}_a$ sečteny a tím se obě modality sloučí do jedné. Tahle architektura je velmi jednoduchá a ukázala se jako dost efektivní.



Obrázek 5.1: Kombinace hlasového embeddingu e_v a obličejového embeddingu e_f za pomocí jednoduché soft-attenion [23]

Alternativou k této metodě je gated multi-modal fusion, jedná se o úpravu bran využívaných v rekurentních architekturách jako GRU nebo LSTM, využívanou pro řízení toku dat [19]. Architektura funguje podobně jako ta předchozí v tom, že mapuje embeddingy do společného vektorového prostoru. Nicméně po té na ně aplikuje *tanh* funkci, čímž jejich hodnoty dostane do normalizovaného rozsahu [-1,1]. A místo počítání attention k modalitám, skonkatenované embedingy e_a a e_f taktéž namapuje do společného prostoru $\tilde{e}$. A na výsledný embedding aplikuje *sigmoid* funkci, která hodnoty namapuje do intervalu [0,1]. Tenhle embedding z udává jaké modalitě dávat jakou váhu v každé z jejich dimenzí. Embeddingem z se vynásobí $\tilde{e}_a$ a embedding $\tilde{e}_f$ vynásobí $1 - z$. Výsledné vektory jsou sečteny a tím je fúze u konce. Z výsledků publikovaných v [19] lze vidět, že obě metody jsou téměř stejně účinné.

Je nutné poznamenat, že všechny publikace o audiovizuální verifikaci osoby používají jako vizuální data pouze jeden snímek obličeje nebo v případě [19] průměr obličejových embeddingů. Audiovizuální verifikace ve videu, která by využívala efektivně vlastnosti videa oproti fotografii, je tedy zatím ponechána budoucím studiím.

Kapitola 6

Návrh řešení, implementace a vyhodnocení

V téhle kapitole popíši celý proces návrhu řešení a jeho testování. Návrh obsahuje výběr knihoven, implementačních nástrojů a výběr datové sady. Zároveň budu ukazovat výsledky testování a jak jsem poznatky z výsledků interpretoval a použil pro další testy.

6.1 Výběr nástrojů

6.1.1 Knihovny

Jako hlavní vývojové prostředí jsem se rozhodl využít framework **PyTorch**, což je populární a výkonný open-source framework pro vývoj hlubokých neuronových sítí v Pythonu. PyTorch nabízí širokou škálu funkcí a nástrojů pro výzkum a implementaci různých modelů strojového učení, včetně konvolučních neuronových sítí, rekurentních sítí, transformérů a mnoho dalšího. PyTorch se vyznačuje svou schopností automaticky sledovat gradienty a provádět zpětnou propagaci chyb, což výrazně usnadňuje trénování složitých modelů. Díky tomu je možné snadno optimalizovat parametry modelu pomocí různých optimalizačních algoritmů, jako je například stochastický gradientový sestup. Navíc PyTorch nabízí nativní podporu pro využití grafických karet GPU s podporou **CUDA**, což umožňuje výpočetně náročné operace provádět rychleji a efektivněji. Díky těmto vlastnostem je vytváření modelů s PyTorch velmi rychlé a efektivní.

6.1.2 Výpočetní centrum

Veškeré trénování modelů bylo vykonáno na finském superpočítači **LUMI**. LUMI je jedno z nejmodernějších výpočetních zařízení v Evropě, které poskytuje obrovský výpočetní výkon pro vědecký výzkum a inovace v různých oblastech včetně umělé inteligence. LUMI má k dispozici výpočetní uzly skládající se z grafických karet AMD MI250X, na nichž jsem prováděl své výpočty.

6.2 Datové sady

Datových sad vhodných pro trénování modelů na audiovizualní verifikaci osob není mnoho, většina datových sad se zaměřuje pouze na jeden typ dat jako je hlas nebo snímek obličeje, avšak existují také datové sady vhodné pro audiovizuální verifikaci. Jako datovou sadu jsem

vybral dataset **Voxceleb**[1]. Voxceleb je veřejně dostupný dataset, shromážděný z rozhovorů se slavnými osobnostmi nahranými na server YouTube, běžně využívaný pro trénování řečových modelů. Avšak obsahuje též vizuální data z rozhovorů. Jedná se o rozsáhlý dataset rozdělen na *dev* část obsahující 148 tisíc promluv z 21 tisíc různých videí od 1211 různých osob a *eval* část obsahující 4874 promluv od 40-ti různých osob. Audiovizuální verze téhle datové sady se na internetu, bohužel, nachází pouze ve formě, kde je k řečové promluvě přiřazená pouze jedna fotografie nebo snímky vyjímané s vzorkovací frekvencí jeden za sekundu. Tahle verze datasetu[2] byla využita pro trénování modelů pracujících pouze nad jednou fotografií. Pro modely pracující nad videem byl celý Voxceleb stáhnut z YouTube, odkazy na videa, příslušná čísla snímku v záznamu a boundind boxy správného obličeje byly získány z veřejně dostupných stránek[3]. Část videí již bohužel na YouTube není nebo jsou nedostupné. Pro eval sadu to přesně znamená, že mi chybí 18 % videí, jejichž promluvy se vyskytují v 42 % verifikačních párů. Výsledky získané nad touto sadou tedy nejdou plně porovnat s výsledky získaných z celého datasetu.

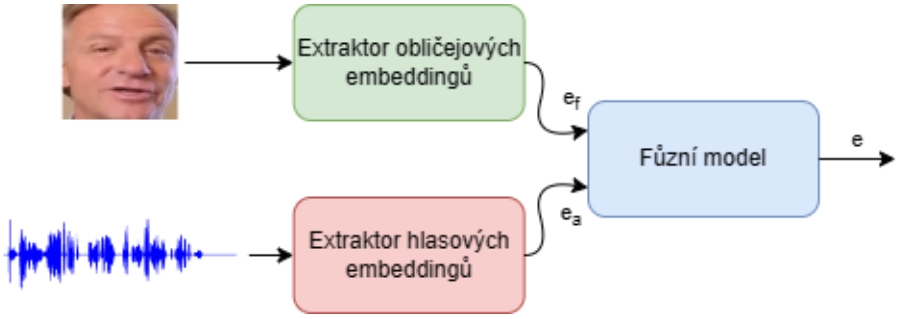
6.2.1 Příprava dat

Pro veškeré trénování a testování byla sjednocena délka hlasové záznamu na 3 vteřiny. To je nutné především při trénování, kdy jednotná velikost záznamu v jedné dávce umožní rychlejší trénování. Z hlasového záznamu bylo vždy vystřiženo okno 3 vteřin (zbytek audia se zahodil) a bylo uloženo v *wav* souboru s vzorkovací frekvencí 16 kHz. Získané snímky obličeje byly ještě vždy ořezány pomocí detektoru obličeje MTCNN implementovaného v PyTorch dostupného z [4] a velikost vyřezaného obličeje byla vždy změněna na 112 x 112 pixelů a obrázky byly uloženy ve formátu *jpg*. Pro trénování nad videem byla data taktéž předpřipravena před samotným trénováním pro urychlení procesu. Videa mají frekvenci 25 snímků za vteřinu. V každém snímku byl stejným způsobem jako u fotek získán výřez obličeje. Kvůli kvótám maximálního počtu souborů ve výpočetním centru bylo všech 75 obličeje jedné promluvy uloženo za sebou do jednoho obrázku formou koláže. Při načítání byly hodnoty pixelů převedeny do rozsahu [0,1] a hodnoty byly normalizovány, aby průměrná hodnota pixelu a směrodatná odchylka měla hodnotu 0.5.

6.3 Návrh experimentů

Navržené audiovizuální modely se vždy skládají ze tří částí, předtrénovaného modelu pro extrakci obličejových embeddingů, předtrénovaného modelu pro extrakci řečových embeddingů a mnou vytrénovaného modelu pro fúzi obou embeddingů. Pro extrahování obličejových embeddingů jsem se rozhodl využít extraktor dostupný z [5], extraktor využívá architektury Inception, konkrétně Inception Resnet V1 a byl vytrénován nad datasetem VGGFace2. Jedná se o velmi hlubokou síť využívající ResNet spojení a Inception bloků. Síť produkuje embeddingy o délce 512. Jako extraktor řečových embeddingů mi sloužil audio transformer WavLM base +, doplnění o pooling backend MHFA s 64-mi hlavami a délkou výstupního embeddingu 256 natrénován nad sadou Voxceleb 2.

[1] https://www.robots.ox.ac.uk/ vgg/data/voxceleb/vox1.html
[2] https://www.robots.ox.ac.uk/ vgg/research/CMBiometrics/
[3] https://mm.kaist.ac.kr/datasets/voxceleb/
[4] https://github.com/timesler/facenet-pytorch/tree/master
[5] https://github.com/timesler/facenet-pytorch/tree/master



Obrázek 6.1: Obecné znázornění audiovizuálního modelu pro verifikaci řečníka

Jako nejjednodušší základní model jsem použil obyčejnou konkatenaci audio a obličejového embbedingu. Takový model se nemusí trénovat a poslouží k porovnání výkonnosti složitějších modelů.

a) Třívrstvá neuronová síť

Jako první síť jsem navrhl skonkatenovat oba dva embeddingy a předložit je třem plně propojeným vrstvám neuronové sítě s dávkovou normalizací (batch norm) a ReLU aktivační funkcí po první a druhé vrstvě. Výstup první vrstvy má délku 1024, výstup druhé a třetí délku 512. Hloubku sítě tři vrstvy jsem navrhl, abych zjistil, jestli hlubší kombinace audio a vizuálního embeddingu přinese nějaké výhody proti obyčejné konkatenaci, a jestli má znalost reprezentací obou modalit schopnost se navzájem doplňovat.

b) Soft attention fúze

Jako druhý přístup jsem se rozhodl otestoval fúzi modalit využívající soft attention. Fúze by měla být schopná poznat, která modalita má lepší diskriminativní vlastnost a podle toho jí dát větší váhu. Transformační vrstvy, které transformují embeddingy e_a a e_f do společného prostoru $\tilde{e}$, jsou plně propojené vrstvy s dimenzionalitou výstupu 512 a obsahují bias:

$$\begin{aligned} \tilde{e}_a &= W_a^T + b_a \\ \tilde{e}_f &= W_f^T + b_f. \end{aligned} \tag{6.1}$$

Podobně attention vrstva je plně propojená vrstva, avšak s výstupní dimenzionalitou pouze 2. Na výstup attention vrstvy je ještě aplikován softmax.

$$\alpha_{\{a,f\}} = \text{softmax}(W_a^T[e_a, e_f] + b_a) \tag{6.2}$$

Jako finální výstupní embedding s délkou 512 se vezme vážený součet obou transformovaných embeddingů.

$$e = \alpha_a \tilde{e}_a + \alpha_f \tilde{e}_f \tag{6.3}$$

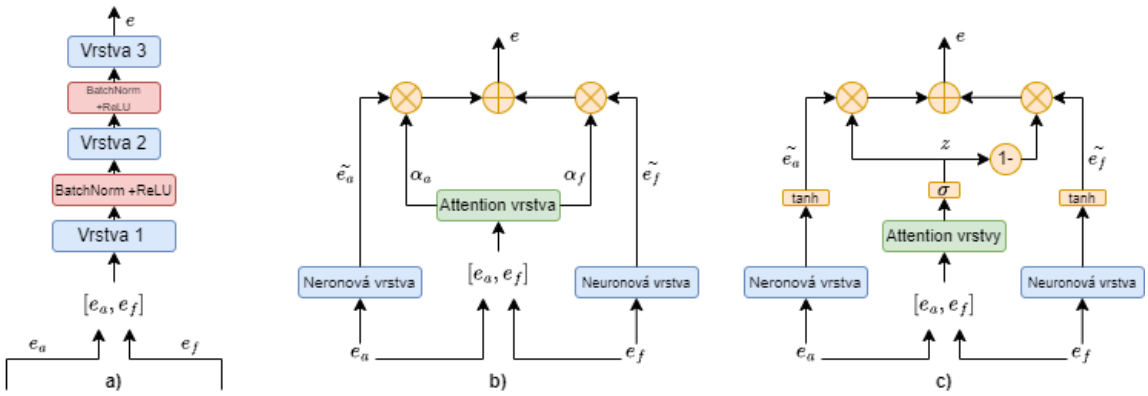
c) Gated fúze

Třetím testovaným fúzním model bude gated multi-modal fúze. Která rovněž počítá attention mezi modalitami, ale oproti soft-attention počítá váhy pro každou z hodnot ve společném vektorovém prostoru $\tilde{e}$. Konkrétně skonkatenované embeddingy e_a a e_f předložím dvěma plně propojeným neuronovým vrstvám s výstupními délkami 32 a 512, mezi

které je ještě vložena normalizace dávky a ReLU aktivační funkce. Na výsledný embedding z je použita sigmoid funkce a poté se všechny embeddingy zkombinují následujícím způsobem:

$$e = z \odot \tanh(\tilde{e}_a) + (1 - z) \odot \tanh(\tilde{e}_f) \tag{6.4}$$

Samotné transformační neuronové vrstvy jsou stejné jako u soft-attention.



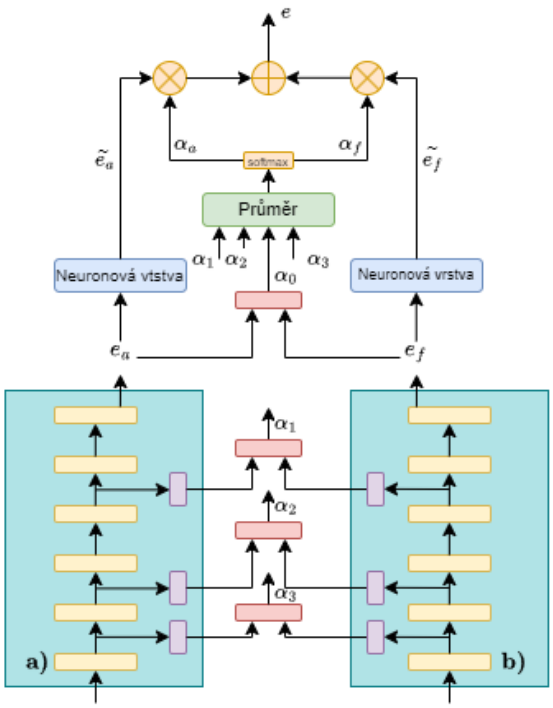
Obrázek 6.2: Znázornění fúzních modelů: **a)** konkatenace + 3 neuronové vrstvy, **b**) soft attention, **c)** multi-modal gated attention

U všech použitých návrhů jsou ještě před samotnou fúzí embeddingy e_a e_f normalizované pomocí L2 normalizace, aby se dostaly do stejného rozsahu a vstupovaly do fúzního systému s hodnotami stejné velikosti.

Soft attention nižších vrstev

Nedostatkem fúzních modelů využívající attention postavenými nad dvěma separátními extraktory embeddingů je, že extraktory jsou trénovány, aby dokázali reprezentovat diskriminativní vlastnosti dané modality užitečné pro verifikaci osoby. Výsledné embeddingy nemusí tedy dost dobře nést informaci o kvalitě původních dat [10]. Následující navržená architektura pramení z předpokladu, že kvalita zdrojových dat a tím i výsledného embeddingu je lépe pozorovatelná v nižších vrstvách architektury extraktoru [10]. Soft attention mezi modalitami se pokusím počítat již na nižších úrovní extraktorů příznaku a to hned na několika strategicky zvolených místech. Nakonec mezi attention hodnotami získaných z nižších úrovní architektury a z attention hodnoty získané z klasické soft attention nad embeddingy vypočítám průměr. Pomocí attention průměru převedeného na procenta pomocí softmaxu vypočítám vážený součet transformovaných embeddingů e_a a e_f. U audio extraktoru embeddingů jsem jako reprezentaci pro výpočet attention použil průměr mezi všemi výstupy WavLM transformeru po dané vrstvě. U extraktoru obličejových příznaků to není tak přímočaré, protože používám konvoluční architekturu. Embeddingy pro reprezentaci nižších vrstev jsem získal průměrem konvolučních map na jedné úrovni a zarovnáním výsledné mapy do jednoho embeddingu. Reprezentace obou modalit na nízké vrstvě je zkonkatenována a pomocí plně propojené vrstvy, která je pro každou nižší attention vrstvu separátní, je vypočítaná attention obou modalit. Lower level attention počítám na třech místech, u audio extraktoru se počítá po druhém, čtvrtém a šestém transformer enkodér bloku. U extraktoru obličejových příznaků byla attention počítána po dvou a pěti konvo-

lučních vrstvách, a po šesti konvolučních vrstvách a jednom incpetion bloku. Pro představu konvoluční mapy měly rozměry 32x53x53, 192x24x24 a 256x11x11.



Obrázek 6.3: Znázornění fúzního modelu počítající attention v nízkých vrstvách audio extraktoru **a)** a extraktoru obličejových příznaků **b)**; červeně jsou znázorněné attention bloky, fialově jsou bloky extrahující embeddingy pro attention výpočet

Augmentace dat

Pro natrénování a testování byl mimo čistého datasetu použit též augmentovaný dataset Voxceleb 1. Augmnetování bylo použito jako simulace poškození jedné z modalit. Byla použita agresivní augmentace, kdy s 80 % šancí byla jedna z modalit poškozená. U snímků obličeje bylo náhodně vybráno mezi dvěma formami augmentace. Buď ze snímku obličeje bylo náhodně vyřezáno okno 70 x 70 pixelů, které bylo následně zvětšeno zpět na 112 x 112 pixelů, což znamená, že v snímku bude pouze část obličeje a sníží se kvalita snímku. Druhá možnost je rozmazání snímku. Byl využit takzvaný Gaussian blur filtr o rozměrech 7 x 7 se směrodatnou odchylkou vybranou s rovnoměrným rozdělením z rozsahu 0.1 až 3. Pro augmentaci audia bylo taktéž náhodně vybráno ze dvou možností. Za prvé byl k hlasové nahrávce přidán náhodný zvuk z datasetu musan. Konkrétně byly použity zvuky pouze kategorie hluk nebo muzika. Nebo, za druhé, byl k hlasové nahrávce přidán náhodně generovaný šum.

Obrázek 6.4: Ukázka augmentace snímku obličeje, první obrázek je originální snímek, druhý je vystřižená část obrázku, třetí je rozmazání původního snímku

Ztrátová funkce

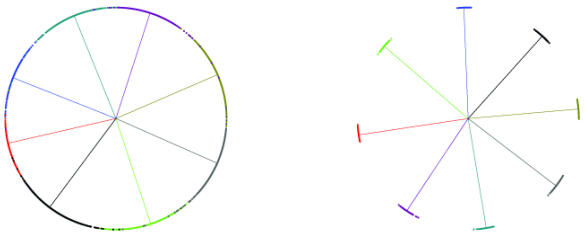
Pro trénování jsem zkoušel dva typy ztrátové funkce. Jako první jsem použil klasickou softmax cross-entropy ztrátu, která se běžně používá při klasifikaci v rámci neuronových sítí. Druhým typem ztrátové funkce, který jsem zkoušel, je *Additive Angular Margin Loss* nebo-li ArcFace, zkráceně AAM. AAM se snaží vzorky stejné třídy shlukovat k sobě a jednotlivé třídy od sebe oddělit. Funkce vychází z klasické softmax ztráty, kterou autoři [4] AAM definují takto:

$$L = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{e^{W_{y_i}^T+b_{y_i}}}{\sum_{j=1}^{n}e^{W_j^T x_i+b_j}} \tag{6.5}$$

kde $x_i \in \mathbb{R}^d$ je příznak vzorku i, patřící třídě y_i. W_j^T značí sloupec j váhové matice W a b_j je bias, N je velikost dávky a n je počet tříd. Autoři danou funkci upravili na

$$L = -\frac{1}{N}\sum_{i=1}^{N}\frac{e^{s(\cos(\theta_{y_i}+m))}}{e^{s(\cos(\theta_{y_i}+m))}+\sum_{j=1,j\neq y_i}^{n}e^{s\cos\theta_j}} \tag{6.6}$$

Autoři tedy odstranili bias. $W_j^T x_i$ upravili na $\|W_j\|\,\|x_i\|\cos\theta_j$, kde θ_j je úhel mezi váhou W_j a příznakem x_i. Díky L2 normalizaci došlo k nastavení váhy $\|W_j\| = 1$ a velikost $\|x_i\|$ je nastavena na s, což je parametr funkce. A m je úhel přidán mezi W_{y_i} a x_j [4].



(a) Softmax (b) ArcFace

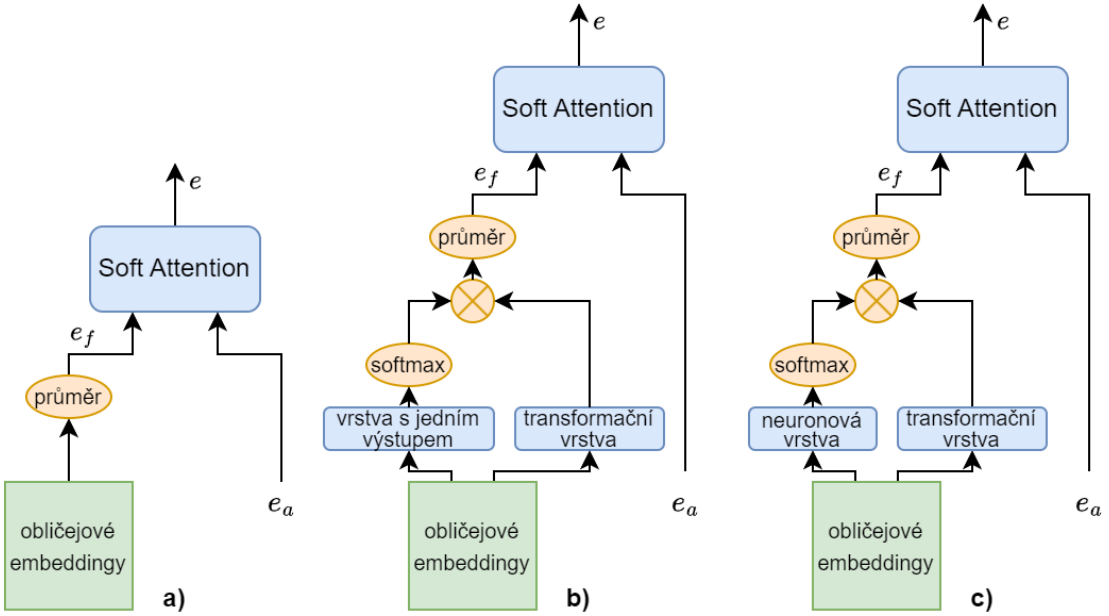
Obrázek 6.5: 2D znázornění ArcFace ztráty pro 8 různých tříd [4].

Zvolil jsem tyto ztrátové funkce, abych porovnal jejich účinnost při trénování mého modelu, a zjistil, která z nich lépe odpovídá mému cíli dosáhnout, co nejlepších výsledků.

Rozpoznání ve videu

Zjištěné poznatky z rozpoznávání osoby ze snímku obličeje a hlasu řídily následující návrhy modelů pro rozpoznávání osoby ve videu. V mísení modalit se jako velmi efektivní

metoda ukázala soft attention, proto se modely navržené pro video na ni zakládají. Oproti předchozím modelům se videu musíme vypořádat s větším množstvím snímků obličeje. Fúznímu modelu je namísto jednoho obličejového embeddingu předložen obličejový embedding z každého snímku videa. Jako nejjednodušší naivní způsob se jeví udělat průměr mezi obličejovými embeddingy napříč videem a výsledný embedding skonkatenovat s audio embeddingem. Průměr více obličejových embeddingů přináší robustnější reprezentaci obličeje než embedding z jednoho snímku, avšak celkový výsledek může pokazit pár nevhodných snímků, ať už z důvodu poškození kvality nebo třeba jenom otočením hlavy osoby. Navržené modely se tedy snaží zjistit, které obličejové embeddingy mají jakou kvalitu, a podle toho s nimi nakládat.



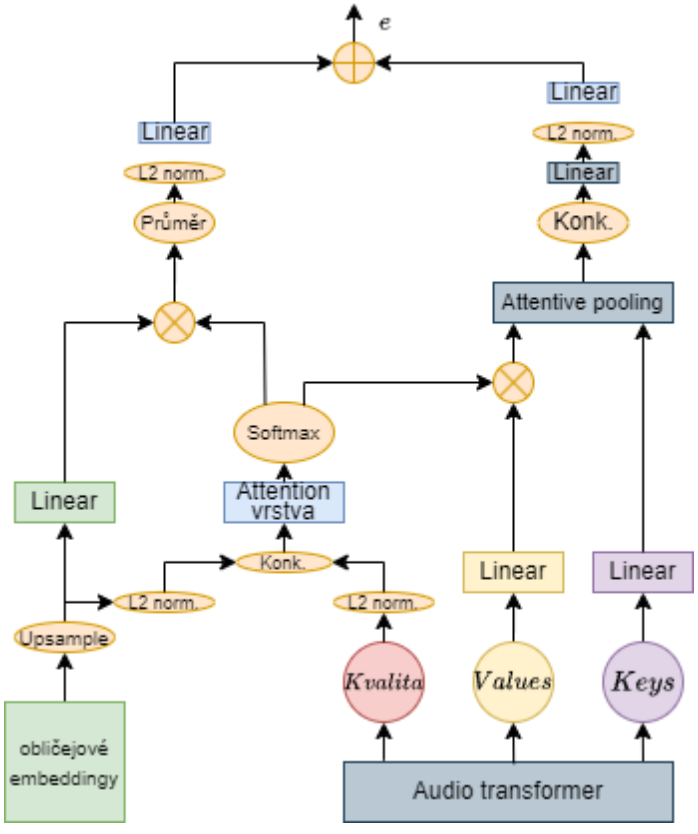
Obrázek 6.6: Navržené modely pro rozpoznávání osoby ve videu

Model **a)** je jednoduchý model, který vypočítá průměr mezi obličejovými embeddingy a následně obličejový embedding e_f a audio embedding e_a sloučí pomocí soft attention popsané výše v téhle kapitole. U každého modelu se před soft attention fúzí embeddingy ještě L2 normalizují.

Model **b)** se snaží zjistit váhu pro každý obličejový embedding podle jeho diskriminativních vlastností a díky tomu v ideální případě ignorovat špatné embeddingy. Využívá k tomu neuronovou vrstvu s pouze jedním výstupem. Každý embedding si tedy vygeneruje vlastní váhu. Následně je na váhy embeddingů aplikován softmax, čímž se získá skutečná váha. Původní obličejový embeddingy se transformují jednou lineární neuronovu vrstvou, se stejnou dimenzí vstupu jako výstupu. Transformované embeddingy se vynásobí se svou váhou a vypočítá se průměr mezi všemi embeddingy.

Model **c)** je podobný modelu b) s tím rozdílem, že nepočítá váhu každého embeddingu zvlášť, ale pomocí jedné sítě, která na vstupu bere všechny skonkatenované obličejové embeddingy a na výstupu vydá váhy pro všechny embeddingy. Model má jasnou nevýhodu, a to že přijímá pouze vstup fixní délky, použité při trénování. Oproti modelu b) má ale výhodu, že embeddingy bere jako sekvenci, může se tedy naučit, že pokud je jeden embedding špatný tak snížit váhu i okolním embdeddingům.

Poslední navržený model se videu pokouší počítat soft attention navzájem mezi každým video snímkem a příslušným audio rámcem. Dočasné poškození jedné modalit by tedy mělo mít minimalizovaný odraz na výsledek rozpoznávání. Protože audio embedding reprezentuje celou promluvu, musel jsem pracovat na úrovní MHFA, které vykonává attentive pooling mezi jednotlivými rámci. Jako první krok se obličejové embeddingy upsamplují na frekvenci výstupů audio transformeru, u videa s 25 snímky za vteřinu je frekvence audio výstupů přibližně dvakrát větší. Stejným způsobem jako MHFA získává *Values* a *Keys*, a to je vážený součet výstupů transformer bloků napříč všemi bloky, kde váhy jsou naučitelné parametry, získávám novou hodnotu *kvality* pro každý frame. Hodnota by měla zachytit diskriminační schopnost daného framu. L2 normalizované kvality embeddingy a obličejové embeddingy se skonkatenují. Následně se pomocí jedné lineární vrstvy s dvěma výstupy vypočítají attention hodnoty. Z těch se pomocí softmaxu vypočítá váha pro každou z modalit v daném rámci. V případě videa se obličejové embeddingy transformují pomocí jedné vrstvy bez změny dimenzionality a transformované embeddingy jsou vynásobeny svou váhou. Mezi všemi váženými obličejovými embeddingy se udělá průměr napříč videem. Výsledný embedding se L2 normalizuje a projde poslední lineární vrstvou beze změny dimenzionality. S váhou audio snímku se vynásobí transformované Values. MHFA následně pokračuje beze změny. Výsledný audio embedding se L2 normalizuje a transformuje na stejnou délku jako obličejový embedding. Výsledný audio a video embedding se sečte.



Obrázek 6.7: Fúze využívající soft attention mezi modalitami v každém video snímku a frame-wise výstupu transformer bloků

6.3.1 Testování a výsledky

Navržené architektury byly vytrénovány a poté porovnány mezi sebou pro získání poznatků. Všechny modely byly trénovány nad dev částí datasetu Voxceleb 1 a testovány nad jeho test částí podle oficíalního trialu. Všechny modely byly trénovány dvacet epoch a pro prezentaci výsledků byly vybrány modely s nejnižším EER na testovací sadě po dané epoše. Pro trénování byla nastavena velikost jedné dávky na 80, což představovalo ideální kompromis mezi rychlostí trénování a rychlostí konvegence modelu. Nejdůležitější trénovací hyperparameter **learning rate** byl testováním určen na hodnotu 0.005. Každou novou epochu byl learning rate zmenšen vynásobením faktorem 0.7. Pro optimalizaci vah byla využita metoda **SGD** (Stochastic Gradient Descent) s parametrem momentum nastaven na hodnotu 0.9. Přičemž váhy extraktorů embeddingů byly zamražené a neměnily se. Pro vyhodnocování výsledků slouží dvě klasické metriky equal error rate a minDCF, minDCF byla počítána pro $c_{miss} = 1$, $c_{fa} = 1$ a pravděpodobnost P_{target} byla nastavena na 0.01. Pro porovnání výsledků byly zjištěny výsledky unimodálních extraktorů nad testovací sadou a výsledky při konkatenaci audio a video embeddingu viz. tabulka 6.1.

	EER	**minDCF**
Audio	1.610	0.200
Video	3.387	0.357
Kombinace	0.499	0.059

Tabulka 6.1: Test použitých extraktorů nad vyhodnocovací sadou Voxceleb1 test; Audio je použitý extraktor MHFA_64 WavLM Base+ nad 3 vteřinovým segmentem; video je použitý Facenet Inception_resnet_v1 nad jedním snímkem obličeje; Kombinace je konkatenace obou embeddingů

Audio a snímek obličeje

Při testování jsem jako první chtěl zjistit, jaká ztrátová funkce má nejlepší výsledky. Během zjišťování ideální hodnoty learning rate jsem testy prováděl s oběma pro použití navržených ztrátových funkcí. V tabulce 6.2 jsou výsledky modelů využívajících soft attention pro různé learning rate a obě ztrátové funkce. Ačkoliv žádný z learning rate použitých v tabulce 6.2 nebyl vybrán jako optimální a learning rate scheduling stategie byla nastavená vadně, tak výsledky dávají jasný přehled o optimální ztrátové funkci. Softmax ztráta vykazuje řádově horší výsledky oproti AAM. Ukázalo se, že využití správné ztrátové funkce je pro dobrý výsledek stejně zásadní jako výběr fúzního modelu, protože modely využívající softmax ztrátu při trénování se zhoršily oproti naivní konkatenaci. Pro AAM byly použity parametry scale = 30 a margin = 0.2. Proto pro všechny nadcházející experimenty se používá pouze AAM ztrátová funkce.

	SoftMax loss		**AAM loss**	
lr	**EER**	**minDCF**	**EER**	**minDCF**
0.1	2.248	0.268	0.382	0.059
0.01	3.3200	0.350	0.356	0.060
0.001	1.170	0.148	0.356	0.064

Tabulka 6.2: Výsledky pro různé hodnoty learning rate a různé ztrátové funkce s modelem soft attention; parametry AAM: scale=30, margin=0.2

Díky správně vybrané ztrátové funkci testování samotných fúzních modelů začalo přinášet pozitivní výsledky 6.3. Z navržených fúzních modelů nejhůře dopadl model využívající fúzi pomocí tří neuronových vrstev. Tento model se dokonce zhoršil proti naivní konkatenaci. Z toho jsem vyvodil, že snaha mísit modality více vrstvami transformací nepřináší kýžené výsledky. Tedy, že lepší je vypočítat váhy pro jednotlivé modality a výsledné embeddingy pouze sloučit do jednoho. Na druhé straně ostatní fúzní modely přinesly zlepšení oproti konkatenaci. Soft attention dopadl o trochu lépe než multi-modal gated attention. A vzhledem k tomu, že soft attention je jednodušší architektura, dá se říci, že pro verifikaci osoby je vhodnější než gated attention. Model počítající průměr attention hodnot napříč nižšími vrstvami a výslednými embeddingy přinesl ještě o něco lepší výsledky než soft attention. Tím potvrdil, že informace o kvalitě zdrojových dat se dají lépe zjistit z nižších vrstev architektury.

typ fúze	**EER**	**minDCF**
konkatenace	0.499	0.059
3 neuronové vrstvy	1.169	0.146
soft attention	0.350	0.060
gated attention	0.388	0.069
lower levels soft attention	0.340	0.052

Tabulka 6.3: Výsledky fúzních modelů nad testovací sadou Voxceleb 1 test

Další řada experimentů se pokusili zjistit, jestli rozmražení extraktorů embeddingu a jejich joint trénování společně s fúzním modelem přinese zlepšení oproti trénování s zmraženými extraktory. Fúzní model s třemi neuronovými vrstvami již testován nebyl. Předchozí testy jasně ukázaly, že je podřadný k ostatním modelům. Abychom předešli přetrénování a zachování kvality extraktorů byl použit *layer-wise learning rate decay*. Každé vrstvě extraktorů embeddingů byl přiřazen jiný learning rate, aby si nižší vrstvy zachovaly schopnost kvalitně reprezentovat data a vyšší vrstvy se mohly přizpůsobit novému úkolu. Pro dvanáct bloků WavLM modelu byly learning raty vypočítány podle vzorce přebraného z [16].

$$\mathrm{LR}_l = \mathrm{LR}_1 \times \xi^{l-1} \tag{6.7}$$

LR_1, learning rate nejnižšího enkodér bloku, je nastaven 1×10^{-5}. Násobící faktor ξ byl nastaven na hodnotu 1.45. Hodnoty learning rate pro enkodér bloky byly tedy v rozsahu 0.00001 až 0.0006. Learning rate pro MHFA extraktoru audio embeddingů byl nastaven na 0.001. Konvoluční vrstvy před transformer modelem zůstaly zmražené. Extraktor obličejových embeddingů nešel jednoduše rozdělit na jednotlivé vrstvy, tak byl rozdělen na 13 příhodně vycházejících částí. A těm byl přidělen learning rate podle stejného vzorce, ale jako hodnota násobícího faktoru ξ byla použita hodnota 1.4.

Tabulka 6.4 ukazuje, že u každého modelu došlo k malému zlepšení. Zlepšení ovšem přišlo za cenu pomalejšího trénování. Trénovací program totiž musel počítat gradienty a optimalizovat parametry obou extraktorů, kterých je dohromady přes 100 milionů. Největší zlepšení zaznamenal model počítající attention v nižších vrstvách extraktorů.

typ fúze	bez joint tréninku		s joint tréninkem	
	EER	**minDCF**	**EER**	**minDCF**
soft attention	0.350	0.060	0.324	0.059
gated attention	0.388	0.069	0.388	0.063
lower levels soft attention	0.340	0.052	0.303	0.050

Tabulka 6.4: Rozdíly mezi tréninkem modelu s a bez joint tréninku

Poslední várka experimentů nad audiem a snímkem obličeje se týkala augmentace dat. Nad augmentovaným datasetem popsaným v kapitole 6.3 jsem pro porovnání vyhodnotil předtrénované extraktory viz. tabulka 6.5. Je vidět, že augmentovaný dataset je značně náročný, výsledky se velmi zhoršili oproti výsledkům na čistým datasetem.

	EER	**minDCF**
Audio	4.829	0.391
Video	9.945	0.538
Kombinace	2.291	0.201

Tabulka 6.5: Výsledky základních extraktorů nad augmentovaným datasetem Voxceleb 1 Test; modely jsou totožné s těmi v tabulce 6.1

Tabulka 6.6 ukazuje vyhodnocení modelů trénovaných nad čistým datasetem nad augmentovaným datasetem. Modely ukazují zlepšení oproti naivní konkatenaci, avšak zhruba jenom a půl procenta EER. Modely vytrénované nad čistými daty mají tedy schopnost rozlišovat, který z embeddingů má lepší diskriminativní vlastnosti, avšak nejsou schopné spolehlivě rozlišit, když je jedna z modalit významně poškozená.

typ fúze	bez joint tréninku		s joint tréninkem	
	EER	**minDCF**	**EER**	**minDCF**
3 neuronové vrstvy	3.595	0.350	—	—
soft attention	1.754	0.177	1.616	0.150
gated attention	1.781	0.176	1.553	0.195
lower levels soft attention	1.701	0.174	1.537	0.158

Tabulka 6.6: Modely trénované nad čistým datasetem vyhodnocené nad augmentovanou datovou sadou

Při testování modelů nad augmentovaným datasetem se ukázala důležitost joint tréninku, kdy všechny modely trénované s joint tréninkem mají nižší EER zhruba o 0.5 % oproti svým protějškům bez joint trénování 6.7. Výsledky prezentují uspokojivě zvýšenou odolnost na poškozená data.

typ fúze	bez joint tréninku		s joint tréninkem	
	EER	**minDCF**	**EER**	**minDCF**
soft attention	1.504	0.163	1.132	0.151
gated attention	1.478	0.156	1.021	0.134
lower levels soft attention	1.489	0.178	1.010	0.129

Tabulka 6.7: Modely trénované nad augmentovanými daty vyhodnocené nad augmentovanou datovou sadou

Při vyhodnocování modelů trénovaných nad augmentovanými daty čistým datasetem 6.8 se ukázalo, že modely dosahují mírně horších výsledků než modely trénované nad čistým datasetem. S výjimkou gatted attention s joint tréninkem. Je tedy na zvážení, jestli přidaná robustnost modelu nad poškozenými daty stojí za jemně zhoršenou přesnost nad daty čistými.

typ fúze	**bez joint tréninku**		**s joint tréninkem**	
	EER	**minDCF**	**EER**	**minDCF**
soft attention	0.489	0.078	0.425	0.080
gated attention	0.478	0.075	0.382	0.060
lower levels soft attention	0.446	0.070	0.366	0.057

Tabulka 6.8: Modely trénované nad augmentovanými daty vyhodnocené nad čistou datovou sadou

Obrázek 6.8: Procentuální vyjádření kolik pozornosti věnoval attention layer (u soft attention modelu) snímkům obličeje pro různé snímky proti stejnému hlasovému záznamu

Video modely

Modely navržené pro fúzi audia a videa byly taktéž trénovány nad čistým i augmentovaným datasetem. Augmentační strategie byla stejná jako u fúze snímku a audia. Ale místo jednoho snímku byla augmentovaná sekvence snímku ve videu. Ve třívteřinovém videu byl náhodně vybrán začátek sekvence a ze zbylé části videa byl náhodně vybrán konec sekvence. Pro porovnání jako naivní základní způsob fúze sloužil průměr obličejových embeddingů skonkatenovaný s audio embeddingem viz. tabulka 6.9.

	čistá data		**augmentovaná data**	
	EER	**minDCF**	**EER**	**minDCF**
Audio	1.629	0.195	5.184	0.426
Video	1.163	0.112	1.520	0.123
Konkatenace	0.187	0.024	0.450	0.055

Tabulka 6.9: Výsledky jednoduchých modelů nad testovacími sadami; Audio je embedding z hlasu řečníka, Video je průměr všech obličejových embeddingů, Konkatenace je konkatenace obou Audio a Video embeddingu

Testováním nad augmentovaným datasetem ukázalo, že průměr všech obličejových snímků sám o sobě stačí jako dobrý způsob boje se ztrátou kvality dat. K překvapení naopak model a), jednoduchá soft attention mezi průměrem obličejových embeddingů napříč videem a audio embeddingem, se zhoršil proti konkatenaci 6.10. U robustnější reprezentace obličeje napříč videem bylo tedy náročnější určovat kvalitu videa. Modely a) a b), které přidělovaly

každému obličejovému embeddingu váhu a podle ní počítaly vážený průměr obličejových embeddingů a následně počítaly soft attention mezi audio a video embeddingem, se oba zlepšily proti konkatenaci. Úspěšně se tedy podařilo zjišťovat kvalitu video snímků a podle ní dělat průměr. Nejlépe dopadl model, který počítá soft attention navzájem mezi rámci jednotlivých modalit.

	EER	**minDCF**
konkatenace	0.450	0.055
fúzní video model **a)**	0.497	0.034
fúzní video model **b)**	0.384	0.038
fúzní video model **c)**	0.413	0.035
cross-modal frame soft attention	0.369	0.030

Tabulka 6.10: Vyhodnocení modelů trénovaných nad augmentovanými daty nad augmentovaným test trialem; konkatenace je konkatenace průměru obličejových embeddingů a audio embeddingu, model a) je soft attention model využívající síť s jedním výstupem pro výpočet váhy obličejového embeddingu, c) je model využívající síť pro fixní délku videa, cross-modal frame soft attention počítá soft attention mezi rámci obou modalit navzájem

Při vyhodnocování modelů trénovaných nad augmenotvanými daty nad čistým test trialem. 6.11 se jako nejlepší ukázala naivní jednoduchá konkatenace průměru obličejového embeddingu s audio embeddingem. Průměr obličejových embeddingů přináší robustní reprezentaci osoby a audio embedding jako takový nejde více vylepšit. Z natrénovaných modelů se jako nejlepší ukazuje model počítající soft attention navzájem mezi obličejovými embeddingy a frame-wise výstupu audio transformeru bloků.

	EER	**minDCF**
konkatenace	0.187	0.024
fúzní video model **a)**	0.262	0.023
fúzní video model **b)**	0.309	0.010
fúzní video model **c)**	0.253	0.018
cross-modal frame soft attention	0.225	0.012

Tabulka 6.11: Modely trénované nad augmentovanými daty a vyhodnocené nad čistými daty

U modelů trénovaných a nad čistými daty jsem neočekával výrazné zlepšení proti naivní metodě konkatenace průměru embeddingů obličeje a audio embeddingu, protože při dobré kvalitě obou modalit jsem si nemyslel, že má fúze čím přispět. Navíc se již pohybujeme v nízkých desetinách jednoho procenta EER, takže případné zlepšení bude spíše nepatrné. Ze čtyř navržených modelů se dva zlepšily proti konkatenaci. Soft attention mezi rámci se opět ukázala jako nejlepší metoda 6.12, kdy se proti konkatenaci zlepšila o tři setiny procenta na 0.159 % EER. V poměru ke konkatenaci se jedná o patnáctiprocentní zlepšení. Jednoduchý model a), který počítá soft attention mezi audio embeddingem a průměrem obličejových embeddingů, se významně zhoršil. Soft attention, která se u snímků obličeje ukázala jako jednoduchá výkonná metoda, se tedy nedá dobře aplikovat na průměr obličejových embeddingů.

	EER	**minDCF**
konkatenace	0.187	0.024
fúzní video model **a)**	0.215	0.018
fúzní video model **b)**	0.168	0.011
fúzní video model **c)**	0.197	0.011
cross-modal frame soft attention	0.159	0.009

Tabulka 6.12: Modely trénované nad čistými daty a vyhodnocené nad čistými daty

Kapitola 7

Závěr

Cílem bakalářské práce bylo navrhnout a vytvořit modely pro audiovizuální verifikaci osob ve videu nebo ze snímku obličeje a záznamu hlasu. Také se zaměřit na efektivní kombinaci audio a video embeddingů získaných z předtrénovaných extraktorů embeddingů. Výsledný model by měl být odolný proti poškození jedné z datových modalit.

Bylo vytvořeno mnoho fúzních modelů embeddingů ze snímku obličeje a nahrávky hlasu. Nejlepšími výsledky se prezentoval model počítající soft attention mezi modalitami již na nízkých úrovní extraktorů embeddingů. Model dosáhl zlepšení z 0.499 % na 0.340 % EER proti naivní konkatenaci embeddingů na uznávaném testovacím trialu Voxceleb1-O. Taktéž byly vytvořeny modely trénované a testované nad augmentovanou verzí datasetu, která simuluje poškození jedné z modalit, u nejlepšího z nich se výsledky zlepšily z 2.291 % na 1.010 % EER proti konkatenaci embeddingů.

U testování nad videem se ukázala jednoduchá metoda průměru mezi obličejovými embeddingy a konkatenace s audio embeddingem jako velmi efektivní a odolná proti poškození modalit. Avšak navrženým modelů se podařilo jednoduchou metodu nad augmentovanými daty překonat. Nejlepší výsledky poskytl model, který počítal váhu mezi modalitami v každém video snímku a frame-wise výstupu transformer bloků. Nad augmentovanými daty se oproti konkatenaci jednalo o zlepšení z 0.450 % na 0.369 % EER. U čistých dat se stejnému modelu podařilo zaznamenat zlepšení z 0.187 % na 0.159 % EER.

Výsledky práce by se zajisté daly vylepšit technickými detaily s velkým významem jako je trénink nad rozsáhlejší datovou sadou Voxceleb 2, představení optimálnější learning rate scheduling strategie nebo zvolit jinou strategii augmentace dat při trénování. Do budoucna by se navržené modely mohly naučit, místo přidělování nízkých vah poškozeným částem modalit, ony nevhodná data přímo zahazovat. Video modely by také bylo vhodné vyhodnocovat nad těžším datasetem s více trialy kvůli statistické signifikantnosti výsledků.

Literatura

[1] Atito, S., Awais, M. a Kittler, J. *SiT: Self-supervised vIsion Transformer*. 2022.

[2] Behera, G. S. *Face Detection with Haar Cascade*. Dostupné z: https://towardsdatascience.com/face-detection-with-haar-cascade-727f68dafd08.

[3] Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J. et al. *Emerging Properties in Self-Supervised Vision Transformers*. 2021.

[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. DOI: 10.1109/tpami.2021.3087709. ISSN 1939-3539. Dostupné z: http://dx.doi.org/10.1109/TPAMI.2021.3087709.

[5] Devlin, J., Chang, M.-W., Lee, K. a Toutanova, K. *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. 2019.

[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.

[7] Gradilla, R. *Multi-task Cascaded Convolutional Networks (MTCNN) for Face Detection and Facial Landmark Alignment*. Dostupné z: https://medium.com/@iselagradilla94/multi-task-cascaded-convolutional-networks-mtcnn-for-face-detection-and-facial-landmark-alignment-7c21e8007923.

[8] He, K., Zhang, X., Ren, S. a Sun, J. Deep Residual Learning for Image Recognition. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, s. 770–778. DOI: 10.1109/CVPR.2016.90.

[9] Hsu, W.-N., Bolte, B., Tsai, Y.-H. H., Lakhotia, K., Salakhutdinov, R. et al. *HuBERT: Self-Supervised Speech Representation Learning by Masked Prediction of Hidden Units*. 2021.

[10] Hörmann, S., Moiz, A., Knoche, M. a Rigoll, G. Attention Fusion for Audio-Visual Person Verification Using Multi-Scale Features. In: *2020 15th IEEE International Conference on Automatic Face and Gesture Recognition (FG 2020)*. 2020, s. 281–285. DOI: 10.1109/FG47880.2020.00074.

[11] Kamil, O. Frame Blocking and Windowing Speech Signal. Prosinec 2018, sv. 4, s. 87–94.

[12] Li, X., Xu, Y., Lv, Q. a Dou, Y. Affine-Transformation Parameters Regression for Face Alignment. *IEEE Signal Processing Letters*. 2016, sv. 23, č. 1, s. 55–59. DOI: 10.1109/LSP.2015.2499778.

[13] Muda, L., Begam, M. a Elamvazuthi, I. Voice Recognition Algorithms using Mel Frequency Cepstral Coefficient (MFCC) and Dynamic Time Warping (DTW) Techniques. *CoRR*. 2010, abs/1003.4083. Dostupné z: http://arxiv.org/abs/1003.4083.

[14] Nossier, S., Wall, J., Moniri, M., Glackin, C. a Cannings, N. A Comparative Study of Time and Frequency Domain Approaches to Deep Learning based Speech Enhancement. In:. červenec 2020, s. 1–8. DOI: 10.1109/IJCNN48605.2020.9206928.

[15] Oh, T.-H., Dekel, T., Kim, C., Mosseri, I., Freeman, W. T. et al. *Speech2Face: Learning the Face Behind a Voice*. 2019.

[16] Peng, J., Plchot, O., Stafylakis, T., Mosner, L., Burget, L. et al. *An attention-based backend allowing efficient fine-tuning of transformer models for speaker verification*. 2022.

[17] Plchot, O. *Extensions to Probabilistic Linear Discriminant Analysis for Speaker Recognition*. Brno, CZ, 2014. Ph.D. thesis. Brno University of Technology, Faculty of Information Technology. Dostupné z: https://www.fit.vut.cz/study/phd-thesis/347/.

[18] Prasad, N. V. a Umesh, S. Improved cepstral mean and variance normalization using Bayesian framework. In: *2013 IEEE Workshop on Automatic Speech Recognition and Understanding*. 2013, s. 156–161. DOI: 10.1109/ASRU.2013.6707722.

[19] Qian, Y., Chen, Z. a Wang, S. Audio-Visual Deep Neural Network for Robust Person Verification. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*. 2021, sv. 29, s. 1079–1092. DOI: 10.1109/TASLP.2021.3057230.

[20] Raj, B. *A Simple Guide to the Versions of the Inception Network*. Dostupné z: https://towardsdatascience.com/a-simple-guide-to-the-versions-of-the-inception-network-7fc52b863202.

[21] Schneider, S., Baevski, A., Collobert, R. a Auli, M. Wav2vec: Unsupervised Pre-Training for Speech Recognition. In:. Září 2019, s. 3465–3469. DOI: 10.21437/Interspeech.2019-1873.

[22] 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)*. 2015, s. 815–823. DOI: 10.1109/CVPR.2015.7298682.

[23] Shon, S., Oh, T.-H. a Glass, J. *Noise-tolerant Audio-visual Online Person Verification using an Attention-based Neural Network Fusion*. 2018.

[24] Simonyan, K. a Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. *CoRR*. 2014, abs/1409.1556. Dostupné z: https://api.semanticscholar.org/CorpusID:14124313.

[25] Snyder, D., Garcia Romero, D., Sell, G., Povey, D. a Khudanpur, S. X-Vectors: Robust DNN Embeddings for Speaker Recognition. In: *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2018, s. 5329–5333. DOI: 10.1109/ICASSP.2018.8461375.

[26] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S. et al. Going deeper with convolutions. In: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015, s. 1–9. DOI: 10.1109/CVPR.2015.7298594.

[27] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L. et al. Attention is all you need. *Advances in neural information processing systems*. 2017, sv. 30.

[28] Viola, P. a Jones, M. Rapid object detection using a boosted cascade of simple features. In: *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*. 2001, sv. 1, s. I–I. DOI: 10.1109/CVPR.2001.990517.

[29] Waibel, A., Hanazawa, T., Hinton, G., Shikano, K. a Lang, K. Phoneme recognition using time-delay neural networks. *IEEE Transactions on Acoustics, Speech, and Signal Processing*. 1989, sv. 37, č. 3, s. 328–339. DOI: 10.1109/29.21701.

[30] Wang, H., Liang, C., Wang, S., Chen, Z., Zhang, B. et al. Wespeaker: A Research and Production Oriented Speaker Embedding Learning Toolkit. In: *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2023, s. 1–5. DOI: 10.1109/ICASSP49357.2023.10096626.

[31] Zeiler, M. D. a Fergus, R. Visualizing and Understanding Convolutional Networks. *ArXiv*. 2013, abs/1311.2901. Dostupné z: https://api.semanticscholar.org/CorpusID:3960646.

[32] Zhang, K., Zhang, Z., Li, Z. a Qiao, Y. Joint Face Detection and Alignment Using Multitask Cascaded Convolutional Networks. *IEEE Signal Processing Letters*. 2016, sv. 23, č. 10, s. 1499–1503. DOI: 10.1109/LSP.2016.2603342.

[33] Zhang, Z., Huang, H. a Wang, K. Using Deep Time Delay Neural Network for Slot Filling in Spoken Language Understanding. *Symmetry*. Červen 2020, sv. 12, s. 993. DOI: 10.3390/sym12060993.