SUR – dokumentace k projektu

Dvořák Martin (xdvora2l)
Halva Vladislav (xhalva04)

2020

1 Úvod

Tato dokumentace popisuje implementaci a vyhodnocení systému pro verifikace osoby z obrázku a zvukové nahrávky v rámci projektu předmětu SUR. Systémy jsou implementovány v jazyce *Python3*. Pro trénování byla použita *pouze data dodaná v rámci zadání projektu*. Pro zvětšení toho datasetu byly použity metody pro rozšíření datových sad (data augmentation) pro obrázky i audio nahrávky. Pro verifikaci na základě obrázku byla použita konvoluční neuronová síť vytvořená pomocí knihovny PyTorch, viz sekce 2. Pro verifikaci podle zvukové nahrávky byl použit GMM klasifikátor pracující s MFCC koeficienty extrahovanými z nahrávek, viz sekce 3. Výsledky těchto dvou modelů byly dále kombinovány a vytvořen výstup zohledňující výsledky obou dvou modelů, viz sekce 4.

2 Zpracování obrazu

Kapitola popisuje použité a vyzkoušení přístupy k řešení problému verifikace osoby z obrazu. Vybraný a použitý přistup představuje malá konvoluční neuronová síť struktury *LeNet*. Implementována byla pomocí knihovny *PyTorch*.

2.1 Rozšíření obrázkové datové sady

Z důvodu výběru konvoluční neuronové sítě je počet přiložených dat k trénování a vyhodnocení sítě nedostatečný a nevyrovnaný. Zadání udává stejnou apriorní pravděpodobnost obou tříd, proto je vhodné mít řádově stejný počet dat pro jednotlivou třídu. Datový set byl rozšířen pomocí níže popsaných transformací. Transformace byly rozděleny do dvou tříd: *změna barev a přidání šumu*, *posunutí a otočení obrazu*. Na jeden obraz se aplikovaly všechny permutační možnosti, tak že transformace byly vybírány z jiných tříd. Výsledné data pro jednotlivé třídy byly zamíchány a náhodně vybrány počty, tak aby řádové obsahovaly stejný počet hodnot. Použité transformace:

- Zrcadlení obrazu přes souřadnou osu Y.
- Rotace obrazu o náhodný úhel z rozmezí $< 8; 20 >$ nebo $< -20; -8 >$ stupňů.
- Přidání šumu do obrazu, které se řídí Gaussovským rozdělením.
- Přidáním šumu do obrazu, které se řídí uniformním rozdělením.
- Vyhlazení obrazu filtrem 5×5.
- Změna kontrastu na náhodnou hodnotu z intervalu $< 0,75; 3.0 >$.
- Změna jasu na náhodnou hodnotu z intervalu $< 25; 75 >$.

Definované transformace byly implementovány pomocí knihoven *cv2* a *skimage*.

2.2 Extrakce příznaků z konvoluční neuronové sítě.

První použitý způsob zpracování obrazu byl vytvoření plně konvoluční sítě, která obsahovala tři vrstvy a výstupní počet hodnot byl roven 128. Za konvoluční vrstvou v čase učení se nacházely dvě plně propojené vrstvy, které zajišťovaly korektní propagaci chyby a učení sítě. Bylo naučeno 20 sítí tak, tak aby dosahovali na testovací sadě přesnost minimálně 95 %. Na získaných příznacích bylo provedena analýza *PCA* a *LDA*. Tyto analýzy nedosahovali kvalitních výsledků, proto byl tento postup ukončen.

Druhý použitý postup počítal nad každým pixelem obrazu diskrétní cosínovou transformaci. Následně byly vytaženy tyto hodnoty filtrem o rozměrech 8×8 s krokem o velikosti 4. Tímto bylo vygenerováno pole příznaků o rozměrech 361×64 z jednoho obrázku. Nad příznaky byla provedena taktéž *PCA* a *LDA* analýza. Dosažené výsledky obsahovaly nízkou přesnost na testovací sadě.

2.3 Konvoluční neuronová síť

Problém verifikace byl převeden na problém klasifikace a následně použití metody *finetuning* naučena plně propojená část sítě na problém verifikace.

2.3.1 Struktura sítě

Výsledná použitá sít je tvořena strukturou *LeNet* se čtyřmi konvolučními vrstvami, které prokládají vrstvy *max pooling* s velikostí jádra 2×2. Následují tři plně propojené vrstvy o rozměrech 512, 64 a 20 neuronů pro klasifikaci a 256, 32, 2 pro verifikaci. Mezi konvolučními i plně propojenými vrstvami se nachází aktivační vrstva tvořená funkcí *ReLU*. Pro použití neuronové sítě byla zvětšena datová sada na tisíce vzorků pomocí dříve popsaných metod *augmentace*.

2.3.2 Trénovací algoritmus

Složky **dev* a **train* byly spojeny do jedné, nad těmito daty byla provedena augmentace a následně byly rozděleny v poměru 9:1 na trénovací a testovací sadu. *Loss* funkce pro úlohu verifikace byla *křížové entropie*, pro úlohu klasifikace *křížová entropie pro více tříd*. Propagaci chyby realizuje učící algoritmus *Adam* s nastaveným učícím koeficientem na hodnotu $0,001$. Velikost dávky byla nastavena na 16 obrázků. Učení bylo provedeno v prostředí *Google Colabory* s grafickým akcelerátorem. Doba trénování sítě byla kolem jedné minuty. Počet epoch, které bylo potřeba k na trénování sítě byl z intervalu $< 15; 25 >$.

2.4 Experimenty a vyhodnocení

Následující tabulka 1 popisuje vyhodnocení jednotlivých sítí, počet epoch potřebných na trénování, dosaženou přesnost na úloze klasifikace a verifikace následně počet falešných přijmutí i odmítnutí. Jednotlivé sítě jsou zapsány zkráceně například: Con 16/32/64/128 FC 512/64/20 popisuje jít složenou ze čtyř konvuločních vrstev, které mají počet kanálů: 16, 32, 64, 128 a v plně propojených vrstvách 512, 64, 20 neuronů. Trénovací parametry jsou nastaveny pro všechny sítě stejně viz předchozí odstavec, pouze počet epoch se bude měnit. Aktivační funkce je použita pouze funkce *ReLU*.

Síť	Epochy	Přesnost (%)	False alarm (%)	False rejection (%)
Con 16/32/64/128 FC 512/64/20	20	99,7	0,28	0,0
Con 16/32/64/128 FC 256/32/2	20	99,8	0,17	0,0
Con 16/32/64 FC 256/64/20	20	94,5	2,4	5,6
Con 16/32/64/128 FC 1024/256/64/20	20	98,1	1,8	2,4
Con 16/32/64 FC 1024/256/64/20	20	99,0	1,1	0,0

Table 1: Výsledky vybraných nastavení CNN sítí

3 Zpracování řeči

3.1 Rozšíření sady trénovacích zvukových nahrávek

Vzhledem k tomu, že trénovací sada obsahuje několikanásobně méně nahrávek hledané osoby, než osob ostatních byla na tyto nahrávky aplikována sada transformací, díky níž bylo docíleno vyváženého počtu trénovacích dat obou tříd pro verifikaci, tedy cílových "target" a necílových "non-target". Pro vytvoření modifikované nahrávky byly vždy aplikovány dvě náhodně zvolené transformace z následujících:

- Přidání Gaussovského šumu o střední hodnotě $\mu = 0$ a rozptylu $\sigma = 1$
- Posunutí zvukového signálu (time shift)
- Roztažení časové osy (time stretch) - zpomalení signálu
- Žádná transformace

3.2 Extrakce MFCC koeficientů

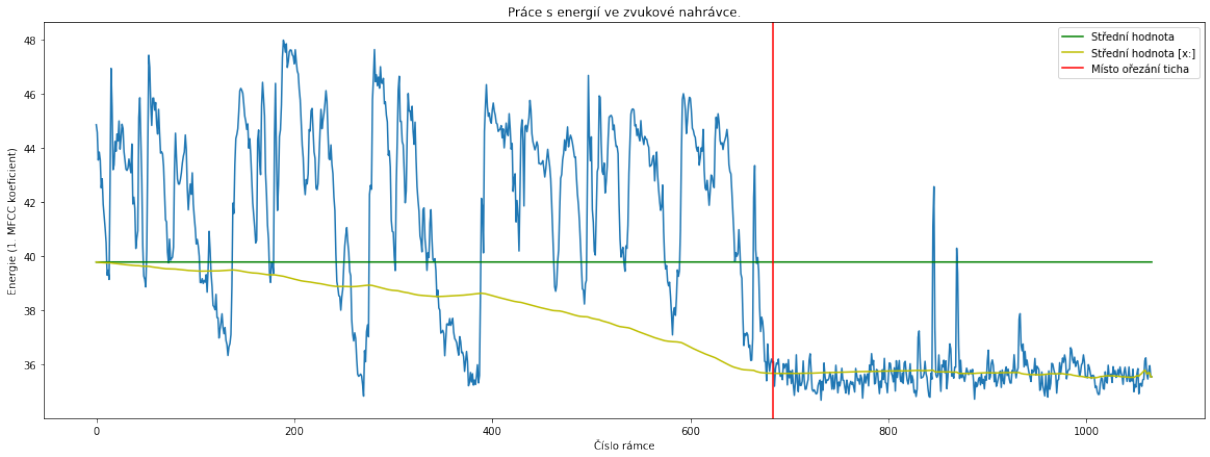
Z případně rozšířené datové sady byly následně extrahovány MFCC koeficienty, s následujícími parametry: *velikost okna = 400 vzorků, překryv rámců = 250 vzorků, počet frekvencí pro výpočet diskrétní Furierovy transformace = 512, vzorkovací frekvence vstupu = 16000 Hz, počet "Mel filter bank" = 23, počet kepstrálních koeficientů = 13.*

3.2.1 Výběr rámců dle energie

Z koeficientů extrahovaných pro každou nahrávku je následně vždy odstraněno prvních 190 rámců, které neobsahují řeč (ticho).

Ve zbylých rámcích je určena střední hodnota energie (prvního MFCC koeficientu), které je použita jako práh pro ořezání ticha na konci nahrávky. Rámce jsou procházeny od posledního a za ticho je považováno vše do pátého rámce přesahujícího energií střední hodnotu (empiricky určeno). Nahrávka je odříznuta 18 rámců za takto zvolenou hranicí (opět empiricky určeno) viz graf 1.

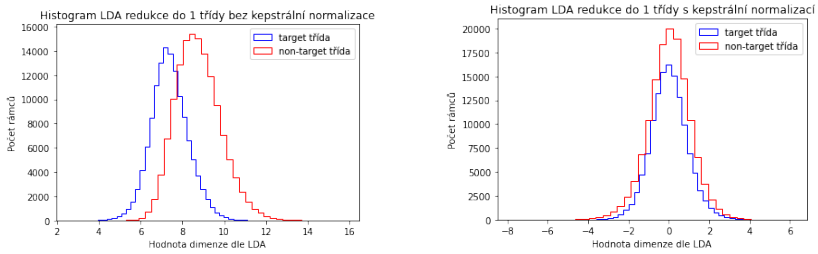
Pro ořezání ticha bylo dále experimentování s různými prahy odvozenými od maximální hodnoty energie v signálu, fixní hodnoty nebo střední hodnoty počítané pro každý rámec vždy pouze z po něm následujících rámců (viz graf 1), ale práh odvozený od střední hodnoty pracoval nejlépe.



Graf 1: Výběr rámců zvukové nahrávky na základě energie.

3.2.2 Kepstrální normalizace

Bylo experimentováno také s kepstrální normalizací, tedy odečtením střední hodnoty MFCC koeficientů napříč rámci od jejich hodnoty. Tato úprava se ukázala jako neefektivní a naopak snížila rozlišitelnost mezi třídami a nebyla tedy použita viz grafy 2.



Graf 2: Změna rozlišitelnosti cílové a necílové třídy použitím kepstrální normalizace po aplikaci LDA.

3.3 GMM klasifikátor

Pro verifikaci cílového řečníka byl použit GMM klasifikátor se dvěma třídami - první reprezentující hledaného řečníka, druhá ostatní řečníky.

3.3.1 Experimenty a vyhodnocování

V rámci experimentů s GMM klasifikátorem byly vždy jako trénovací data požity nahrávky v původních adresářích `*_train` a pro testování data v adresářích `*_dev`. S každým nastavením klasifikátoru a také variantou přípravy vstupních dat bylo provedeno několik trénování a vyhodnocení úspěšnosti na testovacích datech. Z těchto běhů byla následně určena průměrná úspěšnost varianty.

V prvních fázích byl použit GMM klasifikátor se 40 komponentami pro každou třídu a diagonální kovariační maticí, tento model byl vždy trénován ve 30 iteracích. Na tomto modelu byly dále sledovány výsledky pro různé transformace vstupních dat jako ořezání tichých míst různými metodami, kepstrální normalizace a transformace vstupních dat do jedné dimenze pomocí LDA. Nejlepší výsledky byly dosaženy s ořezáním dat podle střední hodnoty energie viz sekce 3.2.1, bez kepstrální normalizace a bez použití LDA viz 2.

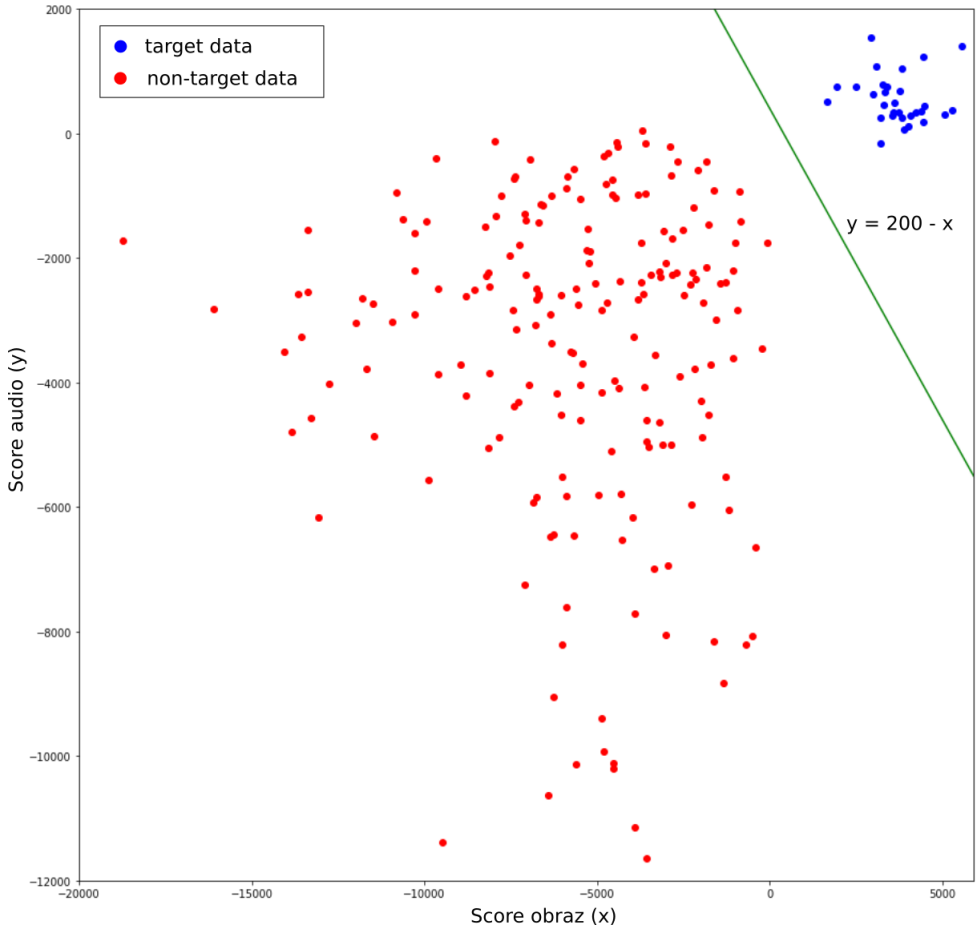
Následně byla místo diagonální použita plná kovariační matice a testovány různé počty komponent pro jednotlivé třídy a počty trénovacích iterací. Nejlepších výsledků bylo dosaženo při použití 64 komponent pro každou třídu, pouze 10 trénovacích iterací plná kovariační matice a ořezání ticha ve vstupních datech. S těmito parametry bylo na testovacích datech dosaženo úspěšnosti 97.2%. Horší výsledky při vyšším počtu trénovacích iterací jsou nejspíše způsobeny přetrénováním a tak horším zobecnění.

Komp	Iterace	Kov.	Úprava dat	Přesnost (%)	False alarm (%)	False rejection (%)
40	30	diagonální	žádná	90	0	70
40	30	diagonální	k. norm.	85	0	100
40	30	diagonální	ořezání ticha	87	0	90
40	30	plná	ořezání ticha	94	0	7
64	20	plná	ořezání ticha	95.7	0	6
64	10	plná	ořezání ticha	97.2	1.6	4
128	20	plná	ořezání ticha	93.7	0	10

Table 2: Výsledky vybraných nastavení GMM klasifikátoru a předzpracování dat (počet komponent pro každou třídu, počet trénovacích iterací, úprava dat (k. norm. = kepstrální normalizace), false alarm, false rejection).

4 Kombinovaný klasifikátor

V případě kombinované klasifikace, resp. verifikace jsou použity výsledky obou výše popsaných modelů pro verifikaci na základě obrazu a zvuku. Score navrácené těmito modely je zkombinováno (chápáno jako dvou-dimenzionální parametr) a následně je učiněno rozhodnutí na základě této kombinace. Lineární rozhodovací hranice byla empiricky stanovena na přímku $score\_audio = 200 - score\_obraz$ viz graf 3. Pokud je tedy $score\_audio + score\_obraz > 200$ je verifikace úspěšná (1). V opačném případě se rozhodne na základě hodnoty $score\_audio$, která vykazuje obecně lepší rozlišovací schopnost. Hranice byla stanovena na 0. Verifikace je tedy dále úspěšná, pokud $score\_audio > 0$, v opačném případě neúspěšná (0). Takto nastavený systém má na trénovací i testovací sadě 100% úspěšnost. Dle grafu 3 by takto zvolená rozhodovací hranice neměla zapříčinit nízkou schopnost zobecňovat vzhledem k poměrně zřejmé diverzitě tříd v prostoru score jednotlivých modelů.



Graf 3: Rozhodovací hranice u kombinovaného modelu pro verifikaci.

5 Spuštění systému

Pro spuštění systému jsou nutné knihovny: *numpy*,*PyTorch*, *cv2*, *skimage*, *librosa*, *scipy*, *pickle*, *tqdm*.

Systém byl spouštěn v prostředí *Google Colaboratory*, konvoluční neuronová síť byla pomocí *GPU*.

Vstupní bod *evaluace* klasifikátorů se nachází v souboru *personVerification.py*, ve kterém se nachází implementace CNN a GMM klasifikátoru. Dále tento soubor vytváří rozhraní pro sběr výsledků od těchto klasifikátorů a zapisuje je do souborů s výsledky. Spuštěním tohoto souboru se provede v metodě main vytvoření modelů, načtení natrénovaných parametrů a následně vyhodnocení evaluační sady. Vstupní

funkce main lze spustit s právě jedním parametrem a to je relativní cesta k adresáři s evaluačními daty. Proces verifikace evaluačních dat je zobrazován *progress barem*.

Soubor *picture_CNN.txt* obsahuje výsledky na *evaluační sadě* pro klasifikátor používající konvoluční neuronovou síť. Soubor *audio_GMM.txt* obsahuje výsledky pro audio GMM klasifikátor.
A soubor *combined_Picture_CNN_Audio_GMM.txt* představuje kombinované výsledky, které považujeme za preferované ve vyhodnocení jejich správnosti.

Natrénované modely jsou obsaženy ve složce *models*. Při ohodnocení evaluačních dat jsou tyto modely načteny, aby nebylo nutné provádět fázi učení.

6 Rozšíření práce

Systém pro zpracování zvuku by mohl být implementovaný pomocí rekurentní nebo konvoluční neuronové sítě. Pro systém zpracování obrazu by mohl být použit před-trénovaný model pro extrakci příznaků, a nad těmito příznaky by mohla být provedena *PCA*, či *LDA* analýza a postavena neuronová síť s plně propojenými vrstvami. Rozhodovací hranice kombinovaného systému by mohla být implementována pomocí lineárního či komplikovanějšího klasifikátoru.