Vysoké učení technické v Brně
Fakulta informačních technologií

SUR - Strojové učení a rozpoznávání
1. Projekt – IKR - evaluace

Michal Kabáč (xkabac00)
Daniel Kavuliak (xkavul01)
30. apríla 2022

1 Oboznámenie sa s dátovou sadou

Na začiatku sme preskúmali dátovú sadu a zistili sme, že obsahuje 4 základné priečinky - target_train, target_dev, non_target_train, non_target_dev. Priečinky obsahovali nahrávky vo formáte wav a fotky tvárí. Na začiatok sme začali s predspracovaním nahrávok a fotiek.

1.1 Predspracovanie nahrávok

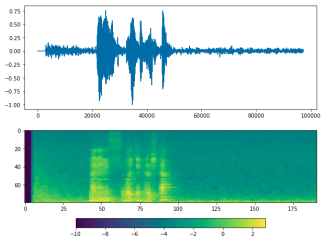
Na vstupe sme si určili absolútnu cestu ku nahrávkam. Následne sme prechádzali všetky súbory v tomto priečinku a ak končili príponou .wav pomocou knižnice *malaya_speech* sme ich načítali. Následne sme si pre nahrávky vykreslili spektrogram, ktorý môžeme vidieť na obrázku 1. Z obrázku sme videli že v nahrávkach sa nachádza ticho na začiatku a na konci nahrávky. Preto sme z knižnice *librosa* použili funkciu *effects.trim*, ktorej sme ako parameter dali signál a hlasitosť v decibeloch. Ak sa na začiatku alebo konci nahrávky vyskytuje signál s hodnotou nižšou, ako je nami zadaná hodnota v decibeloch, tak sa z nahrávky oreže. Po použití tejto funkcie sme si znova vykreslili spektrogram. Ukážka spektrogramu po orezaní ticha zo začiatku a konca nahrávky je na obrázku 2. Ďalej sme chceli ticho odstrániť aj zo stredu nahrávky pomocou knižnice librosa. Ak sa v strede nahrávky vyskytovalo ticho, nahrávku sme na tomto tichu rozdelili a vytvorili sme zoznam položiek typu *AudioSegment*. Následne sme tento zoznam položiek spojili, čím sme dostali výslednú nahrávku, v ktorej boli orezané tiché miesta. Na takto spracovanú nahrávku sme následne zavolali funkciu *feature.mfcc* z knižnice librosa. Ako parameter n_mfcc ktorý značí koľko mfcc príznakov nám funkcia vráti sme zadali číslo 13.

1.2 Rozšírenie dátovej sady pre rečové nahrávky

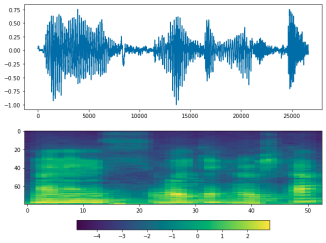
Keďže trénovacia aj validačná sada obsahovali malé množstvo rečových nahrávok, rozhodli sme sa počet nahrávok rozšíriť pomocou data augumentation. Na rozšírenie počtu nahrávok nám poslúžila knižnica *audiomentations*. Knižnica audiomentations obsahuje rôzne funkcie ako napríklad:

- AddGaussianNoise - funkcia pridá do nahrávky gaussovský šum (obrázok 3)
- Gain - zvýši sa alebo zníži hlasitosť nahrávky pomocou vynásobenia náhodným faktorom amplitúdy.
- Time stretch - roztiahnutie signálu bez zmeny výšky tónu (obrázok 4)
- RoomSimulator - simuluje rozprávanie človeka v uzavretej miestnosti (obrázok 5)
- BandStopFilter - zamaskuje nejaké frekvenčné pásmo na spectograme (obrázok 6)

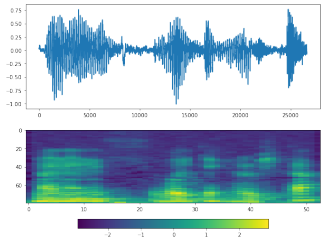
a mnoho ďalších. Pre tieto funkcie sa môžu nastavovať rôzne parametre. Taktiež je možné nastaviť pravdepodobnosť s akou má byť táto funkcia vykonaná a funkcie je možné kombinovať.



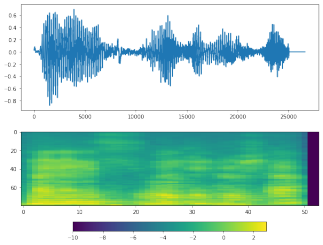
Obr. 1: Vykreslenie spectogramu pre záznam z trénovacej sady.



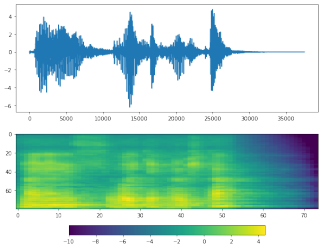
Obr. 2: Vykreslenie spectogramu po orezaní ticha zo začiatku a konca nahrávky.



Obr. 3: Pridanie gausovského šumu do nahrávky.



Obr. 4: Roztiahnutie nahrávky pomocou time stretch.



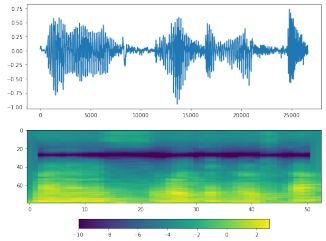
Obr. 5: Použitie funkcie room simulátor na nahrávku.

2 Trénovanie detektorov pre reč

Pre detekciu osoby z reči sme použili nasledujúce 2 prístupy: GMM a SVM.

GMM - boli natrénované 2 GMM. Jeden model bol natrénovaný pre osobu ktorú chceme detekovať, druhý model bol natrénovaný na osobu, ktorú detekovať nechceme. Pri tomto trénovaní vstup modelu tvorili MFCC príznaky. GMM sme použili z knižnice *sklearn*. Rozhodnutie či sa jedná alebo nejedná o detekovanú osobu sme rozhodoval na základe pomeru priemernej logaritmickej pravdepodobnosti z každého GMM modelu pre danú nahrávku. Ak bola hodnota nižšia ako nami zvolená hranica rozhodli sme že ide o hľadanú osobu. Ak bola hodnota vyššia rozhodli sme že to hľadaná osoba nebude.

Ostatné modely. Skúsili sme experimentovať aj s ďalšími modelmi, avšak tieto modely potrebovali ako vstup vektor príznakov. Spočítali sme priemer a štandardnú odchýlku zo získaných MFCC príznakov pre každú nahrávku. Taktiež sme spočítali hodnoty skewness, kurtosis, varianciu. Hodnota skewness určuje symetrickosť rozloženia a hodnota kurtosis je parameter špicatosti kombinovaných váh chvostov vo vzťahu k zvyšku distribúcie. Po testovaní na všetkých kombináciach týchto príznakov sa ako najlepšie ukázali priemer a štandartná odchylka, preto sme vyhodnocovanie robili na týchto 2 príznakoch. Ak sa jednalo o hľadanú osobu ako label sme nastavili hodnotu 1, inak bola hodnota nastavená na 0. Na takto upravených nahrávkach sme trénovali Support vector machine, Random forrest, K najbližších susedov, Decision tree, XGBoost. Pri všetkých modeloch sme skúšali optimalizáciu hyperparametrov pomocou Random Search. Jednotlivé nastavenia random searchu je možné vidieť v zdrojovom kóde. Taktiež sme sa pomocou ensemble stackingu pokúsili natrénovať model, ktorý sa



Obr. 6: Maskovanie frekvencie pomocou band stop filtra.

rozhoduje na základe predikcií ostatných modelov. Ako výsledný model sme určili logistickú regresiu, ktorá bola natrénovaná na predikciách optimalizovaných modelov SVM, KNN, Random forest, Decision Tree a XGBoost.

3 Výsledky pre reč

Pri GMM sme výslednú hodnotu pre porovnávanie s hranicou vypočítali ako pomer logaritmickej pravdepodobnosti $\frac{GMM(detektor)}{GMM(non\_detektor)}$. Kde GMM(detektor) je GMM natrénované na osobe ktorá je hľadaná a GMM(non_detektor) je natrénované na všetkých ostatných osobách. Následne sme vyhľadali maximálnu hodnotu pomeru logaritmickej pravdepodobnosti na validačných dátach pre hľadanú osobu a minimálnu hodnotu pre osoby ktoré hľadané nie sú. Pri tomto pomere sme pozorovali, ako vzdialené sú tieto hodnoty od hodnoty 1 (Hodnota 1 bola určená ako rozhodovací prah).

Na začiatku sme experimentovali čisto iba s MFCC príznakmi bez orezania ticha a data augumentation. Výsledky týchto experimentov môžeme vidieť na obrázku 7. Kde prvý stĺpec reprezentuje počet gausovských komponent v rámci jedného GMM. Počty komponent boli rovnaké pre oba GMM. Môžeme vidieť že v priemere mali najväčší rozdiel pomerov GMM pri 15 komponentách.

Počet komponent	Pomer dáta detect	Pomer dáta non detect	Priemerný počet chýb
5	0,993	1,015	0
10	0,991	1,020	0
15	0,991	1,022	0
20	0,998	1,023	0,4

Obr. 7: Obrázok zachytávajúci priemerné pomery z 5 behov GMM pre validačné dáta hľadanej osoby a dáta ostatných osôb iba na MFCC príznakoch bez ďalšieho predspracovania.Priemerný počet chýb značí, koľko krát v priemere bola chybná detekcia pre hranicu 1.

Následne sme experimentovali aký efekt bude mať orezanie ticha v nahrávke. Testovali sme s rovnakým počtom gaussovských komponent ako v prípade vyššie. Výsledky môžeme vidieť na obrázku 8. Vidíme, že orezanie ticha v nahrávke malo značný vplyv na maximalizáciu rozdielu rozhodovacích hraníc. Taktiež vidíme že maximálna hodnota pomeru pre hľadanú osobu je pri 15 komponentách 0,973 a minimálna hodnota pre nehľadanú osobu 1,021. Tieto pomery sú zpriemerované z 5 behov minimálnych a maximálnych hodnôt.

Počet komponent	Pomer dáta detect	Pomer dáta non detect	Priemerný počet chýb
5	0,975	1,001	0
10	0,972	1,013	0
15	0,973	1,021	0
20	0,986	1,021	0

Obr. 8: Obrázok zachytávajúci pomery logaritmickej pravdepodobnosti 2 GMM modelov pre validačné dáta hľadanej osoby a dáta ostatných osôb pri odstránení ticha z nahrávok.

Ďalšie experimenty boli spojené s rozšírením dátovej sady. Výsledky pre jednotlivé modely môžeme vidieť v tabuľke 9. Výsledky sme dosiahli s krížovou validáciou na hodnote 10. Pri krížovej validácii sa nebrali do úvahy jednotlivé sedenia. Ako môžeme vidieť po augumentácií sa niektorým modelom znížil počet chýb na validačnej sade. Pri modeli SVM však došlo k zhoršeniu výsledkov a väčšiemu počtu chýb. Do budúcna by bolo vhodné pozrieť sa na jednotlivé funkcie z knižnice audiomentations pre rozšírenie dátovej sady na reč a skúsiť vyhodnotiť každú funkciu na jednotlivých modeloch, ako aj ich kombinácie. Na základe týchto vyhodnotení by sme potom mohli určiť, ako jednotlivé metódy prispievajú k zlepšeniu modelu. Na tento experiment sme však už nemali čas.

Názov modelu	Hodnota Accuracy najlepšieho modelu pri Random search			Počet chýb na validačnej sade		
	Neaugumentovaná sada	1x rozšírená sada	2x rozšírená sada	Neaugumentovaná sada	1x rozšírená sada	2x rozšírená sada
SVM	93,57	91,23	90,13	0	1	1
KNN	92,14	89,9	90,1	3	3	2
Random forrest	94,28	90,31	90,12	3	1	2
Decision Tree	85,26	0,82	91,12	11	10	5
XGBoost	92,21	90,3	91,13	6	4	2
Ensamble model	0,91	0,90	0,89	0	0	1

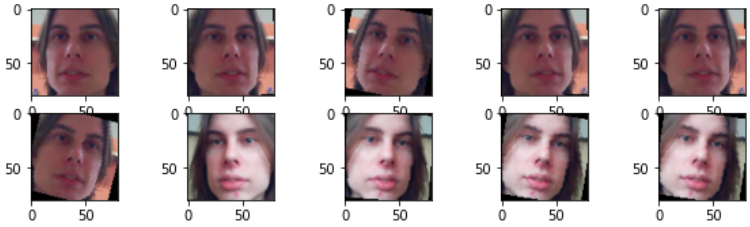
Obr. 9: Obrázok zachytávajúci metriky accuracy najlepších modelov z Random search a počet chýb týchto modelov na evaluačných sadách.

Zhrnutie. Ako najlepšie detektory sa ukázali SVM a rozhodovanie na základe pomeru likelihood pre GMM natrénovaný na hľadanej osobe a GMM natrénovaný na ostatných osobách s rozhodovacou hranicou. Preto sme tieto 2 prístupy vybrali na evaluáciu výsledných testovacích dát. SVM sme natrénovali na spojenej evaluačnej a trénovacej sade pomocou krížovej validácie na hodnote 10. Pre rozhodovanie na základe pomerov logaritmickej pravdepodobnosti GMM modelov s rozhodovacou hranicou sme sa rozhodli otestovať 2 spôsoby. Prvý spôsob bol natrénovaný na augmentovaných dátach z trénovacej sady. Následne bol vyhodnotený na dátach evaluačnej sady. Z tejto evaluačnej sady sme vybrali maximálnu hodnotu pomeru pre hľadanú osobu, minimálnu hodnotu pomeru pre ostatné osoby a vypočítali sme ich priemer. Tento priemer tvoril rozhodovaciu hranicu. Druhý prístup bol založený na natrénovaní spojených validačných a evaluačných dát (jedno GMM pre hľadanú osobu a 1 GMM pre ostatné osoby). Ako hranica bola pevne zvolená hodnota 1.

4 Načítanie a predspracovanie obrázkov tvárí

Obrázky sme načítavali v RGB forme pre model SVM alebo grayscale forme pre model RandomForest. Obrázky tvárí v trénovacej dátovej sade sme rozšírili pomocou dátovej augmentácie z pôvodných obrázkov. Na každý obrázok sme vytvorili 5 nových obrázkov, takže počet obrázkov v trenovacej sade sa zvýšil o z 140 na 840 obrázkov. Obrázky sme generovali pomocou následujúcich metód:

- horizontálne preklopenie - prevrátenie obrázka okolo zvislej strednej čiary obrázka
- rotovanie - náhodná rotácia obrázka v rozmedzí -20 až 20 stupňov



Obr. 10: Vzorka generovanej sady

Následne sme tieto obrázky previedli z formátu PIL Image do tenzoru. V následujúcich podsekciách si popíšeme ďalšie predspracovania pre spomenuté modely.

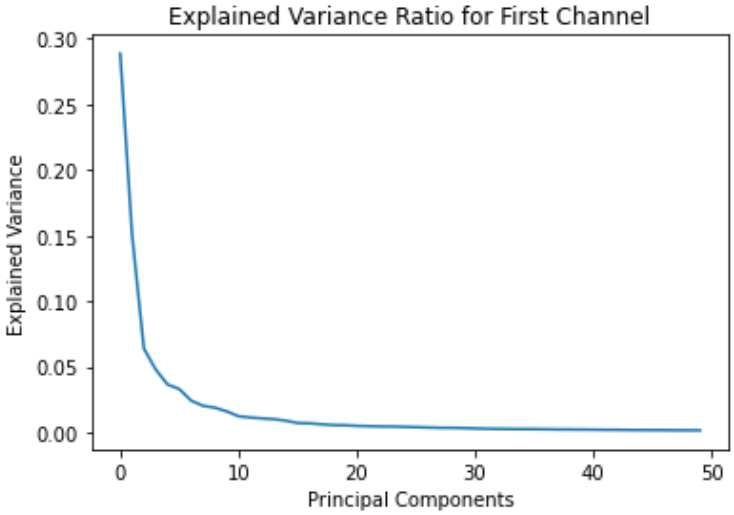
Predspracovanie pre SVM

Pre model SVM sme sa rozhodli vykonať PCA analýzu pre každý kanál zvlášť, chceme extrahovať príznaky, ktoré dokopy budú tvoriť aspoň 80 percent variancie daného kanálu. Takýmto spôsobom sa zbavíme zbytočných príznakov v jednotlivých kanáloch, ktoré majú príliš malú varianciu, takže neobsahujú veľa informácii. V obrázkoch 11, 12 a 13 môžeme vidieť krivku pomeru variancií príznakov každého kanála.

Ako vidíme na obrázkoch 11, 12 a 13, tak prvých desať príznakov májú veľký rozptyl dát, ostatné príznaky majú malú varianciu, čiže neobsahujú moc informácii o dátach. Aj keď prvých desať príznakov obsahujú významnejšie informácie o rozptyle dát v nich, tak si ponecháme prvých 50 príznakov, aby model SVM vedel pomocou rozhodovacej funkcie lepšie rozdeliť rozloženie jednej triedy od druhej.

Pre tento model sme sa taktiež rozhodli vykonať PCA analýzu nad samotnými obrázkami, nie samostatne nad kanálmi. Pomer variancií príznakov v obrázkoch je rovnaká ako pri kanáloch.

Rozdiel v týchto dvoch prístupoch je v počte príznakov, ktoré nám zostanú po redukcií. V prvom prístupe sa snažíme zachovať príznaky každého kanála, výsledný počet príznakov je 150. Takto si zachováme dôležité príznaky jednotlivých kanálov, ktoré môžu lepšie prispieť v rozdelení oboch rozložení dát daných tried. V druhom prístupe budeme mať iba 50 príznakov, takto sa môžeme zbaviť kanálov, ktoré neprispievajú v lepšom rozdelení oboch spomínaných rozložení.



Obr. 11: Pomer variancií príznakov v prvom kanáli

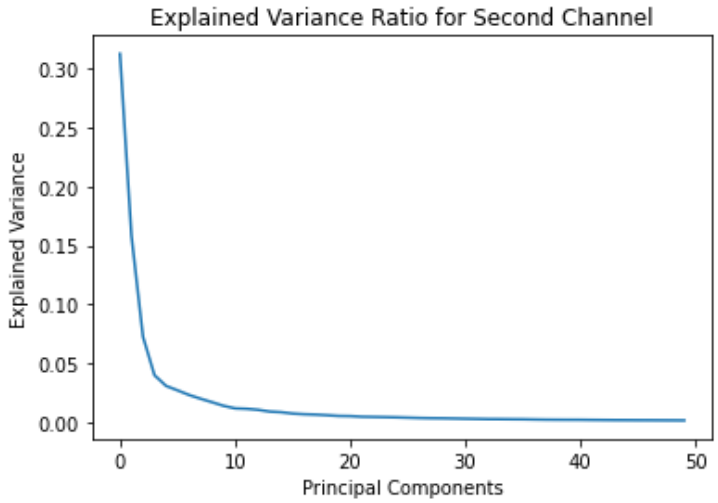
Predspracovanie pre RandomForest

Pre model RandomForest sme generovali obrázky rovnakým spôsobom ako pri modeli SVM, avšak rozdiel je v tom, že obrázky načítavame v grayscale móde, nie v RGB. Toto predspracovanie vykonáme následujúcim spôsobom: najprv od všetkých hodnôt trénovacej sady odčítame stredné hodnoty príznakov a následne od tohto výsledku odčítame stredné hodnoty obrázkov. Prvý krok robíme kvôli tomu, aby všetky obrázky mali priemernú úroveň sivej farby. Druhý krok vykonávame, aby boli dátové objekty vycentrované okolo počiatku súradnicovej sústavy. Rozdiel medzi obrázkami uvidíme na obrázku 14.

5 Trénovanie a výsledky detektorov pre tváre

Modely pre detekovanie tvárí sme trénovali následujúcim spôsobom: vykonali sme krížovú validáciu nad každým modelom, následne sme vybrali model, ktorý mal najväčšiu úspešnosť počas krížovej validácie a nakoniec sme tento model otestovali na testovacej sade. Krížovú validáciu vykonávame z dôvodu malej dátovej sady.

Model SVM, do ktorého išli obrázky predspracované prvým spôsobom spomenutým v predchádzajúcej sekcií, mal následujúce úspešnosti v krížovej validácii (úspešnosti v celom projekte sme merali metrikou správnosti): 0.76785714, 0.9047619 , 0.75595238, 0.85714286, 0.73809524. Podľa týchto úspešností sme vybrali model z druhej iterácie krížovej validácie a následne sme ho otestovali na testovacej sade. Na tejto sade dosiahol úspešnosť 0.8285714285714286. Model SVM, do ktorého išli obrázky predspracované druhým spôsobom, mal v krížovej validácii tieto úspešnosti: 0.74404762, 0.9047619, 0.79761905, 0.81547619, 0.7202381. Na základe týchto úspešností sme vybrali a otestovali model z druhej iterácie, ktorý má úspešnosť na testovacej sade 0.8428571428571429. Na základe rozdielu úspešností medzi týmito dvoma modelmi môžeme usúdiť, že pravdepo-



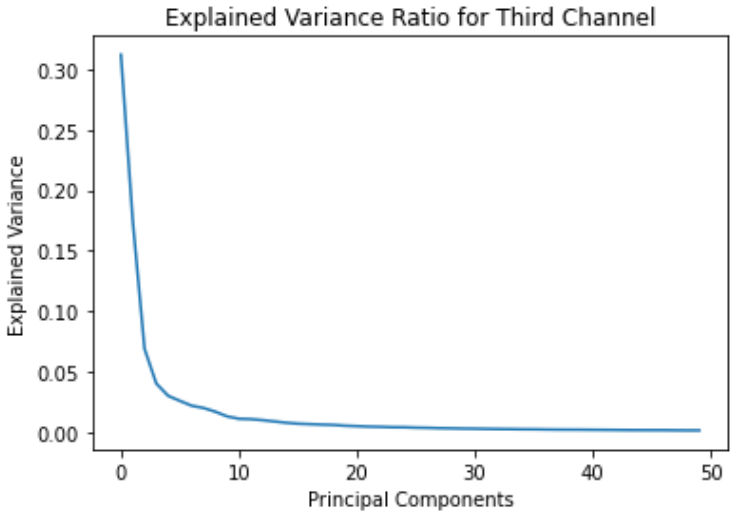
Obr. 12: Pomer variancií príznakov v druhom kanáli

dobne nezáleží na príznakoch všetkých kanálov, niektorý z tých kanálov nemusí byť užitočný ako ostatné kanály ("pravdepodobne"preto, lebo rozdiel úspešností je malý).

Po predspracovaní obrázkov pre model RandomForest sme vykonali krížovú validáciu tohto modelu a má následujúce úspešnosti: 0.96428571, 0.9702381, 0.9702381, 0.97619048, 0.94642857. Vybrali sme model zo štvrtej iterácie a ten mal na testovacej sade úspešnosť 0.8142857142857143.

Zhodnotenie

Na základe týchto troch úspešností môžeme usúdiť, že lepší model je SVM, kde vykonávame PCA analýzu na celom obrázku. Tento model môže byť lepší z toho dôvodu, že môžeme pomocou spomenutého prístupu vyhodiť z nejakého kanála vyhodiť viac príznakov ako v ostatných kanáloch. Model RandomForest mal najhoršiu úspešnosť. V modeli RandomForest najprv prebehne klasifikácia dátového objektu v každom rozhodovacom strome a následne prebehne hlasovanie, ktoré rozhodne o finálnej klasifikácii dátového objektu. Aj napriek tejto skutočnosti, dosahuje najhoršie výsledky zo všetkých troch modelov. Ako vidíme na výsledkoch všetkých krížových validácii, tak najlepšie výsledky z krížových validácii sú vždy vyššie ako výsledky na testovacej sade. Môže to byť nadhodnotené z dôvodu nerešpektovania nahrávacích sedení, obrázky z jedného sedenia sú skoro rovnaké a môžu sa nachádzať aj v trénovacej, aj validačnej sade. V budúcnosti by sme vykonali vlastnú krížovú validáciu, ktorá rešpektuje tieto sedenia a vykonali by sme aj optimalizáciu hyperparametrov. Pri modeli RandomForest bol volený iba počet stromov a pri modeloch SVM iba maximálny počet iterácii dokedy môže hľadať vhodné riešenie (maximálny počet iterácii bol 10). V budúcnosti by sme vedeli zmeniť počet rozhodovacích stromov pri modeli RandomForest alebo zvyšiť hodnotu regularizačnej konštanty pri modeli SVM (defaultne je hodnota 1). Spravili sme evaluáciu na modeli RandomForest a SVM, ktorého vstupom boli predspracované dáta druhým spôsobom. Na záver



Obr. 13: Pomer variancií príznakov v treťom kanáli

sme sa snažili spojiť model RandomForest pre obrázky s modelom GMM pre nahrávky, takýto spojený model sa rozhoduje na základe skóre, ktoré predstavuje istotu, či ide o hľadanú osobu.

6 Spustenie a reprodukcia získaných výsledkov

Pre spustenie je potrebné mať nainštalované knižnice, ktoré sa importujú na začiatku programu. Všetky riešenia boli naimplementované v jazyku Python.

Na trénovanie GMM pre reč je potrebné spustiť run_gmm.sh. V tomto kóde je potrebné si dodefinovať cesty k jednotlivím priečinkom s dátami. V premennej BASE stačí zadať cestu, kde máme uložené dáta a náš zdrojový kód. Následne zadáme cesty k súborom. Taktiež zadáme či chceme aby sa nám uložili natrénované modely a cestu ich uloženia. Do premennej save-results-dir zadáme kde chceme aby sa uložili výsledky. V prípade že chceme aby dáta boli augumentované, je potrebné v kóde gmm.py nastaviť hodnotu augumentation_data na true. Script spustíme príkazom bash run_gmm.sh.

Rovnako aj SVM pre reč je možné spustiť príkazom bash run_svm.sh. V scripte run_svm.sh sa nachádzajú rovnaké premenné ako v scripte run_gmm.sh.

Spustenie kombinovaného modelu z modelov RandomForest pre obrázky a GMM pre nahrávky, vykonáme pomocou príkazu "bash combine_models.sh". Do skriptu je potrebné zadať cesty k textovým súborom, blžšie inštrukcie sa nachádzajú priamo v scripte.

Ostatné zdrojové kódy sú uložené v jupyter notebookoch z dôvodu vizualizácie postupu a lepšieho prehľadu. Cesty k dátam a cestu k uloženým výsledkom pre jupyter notebooky je potrebné zadať priamo v kóde.

Obr. 14: Ukážka obrázkov pred predspracovaním (horný rad) a po predspracovaní (dolný rad)

7 Návrhy na zlepšenie

Jedným z návrhov na zlepšenie by bola implementácia krížovej validácie pre jednotlivé sedenia. Ďalším návrhom na zlepšenie by bolo vyskúšanie viacerých modelov s rôznymi nastaveniami parametrov. Zlepšenie by tiež mohlo priniesť použitie predtrénovanej neurónovej siete, ktorá však v tomto projekte bola zakázaná. Lepšie výsledky by tiež mohlo priniesť experimentovanie s augumentáciou dát a hľadanie takých parametrov pri rozširovaní, ktoré by na modely najlepšie vnímali. Prípadne výber najlepších metód pre augmentáciu a nepoužívanie napr. všetkých 19 spôsobov pri zvuku.