

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

DIPLOMOVÁ PRÁCE

Brno, 2019

Bc. Pavol Lieskovský



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV TELEKOMUNIKACÍ

DEPARTMENT OF TELECOMMUNICATIONS

PARAMETRIZACE TVÁŘE POMOCÍ VIDEOSEKVENCE

FACE PARAMETERIZATION USING VIDEOSEQUENCE

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. Pavol Lieskovský

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Martin Rajnoha

BRNO 2019

Diplomová práce

magisterský navazující studijní obor **Telekomunikační a informační technika**

Ústav telekomunikací

Student: Bc. Pavol Lieskovský

ID: 173689

Ročník: 2

Akademický rok: 2018/19

NÁZEV TÉMATU:

Parametrizace tváře pomocí videosekvence

POKYNY PRO VYPRACOVÁNÍ:

Seznamte se s metodami pro analýzu obrazů. Vytvořte program na parametrizaci tváře osob. Parametry by měly být především plocha úst, výška a šířka tváře, různé vzdálenosti např. mezi očima, obočím apod. Dále doplňte parametrizaci o rozpoznávání pravděpodobností emocí. Algoritmus rozšiřte o analýzu celé videosekvence mluvící osoby, ze které se vytvoří časová řada s danými parametry pro každý snímek. Synchronizujte videosekvence s korespondujícími zvukovými nahrávkami. Vytvořte algoritmus pro měření změn (rychlost a velikost) parametrů. Proveďte analýzu vytvořených časových řad a vyhledejte nejvýznamnější parametry pro klasifikaci Parkinsonovi nemoci. Dále natrénujte model pro odhad progresu onemocnění.

DOPORUČENÁ LITERATURA:

[1] SCHROFF, Florian, KALANICHENKO, Dmitry a PHILBIN, James. Facenet: A unified embedding for face recognition and clustering. In Proceedings of the IEEE conference on computer vision and pattern recognition, s. 815–823, 2015.

[2] JANKOVIC, Joseph. Parkinson's disease: clinical features and diagnosis. In Journal of neurology, neurosurgery & psychiatry, 79(4), s. 368–376, 2008.

Termín zadání: 1.2.2019

Termín odevzdání: 16.5.2019

Vedoucí práce: Ing. Martin Rajnoha

Konzultant:

prof. Ing. Jiří Mišurec, CSc.
předseda oborové rady

UPOZORNĚNÍ:

Autor diplomové práce nesmí při vytváření diplomové práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

ABSTRAKT

Táto práca sa zaoberá problémom parametrizácie tváre rozprávajúcej osoby na videu a odhadom Parkinsonovej choroby a progresu jej symptómov na základe parametrov tváre. Popisuje syntax a funkciu programu, ktorý bol v rámci tejto práce vytvorený a rieši problém parametrizácie tváre. Program spracované dáta formátuje do časovej rady parametrov vo formáte JSON. Z týchto dát bol vytvorený dataset, na základe ktorého boli v rámci práce natrénované modely umelej inteligencie určené na predikciu Parkinsonovej choroby a jej symptómov u pacienta. Proces tréningu modelov a ich výsledky sú zdokumentované v práci.

KĽÚČOVÉ SLOVÁ

parametrizácia tváre, spracovanie videa, Parkinsonova choroba, klasifikácia, regresia

ABSTRACT

This work deals with the problem of face parameterization from the video of a speaking person and estimating Parkinson's disease and the progress of its symptoms based on face parameters. It describes the syntax and function of the program that was created within this work and solves the problem of face parameterization. The program formats the processed data into a time series of parameters in JSON format. From these data, a dataset was created, based on which artificial intelligence models were trained to predict Parkinson's disease and the progress of its symptoms. The process of model training and their results are documented within this work.

KEYWORDS

face parameterization, video processing, Parkinson's disease, classification, regression

LIESKOVSKÝ, Pavol. *Parametrizace tváře pomocí videosekvence*. Brno, 2019, 83 s. Diplomová práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací. Vedúci práce: Ing. Martin Rajnoha

VYHLÁSENIE

Vyhlasujem, že som svoju diplomovú prácu na tému „Parametrizace tváře pomocí videosekvence“ vypracoval samostatne pod vedením vedúceho diplomovej práce, využitím odbornej literatúry a ďalších informačných zdrojov, ktoré sú všetky citované v práci a uvedené v zozname literatúry na konci práce.

Ako autor uvedenej diplomovej práce ďalej vyhlasujem, že v súvislosti s vytvorením tejto diplomovej práce som neporušil autorské práva tretích osôb, najmä som nezasiahol nedovoleným spôsobom do cudzích autorských práv osobnostných a/alebo majetkových a som si plne vedomý následkov porušenia ustanovenia § 11 a nasledujúcich autorského zákona Českej republiky č. 121/2000 Sb., o práve autorskom, o právach súvisiacich s právom autorským a o zmene niektorých zákonov (autorský zákon), v znení neskorších predpisov, vrátane možných trestnoprávných dôsledkov vyplývajúcich z ustanovenia časti druhej, hlavy VI. diel 4 Trestného zákoníka Českej republiky č. 40/2009 Sb.

Brno

.....

podpis autora

POĎAKOVANIE

Rád by som poďakoval vedúcemu diplomovej práce páňovi Ing. Martinovi Rajnohovi za odborné vedenie, konzultácie, a podnetné návrhy k práci.

Brno

.....

podpis autora

Obsah

Úvod	11
1 Parkinsonova choroba (PCH)	12
1.1 Dopamín	12
1.2 Príznaky Parkinsonovej choroby	12
1.2.1 Motorické príznaky	13
1.2.2 Nemotorické príznaky	16
2 Metódy parametrizácie obrazu	17
2.1 Segmentácia obrazu	17
2.1.1 Predspracovanie obrazu	17
2.1.2 Metódy segmentácie	19
2.2 Metódy na identifikáciu osôb pomocou tváre	26
2.2.1 Metódy založené na vzhľade	27
2.2.2 Metódy založené na črtoch	27
2.2.3 Metódy založené na vzoroch	28
2.2.4 Metódy založené na častiach	28
3 Umelá inteligencia	30
3.1 Typy strojového učenia	30
3.1.1 Učenie bez učiteľa	30
3.1.2 Učenie s učiteľom	30
3.2 Klasifikácia a Regresia	33
3.2.1 Klasifikácia	33
3.2.2 Regresia	33
4 Parametrizácia tváre osoby na videu	35
4.1 Riešenie s využitím segmentácie obrazu	35
4.1.1 Fáza inicializácie	35
4.1.2 Detekcia úst	36
4.1.3 Detekcia hrán	37
4.1.4 Segmentácia úst	37
4.1.5 Úspešnosť riešenia	38
4.2 Využitím metód na identifikáciu tváre	39
4.2.1 Spracovanie argumentov a otvorenie súborov	40
4.2.2 Cyklus detekcie parametrov tváre v snímkach	40
4.3 Výsledky	42

5	Dataset	45
5.1	Výpočet časov cvičení pre video súbor	45
5.2	Extrakcia spracovaných parametrov tváre pre snímky konkrétneho cvičenia	47
5.3	Konverzia parametrov tváre cvičenia do vektorov vlastností	47
5.3.1	Tvorba vektorov vlastností SD–RSD	48
5.3.2	Tvorba vektorov vlastností časových úsekov	49
5.4	Výber cvičení a dát pre odhad	51
5.5	Popis datasetu	52
5.5.1	Časť obsahujúca vektory vlastností SD–RSD	52
5.5.2	Časť obsahujúca vektory vlastností časových úsekov	53
6	Trénovanie modelov na predikcia prítomnosti PCH a progresu jej symptómov u pacienta	54
6.1	Predikcia na časti datasetu SD–RSD	54
6.1.1	Trénovanie a hodnotenie modelov	54
6.1.2	Výsledky	57
6.2	Predikcia na časti datasetu časových úsekov	61
6.2.1	Klasifikačné modely	61
6.2.2	Regresné modely	61
6.2.3	Výsledky	61
6.3	Porovnanie výsledkov	63
7	Záver	65
	Literatúra	67
	Zoznam symbolov, veličín a skratiek	73
	Zoznam príloh	74
A	Graf úspešnosti predikcie prítomnosti PCH pre jednotlivé cvičenia vo vzťahu so vstupnými parametrami(SD–RSD data)	75
B	Výsledky predikcie progresu PCH a jej symptómov u pacienta(SD–RSD dáta)	77
C	Výsledky predikcie progresu PCH a jej symptómov u pacienta(dáta časových úsekov)	81
D	Obsah příloženého CD	83

Zoznam obrázkov

1.1	Dopaminergné dráhy[6].	13
1.2	Motorické príznaky Parkinsonovej choroby[8].	14
2.1	Prahovanie (vľavo originál, vpravo výsledok)[16].	20
2.2	Detekcia hrán (vľavo originál, vpravo výsledok).	21
2.3	Náhľad stromovej štruktúry algoritmu delenia a spájania oblastí[10]. .	23
2.4	Segmentácia na základe oblasti (vľavo originál, v strede zrno, vpravo výsledok)[19].	23
2.5	Ukážka dendogramu hierarchického zhukovania[10].	24
2.6	Segmentácia algoritmom k -means (vľavo originál, vpravo výsledok)[22].	25
2.7	Typické prvky systému rozpoznávania tváre[23].	26
2.8	Topologický graf tváre[27].	28
2.9	Konštrukcia podoblastí a výpočet podobnosti metódy používajúcej SIFT črty[28].	29
3.1	Kroky K -fold krížovej validácie[33].	32
3.2	Grafické znázornenie klasifikácie a regresie[35].	33
4.1	Kroky detekcie úst.	36
4.2	Výsledok detekcie hrán a segmentácie.	37
4.3	Body popisujúce tvár získané metódou <code>face_landmarks(...)</code> [45]. . .	42
4.4	Grafický výstup detekcie črtov tváre.	44
A.1	Graf úspešnosti predikcie prítomnosti PCH pre jednotlivé cvičenia vo vzťahu so vstupnými parametrami.	76

Zoznam tabuliek

5.1	Popis predikovaných hodnôt zaradených do datasetu.	51
5.2	Popis cvičení vybraných na tvorbu datasetu.	52
6.1	Najúspešnejšie modely predikcie prítomnosti choroby u pacienta na základe SD–RSD dát.	57
6.2	Päť vstupných parametrov s najlepšou priemernou úspešnosťou na trénovaného modelu pre predikciu prítomnosti choroby u pacienta na základe SD–RSD dát.	58
6.3	Najlepšie modely odhadu progresu symptómov pacienta pre jednotlivé symptómy podľa MAEPZR (SD–RSD dáta).	60
6.4	Najúspešnejšie modely predikcie prítomnosti choroby u pacienta na základe dát časových úsekov.	62
6.5	Najlepšie modely odhadu progresu symptómov pacienta pre jednotlivé symptómy podľa MAEPZR (dáta časových úsekov).	63
B.1	Najlepšie modely odhadu progresu symptómov pacienta pre jednotlivé symptómy podľa MAE (SD–RSD dáta).	78
B.2	Najlepšie modely odhadu progresu symptómov pacienta pre jednotlivé symptómy podľa MAPE (SD–RSD dáta).	79
B.3	Najlepšie modely odhadu progresu symptómov pacienta pre jednotlivé symptómy podľa RMSE (SD–RSD dáta).	80
C.1	Najlepšie modely odhadu progresu symptómov pacienta pre jednotlivé symptómy podľa MAE (dáta časových úsekov).	81
C.2	Najlepšie modely odhadu progresu symptómov pacienta pre jednotlivé symptómy podľa MAPE (dáta časových úsekov).	82
C.3	Najlepšie modely odhadu progresu symptómov pacienta pre jednotlivé symptómy podľa RMSE (dáta časových úsekov).	82

Zoznam výpisov

4.1	Telo hlavnej metódy Python skriptu.	39
4.2	Telo metódy <code>segmentate_video(...)</code>	40
4.3	Ukážka JSON formátu výsledných dát (pre 1 snímku).	42
5.1	Telo funkcie <code>cutExerciseAudioAndSaveIt(...)</code>	46
5.2	Telo funkcie <code>extractFeatureVectors(...)</code>	49
5.3	Telo funkcie <code>computeVectors(...)</code>	50
6.1	Telo metódy <code>knn(...)</code>	55
6.2	Telo metódy <code>regression(...)</code>	56

Úvod

V posledných desaťročiach napredujú *Informačné technológie* (IT) nezastaviteľným tempom. Pravidelne vznikajú nové technológie, ktoré nachádzajú svoje uplatnenie v čoraz viac odvetviach a na miestach, kde to pred pár rokmi nebolo možné. Mnohé procesy sa dajú vďaka novým technológiám automatizovať a tým pádom dochádza k úspore času. Okrem automatizácie sa využívajú IT aj pri návrhoch nových riešení, ktoré môžu vznikáť vďaka technologickému pokroku. IT sa v dnešnej dobe využívajú v každom jednom odvetví, od poľnohospodárstva cez bankovní sektor až po medicínu. V medicíne sú IT využívané napríklad pri úschove informácií v databázových systémoch, určovaní diagnóz, prevádzaní operácií ale aj pri vyhodnocovaní vyšetrení. Značné množstvo lekárskeho vyšetrení má grafický výstup v podobe obrazu, ktorý je človeku ľahšie pochopiteľný. Pri generovaní a úprave týchto výstupov sa využívajú techniky na spracovanie obrazu, ktoré sú taktiež využívané aj pri samotnej analýze výstupov. Pri analýze výsledkov je používaná aj umelá inteligencia, ktorá nachádza svoje využitie v mnohých softwéroch určených na diagnostikáciu rôznych chorôb[1].

Jednou z chorôb, ktoré sa snažia vedci diagnostikovať pomocou využitia IT je aj *Parkinsonova choroba* (PCH). Jedná sa o druhé najčastejšie sa vyskytujúce neurodegeneratívne ochorenie. Vzniká na základe zániku dopamínerných neurónov, čoho dôsledkom je nedostatok dopamínu v organizme. Prejavuje sa mnohými motorickými ale aj nemotorickými príznakmi, ktoré znižujú kvalitu života pacientov[2].

Myšlienkou tejto práce je vytvorenie programu, ktorý s využitím techník spracovania obrazu a najnovších pokrokov v umelej inteligencii dokáže z videa zachytávajúceho rozprávajúcu osobu vytvoriť časovú radu dát popisujúcich kľúčové črty tváre v čase. Táto rada dát bude následne použitá ako vstup pre umelú inteligenciu diagnostikujúcu štádium PCH u sledovaného pacienta. Prínos takéhoto programu v medicíne spočíva v automatizácii pomerne komplexného problému diagnostikácie štádia PCH na základe objektívneho štatistického zhodnotenia pomocou matematického modelu. Týmto spôsobom by sa dal proces diagnostikácie značne urýchliť a taktiež by sa zamedzilo vplyvu subjektivity, ktorá je do tohoto procesu vnášaná človekom.

Práca je štrukturovaná nasledovne: Kapitola 1 sa zaoberá popisom PCH. Kapitola 2 je venovaná rozboru metód parametrizácie obrazu. Za ňou nasleduje kapitola 3 popisujúca umelú inteligenciu, pričom je zameraná na popis metód využitých v praktickej časti práce. V 4. kapitole sú popísané navrhnuté riešenia parametrizácie tváre, ich implementácia a ich úspešnosť. 5. kapitola obsahuje popis tvorby datasetu a jeho výslednú formu. Tento dataset bol následne použitý pre tréning modelov umelej inteligencie určených na predikciu PCH a jej symptómov u pacienta popísaných v kapitole 6. Táto kapitola obsahuje taktiež popis dosiahnutých výsledkov predikcií.

1 Parkinsonova choroba (PCH)

PCH je svojou prevalenciou druhé najčastejšie sa vyskytujúce neurodegeneratívne ochorenie hneď po Alzheimerovej chorobe. Postihuje najmä starších ľudí, nakoľko priemerný vek kedy sa PCH začína u ľudí prejavovať je 60 rokov. V priemere postihuje asi 1% populácie vo veku 65 rokov, pričom riziko vzniku rastie priamo úmerne s vekom. Vo veku 85 rokov je prevalencia PCH 4–5%. Prvý krát bola popísaná v roku 1817 Jamesom Parkinsonom v knihe s názvom „*An essay on the shaking palsy*“[3].

Jedná sa o chronické progresívne ochorenie centrálného nervového systému, ktoré zaraďujeme do skupiny porúch pohybového systému a psychických funkcií. Vzniká na podklade degeneratívneho zániku dopamínerných neurónov v mozgovej oblasti *substantia nigra pars compacta*, ale aj v ďalších oblastiach mozgu. Dôsledkom straty týchto neurónov je nedostatok dopamínu v organizme. Pri PCH môžu nastať abnormality aj u ďalších neurotransmiterov. Príčina straty buniek produkujúcich dopamín zatiaľ nie je jasná, rovnako ako sa nevie prečo sa u niektorých ľudí PCH rozvinie a u ostatných nie[4][5].

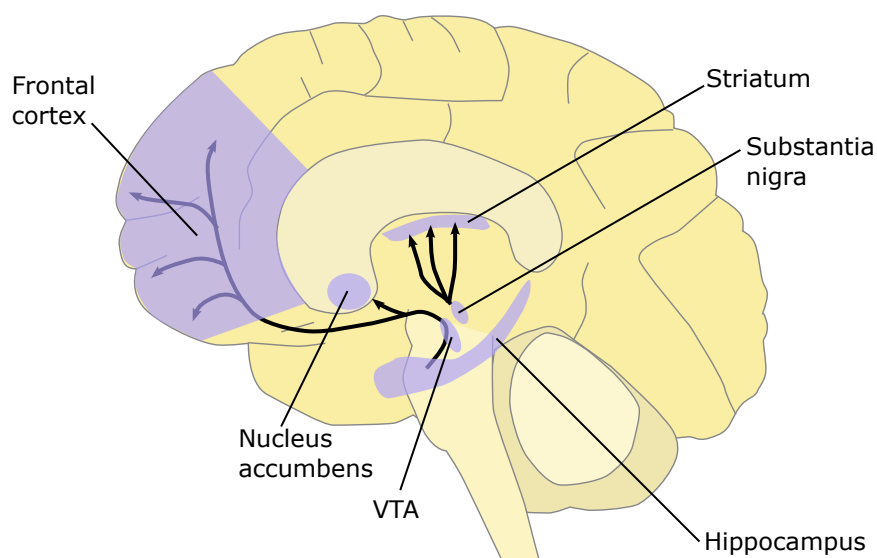
PCH má priamy dopad na kvalitu pacientovho života. Pacient trpiaci touto chorobou sa môže javiť ako menej inteligentný, má problém s vyjadrovaním, s gestikuláciou, javí sa ako by bol bez emócií a má pohybové problémy[5].

1.1 Dopamín

Dopamín je chemická látka prirodzene vznikajúca v mozgu, ktorá zastáva dôležitú úlohu pri prenose nervových impulzov z jednej nervovej bunky do druhej na najmenej šiestich nervových dráhach. Tieto dráhy sú označované ako dopaminergné a jedna z nich je dráha spájajúca oblasť *substantia nigra* s oblasťou *striatum* (viď Obr. 1.1). *Substantia nigra* zastupuje významnú úlohu práve v plánovaní a programovaní pohybu, kedy sa myšlienka pomocou nervových impulzov mení na vôľou riadený pohyb. Nedostatok dopamínu v tejto oblasti má za následok nesprávny prenos nervových impulzov po danej dopaminergnej ceste, čoho výsledkom sú poruchy pohybu charakteristické pre PCH. Tieto poruchy pohybu sa označujú ako príznaky motorické. Abnormality u ostatných neurotransmiterov mimo dopamín sú zodpovedné za príznaky nemotorické[6][7].

1.2 Príznaky Parkinsonovej choroby

PCH je charakteristická veľkým počtom motorických a nemotorických príznakov, ktoré sa zvyčajne začnú prejavovať pri strate 50 až 80 % neurónov produkujúcich



Obr. 1.1: Dopaminergné dráhy[6].

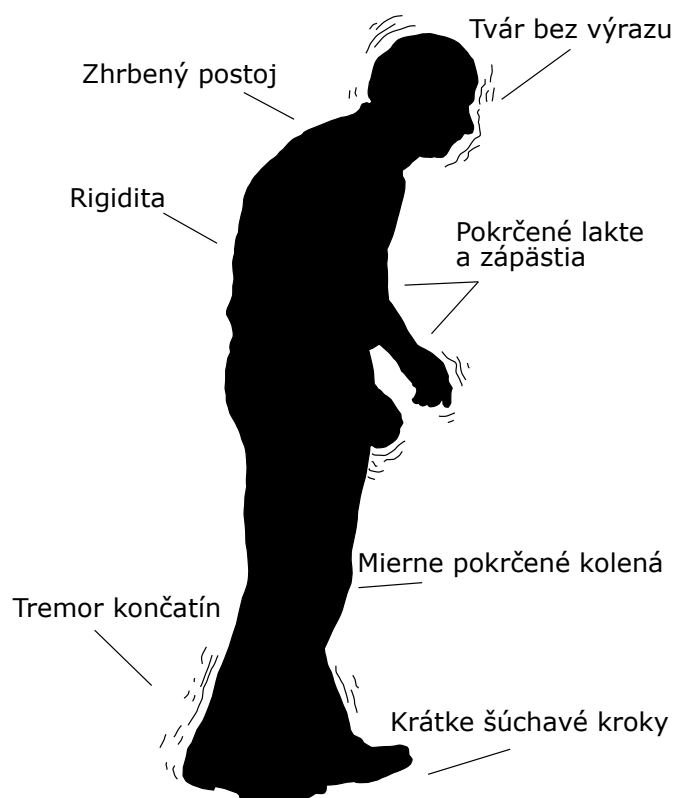
dopamín. Príznaky sa postupom času s klesajúcim počtom dopaminergných neurónov zhoršujú, avšak tento priebeh zhoršovania nie je v čase lineárny. Rýchle zmeny a nástup symptómov indikujúcich PCH sú pozorovateľné najmä zo začiatku choroby, postupom času sú zmeny pomalšieho charakteru. Táto práca sa zaoberá najmä príznakmi motorickými, nakoľko práve tie ovplyvňujú mimiku tváre[2][4].

1.2.1 Motorické príznaky

Z hľadiska motoriky pacienta, je PCH charakterizovaná syndrómom všeobecne pomenovaným ako *parkinsonizmus*, ktorý zahŕňa štyri kardinálne príznaky. Sú nimi bradykinézia, tremor, rigidita a poruchy chôdze a postoja. U pacienta trpiaceho PCH nemusia byť pozorované všetky tieto príznaky, aspoň nie súčasne v jednom časovom období[5]. Okrem vymenovaných motorických príznakov sa vyskytujú aj ďalšie. Najcharakteristickejšie z príznakov sú zobrazené na Obr. 1.2 a budú bližšie popísané v nasledujúcich odstavcoch.

Bradykinézia

Alebo taktiež spomalenosť pohybov je najcharakteristickejším príznakom PCH. Zahŕňa poruchu naplánovania, zahájenia a prevedenia pohybu. Pohyby sú spomalené, prevádzané s menším rozsahom. Reč je tichá a taktiež spomalená, môže sa vyskytovať dysartria (problém s vyslovovaním hlások „l“, „p“, „r“ a „b“), palilalia (opakovanie slabiky alebo celého slova na začiatku vety) alebo aj tachyfémia (postupné zrýchľovanie reči až nezrozumiteľná reč na konci výdychu). Mimika tváre je nehybná,



Obr. 1.2: Motorické príznaky Parkinsonovej choroby[8].

maskovitá, nereaguje na emócie. Taktiež je menej časté žmurkanie očí. Najväčším problémom je prvotné zahájenie pohybu, pri ktorom sa pacient musí sústrediť aby vykonal pohyb, ktorý sa u zdravého človeka vykonáva automaticky[2][5].

Tremor

Kľudový tremor je rytmický oscilačný nedobrovoľný pohyb, ktorý sa prejavuje keď je daná časť tela v kľude a stabilizovaná podložkou. Frekvencia pohybu je zvyčajne v rozmedzí 3–6 Hz, rozsah pohybu je variabilný od menej ako 1cm až po viac ako 10cm. Najcharakteristickejšim je tremor na ruke, ktorý vzniká šúchaním palca o ukazovák a zvyčajne prechádza z jednej ruky do druhej. Tremor charakteristický pre PCH zahŕňa pohyby pier, brady, sánky a nôh. Na rozdiel od obvyčajného tremoru nezahŕňa krk, hlavu a hlas. Tremor mizne počas spánku a počas aktivity[5][9].

Rigidita

Rigidita, teda stuhlosť, je charakteristická zvýšeným odporom pri pohybe. Pri pasívnom pohybe je odpor citelný pri flexii aj extenzií a počas celého rozsahu pohybu.

S rastúcou rýchlosťou pohybu sa odpor nemení. Najviac sa prejavuje na svaloch ohýbajúcich končatiny a trup. Tento fakt sa prejavuje na držaní tela pacientov, ktorý majú trup v predklone, čo býva patrné najmä pri chôdzi. Niekedy majú pri chôdzi pocit ťahu do predu. Rigidita spôsobuje väčšiu námahu potrebnú na vykonanie pohybu, čo sa u pacienta prejavuje na zmenšenej rýchlosti vykonávania pohybov[9].

Poruchy chôdze a postoja

Poruchy držania postoja zapríčinené stratou posturálnych reflexov sú vo väčšine prípadov znakom neskorších fáz PCH a zvyčajne sa neprejavujú ako prvý motorický príznak. Chôdza pacientov je šúchavého charakteru, s malými krokmi. Horné končatiny sú pri chôdzi pripažené a v klude. Pacienti majú problém so zmenou smeru pohybu a otočením sa, ktoré prevádzajú pomaly a pomocou niekoľkých malých krokov. Niektorí pacienti majú problém ako so zahájením tak aj so zastavením chôdze. Chôdza pacientov môže pôsobiť dojmom, že sa pacient snaží malými krokmi bežať do predu a niekedy nastáva aj takzvané „zamrznutie“ (akinetický freezing), kedy pacient pri chôdzi náhle zastaví. Pacient obvykle trpí aj problémom s vyrovnaním rovnováhy, kedy sa snaží rovnováhu vyrovnať drobnými krokmi. Strata posturálnych reflexov vedie k pádom, ktorých frekvencia koreluje s pokrokom štádia choroby u pacienta. V pokročilom štádiu sú u pacientov pozorované pády bez reakcie. Nakoľko sa PCH zvyčajne vyskytuje u ľudí vo veku nad 60 rokov, pri páde je väčšia šanca vzniku fraktúry kosti[2][5][9].

Akinetický freezing

Akinetický freezing, taktiež nazývaný ako blok motoriky, je jeden z najviac obmedzujúcich symptómov PCH. Napriek tomu, že je charakteristický, nevyskytuje sa u všetkých pacientov. Výskyt je približne u 50% pacientov, pričom častejšie sa vyskytuje u mužov a s menšou pravdepodobnosťou sa vyskytuje v prípade, že pacientov hlavný syndróm je tremor. Akinetický freezing sa prejavuje najmä na dolných končatinách pri chôdzi, ale taktiež môže postihnúť aj ruky a očné viečka. Prejavuje sa ako náhla neschopnosť pohybu trvajúca zpravidla menej ako 10 sekúnd. Zväčša sa prejavuje pri zahájení chôdze, alebo pri špecifických situáciách akými sú napríklad zmena smeru pohybu, prechod úzkym priestorom alebo chôdza rušným miestom. Na pacienta má značný sociálny dopadom a je jedným z hlavných dôvodov pádov. Chôdza do rytmu hudby alebo aj prekračovanie rôznych objektov patria medzi spôsoby, ktoré si pacienti vyvynuli na prekonanie tohoto príznaku[5].

1.2.2 Nemotorické príznaky

PCH bola v minulosti spájaná hlavne s motorickými príznakmi, avšak v posledných rokoch vzrástol záujem aj o nemotorické príznaky, ktoré sa využívajú pre diagnostiku a taktiež majú priamy dopad na kvalitu života pacienta. Niektoré z nemotorických príznakov sa prejavujú niekoľko rokov až desaťročí pred prejavom prvých motorických, čoho sa snažia vedci využiť pri potencionalnej diagnostikácii skorého štádia choroby u pacienta. Sú nimi napríklad rýchli pohyb očí, poruchy v správaní alebo aj depresie. Príznaky akými sú napríklad demencia alebo halucinácie sa prejavujú v neskorších štádiách choroby, čo slúži na odlíšenie PCH od ostatných chorôb[9].

Depresia

Depresia je chorobný stav prejavujúci sa skleslou náladou sprevádzanou zníženou aktivitou a energiou jedinca. Podľa intenzity a množstva príznakov sa môže deliť na miernu, strednú alebo ťažkú. Najčastejšie udávaná prevalencia depresie pri PCH je 40%[2].

Psychóza

Psychóza je definovaná ako porucha percepcie (zmyslového vnímania) a myslenia, obvykle zahrňujúca aj halucinácie, ilúzie a symptómy paranoie. Vizuálne halucinácie sú najčastejším psychotickým prejavom pri PCH. Zvyčajne im predchádzajú farebné živé sny a nočné mory. Trpí nimi približne 20–40% pacientov v prevažne pokročilom štádiu PCH. Obecne platí, že všetky antiparkinsonika môžu vyvolať alebo zhoršovať psychotické symptómy[2].

Demencia

Neuropsychologický profil demencie je pri PCH definovaný ako progresívny syndróm s pamäťovým deficitom a poruchou abstraktného myslenia. Riziko rozvoja demencie je u človeka trpiaceho PCH 1,7–5,9 krát väčšie než u bežnej populácie v rovnakom veku, pričom už v skorých štádiách choroby sú pozorovateľné kognitívne poruchy mierneho stupňa[2].

Poruchy spánku

Sú častým príznakom, ktorým v priebehu choroby trpí až 96% pacientov. Patrí medzi ne napríklad fragmentácia spánku. Antiparkinsoniká môžu viesť k farebným snom a nočným morám, ktoré sú predzvestou halucinácií (viď. 1.2.2). Vyskytujú sa taktiež príznaky viazané na REM fázu spánku, medzi ktoré patria napríklad výkriky, nadávanie, kopanie alebo aj skákanie[2][5].

2 Metódy parametrizácie obrazu

Parametrizácia obrazu označuje proces, ktorým sme schopný v digitálnom obraze označiť a sledovať rôzne parametre. V prípade, že sa jedná o video, sme v jeho sekvencií snímok schopný zaznamenávať zmenu týchto parametrov v čase. Keďže sa táto práca zaoberá parametrizáciou tváre, jednotlivé parametre budú čo možno najdetailnejšie sledovať a popisovať tvár nachádzajúcu sa na snímku.

Pri parametrizácii sú využívané rôzne techniky počítačového spracovania obrazu. Pri implementácii riešenia tejto práce bola využitá segmentácia obrazu, techniky predspracovania obrazu pre samotnú segmentáciu a metódy na identifikáciu osôb pomocou tváre, pod ktoré spadajú aj funkcie na vytvorenie jednotlivých bodov opisujúcich jej najvýraznejšie črty[10].

2.1 Segmentácia obrazu

Segmentácia obrazu je jednou z najpoužívanějších techník v spracovaní obrazu a počítačovom videní. Taktiež zohráva významnú rolu v rozpoznávaní objektov v obraze. Je používaná napríklad v medicíne, systémoch riadenia dopravy ale aj pri rozoznávaní odtlačkov prstov alebo objektov na satelitných snímkach.

Pod segmentáciou obrazu sa rozumie rozdelenie obrazu na regióny alebo kategórie, ktoré odpovedajú odlišným objektom alebo častiam daného obrazu. Pod dobrou segmentáciou sa rozumie taká, pri ktorej:

- pixely v jedej kategórii majú podobné hodnoty na stupnici šedej a tvoria spojitý región v obraze,
- susedné pixely patriace do rozličných regiónov majú rozdielne hodnoty na stupnici šedej[11].

2.1.1 Predspracovanie obrazu

Pred samotnou segmentáciou obrazu je často nutné obraz vhodne predspracovať tak, aby bol čo najvhodnejšie upravený pre zvolenú segmentačnú metódu. Medzi tieto úpravy patrí napríklad prevod do stupnice šedej, redukcia šumu ale aj zaostrenie obrazu.

Prevod do stupnice šedej

Prevod obrazu do stupnice šedej je jednou z operácií, ktorú využíva mnoho metód. Obraz v stupnici šedej je taký, v ktorom nesie každý pixel iba jednu hodnotu reprezentujúcu množstvo svetla – intenzitu. V prípade obrazu, ktorého pixel má 8 bitov,

sú hodnoty šedej v rozsahu 0-255, pričom 0 označuje najnižšiu (čierna) a 255 najvyššiu intenzitu (biela). Prevod farby pixlu je v prípade lineárneho RGB farebného modelu vyrátaný podľa rovnice:

$$Y=0,2126R+0,7152G+0,0722B. \quad (2.1)$$

Z rovnice vidíme, že sa jedná o vážený súčet, pričom jednotlivé váhy sú určené na základe vlastností ľudského oka. To je najviac senzitívne na zelenú farbu, a najmenej na farbu modrú. V prípade, že farba nie je v lineárnom RGB modeli, je potrebné ju najprv transformovať do lineárneho modelu[12].

Redukcia šumu

Šum je náhodná zmena intenzity obrazu a je pozorovateľný v podobe určitej zrnitosti. Najvýraznejšie pôsobí v homogénnych oblastiach, naopak v nehomogénnych častiach obrazu sa môže strácať. Šum znamená, že jednotlivé pixely zobrazujú rozdielne hodnoty intenzity od skutočných. Algoritmy na redukciu šumu fungujú na báze redukcie alebo odstránenia viditeľnosti šumu vyhladením obrázku, pri ktorom však môže dôjsť k prekrytiu jemných – nízko kontrastných detailov. Existuje viac druhov a zdrojov šumu. Zvyčajne sa vyskytujúce druhy šumu v obraze sú napríklad impulzný, aditívny alebo multiplikatívny šum. Redukcia šumu je vykonávaná pomocou filtrov. Tieto filtre delíme na lineárne, adaptívne a nelineárne

Medzi lineárne filtre patrí napríklad Gaussov filter. Lineárne filtre majú tendenciu rozmazávať ostré hrany, mazať tenké čiary a odstranovať jemné detaily. Ich podstata je v nastavení hodnoty každého pixlu na priemernú alebo váženú hodnotu jeho susediacich pixlov.

Pri adaptívnych filtroch sa aplikuje filter jednotlivo pre oblasti obrazu. Wienerova funkcia používa lineárny Wienerov filter, avšak aplikuje ho adaptívne v závislosti na rozmanitosti oblastí. V prípade veľkej rozmanitosti vykoná vyhladzovanie v menšej miere, naopak ak je rozmanitosť nízka znamená to väčšiu homogenitu a je aplikované výraznejšie vyhladenie oblastí.

Medzi nelineárne filtre radíme napríklad mediánové typy filtrov. Mediánový filter funguje na takmer rovnakom princípe ako filter lineárny, avšak hodnota pixlu nie je nahradená priemerom hodnôt s ním susediacich pixlov, ale mediánom týchto hodnôt. Medián je menej senzitívny na prípadné extrémne odlišné hodnoty susediacich pixlov. Tento typ filtru je jednoduchý na implementáciu a prináša značne lepšie výsledky[13].

Zaostrenie obrazu

Ludské oko je citlivé na hrany a jemné detaily obrazu, ktoré sú väčšinou zložené z vysokých frekvencií. Vysoké frekvencie sú v častiach obrazu, ktoré sú nehomogénne (zmena farby, hrana, ...). Naopak nízke frekvencie sú v oblastiach homogénnych. Vizualna kvalita obrazu môže byť značne degradovaná v prípade potlačenia alebo odstránenia vysokých frekvencií. V prípade zvýraznenie vysokých frekvencií dochádza k zvýšeniu vizuálnej kvality obrazu. Zaostrenie obrazu je technika, ktorá zvýrazňuje hrany a detaily pomocou zvýraznenia oblastí s vysokou frekvenciou. Pri procese zaostrenia sa najprv pomocou filtru typu hornej priepusti extrahujú z obrazu oblasti s vysokými frekvenciami. Následne sa na základe požadovanej hodnoty zaostrenia výstup filtru upraví a pričíta k pôvodnému obrazu. Týmto spôsobom dostávame obraz so zvýraznenými областami vysokých frekvencií. Homogénne oblasti obrazu ostávajú bez zmeny[14].

2.1.2 Metódy segmentácie

V súčasnosti existuje mnoho algoritmov využívajúcich rôzne metódy segmentácie. Metódy sa podľa spôsobu segmentácie delia na štyri základné:

- prahovanie (thresholding),
- detekcia hrán (edge detection),
- na základe oblasti (region-based),
- zhľukovanie na základe vlastností (feature based clustering)[15].

Prahovanie

Prahovanie je jednou z najjednoduchších a najvyužívanejších metód segmentácie, ktorá použitím jednej hodnoty prahu t rozdelí všetky pixely obrázku do dvoch kategórií. Pixel obrazu na pozícii (i,j) s hodnotou na stupnici šedej označenej ako f_{ij} je zaradený do prvej skupiny ak

$$f_{ij} \leq t. \quad (2.2)$$

V prípade, že danú rovnosť nespĺňa je zaradený do skupiny druhej[11]. Z popisu vyplýva, že kľúčovým faktorom pre dosiahnutie požadovaných výsledkov je pri tejto metóde výber adekvátnej hodnoty prahu. Tá je v mnohých prípadoch volená manuálne technikou pokus – omyl, kedy vedec skúša rôzne hodnoty prahu a analyzuje dosiahnuté výsledky. Druhým spôsobom zvolenia prahu je adaptívna metóda, kedy je prah vypočítaný automaticky. Jednou z možností adaptívneho výpočtu prahu je výpočet prahu na základe hodnôt histogramu stupnice šedej¹. Ďalšou z metód na

¹Histogram v digitálnej fotografii označuje grafické znázornenie množstva jednotlivých jasových zložiek (0-255) v obraze.

adaptívny výpočet prahu je metóda, ktorá z jednotlivých pixlov vyberie jeden s najnižšou a jeden s najvyššou hodnotou na stupnici šedej. Z nich sa následne vyráta ich priemerná hodnota, ktorá je použitá ako prah.

Prahovanie sa delí na globálne a lokálne. Pri globálnom prahovaní sú všetky pixely obrazu rozdelené na základe jednej hodnoty prahu buď do skupiny označujúcej objekt záujmu alebo do skupiny pozadia. Pri lokálnom je obraz rozdelený na niekoľko oblastí a hodnota prahu je adaptívne vyrátaná zvlášť pre každú jednu z oblastí. V prípade niekoľkých odlišných hodnôt prahov sú výsledkom prahovania viac ako dve skupiny, pod ktoré spadajú pixely obrazu[11][15].



Obr. 2.1: Prahovanie (vľavo originál, vpravo výsledok)[16].

Detekcia hrán

Detekcia hrán je často využívaný proces, ktorým sme schopný v obraze nájsť a zvýrazniť hrany. Hrany zvyčajne opisujú tvar objektov a tým pádom dokážeme kvalitnou detekciou hrán zvýrazniť a oddeliť jednotlivé objekty obrazu. Hrana je v digitálnom obraze definovaná ako značná lokálna zmena intenzity. Výsledkom detekcie hrán by mal byť obraz, na ktorom sú spojitými čiarami zobrazené jednotlivé hrany, opisujúce dôležité črty obrazu. Tieto črty môžu byť následne spracované algoritmami počítačového videnia.

Pri hranovej detekcii rozlišujeme na základe profilu zmeny intenzity štyri rôzne typy hrán:

- skoková – zmena intenzity je náhla a skoková z jednej hodnoty na druhú,
- rampová – zmena intenzity nieje okamžitá ale postupná z jednej hodnoty na druhú,
- hrotová – intenzita sa skokovo mení z jednej hodnoty na druhú ale po krátkej vzdialenosti sa znova vracia na pôvodnú hodnotu,

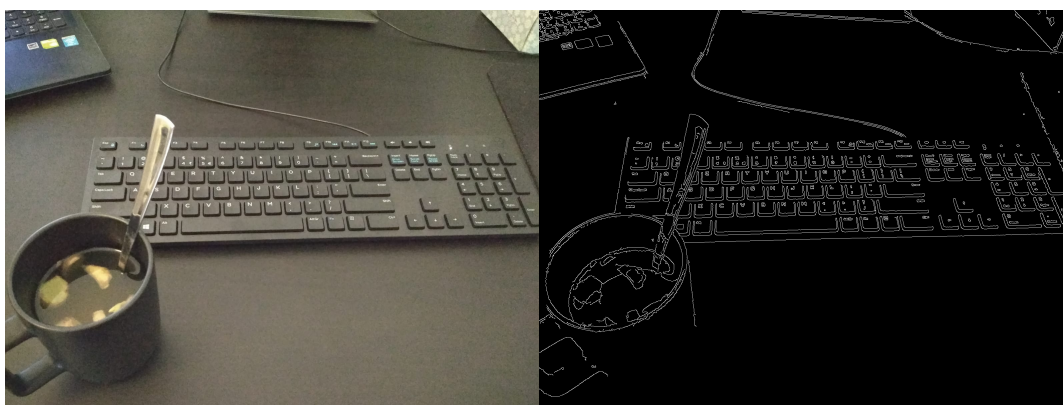
- strechová – v podstate sa jedná o dve rampové hrany, pričom prvá predstavuje zmenu intenzity a druhá jej návrat do pôvodnej hodnoty.

Detekcia týchto hrán je zvyčajne prevádzaná v štyroch krokoch:

- vyhladenie – potlačenie čo možno najviac šumu bez poškodenia hrán,
- zaostrenie – aplikovanie filtru na zvýšenie kvality hrán,
- detekcia – rozhodnutie, ktoré hranové pixely by mali byť označené ako šum a ktoré by sa mali zachovať,
- lokalizácia – určenie presnej polohy hrany, zúženie hrany a spojenie jej častí do jednej[17].

Existuje veľa metód detekcie hrán, avšak väčšina z nich spadá do dvoch kategórií – založené na vyhľadávaní a založené na prechode nulou. Metódy založené na vyhľadávaní využívajú na detekciu hrán meranie intenzity gradientu v bodoch obrazu. Na to je potrebná derivácia obrazu prvého rádu. Pri gradiente vieme vypočítať jeho veľkosť a smer, pričom veľkosť udáva mieru odlišnosti intenzity od susedných pixlov (miera pevnosti hrany) a smer gradientu udáva smer najväčšej zmeny intenzity. Metódy založené na prechode nulou hľadajú za účelom nájdenia hrán prechody nulou v derivácií druhého rádu obrazu.

Metódy na detekciu hrán sa líšia hlavne v spôsobe vyhladzovania a v spôsobe akým je meraná miera pevnosti hrany. Nakoľko sa veľa metód spolieha na výpočet gradientov obrazu, líšia sa taktiež v typoch filtrov používaných na výpočet odhadu gradientov, akými sú napríklad Sobel, Laplacian alebo Canny[18].



Obr. 2.2: Detekcia hrán (vľavo originál, vpravo výsledok).

Na základe oblasti

Metódy segmentácie na základe oblasti spoliehajú na fakt, že susedné pixely v rámci jedného regiónu majú podobné hodnoty. Využívanou procedúrou je porovnávanie jedného pixlu s jeho susediacimi a ak je splnené kritérium podobnosti, susedný pixel môže byť pridaný do rastúceho zhluku pixlov tvoriacich oblasť. Výber kritéria podobnosti je kľúčový pre výsledok, rovnako ako aj šum, ktorý vplýva na výsledky negatívne. Niektoré z algoritmov sú napríklad rastúca oblasť so zrnom, rastúca oblasť bez zrna, delenie a spájanie oblastí a rýchle skenovanie.

Algoritmus rastúcej oblasti so zrnom je jedným z najjednoduchších algoritmov segmentácie na základe oblasti. Segmentácia je vykonávaná na základe porovnávania susedných pixlov s množinou pixlov označených ako zrno. Celý proces algoritmu je nasledovný:

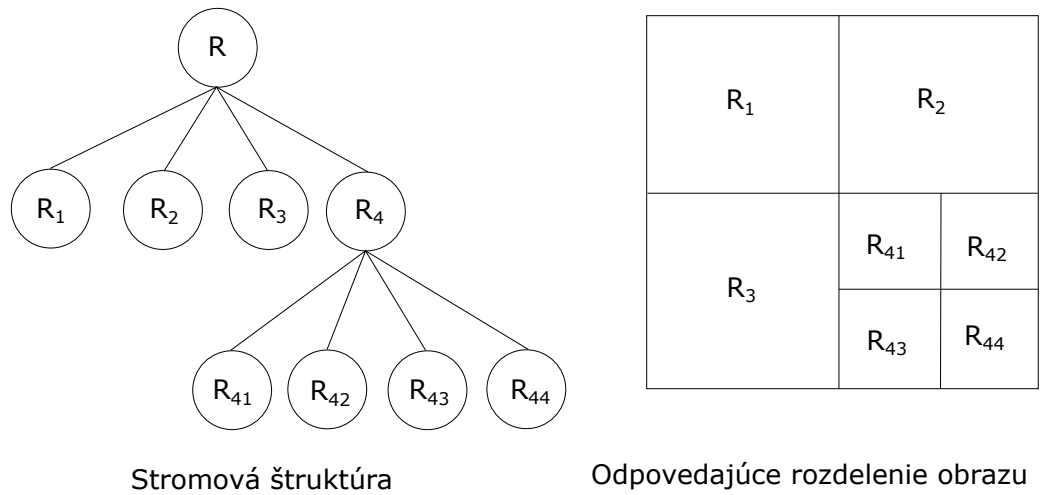
1. Začína sa s určitým počtom pixlov označených ako zrno, ktoré sú zoskupené do n zhlukov označených ako C_1, C_2, \dots, C_n . Pozície týchto pixlov sú označené ako p_1, p_2, \dots, p_3 .
2. Nasleduje výpočet rozdielu medzi hodnotami pixlov zrna p_i a hodnotami s nimi susediacich pixlov. Ak je rozdiel menší ako prah (kritérium podobnosti) definovaný užívateľom, susedný pixel môže byť priradený do zhluku C_i .
3. Nový výpočet hranice každého zhluku C_i a nastavenie tejto hranice ako nové zrno p_i . Zároveň sa musí vyrátať priemerná hodnota pixlov v jednotlivých zhlukoch, ktorá bude slúžiť pre výpočet rozdielu so susednými pixlami.
4. Opakovanie kroku 2 a 3 až pokiaľ bude každý pixel priradený do vhodného zhluku.

Prah je vytvorený užívateľom a zvyčajne je zvolený na základe intenzity, úrovni šedej alebo farebných hodnôt. Takisto iniciálne zrno je vyberané užívateľom, pričom jeho výber má priamy dopad na výsledok. Rôzne sady počiatočných pixlov vedú k rôznym výsledkom segmentácie rovnakého obrazu. Ďalším dôležitým aspektom je počet pixlov v zrne, ktorý je rozdielny v závislosti od obrazu. Najväčším problémom tohoto algoritmu sú vysoké nároky na výpočetný čas.

Algoritmus rastúcej oblasti bez zrna je odvodený od toho so zrnom s tým rozdielom, že výber zrna nie je potrebný. Pri procese segmentácie sa iniciálne zrno zvolí automaticky a tým pádom je tento algoritmus plne automatický a robustnejší.

Delenie a spájanie oblastí je algoritmus, ktorého hlavným cieľom je rozoznať homogénne oblasti obrazu. Je založený na koncepte stromu, v ktorom má každý vrchol štyroch potomkov a koreň stromu odpovedá celému obrazu. Každý vrchol reprezentuje ďalšie rozdelenie na štyri vrcholy potomkov. Tento návrh je zobrazený na obrázku 2.3. Algoritmus funguje na princípe, že celý obraz označíme ako R a určíme predikát P . V prípade, že $P(R)=\text{nepravda}$, rozdelíme obraz R na štyri časti. Ak je P

nepravda pre ktorúkoľvek z častí, ďalej ju delíme na ďalšie štyri časti až dokým neplatí pre každú oblasť R_i $P(R_i)$ =pravda. Po procese delenia nasleduje proces spájania, pri ktorom sú susedné oblasti R_j a R_k spojené v prípade, že $P(R_j \cup R_k)$ =pravda. Maximálny počet úrovní, do ktorých sa môže obraz rozdeliť je nastaviteľný užívateľom, tak isto ako aj kritérium pre delenie a spájanie oblastí, pričom môžu byť odlišné. Nevýhodou je výstup charakteru blokovitých segmentov. Tie sa dajú eliminovať použitím väčšieho počtu úrovní delenia za cenu nárastu výpočtového času[10].



Obr. 2.3: Náhľad stromovej štruktúry algoritmu delenia a spájania oblastí[10].

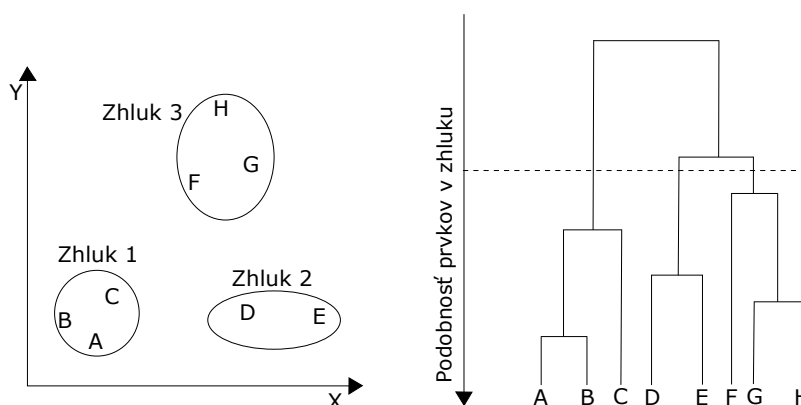


Obr. 2.4: Segmentácia na základe oblasti (vľavo originál, v strede zrnno, vpravo výsledok)[19].

Zhlukovanie na základe vlastnosti

Zhlukovanie vo všeobecnosti označuje proces zoskupovania objektov do skupín, ktoré obsahujú v určitom zmysle podobné dáta. Tým pádom sa dá zhluk definovať ako kolekcia objektov, ktoré sú si navzájom podobné a zároveň sú odlišné od objektov patriacich do ostatných zhlukov. Pri digitálnom obraze je zhlukovanie založené na vlastnostiach jednotlivých pixlov. Metódy zhlukovania sa dajú rozdeliť na dve kategórie: hierarchické a partíciové (z angl. *partitional*)[15].

Koncept hierarchického zhlukovania spočíva v konštrukcii dendogramu² reprezentujúceho vnorené zoskupenia vzorov (pri digitálnej fotografii pixlov) a úrovni podobností, na základe ktorých sa priradujú do jednotlivých zoskupení. Na základe dendogramu sú následne jednotlivé pixely zoskupované a priradované do jednotlivých zhlukov. Pre ukážku dendogramu predpokladajme osem zoskupení pixlov označených ako *A*, *B*, *C*, *D*, *E*, *F*, *G* a *H*, ktoré sú priradené do troch zhlukov zobrazených na obrázku 2.5 vľavo. Dendogram odpovedajúci tomuto stavu je zobrazený na rovnakom obrázku v jeho pravej časti[10].



Obr. 2.5: Ukážka dendogramu hierarchického zhlukovania[10].

Na rozdiel od hierarchického zhlukovania vytvárajúceho zhukovú štruktúru, partíciové zhlukovanie pracuje s jednou partíciou dát. To je využiteľné najmä pri veľkých datasetoch, pri ktorých by výpočet dendogramu potreboval veľký výpočetný čas. Problémom partíciového zhlukovania je potreba zvolenia počtu výsledných zhlukov pred vykonaním samotnej klasifikácie dát. Jedným z najznámejších algoritmov partíciového zhlukovania je algoritmus *k*-means. Kroky tohto algoritmu sú nasledovné:

1. Určenie počtu výsledných zhlukov *N*. Náhodný výber *N* vzorov (pixlov) zo všetkých dát ako *N* centroidov patriacich *N* zhukom.

²Dendogram označuje stromovú štruktúru často používanú na ilustráciu usporiadania zhlukov generovaných hierarchickým zhlukovaním[20].

2. Klasifikácia každého vzoru do centroidu najbližšieho zhluku. Najbližší zvyčajne znamená najpodobnejší v hodnote pixlu, ale môže to byť aj iná hodnota.
3. Prepočet zhlukových centroidov. Stále je ten istý počet N centroidov v N zhlukoch ako po prvom kroku.
4. Opakovanie krokov dva a tri až do kým nie sú splnené kritéria konvergenzie. Typickými kritériami konvergenzie sú: žiadne preradenie vzoru z jedného do druhého zhluku, alebo minimálny pokles v kvadratickej chybe.

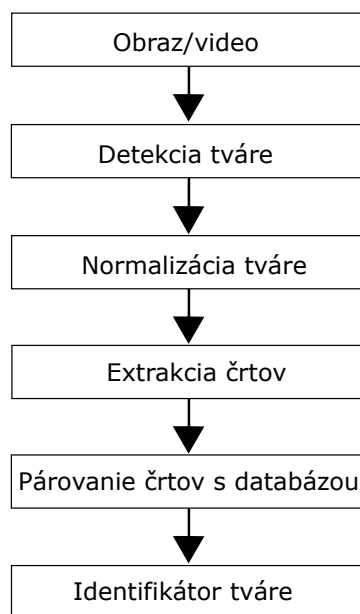
K -means algoritmus je pomerne jednoduchý na implementáciu a zároveň je jeho časová zložitosť definovaná ako $O(n)$, kde n je počet vzorov. Tým pádom je rýchlejší ako hierarchické zhlukovanie. Nevýhodou je citlivosť na zvolenie iníciaľných zhlukov a nemožnosť zobrazenia detailov zhlukovania (pri hierarchickom ich reprezentuje dendogram)[10][21].



Obr. 2.6: Segmentácia algoritmom k -means (vľavo originál, vpravo výsledok)[22].

2.2 Metódy na identifikáciu osôb pomocou tváre

Práca zaoberá spracovaním obrazu, na ktorom sa nachádza tvár, je možné použiť okrem všeobecných metód spracovania obrazu aj metódy používané pri identifikácii osôb pomocou tváre. Pokroky v automatizovanej analýze tváre, párovaní vzorov a počítačovom učení viedli k možnosti vytvorenia systémov automatického rozpoznania tváre. Proces rozpoznania tváre je pre človeka prirodzený, automaticky vykonávaný proces, avšak problém spočíva v zapamätaní si veľkého množstva neznámych tvárí. V oblasti počítačového videnia sa stále jedná o jeden z obtiažnych a zložitých procesov, no aj napriek tomu už existujú rôzne metódy, ktoré zvládajú identifikáciu tváre s využitím rozsiahlych databáz tvár. Tieto metódy nachádzajú svoje využitie napríklad v oblastiach zaoberajúcich sa bezpečnosťou. Systém vykonávajúci počítačovú identifikáciu tváre sa väčšinou skladá zo šiestich prvkov zobrazených na obrázku 2.7. Jedným z kľúčových problémov detekcie tváre je výkonnosť detekcie. Je potrebná extrakcia kľúčových črtov tváre a konštrukcia robustných klasifikátorov. To všetko s ohľadom na výkon. Týmto problémom sa zaoberá čoraz viac odborníkov[23].



Obr. 2.7: Typické prvky systému rozpoznávania tváre[23].

Existuje viacero prístupov a spôsobov akými je identifikácia tváre prevádzaná. Môžeme ich rozdeliť na:

- Metódy založené na vzhľade,
- Metódy založené na črtoch,
- Metódy založené na vzoroch,
- Metódy založené na častiach.

2.2.1 Metódy založené na vzhľade

Taktiež bývajú označované ako holistické. Tieto metódy získavajú značnú pozornosť od širokého spektra výskumníkov. Využívajú informácie z celej oblasti tváre, z ktorých pomocou vhodných transformácií získavajú kompaktnú reprezentáciu vhodnú pre identifikáciu. Medzi najznámejšie prístupy patria *Principal Component Analysis* (PCA)[24], *Linear Discriminant Analysis* (LDA)[25] alebo *Independent Component Analysis* (ICA)[26][27].

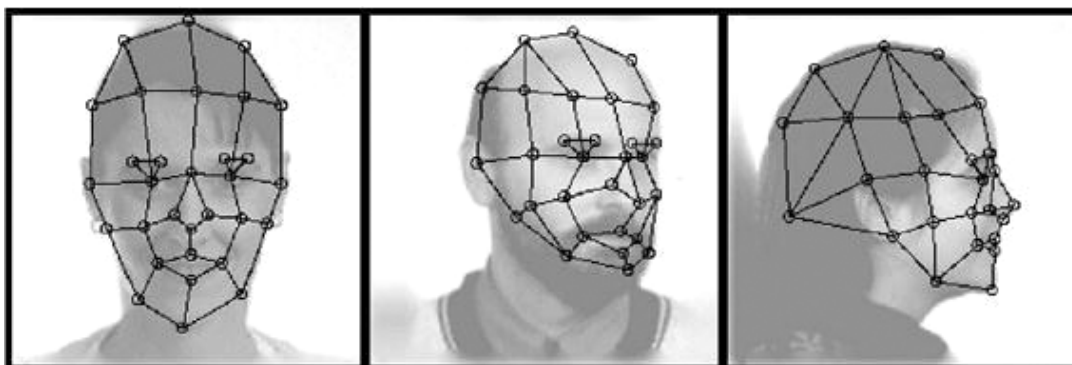
PCA a Eigenfaces

Eigenfaces je pomenovanie pre ortogonálnu množinu tvárí predstavujúcu základ, z ktorého je možné vytvoriť väčšinu tvárí. Jednotlivé tváre v tejto množine sú rozmanité zobrazenia, pričom každá jedna z nich vyzdvihuje určitú črtu tváre. Zostavenie eigenfaces je pomerne jednoduché. Je daný zoznam N tréningových obrazov tvárí. Najprv sa každý z obrazov prevedie na obraz konštantnej veľkosti (napr. 100x100). Následne sa každý z obrazov prevedie na vektor (10000x1). Na základe týchto N D -rozmerných vektorov môžeme aplikovať PCA aby sme získali vhodný základ s redukovanými rozmermi. Predpokladajme, že vyberieme M projekčnú základňu ($M \ll D$), každý D -rozmerný vektor sa môže transformovať na M -rozmerný. Týmto spôsobom sme schopní redukovať nadbytočné informácie v obraze tváre a zanechať iba tie najdôležitejšie, ktoré popisujú odlišujúce črty tváre. Po redukcii ostávajú iba vektory s najväčšími hodnotami. Týmto spôsobom sa z množiny N obrazov získava redukovaná množina, pričom každý jeden obraz tváre pôvodnej tréningovej množiny je možné vyjadriť vhodnou lineárnou kombináciou prvkou výslednej množiny. Čím je tréningová množina rôznorodejšia, tým viac tvárí je výsledná množina schopná reprezentovať. Výsledná množina sa používa pre identifikáciu. Dokázalo sa, že metóda PCA odstraňuje šum a odchýlky z tréningových dát, ale taktiež môže ignorovať niektoré kľúčové faktory[23].

2.2.2 Metódy založené na črtoch

Tieto metódy identifikácie tváre využívajú informácie získané z konkrétnych črtov tváre. Medzi tieto črty patria napríklad oči, nos, ústa, brada alebo aj obrys tváre. Na vyjadrenie vzťahov medzi črtami sa používajú jednoduché topologické grafy. Na rozpoznanie podobných tvárí z databáze sa používa jednoduchá deterministická schéma zhody. Jedným z algoritmov je aj Elastic Bunch Graph Matching algoritmus[27], ktorý rozoznáva tváre pomocou porovnávania vstupného grafu tváre s grafmi uloženými v databáze. Základom tohoto algoritmu je koncept uzlov, kedy každý uzol

vstupného grafu tváre reprezentuje určitý črt tváre. Tieto uzle sú následne prepojené tak, aby vytvorili datovú štruktúru grafu tváre (viď. obr.2.8)[27].



Obr. 2.8: Topologický graf tváre[27].

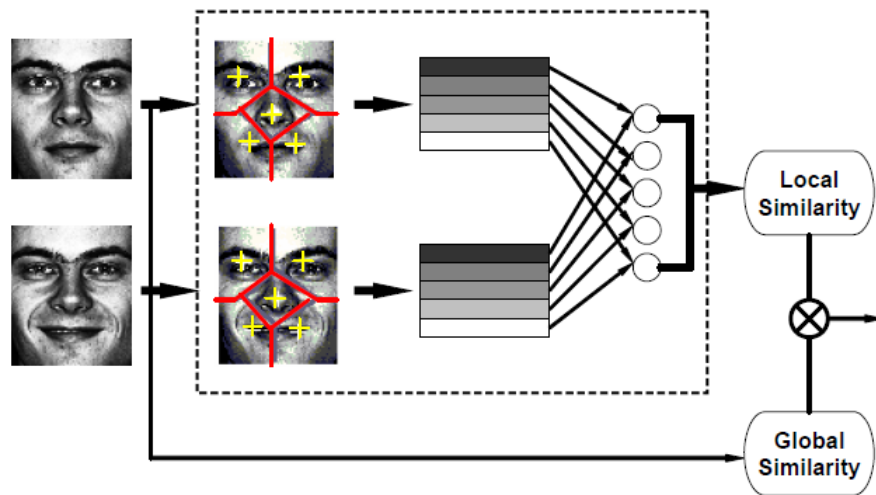
2.2.3 Metódy založené na vzoroch

Predchádzajúce dve metódy rozpoznávania prevádzajú zvyčajne extrakciu črtov všetkých obrázkov tvár uložených v databáze a trénujú klasifikátory alebo definujú metriku na výpočet podobnosti testovanej tváre s každou známou tvárou. Na prekonanie variácií tej istej tváre tieto metódy navyšujú svoju databázu známych tvárí aby vyhovovali oveľa väčšiemu počtu vzorkov a očakávajú, že tým zmenšia rozdiel v rámci jednej triedy a zväčšia rozdiel medzi odlišnými triedami. Na rozdiel od nich, tradičné metódy párovania vzorov sú založené na použití sady symbolických vzorov pre každú triedu (osobu). Porovnanie podobnosti je vyrátané medzi testovaným obrázkom tváre a každou triedou. Trieda s najväčšou podobnosťou je zvolená ako výsledná identifikovaná tvár[28].

2.2.4 Metódy založené na častiach

Tieto metódy sú založené na častiach tváre, ktoré sú značne odolné voči rozdielnosti v osvetlení alebo póze. Na rozdiel od metód založených na črtoch, tieto metódy detekujú v obraze významné časti tváre a kombinujú vzhľad týchto častí s nástrojmi strojového učenia určenými na identifikáciu tváre. Jednou z týchto metód je aj metóda založená na SIFT (scale-invariant feature transform)[29] črtách získaných z obrazu tváre. SIFT je široko a úspešne používaná metóda v detekcii a rozpoznávaní objektov. Skladá sa z dvoch funkcií: detektoru bodu záujmu a deskriptoru oblasti. Na detekciu týchto bodov sa dá použiť algoritmus difference-of-Gaussian (DOG), a deskriptory môžu byť generované na základe orientácie a gradientu vyrátaného

v špecifickej oblasti okolo každého bodu záujmu. Počet jednotlivých črtov získaných pomocou SIFT sa líši od osoby a aj jej výrazu. Aby sa porovnali iba páry črtov s podobnými fyzikálnym významom je každý obraz tváre rozdelený na rovnaký počet podoblastí. Následne sa počíta podobnosť dvojíc jednotlivých podoblastí na základe SIFT črtov vo vnútri oblasti a na koniec sa získavajú priemerné hodnoty podobnosti. Nájdenie jednotlivých podoblastí je realizované pomocou K -means zhlukovacieho algoritmu. Konštrukcia piatich podoblastí je zobrazená na obrázku 2.9. Na obrázku je vidieť, že stredy oblastí vyznačené krížikmi sa nachádzajú na očiach, nose a krajoch úst čo súhlasí s myšlienkou identifikácie tváre, nakoľko sa jedná o oblasti najviac charakterizujúce tvár. Na základe zostavených podoblastí nastáva lokálne a globálne kombinované porovnanie pre rozpoznanie tváre, taktiež zobrazené na tomto obrázku[28].



Obr. 2.9: Konštrukcia podoblastí a výpočet podobnosti metódy používajúcej SIFT črty[28].

3 Umelá inteligencia

Umelá inteligencia je odvetvie počítačovej vedy, ktorej cieľom je vytvoriť stroje a algoritmy, ktorých výpočet sa dá označiť ako inteligentný. V posledných rokoch sa stáva neoddeliteľnou súčasťou technického priemyslu. Medzi hlavné zamerania umelej inteligencie patrí tvorba takých počítačov, ktoré sú schopné učiť sa, riešiť problémy, vnímať okolie, rozhodovať alebo aj manipulovať predmetmi tak, ako ľudia. Jadrom umelej inteligencie je aj znalostné inžinierstvo, nakoľko stroje môžu často konať a reagovať ako ľudia len ak majú dostatok informácií týkajúcich sa daného problému. Aby bola umelá inteligencia výkonná a s čo najmenšou chybovosťou, potrebuje mať informácie o objektoch, kategóriách, vlastnostiach a vzťahoch medzi všetkými dátami, s ktorými pracuje[30].

3.1 Typy strojového učenia

Tvorba inteligencie, rozhodovacích schopností a schopnosti riešiť problémy za pomoci stroju je zdĺhavý proces, ktorého dôležitou súčasťou je aj strojové učenie. Tento proces slúži na naučenie modelu inteligentne a s čo najmenšou chybovosťou riešiť požadovaný problém. Učenie sa delí na učenie bez učiteľa a učenie s učiteľom[30].

3.1.1 Učenie bez učiteľa

Pri učení bez učiteľa má umelá inteligencia k dispozícii len vstupné dáta bez im odpovedajúcich výstupných dát. Cieľom takéhoto procesu je modelovanie štruktúry a vzťahov medzi vstupnými dátami tak, aby sa na ich základe dalo zistiť viac informácií o vstupných dátach. Pri týchto algoritmoch neexistuje správna výstupná hodnota. Dajú sa rozdeliť na klastrovacie a asociačné. Klastrovacie algoritmy riešia problém zoskupovania dát na základe ich vlastností. Asociačné algoritmy sa využívajú v prípadoch, kedy sa hľadajú určité pravidlá popisujúce skupiny dát. Príkladom klastrovania je zoskupovanie zákazníkov na základe nákupného správania. Pri asociácií je to napríklad nájdenie pravidla, ktoré hovorí o tom, že zákazník, ktorý si kúpil X si pravdepodobne kúpi aj Y [31].

3.1.2 Učenie s učiteľom

Pod túto kategóriu spadá väčšina strojového učenia. Toto učenie je také, kde sú vstupné premenné X a im priradené výstupné premenné Y . Úlohou algoritmu je naučiť sa mapovaciu funkciu, pomocou ktorej sa prevedie vstup na jemu odpovedajúcu výstupnú hodnotu. Cieľom je aproximovať funkciu mapovania do takej miery,

že pre nové vstupné hodnoty bude schopná správne predpovedať im odpovedajúce výstupné hodnoty s minimálnou mierou chybovosti. Problémy spadajúce pod učenie s učiteľom sa dajú rozdeliť na problémy regresné a klasifikačné[31].

Dataset

Dataset je množina dát, na ktorých sa algoritmus učí. V týchto dátach hľadá vzory a súvislosti, ktoré zohľadní vo výslednej namodelovanej aproximačnej funkcii. Dataset je vždy tvorený z dvoch skupín prvkov: vlastností a značiek. Vlastnosti označujú hodnoty, na základe ktorých sa odhaduje výsledná značka. Na základe vlastností by mal byť algoritmus schopný nájsť podobnosti alebo rozdiely, podľa ktorých bude rozhodovať. Značky popisujú dáta, ktoré by mal byť algoritmus schopný z daných vlastností odhadnúť. Príklad môže byť klasifikácia ovocia, kedy vlastnosti by boli tvar, farba, veľkosť a odhadovaná značka by bola druh ovocia[32].

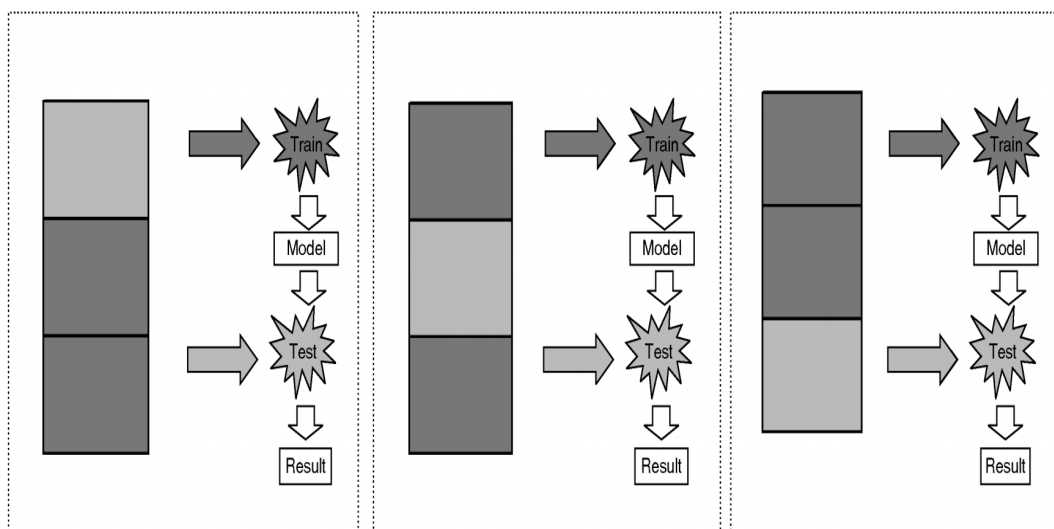
Problémy tréovania

Medzi časté problémy učenia algoritmov patrí tzv. pretrénovanie alebo podtrénovanie. Pri pretrénovaní sa natrénovaný model naučil odhadovať výslednú veličinu na základe vstupných vlastností až príliš detailne. To znamená, že natrénovaný model sa naučí predikovať aj šum alebo náhodné odchýlky dát. To degraduje jeho schopnosť zovšeobecňovať svoje poznatky a predikcie. Pretrénovanie vzniká pri veľkom množstve vstupných vlastností alebo pri nedostatočnom množstve dát pre tréovanie modelu. Podtrénovanie predstavuje opak pretrénovania. V tomto prípade modelu unikajú podstatné spojenia a vzťahy medzi vstupnými dátami. Takto naučený model nemá schopnosť učiť sa a dochádza k ignorovaniu vstupných dát. Odhadovaná hodnota pomocou takéhoto modelu nezávisí od naučených dát[32].

Validácia naučeného modelu

Algoritmy učenia s učiteľom sa skladajú z dvoch fáz. Prvá fáza je tréovacia, kedy sa trénuje model s pomocou tréovacích dát. Druhou fázou je fáza validácie, pri ktorej sa už natrénovaný model validuje na vlidačných dátach, pričom validačné dáta nemohli byť použité pri tréovaní modelu. Tréovacie a validačné dáta sa vytvárajú rozdelením datasetu v určitom pomere, kedy väčšia časť dát je určená na tréovanie a menšia na validáciu. Natrénovaný model je závislý na tom, aké dáta boli použité pre jeho tréovanie a validovanie. Z tohoto dôvodu sa pre komplexnú validáciu modelu odporúča previesť tréovanie a validáciu viac krát, vždy s rozdielnymi tréovacími a validačnými dátami tak, aby každé vstupné dáta boli využité ako pre tréovanie tak pre validáciu. Príkladom takejto validácie je krížová validácia.

Jednou z implementácií krížovej validácie je K -fold krížová validácia. Je založená na tom, že dáta sa najprv rozdelia na k rovnako veľkých segmentov. Následne sa vykoná k iterácií tréovania a validácie, pričom pri každej iterácii sa na validáciu použije rozdielny z k segmentov. Na tréovanie sa v rámci iterácie použije ostávajúcich $k - 1$ segmentov dát. Týmto spôsobom sa pri každej iterácii získa validačná metrika. Po skončení iterácií máme k hodnôt validačných metrik, z ktorých sa dá vyrátať priemerná hodnota určujúca priemernú úspešnosť modelu pre daný dataset. Proces K -fold segmentácie pre $k = 3$ je zobrazený na obrázku 3.1. Pre získanie spoľahlivých výsledkov úspešnosti modelu je potrebné previesť čo najväčšie množstvo iterácií tréovania a následnej validácie na rozdielnych dátach. Pri K -fold krížovej validácii sa vykoná iterácií len k . Z tohoto dôvodu sa v praxi využíva viacnásobný výpočet K -fold validácie, pričom pred každým jedným novým výpočtom sa dáta preusporiadajú do iného poradia[33].

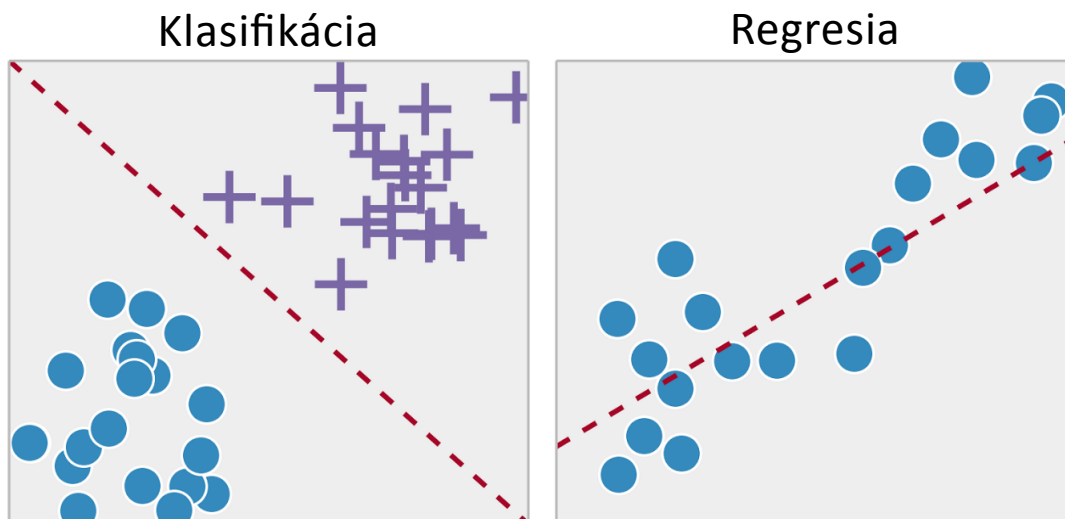


Obr. 3.1: Kroky K -fold krížovej validácie[33].

Ďalším typom krížovej validácie je validácia s názvom „Leave-One-Out“. V prípade datasetu o veľkosti x vstupných vektorov vlastností sa pri tejto validácii vykoná x iterácií tak, aby bol každý jeden vektor vlastností použitý pre validáciu. Všetky ostatné vektory sú v danej iterácii použité ako tréovacie dáta. Táto metóda zodpovedá K -fold validácii s K zvoleným ako x . Metóda je zdĺhavejšia ako K -fold, ale všeobecne je považovaná za spoľahlivejšiu a v prípade malého datasetu využíva pre tréovanie modelu väčšie množstvo dát[34].

3.2 Klasifikácia a Regresia

Táto podkapitola popisuje rozdiel medzi klasifikáciou a regresiou. Tento rozdiel je graficky zobrazený na obrázku 3.2.



Obr. 3.2: Grafické znázornenie klasifikácie a regresie[35].

3.2.1 Klasifikácia

Klasifikácia rieši úlohy, pri ktorých je potrebné dátam na základe vstupných hodnôt X priradiť diskretnú výstupnú hodnotu Y . Ide teda o aproximovanie mapovacej funkcie, ktorá dokáže na základe vstupných dát rozpoznať kategóriu, pod ktorú dané dáta spadajú a správne im ju priradiť. Vstupné dáta musia byť klasifikované do minimálne dvoch kategórií. Problém s dvoma kategóriami sa nazýva binárny, problém s viacerými triedami sa označuje ako viacstupňový klasifikačný problém[36].

Najčastejšie používanou validačnou metrikou pri klasifikácii je percento správne klasifikovaných dát zo všetkých dát využitých pri validácii natrénovaného modelu.

3.2.2 Regresia

Regresia rieši úlohy, pri ktorých je potrebné dátam na základe vstupných hodnôt X priradiť kontinuálnu výstupnú hodnotu Y . Schopnosť odhadu kontinuálnej výstupnej hodnoty je zabezpečená formou natrénovanej aproximačnej funkcie. Pri trénovaní modelu by mal byť vytvorený taký model, ktorého funkcia čo najpresnejšie aproximuje vzťahy medzi vstupnými a výstupnými hodnotami. Čiže po zakreslení

funkcie a bodov vytvorených vzťahom medzi vstupnými a výstupnými dátami do grafu by mala v ideálnom prípade funkcia pretínať všetky zakreslené body. Výstupná premenná s kontinuálnym priebehom je reálna hodnota, ako napríklad celé číslo alebo číslo s desatinnou čiarkou[36].

Validačné metriky

Nakoľko regresia odhaduje kontinuálnu hodnotu, úspešnosť regresného modelu sa musí merať ako chyba jednotlivých validačných predikcií. Existuje mnoho metrík na meranie chyby, ako napríklad *Mean absolute error* (MAE), *Mean absolute percentage error* (MAPE) alebo aj *Root mean squared error* (RMSE). MAE označuje strednú absolútnu chybu, MAPE označuje strednú absolútnu percentuálnu chybu a RMSE označuje strednú kvadratickú chybu. Ich výpočet je definovaný rovnicami:

$$\text{MAE} = \frac{1}{n} \sum_{j=1}^n |y_j - \widehat{y}_j|, \quad (3.1)$$

$$\text{MAPE} = \frac{100\%}{n} \sum_{j=1}^n \left| \frac{y_j - \widehat{y}_j}{y_j} \right|, \quad (3.2)$$

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{j=1}^n (y_j - \widehat{y}_j)^2}, \quad (3.3)$$

kde n je počet výpočtov, y_j predstavuje očakávanú hodnotu a \widehat{y}_j predstavuje reálne vypočítanú hodnotu[37][38]. Ďalšou z často používaných metrík na meranie výslednej presnosti regresného modelu je hodnota R na druhú (R^2). Táto metrika je meradlom toho, ako blízko sú odhadované údaje k čiare reprezentujúcej natrénovanú aproximačnú funkciu. Jedná sa o percentuálny podiel variácie dát zachytenej aproximačnou funkciou k absolútnej variácii dát. Hodnota (R^2) je vždy v rozmedzí 0–100%, pričom 0% znamená, že model nevystihuje žiadnu variabilitu dát a naopak 100% je hodnota popisujúca model, ktorý vystihuje všetkú variabilitu dát[39].

4 Parametrizácia tváre osoby na videu

Cielom tejto časti práce bolo vytvorenie programu, ktorý dokáže z videa zachytávajúceho hovoriacu osobu vytvoriť časovú radu dát popisujúcich kľúčové črty tváre osoby v čase. Dáta tejto rady by mali byť vhodne štrukturované pre nasledujúce spracovanie či už pomocou umelej inteligencie, ktorá z dát môže odhadovať štádium PCH u osoby sledovanej na videu, alebo pomocou vizualizácie na hľadanie korelácií medzi parametrami, prípadne konvertovanie a výber jednotlivých črtov. Pri návrhu a implementácii riešenia boli zvolené dva prístupy: jeden pomocou segmentačných algoritmov a druhý pomocou metód identifikovania osoby pomocou tváre. V nasledujúcich dvoch podkapitolách sú detailne rozobrané obe riešenia, ich postup a aj dosiahnutá úspešnosť daného riešenia.

4.1 Riešenie s využitím segmentácie obrazu

Implementácia tohoto riešenia je prevedená v programovacom jazyku JAVA, s využitím knižnice OpenCV[40]. Knižnica OpenCV poskytuje množstvo nástrojov pre počítačovú úpravu a spracovanie obrazu. Prvotným cieľom riešenia bola implementácia programu, ktorý dokáže v jednotlivých snímkach videa rozprávajúcej osoby zaznamenať plochu pier. Z tejto plochy by sa následne dala dopočítať šírka a výška úst a ich zmena v čase. Toto riešenie sa ukázalo ako nedostatočné. Implementované riešenie sa dá rozdeliť na **fázu inicializácie** a **cyklus výpočtu plochy úst** v snímkoch, ktorý sa skladá z nasledujúcich po sebe idúcich krokov:

1. detekcia úst,
2. detekcia hrán,
3. segmentácia úst.

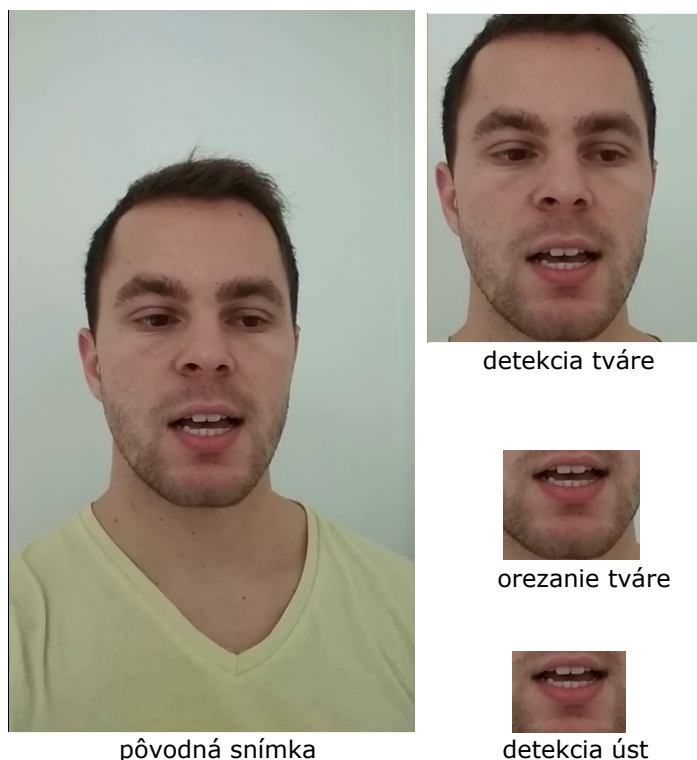
Tieto kroky sa vykonávajú sekvenčne v danom poradí pre každý jeden snímok videa.

4.1.1 Fáza inicializácie

Jedná sa o prvý krok programu, v ktorom dochádza k inicializácii grafického rozhrania a otvoreniu súboru videa z disku. Užívateľovi je zobrazený prvý snímok videa a čaká sa na jeho výber farieb, ktoré budú neskôr použité pre výpočet zrna segmentácie. Voľba je prevádzaná klikaním kurzoru do zvoleného obrázku a jedná sa najmä o farby pier. Zvolenie farieb užívateľ potvrdí tlačítkom. V tomto bode programu sa začne vykonávať cyklus, ktorý sekvenčne prechádza jednotlivé snímky videa a počíta plochu úst.

4.1.2 Detekcia úst

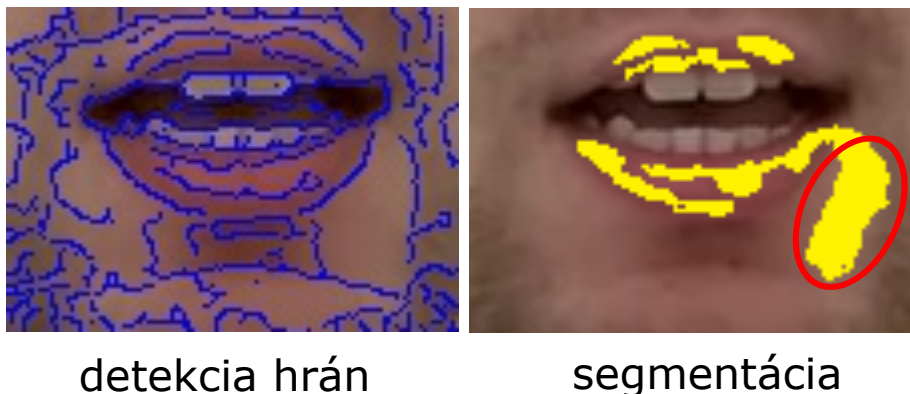
Detekcia úst sa skladá z niekoľkých krokov. Snímka je najprv pomocou funkcie knižnice OpenCV prevedená do stupnice šedej. Následne je v obraze prevedená detekcia tváre s využitím OpenCV kaskádových klasifikátorov HAAR[41]. Najprv je detekovaná tvár a v nej sú následne detekované oči. Týmto spôsobom sa dá do istej miery predísť falošným detekciám. V prípade, že sa tvár nepodari zdetekovať, iterácia cyklu sa končí a prechádza na nasledujúcu snímku videa. Ak tvár detekovaná bola, pokračuje sa v jej ďalšom spracovaní. Z obrazu tváre sa odrežú vrchné dve tretiny a orezaním jednej šestiny šírky z oboch bokov obrazu sa zmenší jeho šírka na šesť osmín pôvodnej šírky. Touto úpravou získame iba spodnú časť tváre a vyhneme sa falošnej detekcii oblasti úst, ktorá bola v niektorých prípadoch detekovaná v oblasti nosu alebo očí. Na orezanom obrázku spodnej časti úst následne znovu aplikujeme detekciu pomocou HAAR kaskádového klasifikátora, ale tentokrát využívame klasifikátor trénovaný na detekciu úst. Ak je detekcia neúspešná cyklus pokračuje nasledujúcou iteráciou, v prípade úspešnej detekcie je oblasť detekovaných úst zväčšená o 10% šírky do strán a o 25% výšky nahor a nadol. Toto zväčšenie je potrebné z dôvodu častej falošnej detekcie jednej z pier namiesto úst. Výsledky krokov detekcie úst sú zobrazené na obrázku 4.1.



Obr. 4.1: Kroky detekcie úst.

4.1.3 Detekcia hrán

V obraze detekovaných úst je aplikovaný Canny-edge algoritmus[17] na detekciu hrán. Detekcia hrán je vykonávaná na obrázku úst prevedenom do stupnice šedej. Funkcia, ktorá prevádza detekciu hrán prijíma ako argument obrázok a prah, ktorý je použitý pri detekcii hrán. Obrázok je najprv zaostrený pomocou funkcie knižnice OpenCV, a následne je funkciou tej istej knižnice vykonaná detekcia hrán. Funkcia na konci vracia objekt typu `Mat`, ktorý reprezentuje obrázok po detekcii hrán v ktorom sú jednotlivé hrany zobrazené bielou farbou a zvyšok obrazu tvoria čierne pixely. Nenulové pixely definujúce hrany sú následne pomocou funkcie `copyNonBlackPixels(...)` prekopírované do farebného obrázku, v ktorom sú vyznačené modrou farbou. Pomocou detekcie hrán sme schopný oddeliť jednotlivé oblasti a zamedziť segmentačnému algoritmu v spojení týchto oblastí, avšak len za predpokladu, že sú hrany definujúce jednotlivé oblasti spojitý. Na obrázku 4.2 je ukážka detekovaných hrán zakreslených v obrázku úst. Z obrázku je patrné, že hrany vo väčšine prípadov netvoria spojitý oblasti, ktoré by výrazne zlepšili funkcionality segmentačného algoritmu.



Obr. 4.2: Výsledok detekcie hrán a segmentácie.

4.1.4 Segmentácia úst

V obrázku úst so zaznačenými hranami je v posledom kroku prevádzaná samotná segmentácia. Pre segmentáciu je využitý algoritmus na základe oblasti, ktorý používa inicializačné zrno, z ktorého sa následne na základe prahu tvorí spojitá oblasť. V inicializačnej fáze užívateľ vybral farby, na základe ktorých je počítané zrno v jednotlivých obrázkoch detekovaných úst. Tie sú vypočítané pomocou funkcie `getFloodPoints(...)`. Argumenty tejto metódy sú obrázok detekovaných úst, počet bodov zrna, ktoré majú byť nájdené, minimálna vzdialenosť bodov zrna, maxi-

málny tolerovaný rozdiel medzi farbou pixelu a farbou zvolenou užívateľom aby sa pixel označil ako zrno a farby zvolené užívateľom. Z popisu argumentov sú patrné podmienky pre pridanie pixelu do zrna:

- vzdialenosť pixelu od každého pixelu tvoriaceho zrno musí byť väčšia alebo rovná hodnote zadanej argumentom,
- chyba pixelu¹ musí byť menšia ako maximálna hodnota chyby zadaná argumentom.

Táto metóda prechádza všetky pixely obrázku a v prípade, že pixel spĺňa podmienky zadané argumentami je pridaný do zoznamu zrna. Zrno je predstavované objektom typu `FloodPoint`, ktorý si nesie informáciu o svojej chybe. V prípade, že zoznam obsahuje maximálny počet daný argumentom, pixel je do zrna pridaný len za podmienky, že je jeho chyba menšia ako chyba niektorého z pixelov zrna. Pixel s väčšou chybou je zo zrna odstránený a do zrna sa pridá nový pixel s menšou chybou.

Po výpočte zrna prichádza na radu aplikovanie samotného segmentačného algoritmu metódou `floodFillCustom(...)`. Táto metóda na začiatku pomocou metódy `getFloodPoints(...)` získa body zrna, a následne ráta pre každý z týchto bodov jeho oblasť. Výpočet oblasti je prevádzaný na základe popisu algoritmu v podsekcii 2.1.2. Rozdiel je v tom, že jednotlivé body zrna netvorí odlišné oblasti, ale všetky oblasti sa spájajú do jednej popisujúcej oblasti pier. Metóda vracia čiernobiely obrázok, v ktorom je bielu farbu označená plocha pier. Táto plocha je následne pre ukážku prekopírovaná žltou farbou do obrázku detekovaných úst. Výsledný obrázok úst so segmentáciou je zobrazený na obrázku 4.2. Na obrázku je vyznačená aj chyba segmentácie.

4.1.5 Úspešnosť riešenia

Z obrázku 4.2 je patrné, že výsledok dosiahnutý touto metódou je nedostatočný. Samotná detekcia úst bola validovaná na piatich videách zachytávajúcich osoby rôzneho pohlavia a rasových skupín s celkovým počtom snímkov 2389. V týchto snímkoch bola úspešnosť detekcie tváre 95%. Úspešnosť detekcie úst bola približne 86%. Väčší problém nastal pri detekcii hrán a pri segmentácii. V prípadoch kedy bolo video nekvalitné, alebo boli časti úst rozmazané (napríklad pri rýchlejšom pohybe úst) dochádzalo k označeniu segmentovanej oblasti aj mimo oblasti pier. V týchto prípadoch si boli pixely patriace perám a patriace oblastiam mimo pier podobné. V oblastiach rozmazania nebola detekovaná hrana, ktorá by zabránila segmentačnému algoritmu v prejdení mimo oblasti pier. Segmentačný algoritmus následne z dôvodu podobnosti pixelov v danej oblasti rozšíril segmentovanú oblasť aj mimo

¹Chyba pixelu v tomto prípade označuje absolútnu hodnotu súčtu rozdielov červenej, zelenej a modrej farby medzi pixelom a najpodobnejšou farbou zvolenou užívateľom.

oblasti pier. Ďalším faktorom bola farebnosť pier – v prípade kedy bola farba pier odlišná od farby pokožky, bola úspešnosť segmentácie vyššia. Naopak v prípade podobnosti farby pier a pokožky dochádzalo k častej chybnnej segmentácii. Ďalším z problémov bol výber zrna, ktorý bol prevádzaný v prvom kroku – v prípade, že sa zmenili svetelné podmienky v priebehu videa, voľba zrna bola značne ovplyvnená. Tento problém by sa však pravdepodobne dal vyriešiť rátaním zrna jednotlivu pre každú snímku úst napríklad na základe histogramu. Nedostatok tohoto riešenia spočíva aj v tom, že riešenie je zamerané iba na detekciu plochy pier. Pridanie ďalších sledovaných črtov je v tomto prípade zložité.

4.2 Využitím metód na identifikáciu tváre

Implementácia s využitím týchto metód sa ukázala ako vhodná a funkčná, a zároveň je program schopný sledovať viac črtov tváre. Riešenie je realizované pomocou skriptu implementovaného v programovacom jazyku Python. Pri riešení je využitých viacero externých knižníc, avšak knižnica, pomocou ktorej je prevádzané samotné detekovanie bodov tváre, je knižnica `face_recognition` založená na implementácii `facenet`[42]. Myšlienka tohoto riešenia spočíva v sledovaní viacerých parametrov tváre, ako napríklad šírka a výška úst, šírka a výška tváre alebo aj vzdialenosť medzi nosom a bradou. Zároveň sa medzi parametre ukladá aj odhad emócie z tváre, ktorý je bližšie popísaný v sekcii 4.2.2. Tieto parametre sú zaznamenané v každej snímke videa rozprávajúcej osoby. Výstupom skriptu sú dáta vo formáte JSON² uložené v súbore zadanom argumentom. Proces skriptu sa dá rozdeliť na nasledujúce kroky:

1. spracovanie argumentov a otvorenie súborov,
2. cyklus detekcie parametrov tváre v snímkach,
3. zápis spracovaných dát.

Vykonávanie týchto krokov je riadené pomocou hlavnej metódy skriptu, ktorej telo je zobrazené na výpisku 4.1.

```
1 def main():
2     path_to_weights = 'model152p.hdf5'
3     emotions_model = get_emotions_model(path_to_weights)
4
5     args = parse_arguments()
6     video = open_video(args.i)
7     output_file = open_output(args.o)
8     json_data = segmentate_video(video, args.v, args.g, emotions_model,
9                                 output_file)
9     output_file.seek(0);
10    output_file.truncate();
11    output_file.write(json_data)
```

²JSON (JavaScript Object Notation) je jedným z často využívaných formátov na výmenu a formátovanie dát. Využíva človekom čitateľný text, ktorý sa skladá z dvojíc kľúč-hodnota[43].

```
12     output_file.close();
13     sys.exit(0)
```

Výpis 4.1: Telo hlavnej metódy Python skriptu.

4.2.1 Spracovanie argumentov a otvorenie súborov

V prvej fáze skript najprv načítava a overuje existenciu súborov na disku a spracováva argumenty zadané užívateľom. Súbor, ktorý sú načítavané sú súbor s videom osoby, súbor s definovanými váhami pre odhad emócií z tváre[44] a súbor pre výstupné dáta skriptu. V prípade, že súbor s videom alebo s definovanými váhami neexistuje, skript vypíše chybovú hlášku a ukončí sa. Argumenty, ktoré skript prijíma sú nasledovné:

- -i cesta k súboru s videom,
- -w cesta k súboru s váhami pre odhad emócií,
- -o cesta k súboru, do ktorého bude uložený výstup,
- -v zvýšenie verbozity skriptu,
- -g zapnutie grafického výstupu pre ukážku a ladenie.

Prvé dva z argumentov sú povinné a bez ich definovania sa skript nepustí. Zvyšné dva argumenty sú voliteľné.

4.2.2 Cyklus detekcie parametrov tváre v snímkach

Po otvorení súborov a spracovaní argumentov nastáva cyklus spracovania jednotlivých snímkov videa. Ten je naimplementovaný pomocou metódy `segmentate_video(...)` (viď. výpisok 4.2)

```
1  def segmentate_video(video, verbose, graphical, emotions_model,
2      output_file):
3      i = 1
4      mouth_count = 0
5      video_data = {'file_name': video.filename, 'frames_count': '?',
6                  'frames': []}
7
8      for image in video:
9          frame_data = segmentate_face(image, graphical, emotions_model)
10         if not (frame_data is None):
11             frame_data['frame_index'] = str(i)
12             video_data['frames'].append(frame_data)
13             mouth_count = mouth_count + 1
14
15         #write data into output after every 10th frame
16         if (i % 10 == 0):
17             video_data['frames_count'] = str(i)
18             output_file.seek(0);
19             output_file.truncate();
20             output_file.write(json.dumps({'video': video_data}))
```

```

19         if verbose:
20             print('processed frames: ' + str(i))
21         i = i + 1
22
23     video_data['frames_count'] = str(i - 1)
24     video_data = json.dumps({'video': video_data})
25     if verbose:
26         print('Processed frames: ' + str(i))
27         print('Mouths segmentated: ' + str(mouth_count))
28         print(video_data)
29
30     return video_data

```

Výpis 4.2: Telo metódy `segmentate_video(...)`.

Táto metóda najprv nainicializuje štruktúru výsledného JSON-u a následne sekvencne iteruje všetkými snímkami videa. Každá jedna zo snímkov je spracovaná ďalšou metódou, ktorej výstup je objekt typu JSON. Tento objekt je následne pridaný spolu s číslom snímku do prvej nainicializovanej štruktúry. Pri každom desiatom spracovanom snímku sa dáta zapisujú do výstupného súboru, čo slúži pre uloženie dát v prípade prerušenia skriptu. Funkcia, ktorá prevádza spracovanie jednotlivých snímkov má pomenovanie `segmentate_face(...)`.

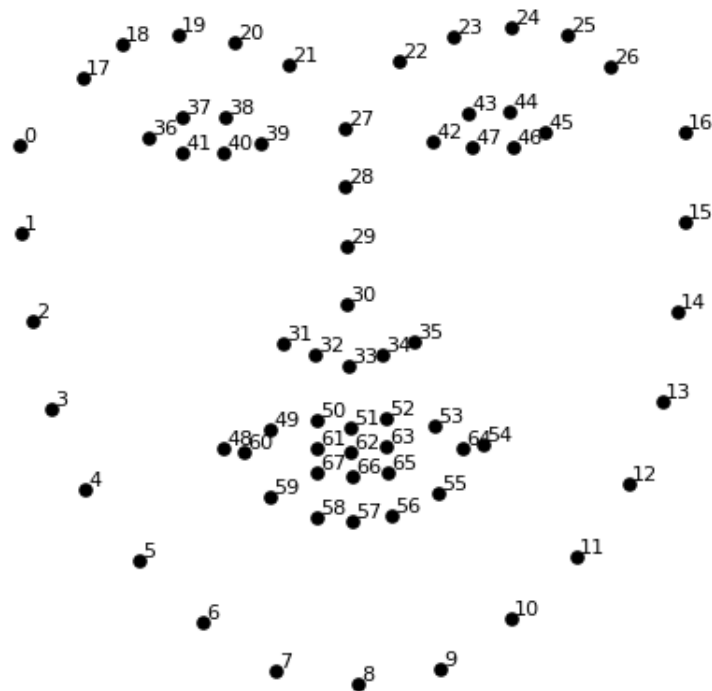
Metóda `segmentate_face(...)`

Funkcionalita tejto metódy sa dá rozdeliť na dve rôzne spracovania tváre. Prvou je odhad emócií a druhou je detekcia črtov tváre.

Odhad emócií je vykonávaný na základe dát pomenovaných ako `fer2013`[44]. Jedná sa o dáta skladajúce sa z 35887 obrázkov tvárí v stupnici šedej o rozmeroch 48x48. Každý z obrázkov je označený emóciou, ktorú tvár na danom obrázku vystihuje. Celkovo sa jedná o sedem typov emócií: hnev, znechutenie, strach, radosť, smútok, prekvapenie a neutrálna. Snímka je pred odhadom emócie transformovaná do rozmerov 48x48 pixelov a následne je na nej prevedený odhad na základe modelu. Výsledkom odhadu je vektor siedmych číselných hodnôt, ktorých súčet je rovný jednej. Hodnoty predstavujú percentuálne zastúpenie každej jednej odhadovanej emócie v snímke, pričom poradie čísla v zozname udáva o akú emóciu sa jedná. Emócie sú v zozname zoradené v zhodnom poradí ako sú vypísané v tomto odseku. Na výpisku 4.3 sú zobrazené dáta snímky tváre, v ktorej bola odhadnutá na 93% emócia prekvapenia, 6% strach, 1% smútok atď.

Pri detekcii črtov tváre sú najprv s využitím metódy `face_landmarks(...)` knižnice `face_recognition` získané body popisujúce tvár. Tieto body sú zobrazené na obrázku 4.3. Body 0–16 opisujú sánku, 17–21 ľavé obočie, 22–26 pravé obočie, 27–35 nos, 36–41 ľavé oko, 42–47 pravé oko, 48–54 a 60, 61, 62, 63, 64 hornú peru

a body 55–59 a 60, 64, 65, 66, 67 dolnú peru tváre. Metóda vracia tieto body v objekte typu mapy–kľúč je textové pomenovanie oblasti (napr. „top_lip“ pre hornú peru) a hodnota je zoznam bodov. Z týchto bodov sú následne pomocou matema-



Obr. 4.3: Body popisujúce tvár získané metódou `face_landmarks(...)` [45].

tických operácií vyrátané výsledné hodnoty, ktoré sú následne naformátované do objektu typu JSON popisujúceho tvár na aktuálnom snímku. Všetky vyrátané hodnoty popisujúce črty tváre aj s ukážkou výsledného formátu JSON-u sú popísané v kapitole 4.3. V prípade zapnutého grafického výstupu táto metóda ešte vykreslí jednotlivé detekované črty do aktuálneho snímku a zobrazí ho užívateľovi. Ukážka je na obrázku 4.4, taktiež v nasledujúcej kapitole.

4.3 Výsledky

Výsledkom tejto časti práce práce sú dáta popisujúce črty a emócie tváre zobrazenej na každom snímku spracovaného videa v sekvenčnom pradí. Formát dát je zobrazený na výpisku 4.3.

```
{
  "video":{
    "file_name": "videos\\video.avi",
```

```

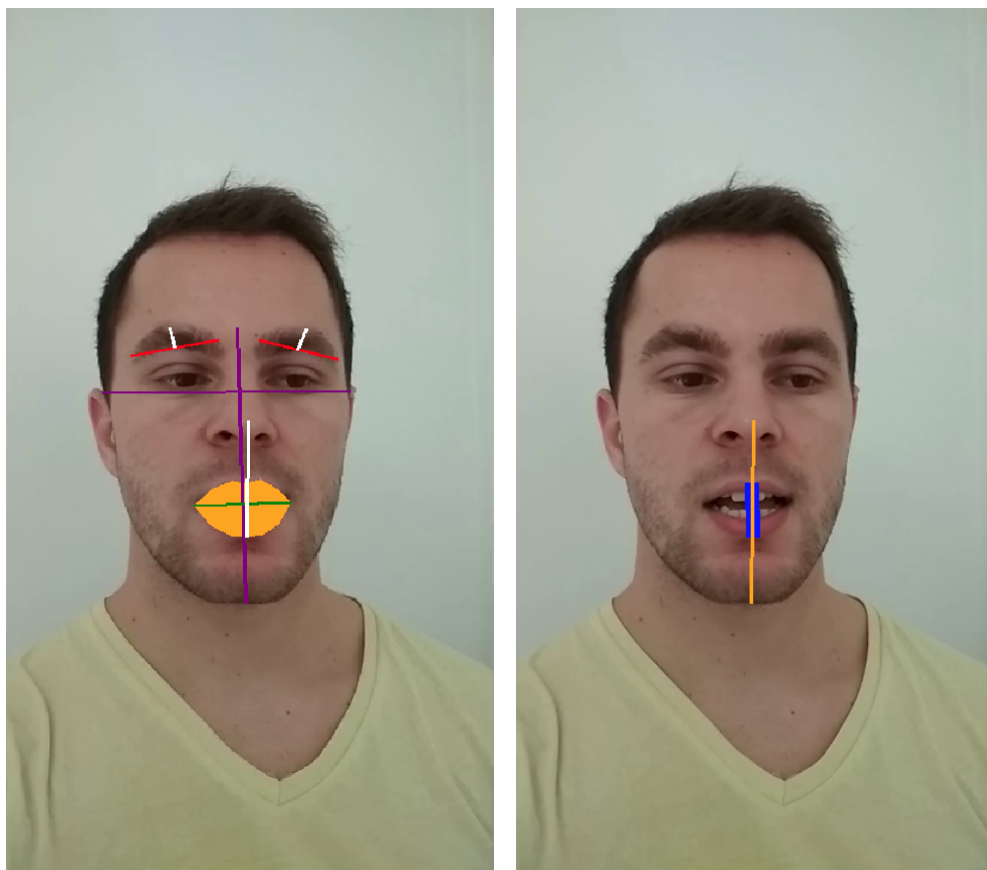
    "frames_count": "1",
    "frames": [
      {
        "frame_index": "1",
        "mouth_area": "2647.500",
        "mouth_width": "80.056",
        "mouth_height": "45.011",
        "face_width": "205.010",
        "face_height": "228.592",
        "left_eyebrow_width": "73.164",
        "left_eyebrow_height": "16.978",
        "right_eyebrow_width": "66.940",
        "right_eyebrow_height": "18.200",
        "dist_nose_mouth": "97.005",
        "dist_nose_chin": "152.013",
        "emotions": "{0.0026075528, 9.9731415e-06,
                     0.058610495, 4.5851215e-05, 0.013636396,
                     0.9250778, 1.2037376e-05}"
      }
    ]
  }
}

```

Výpis 4.3: Ukážka JSON formátu výsledných dát (pre 1 snímku).

Z formátu dát môžeme vidieť, že prvým kľúčom je kľúč „video“. Jemu náležiaci hodnota je objekt popisujúci video. Ten obsahuje popis názvu videa, počet celkových snímok videa, a pole objektov popisujúcich jednotlivé snímky. Každá jedna snímka je popísaná svojím číslom v poradí videa, a črtami detekovanými z tváre osoby. Tieto črty sú: plocha úst, šírka úst, výška úst, šírka tváre, výška tváre, šírka ľavého obočia, výška ľavého obočia, šírka pravého obočia, výška pravého obočia, vzdialenosť nosu a úst, vzdialenosť nosu a brady a hodnoty odhadu emócií. Na obrázku 4.4 sú rôznymi farbami zakreslené jednotlivé detekované črty – z dôvodu prekryvania sú zakreslené do dvoch obrázkov.

Toto riešenie sa ukázalo ako vhodné z viacerých dôvodov. Úspešnosť tohoto riešenia bola validovaná rovnako ako pri segmentácii na piatich videách zachytávajúcích osoby rôzneho pohlavia a rasových skupín s celkovým počtom snímok 2389. Úspešnosť detekovania tváre a jej črtov bola 100 percentná. Ďalším pozitívnym faktorom tohoto riešenia je sledovanie viacerých črtov, na rozdiel od prístupu s využitím segmentácie kde bola sledovaná len oblasť úst. Výhodou je taktiež menšia časová



Obr. 4.4: Grafický výstup detekcie črtov tváre.

náročnosť a jednoduchšia implementácia. Skript funguje ako samostatný celok, ktorého výstup je uložený v súbore vo formáte vhodnom na nasledujúce spracovanie ľubovoľným systémom.

5 Dataset

Cielom práce, ktorý popisuje táto kapitola, bolo vytvorenie datasetu, na základe ktorého môže byť následne natrénovaný model umelej inteligencie schopnej odhadnúť progres PCH a jej jednotlivých symptómov u rozprávajúcej osoby zachytenej na videu. Dáta na tvorbu datasetu boli poskytnuté vedúcim práce v spolupráci s lekármi, ktorí tieto dáta vytvorili a klasifikovali. Tieto dáta boli vytvorené v spolupráci so 145 pacientami. Lekár s každým pacientom vykonal vopred určené cvičenia. Z každého celého sedenia s pacientom bolo nahrané video a zvlášť zvuk. K týmto súborom bol ešte priložený popis stavu a progresu jednotlivých symptómov pacienta a popis zvukovej nahrávky, konkrétne označkovanie časov kedy začína a končí každé jedno cvičenie. Dáta boli najprv analyzované a kvôli neúplnosti bol počet zredukovaný na 138 pacientov. Tieto dáta boli spracované do datasetu pomocou nasledujúcich krokov:

1. parametrizácia tváre osoby na videu v čase pomocou programu popísaného v podkapitole 4.2,
2. prerátanie časov jednotlivých cvičení pre video súbory,
3. extrakcia spracovaných parametrov tváre pre snímky konkrétnych cvičení,
4. konverzia parametrov tváre cvičenia do vektorov vlastností,
5. výber cvičení a dát pre odhad.

Nasledujúce podkapitoly obsahujú detailný popis týchto krokov a sú ukončené podkapitolou popisujúcou výsledný dataset.

5.1 Výpočet časov cvičení pre video súbor

Nakoľko jednotlivé časy cvičení boli definované len pre zvukovú nahrávku, bolo potrebné prepočítanie časov cvičení pre video súbor z dôvodu synchronizácie. Tento prepočet sa týkal každého videa pacienta a preto bol tento proces automatizovaný pomocou programu. Bez znalosti času jednotlivých cvičení vo videu by nebolo možné vytvoriť datasety pre špecifické cvičenia.

Vytvorený program je napísaný v jazyku JAVA, nazvaný ako „LabelSynchronizator“ a púšťa sa s nasledujúcimi argumentami:

- -iv cesta k video súboru,
- -ia cesta k audio súboru,
- -il cesta k súboru obsahujúcemu časy cvičení pre audio súbor,
- -o výstupný súbor obsahujúci prepočítané časy cvičení na video,
- -v zvýšenie verbozity.

Program po spustení spracuje súbor s časmi cvičení pre audio súbor. Následne prechádza jednotlivé cvičenia a keď nájde cvičenie s označením „9.4“, ktoré pred-

stavuje dlhý hovorený čas, vypočíta rozdiel začiatku tohoto cvičenia medzi audio a video súborom. Nakoľko video a audio súbor obsahujú rovnaké nahrávky, ktoré sú voči sebe len vzájomne posunuté, stačí následne vypočítaný rozdiel pripočítať k času začiatku a koncu každého cvičenia vo zvukovej nahrávke. Týmto spôsobom získa program časy cvičení pre video súbor. Nakoniec sú výsledné vypočítané časy cvičení zapísané do výstupného súboru.

Výpočet rozdielu začiatku cvičenia medzi audio a video súborom je najzložitejšia časť tohoto programu. Program najprv pomocou externého programu ffmpeg[46] získa z video nahrávky jej zvukovú stopu. Následne zo vstupného audio súboru vystrihne len úsek daného cvičenia. Z tohoto vystrihnutého úseku program pomocou knižnice musicg[47] vyrába tzv. otlačok. Vo zvukovej stope získanej z video súboru je následne vyhladaná časová pozícia začiatku tohoto otlačku. Po jej nájdení je vyrátaný rozdiel medzi nájdeným časom začiatku a časom začiatku uvedeným v súbore popisujúcom časy cvičení pre audio súbor. Časť programu zobrazujúca telo metódy slúžiacej na vystrihnutie úseku cvičenia z audio súboru je na výpisku 5.1

```
1 private void cutExerciseAudioAndSaveIt(Exercise e, File audio, File
    exerciseAudio, boolean verbose) throws IOException,
    UnsupportedAudioFileException {
2     if (verbose) System.out.println("Cutting exercise " + e.getLabel() + "
        from original audio file " + audio.getAbsolutePath());
3     long start = (long) Math.floor(e.getExerciseStart());
4     long end = (long) Math.ceil(e.getEnd());
5     if (verbose) System.out.println("Trim start second: " + start + ", trim
        end second: " + end );
6
7     AudioInputStream audioInputStream =
        AudioSystem.getAudioInputStream(audio);
8     AudioFormat format = audioInputStream.getFormat();
9
10    int bytesPerSecond = format.getFrameSize() * (int)format.getFrameRate();
11    audioInputStream.skip(start * bytesPerSecond);
12    long framesOfAudioToCopy = (end - start) * (int)format.getFrameRate();
13
14    AudioInputStream exerciseAudioStream = new
        AudioInputStream(audioInputStream, format, framesOfAudioToCopy);
15    AudioSystem.write(exerciseAudioStream, AudioFileFormat.Type.WAVE,
        exerciseAudio);
16 }
```

Výpis 5.1: Telo funkcie cutExerciseAudioAndSaveIt(...).

5.2 Extrakcia spracovaných parametrov tváre pre snímky konkrétneho cvičenia

Ďalším krokom v procese vytvorenia datasetu je extrakcia parametrov tváre spadajúcich pod určité cvičenie. Vytvorený program je napísaný v jazyku JAVA, nazvaný ako „JsonExerciseProcessor“ a púšťa sa s nasledujúcimi argumentami:

- -i cesta k adresáru obsahujúcemu adresáre s dátami jednotlivých pacientov,
- -o cesta k adresáru, do ktorého budú uložené JSON súbory s extrahovanými parametrami tváre daného cvičenia pre každého pacienta,
- -e značka cvičenia, ktorého dáta sa majú extrahovať,
- -v zvýšenie verbozity.

Program si po spustení otvorí vstupný adresár zadaný argumentom a v ňom vyhľadá všetky adresáre, pričom predpokladá, že každý jeden adresár je adresárom obsahujúcim pacientove dáta. Následne iteruje cez všetky adresáre pacientov, pričom v každom z nich si načíta súbor so všetkými parametrami tváre pacienta vytvorený pomocou programu 4.2 a súbor obsahujúci časy začiatku a konca každého cvičenia vo videu vytvorený pomocou programu 5.1. V súbore s popisom časov cvičení nájde program časové údaje o cvičení zadanom vo forme argumentu programu a následne na základe informácie o počte snímok za sekundu, začiatku a koncu cvičenia program extrahuje spomedzi všetkých snímok len tie, ktoré spadajú pod časový interval cvičenia vo videu. Tento proces sa opakuje pre každý jeden adresár s obsahom dát o pacientovi. Po spracovaní všetkých adresárov sa extrahované dáta cvičenia uložia do argumentom zadaného výstupného adresáru. V tomto adresári sa vytvorí súbor obsahujúci dáta cvičenia pre každé jedno video pacienta, tzn. pre 100 videí pacientov to bude 100 súborov, pričom názov súboru sa bude skladať z názvu adresáru pacienta a prípony v tvare názvu cvičenia a koncovky definujúcej typ súboru JSON.

5.3 Konverzia parametrov tváre cvičenia do vektorov vlastností

Posledným automatizovaným krokom v spracovaní dát do datasetu je krok konverzie parametrov tváre konkrétneho cvičenia získaných pomocou programu 5.2 do vektorov vlastností. Vektor vlastností predstavuje jeden prvok výsledného datasetu. Pre vytvorenie vektorov vlastností boli zvolené dva rozdielne prístupy. V prvom z nich sa do vektoru vlastností spracovali hodnoty popisujúce smerodiatnú odchýlku *Standard deviation* (SD) a relatívnu smerodiatnú odchýlku *Relative standard deviation* (RSD) každého parametru tváre v celom videu pre jeho normalizovanú aj normalizovanú rozdielovú hodnotu. Druhý prístup vytvára vektory vlastností v tvare x –

sekundových sekvencií parametrov tváre, pričom parametre sú normalizované a normalizované rozdielové. Oba prístupy sú popísané v nasledujúcich odsekoch.

5.3.1 Tvorba vektorov vlastností SD–RSD

Vytvorený program je napísaný v jazyku JAVA, nazvaný ako „FeatureVectorsProcessor“ a púšťa sa s nasledujúcimi argumentami:

- -i cesta k adresáru obsahujúcemu súbory s parametrami tváre konkrétneho cvičenia pre každého pacienta,
- -o cesta k súboru, do ktorého budú výsledné vektory vlastností zapísané,
- -v zvýšenie verbozity.

Program po svojom spustení najprv načíta a naparsuje všetky súbory nachádzajúce sa vo vstupnom adresári. Každý z týchto súborov obsahuje dáta parametrov tváre konkrétneho pacienta pre konkrétne cvičenie. Po naparovaní súborov má program v pamäti pre každý súbor (každého jedného pacienta) popis jeho parametrov tváre v čase daného cvičenia. Program následne iteruje cez všetky načítané súbory, pričom v každej iterácii najprv dáta normalizuje. Normalizácia je vykonaná na dátach každej jednej snímky delením hodnoty každého parametru s hodnotou parametru popisujúceho šírku tváre. Po normalizovaní každého parametru tváre každej snímky program vytvorí v pamäti dva rovnaké zoznamy všetkých parametrov pre všetky snímky. Jeden z týchto zoznamov je ďalej využitý na výpočet absolútnych medzisnímkových rozdielov parametrov. Týmto spôsobom sú vytvorené dva zoznamy: jeden s normalizovanými dátami a jeden s dátami normalizovanými a rozdielovými. Následne sú z oboch zoznamov pre každú jednu črtu tváre vyrátajú hodnoty SD a RSD podľa vzorcov:

$$SD = \sqrt{\frac{(x_1 - \bar{x})^2 + (x_2 - \bar{x})^2 + (x_3 - \bar{x})^2 \dots + (x_n - \bar{x})^2}{n - 1}}, \quad (5.1)$$

$$RSD = 100 \frac{SD}{\bar{x}}, \quad (5.2)$$

kde x_n predstavuje konkrétnu hodnotu, \bar{x} predstavuje priemernú hodnotu a n je počet hodnôt v zozname[48].

Týmto spôsobom program pre každú jednu črtu tváre získa jej smerodiatnú odchýlku a relatívnu smerodiatnú odchýlku pre jej normalizované hodnoty a pre jej normalizované rozdielové hodnoty. Takže pre každý parameter budú vo výslednom vektore vlastností štyri hodnoty. Tento proces výpočtu vektoru vlastností sa opakuje v každej jednej iterácii, čím v konečnom dôsledku program vyráta vektor vlastností pre každý jeden súbor vo stupnom adresári, čiže pre každého pacienta. Posledným krokom programu je uloženie vektorov vlastností do súboru typu CSV[49] tak, že

jeden riadok zodpovedá jednému vektoru vlastností pričom jeho hodnoty sú od seba oddelené čiarkou. Týmto spôsobom sa vytvoria stĺpce, pričom každý stĺpec obsahuje dáta konkrétnej hodnoty. Prvou hodnotou je názov súboru, z ktorého bol vektor získaný, a to z dôvodu zachovania informácie o pacientovi, ktorému tento vektor náleží. Ostatné stĺpce obsahujú dáta popisujúce parametre tváre videa.

Takto vytvorený súbor predstavuje jeden súbor datasetu, ktorý obsahuje vlastnosti, na základe ktorých sa môže model umelej inteligencie učiť, avšak chýbajú označenia, ktoré by mal predikovať. Na výpisku 5.2 je zobrazené telo metódy, ktorá z načítaných dát parametrov cvičení vytvára vektory vlastností.

```
1 public List<FeatureVector> extractFeatureVectors(HashMap<String,
2     ExerciseJson> exerciseJsonMap, boolean verbose) {
3
4     List<FeatureVector> result = new ArrayList<>();
5
6     for (Map.Entry<String, ExerciseJson> entry : exerciseJsonMap.entrySet())
7     {
8         if (verbose) System.out.println("Computing feature vector for patient:
9             " + entry.getKey());
10
11         FeatureVector v = computeFeatureVector(entry.getValue());
12         v.setName(entry.getKey());
13         result.add(v);
14     }
15     return result;
16 }
```

Výpis 5.2: Telo funkcie `extractFeatureVectors(...)`.

5.3.2 Tvorba vektorov vlastností časových úsekov

Program pre tvorbu vektorov vlastností časových úsekov je napísaný v jazyku JAVA, nazvaný ako „TimePeriodFeatureVectorsProcessor“ a púšťa sa s nasledujúcimi argumentami:

- -i cesta k adresáru obsahujúcemu súbory s parametrami tváre konkrétneho cvičenia pre každého pacienta,
- -o cesta k adresáru, do ktorého budú výsledné vektory vlastností zapísané,
- -s počet sekúnd jedného vektoru vlastností,
- -v zvýšenie verbozity.

Takisto ako pri programe určenom na tvorbu vektorov SD-RSD, program najprv načíta a naparsuje všetky súbory nachádzajúce sa vo vstupnom adresári. Z nich získa parametre tváre každého pacienta. Tieto parametre následne normalizuje rovnakým spôsobom ako pri tvorbe SD-RSD vektorov. Program následne iteruje cez dáta každého jedného súboru (pacienta) a z každej časovej rady parametrov vytvorí vektory vlastností popisujúce dáta časového úseku s trvaním definovaným podľa hodnoty zadanej užívateľom. Pre vytvorenie vektorov vlastností jedného cvičenia sa najprv

z počtu snímok za sekundu videa a z počtu sekúnd zadaného užívateľom vyráta množstvo snímok, ktoré má obsahovať jeden vektor vlastností. Následne program iteruje cez všetky snímky daného videa pre konkrétného pacienta podľa indexu snímky tak, aby bolo poradie iterácie snímok zhodné s poradím snímok vo videu. V každom kroku iterácie program aktuálne parametre snímky pridá do dočasného zoznamu parametrov snímok. V prípade, keď sa po pridaní počet snímok v zozname rovná počtu snímok vyrátaného pre jeden vektor vlastností, program uloží tento zoznam do zoznamu zoznamov snímok a vytvorí pre snímky nový zoznam, do ktorého sa bude pridávať v nasledujúcich iteráciách. Táto metóda je zobrazená na výpisku 5.3. Týmto spôsobom program vytvorí vektory vlastností z každého jedného súboru. Získané vektory vlastností program následne uloží do adresáru zadaného užívateľom. Pre každý vektor sú na koniec vyrátané rozdielové medzisnímkové hodnoty parametrov tváre a takisto uložené do adresáru. Jeden vektor vlastností tak v konečnom výsledku začína názvom súboru, z ktorého bol spracovaný a obsahuje výčet všetkých parametrov tváre každej snímky v sekvenčnom poradí tak ako vo videu. To znamená, že najprv sú vo vektore zapísané parametre tváre prvej snímky, za nimi nasledujú parametre tváre druhej snímky atď. Vektory vlastností sa ukladajú do súborov vo formáte CSV tak isto ako tomu bolo pri vektoroch vlastností SD-RSD. Týmto spôsobom sú vytvorené vektory vlastností, ktoré obsahujú informáciu o istej časovej sekvencií a zmene parametrov tváre počas tejto sekvencie.

```

1 private List<List<FrameJson>> computeVectors(ExerciseJson exJson, int
   framesInVector) {
2     List<List<FrameJson>> result = new ArrayList<>();
3     List<FrameJson> currFrames = new ArrayList<>();
4
5     System.out.println("Total frames: " + exJson.frames.size());
6
7     while (!exJson.frames.isEmpty()) {
8         FrameJson fj = exJson.frames.remove(0);
9
10        currFrames.add(fj);
11        if (currFrames.size() == framesInVector) {
12            result.add(currFrames);
13            currFrames = new ArrayList<>();
14        }
15    }
16
17    System.out.println("Extracted vectors: " + result.size());
18    return result;
19 }

```

Výpis 5.3: Telo funkcie computeVectors(...).

5.4 Výber cvičení a dát pre odhad

Po poslednom zautomatizovanom kroku 5.3 bolo možné vybrať spomedzi všetkých dát pacientov len dáta zodpovedajúce určitým cvičeniam a z nich zostaviť vektory vlastností vo formáte CSV zodpovedajúcom štruktúre datasetu. Tieto vektory vlastností obsahovali len dáta jednotlivých parametrov tváre, avšak bolo ich potrebné doplniť hodnotami (značkami), ktoré mal model umelej inteligencie na základe vlastností predikovať. Vybrané predikované hodnoty pridané do súborov datasetu boli získané z dokumentu popisujúceho stav pacientov podieľajúcich sa na štúdií. Konkrétne sa jednalo o informáciu či pacient trpí PCH, doba trvania PCH u pacienta a ostatné symptómy PCH vyjadrené číselnou hodnotou. Všetky predikované hodnoty sú zapísané v tabuľke 5.1.

Označenie predikovanej hodnoty	Popis predikovanej hodnoty
disease	informácia či pacient trpí chorobou
disease duration of PD	doba trvania PCH u pacienta
UPDRS III	hodnota na zjednotenej stupnici hodnotenia PCH (z angl. Unified Parkinson's disease rating scale)
FOG	miera zmrazovania chôdze (z angl. freezing of gait)
NMSS	hodnota na stupnici nemotorckých príznakov (z angl. non motor symptoms scale)
RBDSQ	hodnota poruchy spánku v REM fáze určená na základe dotazníku skríningu tejto poruchy (z angl. REM sleep behavior disorder (RBD) screening questionnaire)
faciokinesis	faciokinéza
phonorespiration	fonorespirácia
phonetics	fonetika
overall DX index	celkový DX index
BDI	miera depresie podľa zoznamu Beckových depresíí (z angl. Beck depression inventory)
ACE-R	miera fungovania kognitívnych funkcií podľa Addenbrookovho kognitívneho vyšetrenia (z angl. Addenbrooke's Cognitive Examination-Revised)

Tab. 5.1: Popis predikovaných hodnôt zaradených do datasetu.

Selekcia dát podľa cvičenia bola potrebná z dôvodu výberu relevantných cvičení pre odhad PCH. Konkrétne bolo potrebné vybrať cvičenia, pri ktorých bolo vidieť

celú tvár pacienta a príznaky PCH sa pri vykonávaní týchto cvičení mohli prejavíť. Popis vybraných cvičení je v tabuľke 5.2.

Označenie cvičenia	Text hovorený pacientom v rámci cvičenia
7.4 _1	Do čtvrt hodiny tam byla smršť.
7.4 _2	Prohovorte to s ním dopodrobna.
7.4 _2	Pri ústupu pluku duní bubny.
9.2 _1	Kucharští učni nejsou jak zlatnictí.
9.2 _2	Celý večer se učí sčítat.
9.2 _3	ba-ba-ba...
9.2 _4	p-t-k-p-t-k...
9.2 _5	pa-ta-ka-pa-ta-ka...

Tab. 5.2: Popis cvičení vybraných na tvorbu datasetu.

5.5 Popis datasetu

Nakoľko pri tvorbe vektorov vlastností boli zvolené dva prístupy, bude aj popis výsledného datasetu rozdelený do dvoch logických celkov. Ako prvá bude popísaná časť datasetu obsahujúca vektory vlastností SD–RSD, za ktorou bude nasledovať podrobný popis časti datasetu tvorenej z vektorov vlastností časových úsekov. Všetky súbory celého datasetu sú k dispozícii na priloženom CD v adresári s názvom dataset.

5.5.1 Časť obsahujúca vektory vlastností SD–RSD

Dáta popisujúce stav pacientov zúčastnených na štúdií možno rozdeliť do dvoch kategórií: zdraví a chorí pacienti. Pre zdravých pacientov boli v dokumentoch uvedené iba hodnoty uvádzajúce či pacient trpí chorobou, hodnoty popisujúce progres jednotlivých symptómov PCH u týchto pacientov chýbali. Z tohoto dôvodu boli pre každé jedno cvičenie vytvorené dva súbory. Jeden obsahoval dáta všetkých pacientov (vhodný na klasifikáciu či pacient trpí PCH) a druhý obsahoval iba dáta pacientov trpiacich na PCH (vhodný na predikciu symptómov PCH). Týmto spôsobom vzniklo šesťnásť súborov datasetu – dva pre každé cvičenie. Následne boli vytvorené ešte dva ďalšie súbory pričom oba obsahovali dáta všetkých cvičení, avšak jeden obsahoval len dáta pacientov trpiacich na PCH a druhý obsahoval všetky dáta.

Každý jeden súbor je typu CSV, pričom jeden riadok tohoto súboru predstavuje dáta jedného pacienta. Súbory obsahujúce dáta pacientov trpiacich na PCH majú v priemere 85 riadkov, súbory obsahujúce dáta všetkých pacientov 135 riadkov.

Výnimkou sú súbory obsahujúce všetky cvičenia, ktorých počet riadkov je suma všetkých riadkov všetkých cvičení.

Každý riadok obsahuje 49 stĺpcov, pričom prvý stĺpec obsahuje identifikačné číslo pacienta, nasledujúcich 36 stĺpcov obsahuje výčet parametrov tváre a posledných 12 stĺpcov obsahuje číselné hodnoty, ktoré sa má umelá inteligencia naučiť predikovať, viď Tab. 5.1. Každý jeden parameter tváre je popísaný jeho smerodatnou odchýlkou a relatívnou smerodatnou odchýlkou v rámci videa pre jeho normalizované hodnoty a hodnoty normalizované rozdielové.

Súbory tejto časti datasetu sú na priloženom CD uložené v adresáry „SD_RSD“. Názov každého súboru sa skladá z označenia cvičenia (viď Tab. 5.2), ktoré je v prípade, že sa jedná o súbor obsahujúci iba dáta pacientov trpiacich na PCH nasledované slovom „disease“ (po anglicky choroba).

5.5.2 Časť obsahujúca vektory vlastností časových úsekov

Časť datasetu obsahujúca dáta tvorené vektormi vlastností časových úsekov je na CD uložená v adresáry „time_interval“. Táto časť datasetu obsahuje pre každé jedno cvičenie dva súbory. Jeden obsahujúci normalizované parametre tváre a druhý obsahujúci normalizované rozdielové hodnoty parametrov. Taktiež sa v tomto datasete nachádzajú dva súbory obsahujúce dáta zo všetkých cvičení, jeden pre normalizované a druhý pre normalizované rozdielové hodnoty.

Každý jeden súbor je typu CSV, pričom jeden riadok súboru predstavuje dáta popisujúce zmenu sledovaných parametrov tváre jedného časového úseku z videa. Zvolený časový úsek je 2 sekundy. Tento čas bol zvolený na základe analýzy priemernej doby trvania každého cvičenia. Taktiež bolo potrebné zvoliť taký časový úsek, na ktorom by mohla byť viditeľná zmena mimiky tváre pacienta. Dáta popisujúce jeden časový úsek videa (jeden riadok) sú v stĺpcoch. Prvý stĺpec obsahuje informáciu o tom o ktorého pacienta sa jedná. Za ním nasleduje výčet všetkých parametrov tváre každej jednej snímky daného časového úseku v sekvenčnom poradí tak, aby ostalo zachované pôvodné poradie snímok z videa. Pri počte 25 snímok videa za sekundu a časovom úseku trvania jedného vektora 2 sekundy vo výsledku jeden riadok pre normalizované hodnoty obsahuje dáta parametrov pre 50 snímok. V prípade normalizovaných rozdielových hodnôt obsahuje jeden riadok parametre pre 49 snímok – začína sa rozdielom prvej a druhej snímky. Dáta parametrov tváre sú nasledované hodnotami popisujúcimi progres jednotlivých symptómov určených na predikciu.

6 Trénovanie modelov na predikcia prítomnosti PCH a progresu jej symptómov u pacienta

Táto kapitola sa venuje samotnej predikcii PCH a progresu jej symptómov na základe datasetu vytvoreného v rámci práce, ktorý je popísaný v podkapitole 5.5. Keďže dataset je tvorený z dát dvoch rozdielnych typov, bude aj popis tréovania modelov a ich výsledkov rozdelený do dvoch nasledujúcich podkapitol. Ako prvé boli tréované a vyhodnocované presnosti modelov umelej inteligencie na základe dát časti datasetu SD–RSD, následne sa použila časť datasetu obsahujúca dáta časových úsekov cvičení. Každá podkapitola obsahuje popis programu určeného na natréovanie a overenie presnosti modelov a aj celkové výsledky presnosti natréovaných modelov.

6.1 Predikcia na časti datasetu SD–RSD

Časť datasetu označená ako SD–RSD obsahuje dáta popisujúce smerodajnú odchýlku a relatívnu smerodajnú odchýlku každého parametru v rámci celého cvičenia. Práve na základe týchto dát by mala byť umelá inteligencia schopná predikcie prítomnosti PCH a progresu jednotlivých symptómov PCH u pacienta. V rámci práce bol vytvorený program určený na tréovanie a overenie presnosti modelov. Tento program je spolu s výsledkami presnosti modelov popísaný v nasledujúcich sekciách.

6.1.1 Trénovanie a hodnotenie modelov

Táto podkapitola popisuje program vytvorený v rámci práce, ktorý je určený na natréovanie modelov určených pre odhad progresu PCH a jej symptómov. Program je napísaný v jazyku Python a je určený na spracovanie datasetu popísaného v podkapitole 5.5.1. Nakoľko hodnoty datasetu určené na predikciu obsahujú ako diskkrétne tak aj spojité hodnoty, bolo potrebné vytvoriť program, ktorý dokáže natréovať klasifikačné aj regresné modely. Program je taktiež schopný dané natréované modely validovať a určiť ich presnosť s využitím mnohonásobnej K -fold krížovej validácie s preusporiadaním prvkov. Program využíva funkcionality externých knižníc, pričom najvýznamnejšie z nich sú sklearn[50] a pandas[51]. Program na začiatku najprv načíta zvolený dataset a zobrazí o ňom informácie, následne pokračuje v tréovaní a hodnotení klasifikačných alebo regresných modelov. Po natréovaní a ohodnotení

modelov program ukladá informácie o modeloch do výstupného súboru typu Microsoft Excel Workbook s koncovkou „.xlsx“.

Klasifikačné modely

Na tréovanie klasifikačných modelov bol využitý algoritmus *K-nearest neighbors* (KNN)[52], pričom hodnotu, ktorú sa mal model naučiť predikovať bola diskretná binárna hodnota definujúca prítomnosť PCH u pacienta. Algoritmus najprv po načítaní datasetu rozdelí dáta na vstupné a výstupné a následne v cykle iteruje cez všetky parametre tváre a vytvára a validuje pre ne klasifikačný model. Program dokáže taktiež vytvoriť a validovať model len pre vybranú kombináciu parametrov tváre. V každom cykle je najprv nájdený vhodný počet susedov pre KNN algoritmus a následne natrénovaný a validovaný model. Tréovanie a validácia prebieha s využitím 5-fold krížovej validácie s preusporiadaním prvkov, pričom pre každú jej iteráciu je vyrátaná presnosť vytvoreného modelu, ktorá je uložená do pola. Po všetkých iteráciách je pole naplnené hodnotami presností jednotlivých modelov a funkcia vracia priemernú hodnotu týchto presností. Táto funkcia je zobrazená na výpisku 6.1.

```
1 def knn(x, y, k, test_data_percentage, cross_validation_loops):
2     x_digits = np.array(x)
3     Y_digits = np.array(y)
4
5     split_groups = (int)(len(x_digits) / (len(x_digits) *
6         test_data_percentage))
7     k_fold = KFold(n_splits=split_groups, shuffle=True)
8     accuracy_score_arr = np.empty(0)
9
10    for x in range(0, cross_validation_loops):
11        for train_indices, test_indices in k_fold.split(x_digits):
12            x_train, x_test = x_digits[train_indices],
13                x_digits[test_indices]
14            y_train, y_test = Y_digits[train_indices],
15                Y_digits[test_indices]
16
17            scaler = StandardScaler()
18            scaler.fit(x_train)
19            x_train = scaler.transform(x_train)
20            x_test = scaler.transform(x_test)
21
22            classifier = KNeighborsClassifier(n_neighbors=k)
23            classifier.fit(x_train, y_train)
24            y_predicted = classifier.predict(x_test)
25
26            accuracy_score_arr = np.append(accuracy_score_arr,
27                [accuracy_score(y_test, y_predicted)])
28
29    return accuracy_score_arr.mean();
```

Výpis 6.1: Telo metódy `knn(...)`.

Regresné modely

Pre tréovanie regresných modelov boli využité štyri rôzne algoritmy regresie: lineárna regresia[53], rozhodovací strom[54], XGBoost[55], náhodný les[56]. Každý model sa mal naučiť predikovať spojité hodnoty popisujúce dĺžku trvania PCH u pacienta a progres symptómov PCH u pacienta. Popis symptómov je v tabuľke 5.1.

Algoritmus potrebuje ako vstup slovník obsahujúci symptómy, ktoré má predikovať a im priradené parametre tváre, na základe ktorých má predikovať. To z dôvodu selekcie dát z veľkého množstva dát v datasetoch. Následne algoritmus trénuje a odhaduje úspešnosť modelov pre každú jednu zadanú kombináciu symptómu s parametrom tváre. Algoritmus je taktiež schopný rátať kombinácie vstupných parametrov tváre a predikovať výslednú hodnotu symptómu na ich základe. Program iteruje cez všetky zadané kombinácie symptóm–parametre a v každej iterácii vytvára a overuje modely pre všetky štyri zvolené regresné algoritmi. Natréovanie a výpočet úspešnosti jedného modelu je vykonaný pomocou 5–fold krížovej validácie s preusporiadaním prvkov, pričom pre každú jej iteráciu sú vyrátané štyri hodnoty určujúce chybovosť vytvoreného modelu, konkrétne MAE, MAPE, RMSE a R^2 . Tieto metriky sú následne uložené do príslušných polí. Po všetkých iteráciách sú polia naplnené hodnotami jednotlivých metrík chybovosti modelov a funkcia vracia priemernú hodnotu každého jedného pola. Telo funkcie rátajúcej jeden model a jeho chybovosť je zobrazené na výpisku 6.2.

```
1 def regression(regression_model, x, y, test_data_percentage, xval_loops):
2     ...
3     k_fold = KFold(n_splits=split_groups, shuffle=True)
4
5     for x in range(0, cross_validation_loops):
6         for train_indices, test_indices in k_fold.split(x_digits):
7             x_train, x_test = x_digits[train_indices],
8                 x_digits[test_indices]
9             y_train, y_test = Y_digits[train_indices],
10                 Y_digits[test_indices]
11
12             regression_model.fit(x_train, y_train)
13             y_predicted = regression_model.predict(x_test)
14             mae_arr = np.append(mae_arr, [mean_absolute_error(y_test,
15                 y_predicted)])
16             mape_arr = np.append(mape_arr,
17                 [mean_absolute_percentage_error(y_test, y_predicted)])
18             rmse_arr = np.append(rmse_arr,
19                 [sqrt(mean_squared_error(y_test, y_predicted))])
20             r2_score_arr = np.append(r2_score_arr, [r2_score(y_test,
21                 y_predicted)])
22
23     return mae_arr.mean(), mape_arr.mean(), rmse_arr.mean(),
24         r2_score_arr.mean()
```

Výpis 6.2: Telo metódy `regression(...)`.

6.1.2 Výsledky

Modely boli trénované pre dva rozdielne typy odhadovaných dát, diskkrétne a kontinuálne hodnoty. Z tohoto dôvodu boli zvolené na predikciu ako klasifikačné tak regresné modely umelej inteligencie. Klasifikácia bola použitá pri odhade prítomnosti PCH u pacienta, regresia bola použitá pre predikciu progresu štádia ochorenia a ostatných symptómov vypísaných v tabuľke 5.1. Klasifikačné a regresné modely majú rozličné metriky určujúce ich presnosť a z tohoto dôvodu je popis výsledkov rozdelený do dvoch nasledujúcich sekcií.

Predikcia prítomnosti choroby u pacienta

Na predikciu prítomnosti choroby u pacienta bola použitá klasifikácia. Cieľom bolo natrénovať model, ktorý bude na základe parametrov tváre schopný predikovať binárnu hodnotu určujúcu prítomnosť choroby u pacienta. Model sa trénoval na súboroch datasetu obsahujúcich dáta o zdravých aj chorých pacientoch pre každé jedno cvičenie a aj pre súbor obsahujúci dáta všetkých cvičení. Pri každom súbore boli natrénované klasifikačné modely, ktoré mali ako vstupnú hodnotu jeden parameter tváre. Následne bol ešte pre každý súbor natrénovaný model, ktorého vstupné parametre boli všetky parametre tváre. Trénovanie a validácia modelov bola prevedená pomocou 5-fold krížovej validácie s preusporiadaním prvkov. Ako validačná metrika modelu je zvolená percentuálna úspešnosť klasifikovania pacienta. Všetky hodnoty validačných metrik pre každý jeden natrénovaný model sú k dispozícii v podobe súboru klasifikacia.xlsx v adresári výsledkov zapísaného na CD priloženom k tejto práci. Tieto hodnoty boli taktiež vynesené do grafu zobrazeného na obrázku A.1 v prílohe A. Definícia piatich modelov s najlepšou úspešnosťou je zobrazená v tabuľke 6.1. Tabuľka 6.2 zobrazuje päť vstupných parametrov s najlepšou priemernou úspešnosťou natrénovaného modelu.

Cvičenie	Vstupný parameter modelu	Úspešnosť
7.4_2	SD normalizovanej výšky pravého obočia	70%
9.2_4	SD normalizovanej výšky pravého obočia	67%
9.2_5	RSD normalizovanej šírky pravého obočia	66%
7.4_2	SD normalizovanej vzdialenosti nos – brada	65,8%
9.2_4	SD normalizovanej hodnoty výšky ľavého obočia	65%

Tab. 6.1: Najúspešnejšie modely predikcie prítomnosti choroby u pacienta na základe SD–RSD dát.

Vstupný parameter modelu	Úspešnosť
SD normalizovanej výšky pravého obočia	58,4%
RSD normalizovanej rozdielovej hodnoty plochy úst	58%
všetky parametre	58%
SD normalizovanej výšky ľavého obočia	57,6%
SD normalizovanej rozdielovej hodnoty vzdialenosti nos – ústa	57,5%

Tab. 6.2: Päť vstupných parametrov s najlepšou priemernou úspešnosťou natrénovaného modelu pre predikciu prítomnosti choroby u pacienta na základe SD–RSD dát.

Pri grafickom zobrazení je pri pohľade na úspešnosť modelov niektorých vstupných parametrov viditeľná nízka variabilita. Konkrétne model pre parameter s indexom 21, popisujúci relatívnu štandardnú odchýlku normalizovanej rozdielovej hodnoty plochy úst, má najnižšiu variabilitu úspešnosti natrénovaného modelu, a to v hodnote 5%. Ďalším prípadom s nízkou variabilitou úspešnosti modelu je parameter s indexom 30, predstavujúci smerodajnú odchýlku normalizovanej rozdielovej hodnoty šírky pravého obočia. Z tabuliek môžeme vyčítať, že modely založené na predikcii na základe vstupných dát týkajúcich sa výšky a šírky obočí patria medzi najúspešnejšie. Toto zistenie poukazuje na vzťah medzi mimikou obočia pacienta a prítomnosťou choroby.

Predikcia doby trvania a symptómov PCH u pacienta

Pri predikcii doby trvania PCH a jednotlivých symptómov PCH bola použitá predikcia pomocou regresných modelov. Boli zvolené regresné algoritmy: lineárna regresia, rozhodovací strom, XGBoost a náhodný les. Každý jeden natrénovaný model bol validovaný 5-fold krížovou validáciou s preusporiadaním prvkov a vyhodnotený pomocou metrík MAE, MAPE, RMSE a R^2 . Výsledné dáta metrík všetkých modelov sú v prílohe na priloženom CD v adresári s názvom „predikcia“.

Na predikciu doby trvania choroby a symptómov PCH boli použité súbory datasetu, ktoré obsahovali len dáta pacientov trpiacich na PCH. Pre každý jeden súbor boli najprv vyrátané korelácie parametrov tváre k odhadovaným symptómom a dobre trvania. Súbory s výslednými vyrátanými koreláciami sú v prílohe na CD v adresári pomenovanom ako „korelácie“. Následne boli tréované modely pre tie parametre a symptómy jednotlivých cvičení, ktorých miera korelácie bola vyššia ako 25%. Výsledné chybovosti daných natrénovaných modelov boli uložené do súborov.

Taktiež boli natrénované modely, ktoré predikovali symptómy na základe všetkých parametrov tváre daného súboru. Tieto modely boli tréované len pre súbor

cvičenia 9.2_4 a súbor obsahujúci dáta všetkých cvičení. Následne na základe výsledkov boli ešte natrénované modely pre cvičenie 9.2_4, ktoré mali ako vstupné parametre modelu všetky možné kombinácie parametrov tváre s dĺžkou 2 a 3 a odhadovaným symptómom bola faciokinéza.

V tabuľkách v prílohe B sú zobrazené najúspešnejšie modely každého jedného predikovaného symptómu podľa validačných metrík MAE, MAPE a RMSE. Nakoľko hodnoty odhadovaných symptómov sú od seba rozdielne (doba trvania defiovaná od 2 po 22, celkový DX index od 57 po 88), bola zvolená ešte jedna metrika, ktorú v rámci práce nazveme ako MAEPZR (MAE percento z rozsahu). Vyjadruje percentuálnu hodnotu MAE z rozsahu hodnôt symptómu podľa rovnice:

$$\text{MAEPZR} = \frac{MAE}{\frac{x_{max} - x_{min}}{100}}, \quad (6.1)$$

pričom x_{max} vyjadruje maximálnu hodnotu symptómu a x_{min} vyjadruje minimálnu hodnotu symptómu v datasete. Modely symptómov zoradené podľa tejto metriky sú zobazené v tabuľke 6.3, pričom práve táto tabuľka sa dá považovať ako najvhodnejšia pre porovnanie úspešností odhadu progresu jednotlivých symptómov.

Z tabuľky 6.3 vidíme, že model s najlepšou hodnotou metriky MAEPZR je model predikujúci celkový DX index pacienta, s hodnotou metriky 11,1%. Za ním nasleduje model pre predikciu faciokinézy s hodnotou 11,75% a model predikcie fonorespirácie s hodnotou 12%. Model s hodnotou strednej absolútnej chyby 11,1% z rozsahu odhadovaných hodnôt nieje dostatočne presný pre použitie v medicíne, avšak natrénované modely majú hodnotu v podobe informácie o tom, na ktoré parametre tváre, cvičenia a symptómy by sa mohli zamerať ďalšie výskumy. Vo výsledkoch tabuliek najlepších modelov sa vo veľkej väčšine vyskytujú modely trénované na dátach z cvičení 7.4_2, 7.4_3 a 9.2_5. Parametre tváre, na základe ktorých boli trénované najúspešnejšie modely sú rôzne, záleží od sledovaného symptómu.

Symptóm	Parameter	MAEPZR [%]	Cvičenie	Algoritmus
celkový DX index	SD rozdielovej vzdialenosti nos – brada	11,1012	7.4_3	lineárna regresia
faciokinéza	RSD šířky úst	11,7541	7.4_3	lineárna regresia
fonorespirácia	SD rozdielovej šířky ľavého obočia	12,0159	9.2_5	lineárna regresia
fonetika	RSD plochy úst	15,1425	9.2_5	lineárna regresia
ACE-R	SD rozdielovej vzdialenosti nos – brada	15,985	7.4_3	lineárna regresia
doba trvania PCH	všetky	17,119	všetky	lineárna regresia
UPDRS III	všetky	17,750	všetky	lineárna regresia
BDI	RSD rozdielovej vzdialenosti nos – brada	17,758	9.2_2	rozhodovacie stromy
RBDSQ	RSD výšky ľavého obočia	20,42	7.4_2	lineárna regresia
NMSS	RSD vzdialenosti nos – ústa	22,55	9.2_5	lineárna regresia
FOG	RSD šířky pravého obočia	23,34	7.4_3	lineárna regresia

Tab. 6.3: Najlepšie modely odhadu progresu symptómov pacienta pre jednotlivé symptómy podľa MAEPZR (SD–RSD dáta).

6.2 Predikcia na časti datasetu časových úsekov

Táto časť datasetu obsahuje dáta popisujúce zmenu sledovaných parametrov tváre v dvojsekundových úsekoch cvičení.

6.2.1 Klasifikačné modely

Trénovanie klasifikačného modelu bolo potrebné pre odhad prítomnosti PCH u pacienta. Takisto ako pri SD–RSD dátach bol použitý algoritmus KNN. Program pre každý jeden súbor tejto časti datasetu natrénuje a overí presnosť modelu, pričom vstupnými hodnotami modelu sú všetky parametre tváre každej jednej snímky daného časového úseku. Predikovanou hodnotou je binárna hodnota predstavujúca prítomnosť PCH u pacienta. Program ráta pre každé cvičenie model pomocou KNN a krížovej validácie, pričom volí hodnotu K zodpovedajúcu počtu susedov KNN z rozsahu od 1 po 29. Pre každý jeden model je do výstupného súboru uložená jeho úspešnosť, hodnota K a informácia o tom, z akého cvičenia bol model natrénovaný.

6.2.2 Regresné modely

Regresné modely boli využité pre odhad dĺžky trvania PCH a progres symptómov PCH u pacienta, pričom sa jednalo o symptómy popísané v tabuľke 5.1. Regresné algoritmy boli zvolené tie isté ako pri predikcií na základe dát časti datasetu SD–RSD.

Program pre každý súbor tejto časti datasetu trénuje modely určené na odhad progresu každého jedného symptómu. Pre každý symptóm sú natrénované štyri modely – pre každý jeden zo zvolených algoritmov regresie. Trénovanie a validácia je implementovaná s využitím leave-one-out krížovej validácie. Tak isto ako pri klasifikácií sú vstupnými hodnotami modelu všetky parametre tváre každej jednej snímky časového úseku. Výstupnou predikovanou hodnotou je spojitá hodnota popisujúca progres symptómu PCH. Pre každý jeden odhadovaný predikovaný symptóm z každého cvičenia sú do výstupného súboru uložené hodnoty chybovosti všetkých štyroch natrénovaných modelov, informácia o odhadovanom symptóme a označenie cvičenia. Metriky chybovosti sú MAE, MAPE, RMSE a R^2 .

6.2.3 Výsledky

Z dôvodu využitia ako klasifikácie tak aj regresie a ich rozdielnych validačných metrík sú výsledky jednotlivých modelov rozdelené do dvoch častí.

Predikcia prítomnosti PCH u pacienta

V tomto prípade bola využitá klasifikácia, kedy boli trénované modely pre každý jeden súbor datasetu časti časových úsekov s využitím metódy KNN. Odhadovanou veličinou bola prítomnosť PCH u pacienta. Pre každý súbor boli trénované modely s K (počet susedov algoritmu KNN) v rozsahu od 1 po 29. Výsledky úspešnosti všetkých modelov sú k dispozícii na priloženom CD v súbore „klasifikacia.xlsx“ v adresári s názvom „time_interval“. 5 modelov s najlepšou úspešnosťou klasifikácie je zobrazených v tabuľke 6.4.

Cvičenie	K	Úspešnosť
všetky	1	91,1%
všetky s rozdielovými hodnotami	1	90,1%
všetky s rozdielovými hodnotami	3	89,9%
všetky	2	89,7%
všetky	4	89,6%

Tab. 6.4: Najúspešnejšie modely predikcie prítomnosti choroby u pacienta na základe dát časových úsekov.

Najúspešnejší natrénovaný model mal úspešnosť na hodnote 91,1% a jednalo sa o súbor obsahujúci normalizované dáta a K pre KNN nastavené na hodnotu 1. Za ním nasledoval model s úspešnosťou 90,1% natrénovaný na základe dát normalizovaných rozdielových hodnôt všetkých videí. Celkovo sa najúspešnejšie modely predikcie natrénovali pre súbory obsahujúce dáta zo všetkých cvičení, na základe čoho môžeme usúdiť zvyšujúcu sa presnosť s rastúcim množstvom tréningových dát.

Predikcia doby trvania a symptómov PCH u pacienta

Regresné modely určené na odhad doby trvania PCH a jej symptómov boli trénované pre každý jeden súbor tejto časti datasetu, pričom v každom súbore bola predikovaná doba trvania PCH a progres rovnakých symptómov ako pri dátach SD–RSD. Výsledky všetkých natrénovaných modelov sú v prílohe C, v tabuľkách zobrazujúcich informácie o modeloch s najnižšou chybovosťou pre odhad progresu každého symptómu na základe validačných metrík MAE, MAPE a RMSE. Zobrazenie modelov s najnižšou hodnotou validačnej metriky MAEPZR pre každý jeden symptóm je v tabuľke 6.5.

Z tabuľky môžeme vidieť, že tak isto ako pri odhade prítomnosti PCH, aj pri odhade progresu PCH a jej symptómov sa najlepšie modely natrénovali na súboroch

Symptóm	Cvičenie	Hodnota metriky [%]	Algoritmus
celkový DX index	všetky	7,874	XGBoost
faciokinéza	všetky – rozdielové hodnoty	8,36	XGBoost
fonorespirácia	všetky – rozdielové hodnoty	9,054	XGBoost
doba trvania PCH	všetky	10,812	XGBoost
ACE–R	všetky – rozdielové hodnoty	10,921	XGBoost
UPDRS III	všetky – rozdielové hodnoty	11,397	XGBoost
fonetika	všetky – rozdielové hodnoty	11,667	XGBoost
BDI	všetky – rozdielové hodnoty	12,629	XGBoost
RBDSQ	všetky – rozdielové hodnoty	13,2	XGBoost
NMSS	všetky – rozdielové hodnoty	15,447	XGBoost
FOG	všetky – rozdielové hodnoty	16,24	XGBoost

Tab. 6.5: Najlepšie modely odhadu progresu symptómov pacienta pre jednotlivé symptómy podľa MAEPZR (dáta časových úsekov).

obsahujúcich dáta všetkých cvičení. Algoritmus, ktorý bol schopný učiť sa na daných dátach je XGBoost, ostatné vykazovali niekoľkonásobne horšie výsledky. Najúspešnejším predikovaným symptómom je celkový DX index s hodnotou MAEPZR na 7,874%, nasledovaný faciokinézou s hodnotou 8,36%.

6.3 Porovnanie výsledkov

Táto podkapitola slúži na porovnanie výsledkov odhadu prítomnosti PCH u pacienta, doby jej trvania a progresu jednotlivých symptómov na základe parametrov tváre získaných z videa hovoriacej osoby. Na odhad boli použité modely umelej inteligencie, pričom mali k dispozícii dva typy dát. Prvým typom boli hodnoty smerodatnej odchýlky a relatívnej smerodatnej odchýlky v úseku celého cvičenia (označené ako SD–RSD), druhým boli dáta obsahujúce všetky parametre tváre daného cvičenia v dvojsekundových časových úsekoch. Predikcia bola realizovaná pomocou klasifikácie a regresie.

Predikcia prítomnosti PCH u pacienta

Prítomnosť PCH u pacienta bola predikovaná pomocou klasifikácie. Z tabuliek 6.1 a 6.4 zobrazujúcich 5 modelov s najlepšou úspešnosťou pre každý typ dát môžeme vyčítať, že modely trénované na dátach SD–RSD mali najlepšiu úspešnosť predikcie na hodnote 70%. Modely trénované na dátach časových úsekov dosiahli úspešnosti

91,1%. Z tohoto dôvodu sa dáta popisujúce parametre tváre v časovom úseku ukázali ako vhodnejšie na odhad prítomnosti PCH u pacienta oproti dátam popisujúcim smerodajnú a relatívnu smerodajnú odchýlku parametrov tváre v celom videu. Najúspešnejšie modely boli natrénované na súboroch obsahujúcich najväčšie množstvo dát, z čoho vyplýva pravdepodobné zvýšenie úspešnosti pomocou zväčšenia datasetu. Ďalším činiteľom vo výslednej úspešnosti predikcie je dĺžka zvoleného časového úseku, ktorá by pri väčšej hodnote mohla zlepšiť schopnosť učenia umelej inteligencie a zvýšiť výslednú úspešnosť modelu.

Predikcia progresu PCH a jej symptómov u pacienta

Predikcia progresu PCH a jej symptómov bola vykonávaná pomocou algoritmov regresie. Z tabuliek 6.3 a 6.5 zobrazujúcich najúspešnejšie modely odhadu každého jedného symptómu podľa MAEPZR vidíme, že najúspešnejšie predikované symptómy sú v oboch prípadoch celkový DX index a faciokinéza. Pri dátach typu SD-RSD bola ich hodnota MAEZPR na 11,1% a 11,8%. Pri použití dát časových úsekov sa znížila na hodnoty 7,9% a 8,4%. V tomto prípade sa znovu ukázali dáta časových úsekov ako vhodnejšie. Taktiež ako pri odade prítomnosti PCH boli najúspešnejšie modely natrénované na súboroch obsahujúcich dáta zo všetkých cvičení.

7 Záver

Práca obsahuje popis PCH, jej príznakov a aj funkcie dopamínu v organizme, nakoľko práve jeho nedostatok v organizme je hlavnou príčinou príznakov. Ďalej sa venuje metódam segmentácie obrazu a metódam na identifikáciu osoby pomocou tváre. Teoretická časť je ukončená popisom metód umelej inteligencie.

Riešenie parametrizácie tváre z videa hovoriacej osoby bolo implementované dvomi rôznymi prístupmi – metódami počítačového spracovania obrazu a segmentácie a pomocou metód identifikácie osoby na základe tváre. Ako vhodnejšie riešenie sa v tomto prípade ukázalo riešenie pomocou metód identifikácie osoby z tváre, nakoľko má toto riešenie minimálnu chybovosť, sleduje viaceré parametre tváre, odhaduje emóciu tváre, má nižšiu časovú náročnosť a taktiež je jednoduchšie na implementáciu. Medzi sledované parametre tváre patrí napríklad vzdialenosť nos – brada alebo šírka a výška úst. Pripravené dáta sú vhodné pre akékoľvek počítačové spracovanie. Nakoľko sa jedná o dáta popisujúce črty a emócie tváre osoby na videu v čase, využiteľnosť tohoto riešenia má široké spektrum v rôznych oboroch. Tieto číselné hodnoty črt tváre predstavujú anonymizované dáta, ktoré vďaka tomu môžu byť prenášané voľne cez internet. V prípade využitia v medicíne by to predstavovalo možnosť vyšetrenia z domu bez nutnosti prítomnosti lekára.

Z dát získaných pomocou programu parametrizácie tváre bol vytvorený dataset obsahujúci 36 súborov. Prvý typ súborov popisuje smerodajnú a relatívnu smerodajnú odchýlku každého parametru tváre v rámci celého cvičenia, druhý obsahuje dáta popisujúce všetky parametre tváre videa v dvoj sekundových úsekoch v sekvencnom poradí. Na základe týchto dát boli natrénované modely umelej inteligencie určené na odhad prítomnosti a progresu PCH a jej symptómov u pacienta. Najlepšia dosiahnutá úspešnosť predikcie prítomnosti choroby u pacienta bola 91,1%. V odhade progresu symptómov PCH boli využité modely regresie, pričom najúspešnejšie modely boli natrénované pre symptómy celkového DX indexu a faciokinézy. Ich percentuálna hodnotou absolútnej strednej chyby z rozsahu hodnôt popisujúcich progres symptómu je 7,874% a 8,36%. Najúspešnejšie modely boli natrénované na dátach časových úsekov.

Dosiahnutá úspešnosť a chybovosť je pre využitie v medicíne nedostačujúca, avšak natrénované modely a ich výsledky jasne poukazujú na existujúce vzťahy medzi parametrami tváre osoby na videu a odhadom prítomnosti a progresu PCH a jej symptómov u pacienta. V rámci práce boli taktiež navrhnuté ďalšie optimalizácie vytvoreného riešenia, ktoré by mali mať pozitívny vplyv na výsledky predikcií.

Súčasťou výsledkov práce sú taktiež všetky vytvorené nástroje umožňujúce spracovanie nových dát pacientov, ich pridanie do vytvoreného datasetu alebo samotné tréningovanie modelov umelej inteligencie na základe dát tohoto datasetu. Tieto ná-

stroje umožňujú plynule nadviazať na výsledky tejto práce a pokračovať v ďalšom výskume tejto problematiky.

Literatúra

- [1] LTH, T., KL, O. *The Impact of Medical Technology on Healthcare Today* . Hong Kong Journal of Emergency Medicine. Vol 9, Issue 4, pp. 231 - 236. 1.10.2002[online]. Dostupné z URL:<<http://www.hkcem.com/html/publications/Journal/2002-4/231-236.pdf>>.
- [2] REKTOROVÁ, I. *Současné možnosti diagnostiky a terapie Parkinsonovy nemoci*. Neurologia pre prax: Supplement 2. 2009, 10(S2). ISSN 1335-9592. 27. 10.1142/S0218001413560053.
- [3] KOLLÁROVÁ, K, RESSNER, P., KANOVSKÝ, P. *Genetika parkinsonovej choroby*. Neurologie pro praxi [online]. 2007, (8(6): 357-359) [cit.30.9.2018]. Dostupné z URL: <<http://www.neurologiepropraxi.cz/pdfs/neu/2007/06/08.pdf>>.
- [4] MEKYSKA, J., SMÉKAL, Z., KOŠTÁLOVÁ, M., MRAČKOVÁ, M., SKUTILOVÁ, S., REKTOROVÁ, I. *Motorické aspekty poruch řeči u Parkinsonovy nemoci a jejich hodnocení*. Brno, 2011.
- [5] JANKOVIC, J. *Parkinson's disease: clinical features and diagnosis*. Journal of Neurology, Neurosurgery [online]. 2008, extbf79(4), 368-376. DOI:10.1136/jnnp.2007.131045. ISSN 0022-3050 [cit.30.9.2018]. Dostupné z URL: <<http://jnnp.bmj.com/cgi/doi/10.1136/jnnp.2007.131045>>.
- [6] Wikipedia contributors. *Dopamine* [online]. Wikipedia, The Free Encyclopedia. 28.8.2018. [cit. 30.9.2018]. Dostupné z URL:<<https://en.wikipedia.org/w/index.php?title=Dopamine\discretionary{-}{-}{-}&oldid=861573721>>.
- [7] FREUND, Z. *Parkinsonova Nemoc*. Parkinson, o.s.[online],2014, 16s [cit.30.9.2018]. Dostupné z URL:<<http://www.spolecnost-parkinson.cz/res/data/039/004149.pdf>>.
- [8] OLANOW, C.,W., STOCCHI, F. *Safinamide – A New Therapeutic Option to Address Motor Symptoms and Motor Complications in Mid-to Late-stage Parkinson's Disease*[online]. European Neurological Review,2016;11(Suppl. 2): 2–15.[cit. 3.11.2018] Dostupné z URL:<<https://www.touchneurology.com/articles/safinamide-new-therapeutic-option-address-motor-symptoms-and-motor-complications-mid-late>>.
- [9] MASSANO, J., KAILASH, P.B. *Clinical Approach to Parkinson's Disease: Features, Diagnosis, and Principles of Management*. 2012. Cold Spring Harbor perspectives in medicine. 2. a008870. 10.1101/cshperspect.a008870.

- [10] WANG Yu-Hsiang. *Tutorial: Image Segmentation*. Graduate Institute of Communication Engineering National Taiwan University, Taipei, Taiwan, ROC [online]. Dostupné z URL:<<http://disp.ee.ntu.edu.tw/meeting/%E6%98%B1%E7%BF%94/Segmentation%20tutorial.pdf>>.
- [11] GLASBEY, C. A., HORGAN, G. W. *Image Analysis for the Biological Sciences*. John Wiley & Sons, Inc., New York, NY, USA. 1995. ISBN:0-471-93726-6.
- [12] Wikipedia contributors. *Grayscale* [online]. Wikipedia, The Free Encyclopedia. 6.9.2018. [cit. 7.10.2018]. Dostupné z URL:<<https://en.wikipedia.org/wiki/Grayscale>>.
- [13] HAMBAL, A.M., PEI, Z., ISHABAILU, F.L. *Image Noise Reduction and Filtering Techniques*. 2015. International Journal of Science and Research (IJSR)[online]. ISSN: 2319-7064. Dostupné z URL:<<https://www.ijssr.net/archive/v6i3/25031706.pdf>>.
- [14] *Concept of Visual Information*. Module 5.8, IIT KANPUR[online], 2009, [cit. 7.10.2018]. Dostupné z URL:<https://nptel.ac.in/courses/117104069/chapter_8/8_32.html>.
- [15] GURUSAMY, V., KANNAN, S., , NALINI, G. *REVIEW ON IMAGE SEGMENTATION TECHNIQUES*[online]. 2014. Dostupné z URL:<https://www.researchgate.net/publication/273127438_REVIEW_ON_IMAGE_SEGMENTATION_TECHNIQUES>.
- [16] Wikipedia contributors. *Thresholding (image processing)* [online]. Wikipedia, The Free Encyclopedia. 8.5.2018 [cit. 6.10.2018]. Dostupné z URL:<[https://en.wikipedia.org/wiki/Thresholding_\(image_processing\)](https://en.wikipedia.org/wiki/Thresholding_(image_processing))>.
- [17] TRUCCO, E. *Edge detection*. Chapter 4[online] Dostupné z URL:<<https://www.cse.unr.edu/~bebis/CS791E/Notes/EdgeDetection.pdf>>.
- [18] Wikipedia contributors. *Edge detection* [online]. Wikipedia, The Free Encyclopedia. 13.5.2018. [cit. 6.10.2018]. Dostupné z URL:<https://en.wikipedia.org/wiki/Edge_detection>.
- [19] Wikipedia contributors. *Region growing* [online]. Wikipedia, The Free Encyclopedia. 3.5.2018. [cit. 6.10.2018]. Dostupné z URL:<https://en.wikipedia.org/wiki/Region_growing>.
- [20] Wikipedia contributors. *Dendrogram* [online]. Wikipedia, The Free Encyclopedia. 11.4.2018. [cit. 30.9.2018]. Dostupné z URL:<<https://en.wikipedia.org/wiki/Dendrogram>>.

- [21] SATHYA, B., MANAVALAN, R. *Image Segmentation by Clustering Methods: Performance Analysis*. (2011) International Journal of Computer Applications. 29. 27-32. 10.5120/3688-5127.
- [22] NAMEIRAKPAM, D., KHUMANTHEM, M., YAMEB, J.CH. *Image Segmentation Using K -means Clustering Algorithm and Subtractive Clustering Algorithm*. 2015. Procedia Computer Science [online]. ISSN 1877-0509. Dostupné z URL:<<http://www.sciencedirect.com/science/article/pii/S1877050915014143>>.
- [23] WÓJCIK, W., GROMASZEK, K., JUNISBEKOV, M. *Face Recognition: Issues, Methods and Alternative Applications*[online]. Face Recognition IntechOpen, 6.7.2016. DOI: 10.5772/62950. [cit. 21.10.2018]. Dostupné z URL:<<https://www.intechopen.com/books/face-recognition-semisupervised-classification-subspace-projection-and-evaluation-methods/face-recognition-issues-methods-and-alternative-applications>>.
- [24] SMITH, I. L. *A tutorial on Principal Components Analysis*. (2002). [online]. Dostupné z URL:<http://www.cs.otago.ac.nz/cosc453/student_tutorials/principal_components.pdf>.
- [25] THARWAT, A., GABER, T., IBRAHIM, A., HASANNIEN, A. E. (2017). *Linear discriminant analysis: A detailed tutorial*. Ai Communications. 30. 169-190,. 10.3233/AIC-170729. [online]. Dostupné z URL:<https://www.researchgate.net/publication/316994943_Linear_discriminant_analysis_A_detailed_tutorial>.
- [26] HYVARINEN, A., OJA, E. *Independent Component Analysis: Algorithms and Applications*. Neural Networks Research Centre, Helsinki University of Technology. [online]. Dostupné z URL:<<https://www.cs.helsinki.fi/u/ahyvarin/papers/NN00new.pdf>>.
- [27] ROOMI, M., BEHAM, Dr.M.P. *A Review Of Face Recognition Methods*[online]. International Journal of Pattern Recognition and Artificial Intelligence. [cit. 21.10.2018]. Dostupné z URL:<https://www.researchgate.net/publication/274521637_A_Review_Of_Face_Recognition_Methods>.
- [28] CHAO, W.L. *Face recognition*. GICE, National Taiwan University. [cit. 21.10.2018]. Dostupné z URL:<<http://disp.ee.ntu.edu.tw/~pujols/Face%20Recognition-survey.pdf>>.

- [29] LINDEBERG, T. *Scale Invariant Feature Transform*. (2012). 10.4249/scholarpedia.10491. [online]. Dostupné z URL:<https://www.researchgate.net/publication/235355151_Scale_Invariant_Feature_Transform>.
- [30] HABEEB, A. *Introduction to Artificial intelligence* (2017) [online]. Dostupné z URL:<https://www.researchgate.net/publication/325581483_Introduction_to_Artificial_intelligence>.
- [31] BROWNLEE, J. *Supervised and Unsupervised Machine Learning Algorithms* (2016) [online]. Dostupné z URL:<<https://machinelearningmastery.com/supervised-and-unsupervised-machine-learning-algorithms/>>.
- [32] CIBULA, M. *Strojové učenie - Učenie s učiteľom* (2017) [online]. Dostupné z URL:<<http://smnd.sk/mcibula/fungovanie/fungovanie-sl.html>>.
- [33] REFAEILZADEH, P., TANG, L., LIU, H. *Cross-Validation*. Arizona State University [online]. Dostupné z URL:<<http://leitang.net/papers/ency-cross-validation.pdf>>.
- [34] ARLOT, S., CELISSE, A. *A survey of cross-validation procedures for model selection*. Statistics Surveys, Vol. 4 (2010) 40–79. ISSN: 1935-7516. DOI: 10.1214/09-SS054. [online]. Dostupné z URL:<https://www.di.ens.fr/willow/pdfs/2010_Arlot_Celisse_SS.pdf>.
- [35] SONI, D. *Supervised vs. Unsupervised Learning*. (2018) [online]. Dostupné z URL:<<https://towardsdatascience.com/supervised-vs-unsupervised-learning-14f68e32ea8d>>.
- [36] BROWNLEE, J. *Difference Between Classification and Regression in Machine Learning*. (2017) [online]. Dostupné z URL:<<https://machinelearningmastery.com/classification-versus-regression-in-machine-learning/>>.
- [37] CHAI, T., DRAXLER, R.R. *Root mean square error (RMSE) or mean absolute error (MAE)? – Arguments against avoiding RMSE in the literature*. (2014) [online]. Dostupné z URL:<<https://www.geosci-model-dev-discuss.net/7/C473/2014/gmdd-7-C473-2014-supplement.pdf>>.
- [38] DE MZTTENAERE, A., GOLDEN, B., LE GRAND, B., ROSSI, F. *Mean Absolute Percentage Error for regression models*. Neurocomputing, Volume 192, 2016, Pages 38-48, ISSN 0925-2312.
- [39] BARTELS, R. *Re-interpreting R-squared, regression through the origin, and weighted least squares*. (2015) [online]. Dostupné

- z URL:<https://www.researchgate.net/publication/283333191_Re-interpreting_R-squared_regression_through_the_origin_and_weighted_least_squares>.
- [40] BRADSKI, G. *The OpenCV Library*. [online]. Dostupné z URL:<<https://www.geosci-model-dev-discuss.net/7/C473/2014/gmdd-7-C473-2014-supplement.pdf>>.
 - [41] SOO, S. *Object detection using Haar-cascade Classifier*[online]. Institute of Computer Science, University of Tartu.[cit. 3.11.2018] Dostupné z URL:<<https://pdfs.semanticscholar.org/0f1e/866c3acb8a10f96b432e86f8a61be5eb6799.pdf>>.
 - [42] SCHROFF, F., KALANICHENKO, D., PHILBIN, J. *Facenet: A unified embedding for face recognition and clustering*. Proceedings of the IEEE conference on computer vision and pattern recognition, s. 815–823, 2015.
 - [43] Wikipedia contributors. *JSON* [online]. Wikipedia, The Free Encyclopedia. 10.10.2018 [cit. 22.10.2018]. Dostupné z URL:<<https://en.wikipedia.org/wiki/JSON>>.
 - [44] KINLI, F. *[Deep Learning Lab] Episode-3: fer2013*[online]. Dostupné z URL:<<https://medium.com/deep-learning-turkey/deep-learning-lab-episode-3-fer2013-c38f2e052280>>.
 - [45] SHAKIROV, N. *Face recognition: realtime masks development*[online]. Dostupné z URL:<<https://becominghuman.ai/face-recognition-realtime-masks-development-9d3a399b4c3>>.
 - [46] FFmpeg Developers. *ffmpeg tool*. (2016) [Software]. Dostupné z URL:<<http://ffmpeg.org/>>.
 - [47] Google Code Archive. *Musicg*. (2012) [Software]. Dostupné z URL:<<https://code.google.com/archive/p/musicg/>>.
 - [48] BLAND, J.M., ALTMAN, D.G. *Statistics notes: measurement error*. (1996) BMJ. 312 (7047): 1654. doi:10.1136/bmj.312.7047.1654. [online]. Dostupné z URL:<<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2351401/>>.
 - [49] CSV, *Comma Separated Values (RFC 4180)*. (2012) [online]. Dostupné z URL:<<https://www.loc.gov/preservation/digital/formats/fdd/fdd000323.shtml>>.

- [50] Pedregosa et al. *Scikit-learn: Machine Learning in Python*. JMLR 12, pp. 2825-2830, 2011. [online]. Dostupné z URL:<<https://scikit-learn.org/>>.
- [51] *Python Data Analysis Library*. SciPy (2019). [online]. Dostupné z URL:<<https://pandas.pydata.org/>>.
- [52] CUNNINGHAM, P., DELANY, S. *K-Nearest neighbour classifiers*. (2007). [online]. Dostupné z URL:<https://www.researchgate.net/publication/228686398_k-Nearest_neighbour_classifiers>.
- [53] YAN, X. *Linear Regression Analysis: Theory and Computing*, World Scientific. (2009), pp. 1–2, ISBN 9789812834119
- [54] ROKACH, L., MAIMON, O. *Decision Trees*. (2005). 10.1007/0-387-25465-X_9. [online]. Dostupné z URL:<https://www.researchgate.net/publication/225237661_Decision_Trees>.
- [55] CHEN, T., GUESTRIN, C. *XGBoost: A Scalable Tree Boosting System*. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016. doi:10.1145/2939672.2939785.
- [56] LIAW, A., WIENER, M. *Classification and Regression by Random-Forest*. ISSN 1609-3631 (2001). [online]. Dostupné z URL:<https://www.researchgate.net/publication/228451484_Classification_and_Regression_by_RandomForest>.

Zoznam symbolov, veličín a skratiek

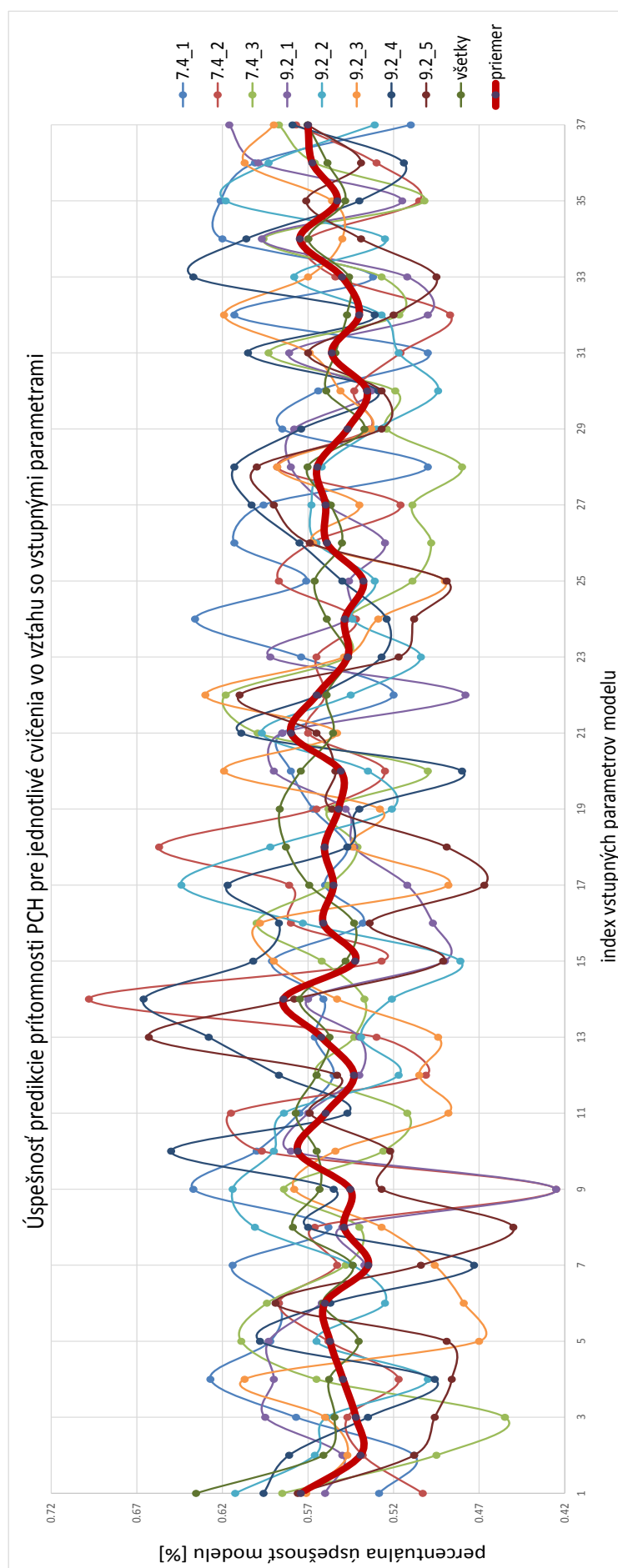
KNN	K-nearest neighbors
MAE	Mean absolute error
MAPE	Mean absolute percentage error
RMSE	Root mean squared error
ICA	Independent Component Analysis
IT	Informačné technológie
LDA	Linear Discriminant Analysis
m	meter
PCA	Principal Component Analysis
PCH	Parkinsonova choroba
RSD	Relative standard deviation
SIFT	Scale-invariant feature transform
SD	Standard deviation

Zoznam príloh

A	Graf úspešnosti predikcie prítomnosti PCH pre jednotlivé cvičenia vo vzťahu so vstupnými parametrami(SD–RSD data)	75
B	Výsledky predikcie progresu PCH a jej symptómov u pacienta(SD–RSD dáta)	77
C	Výsledky predikcie progresu PCH a jej symptómov u pacienta(dáta časových úsekov)	81
D	Obsah príloženého CD	83

A Graf úspešnosti predikcie prítomnosti PCH pre jednotlivé cvičenia vo vzťahu so vstupnými parametrami(SD–RSD data)

Tento graf zobrazuje úspešnosť natrénovaných modelov jednotlivých cvičení v závislosti na zvolených vstupných hodnotách, na základe ktorých mal model predikovať. Vstupné hodnoty sú vyjadrené číselným indexom na ose x od 1 po 37. Index 1 označuje prípad, kedy boli ako vstupný parameter zvolené všetky parametre tváre. Indexy 2 až 37 sú prípady, kedy bol zvolený ako vstupný parameter vždy len jeden parameter tváre. Parametre idú po sebe v poradí: plocha úst, šírka úst, výška úst, šírka ľavého obočia, výška ľavého obočia, šírka pravého obočia, výška pravého obočia, vzdialenosť no s– ústa, vzdialenosť nos – brada. Toto poradie je opakované dva krát, raz pre normalizované hodnoty a raz pre normalizované rozdielové hodnoty. Každý parameter tváre je definovaný dvomi hodnotami v poradí: smerodajná odchýlka a relatívna smerodajná odchýlka parametru daného cvičenia. Vo výsledku to znamená 9 parametrov tváre, dva krát po sebe pre normalizované a normalizované rozdielové dáta, pričom každý parameter je definovaný dvomi hodnotami odchýlok. Celkovo sa to rovná tridsiatim šiestim rôznym typom vstupných hodnôt. V grafe je okrem rád zobrazujúcich úspešnosť modelov jednotlivých cvičení a rady úspešnosti modelov súboru datasetu všetkých cvičení zobrazená aj rada, ktorá zobrazuje priemer hodnôt ostatných rád.



Obr. A.1: Graf úspešnosti predikcie prítomnosti PCH pre jednotlivé cvičenia vo vzťahu so vstupnými parametrami.

B Výsledky predikcie progresu PCH a jej symptómov u pacienta(SD–RSD dáta)

Táto príloha obsahuje výsledky modelov s najlepšimi hodnotami validačných metrík pre každý jeden symptóm. Tieto modely boli vytvorené na základe dát SD–RSD časti vytvoreného datasetu. Výsledky sú vo forme tabuliek, pričom každá tabuľka zobrazuje pre každý jeden symptóm jeho najúspešnejší model podľa validačnej metriky. Stĺpce tabuliek sú zoradené podľa hodnoty metriky od najlepšej po najhoršiu. Ako validačné metriky sú zvolené MAE, MAPE a RMSE.

Symptóm	Parameter	MAE	Cvičenie	Algoritmus
RBD SQ	RSD výšky ľavého obočia	2,45	7.4_2	lineárna regresia
faciokinéza	RSD šířky úst	2,469	7.4_3	lineárna regresia
fonorespirácia	SD rozdielovej šířky ľavého obočia	2,523	9.2_5	lineárna regresia
fonetika	RSD plochy úst	2,726	9.2_5	lineárna regresia
doba trvania PCH	všetky	3,424	všetky	lineárna regresia
BDI	RSD vzdialenosti nos – brada	4,262	9.2_2	rozhodovacie stromy
FOG	RSD výšky pravého obočia	4,434	7.4_3	lineárna regresia
ACE-R	SD rozdielovej vzdialenosti nos – brada	6,07	7.4_3	lineárna regresia
celkový DX index	SD rozdielovej vzdialenosti nos – brada	6,439	7.4_3	lineárna regresia
UPDRS III	všetky	8,697	všetky	lineárna regresia
NMSS	RSD vzdialenosť nos – ústa	18,039	9.2_5	lineárna regresia

Tab. B.1: Najlepšie modely odhadu progresu symptómov pacienta pre jednotlivé symptómy podľa MAE (SD-RSD dáta).

Symptóm	Parameter	MAPE [%]	Cvičenie	Algoritmus
ACE-R	SD rozdielovej vzdialenosti nos – brada	7,37	7.4_3	lineárna regresia
celkový DX index	SD rozdielovej vzdialenosti nos – brada	9,917	7.4_3	lineárna regresia
faciokinéza	RSD šírky úst	11,296	7.4_3	lineárna regresia
fonorespirácia	SD rozdielovej šírky ľavého obočia	11,2976	9.2_5	lineárna regresia
fonetika	RSD plochy úst	11,613	9.2_5	lineárna regresia
BDI	RSD rozdielovej vzdialenosti nos – ústa	51,5684	7.4_2	XGBoost
UPDRS III	všetky	56,853	všetky	lineárna regresia
doba trvania PCH	všetky	63,259	všetky	XGBoost
RBDSQ	všetky	80,9217	všetky	XGBoost
FOG	všetky	105,813	všetky	XGBoost
NMSS	SD rozdielovej vzdialenosti nos – brada	123,107	7.4_3	XGBoost

Tab. B.2: Najlepšie modely odhadu progresu symptómov pacienta pre jednotlivé symptómy podľa MAPE (SD–RSD dáta).

Symptóm	Parameter	RMSE	Cvičenie	Algoritmus
RBDSQ	RSD rozdielovej výšky ľavého obočia	3,053	7.4_2	lineárna regresia
faciokinéza	RSD šírky úst	3,185	7.4_3	lineárna regresia
fonorespirácia	SD rozdielovej šírky ľavého obočia	3,214	7.4_3	lineárna regresia
fonetika	RSD plochy úst	3,337	9.2_5	lineárna regresia
doba trvania PCH	SD rozdielovej výšky pravého obočia	4,146	7.4_1	lineárna regresia
FOG	RSD šírky pravého obočia	5,248	7.4_3	lineárna regresia
BDI	RSD šírky ľavého obočia	5,36	7.4_1	lineárna regresia
ACE-R	SD rozdielovej vzdialenosti nos – brada	7,741	7.4_3	lineárna regresia
celkový DX index	RSD rozdielovej šírky úst	8,461	7.4_1	lineárna regresia
UPDRS III	všetky	10,722	všetky	lineárna regresia
NMSS	RSD šírky pravého obočia	22,494	7.4_3	lineárna regresia

Tab. B.3: Najlepšie modely odhadu progresu symptómov pacienta pre jednotlivé symptómy podľa RMSE (SD–RSD dáta).

C Výsledky predikcie progresu PCH a jej symptómov u pacienta(dáta časových úsekov)

Táto príloha obsahuje výsledky modelov s najlepšimi hodnotami validačných metrík pre každý jeden symptóm v rovnakom tvare, ako v prílohe B. Rozdiel je v type dát, na základe ktorých boli trénované modely predikcie. V tomto prípade zobrazujú tabuľky úspešnosť modelov trénovaných na dátach časových úsekov videa.

Symptóm	Cvičenie	MAE	Algoritmus
RBDSQ	všetky – rozdielové hodnoty	1,584	XGBoost
faciokinéza	všetky – rozdielové hodnoty	1,756	XGBoost
fonorespirácia	všetky – rozdielové hodnoty	1,901	XGBoost
fonetika	všetky – rozdielové hodnoty	2,1	XGBoost
doba trvania PCH	všetky	2,162	XGBoost
BDI	všetky – rozdielové hodnoty	3,03	XGBoost
FOG	všetky – rozdielové hodnoty	3,09	XGBoost
ACE–R	všetky – rozdielové hodnoty	4,15	XGBoost
celkový DX index	všetky	4,567	XGBoost
UPDRS III	všetky – rozdielové hodnoty	5,584	XGBoost
NMSS	všetky – rozdielové hodnoty	12,357	XGBoost

Tab. C.1: Najlepšie modely odhadu progresu symptómov pacienta pre jednotlivé symptómy podľa MAE (dáta časových úsekov).

Symptóm	Cvičenie	MAPE [%]	Algoritmus
ACE-R	všetky – rozdielové hodnoty	5,317	XGBoost
faciokinéza	všetky – rozdielové hodnoty	7,449	XGBoost
celkový DX index	všetky	7,641	XGBoost
fonorespirácia	všetky – rozdielové hodnoty	8,248	XGBoost
fonetika	všetky – rozdielové hodnoty	8,69	XGBoost
BDI	všetky – rozdielové hodnoty	30,281	XGBoost
UPDRS III	všetky – rozdielové hodnoty	33,05	XGBoost
doba trvania PCH	všetky	33,85	XGBoost
RBDSQ	všetky – rozdielové hodnoty	41,817	XGBoost
FOG	všetky – rozdielové hodnoty	49,489	XGBoost
NMSS	všetky – rozdielové hodnoty	76,940	XGBoost

Tab. C.2: Najlepšie modely odhadu progresu symptómov pacienta pre jednotlivé symptómy podľa MAPE (dáta časových úsekov).

Symptóm	Cvičenie	RMSE	Algoritmus
faciokinéza	9.2_2	2,393	rozhodovacie stromy
RBDSQ	7.4_3 – rozdielové hodnoty	2,456	XGBoost
fonetika	9.2_5	2,534	XGBoost
fonorespirácia	7.4_3 – rozdielové hodnoty	2,684	rozhodovacie stromy
doba trvania PCH	7.4_1	2,991	XGBoost
FOG	7.4_3	3,528	rozhodovacie stromy
BDI	9.2_2 – rozdielové hodnoty	4,703	XGBoost
ACE-R	7.4_1 – rozdielové hodnoty	5,295	rozhodovacie stromy
celkový DX index	7.4_1	7,117	XGBoost
UPDRS III	9.2_1 – rozdielové hodnoty	8,644	náhodný les
NMSS	7.4_2	16,959	rozhodovacie stromy

Tab. C.3: Najlepšie modely odhadu progresu symptómov pacienta pre jednotlivé symptómy podľa RMSE (dáta časových úsekov).

D Obsah přiloženého CD

Obsah přiloženého CD sa dá rozdeliť na tri časti: súbory datasetu, zdrojové kódy vytvorených programov a súbory obsahujúce výsledky tréovania jednotlivých modelov. Adresárová štruktúra CD je nasledová:

```
/
├── dataset
│   ├── SD_RSD ..... súbory datasetu SD–RSD dát
│   └── time_interval ..... súbory datasetu s dátami časových úsekov
├── programy ..... zdrojové kódy vytvorených programov
│   ├── FeatureVectorsProcessor
│   │   └── ...
│   ├── JsonExerciseProcessor
│   │   └── ...
│   ├── LabelSynchronizator
│   │   └── ...
│   ├── MouthSegmentate
│   │   └── ...
│   ├── TimePeriodFeatureVectorsProcessor
│   │   └── ...
│   ├── model52p.hdf5 ..... natrénovaný model použitý pre odhad emócií
│   ├── prediction-SD_RSD.py
│   ├── prediction-time_interval.py
│   └── segmentate_mouth.py
└── vysledky ..... adresár obsahujúci súbory popisujúce výsledky modelov
    ├── SD_RSD ..... výsledky pre SD–RSD dáta
    └── time_interval ..... výsledky pre dáta časových úsekov
```