

UNIVERZITA KOMENSKÉHO V BRATISLAVE
FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY



Aplikácia metód výpočtovej inteligencie pre zhlukovanie
osobných fotografií

DIPLOMOVÁ PRÁCA

2018

Bc. Jakub KOVÁČ

UNIVERZITA KOMENSKÉHO V BRATISLAVE
FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY

**Aplikácia metód výpočtovej inteligencie pre zhlukovanie
osobných fotografií**

DIPLOMOVÁ PRÁCA

Študijný program: ekonomicko-finančná matematika a modelovanie

Študijný odbor: 1114 Aplikovaná matematika

Školiace pracovisko: Katedra aplikovanej matematiky a štatistiky

Vedúci práce: prof. Ing. Štefan Kozák, PhD.

Bratislava 2017

Bc. Jakub KOVÁČ



Univerzita Komenského v Bratislave
Fakulta matematiky, fyziky a informatiky

ZADANIE ZÁVEREČNEJ PRÁCE

Meno a priezvisko študenta: Bc. Jakub Kováč
Študijný program: ekonomicko-finančná matematika a modelovanie
(Jednoodborové štúdium, magisterský II. st., denná forma)
Študijný odbor: aplikovaná matematika
Typ záverečnej práce: diplomová
Jazyk záverečnej práce: slovenský
Sekundárny jazyk: anglický

Názov: Aplikácia metód výpočtovej inteligencie pre zhľukovanie osobných fotografií
Application of computational intelligence techniques in clustering personal photographs

Anotácia: V práci sa navrhujú a aplikujú efektívne metódy výpočtovej inteligencie pre rozpoznávanie a zhľukovanie tvárí.

Vedúci: prof. Ing. Štefan Kozák, PhD.
Katedra: FMFI.KAMŠ - Katedra aplikovanej matematiky a štatistiky
Vedúci katedry: prof. RNDr. Daniel Ševčovič, DrSc.
Dátum zadania: 26.01.2017

Dátum schválenia: 27.01.2017
prof. RNDr. Daniel Ševčovič, DrSc.
garant študijného programu

.....
študent

.....
vedúci práce

Podakovanie Rád by som sa chcel poďakovať vedúcej svojej diplomovej práce prof. Ing. Štefanovi Kozákovi, PhD. za pomoc, ochotu a odborné rady pri formulovaní mojich myšlienok do slov. Tiež by som sa chcel poďakovať svojim priateľom za konštruktívnu kritiku mojich nápadov počas viacerých osobných diskusií a tiež pomoc pri kontrole gramatických chýb.

Abstrakt

KOVÁČ, Jakub: Aplikácia metód výpočtovej inteligencie pre zhlukovanie osobných fotografií [Diplomová práca], Univerzita Komenského v Bratislave, Fakulta matematiky, fyziky a informatiky, Katedra aplikovanej matematiky a štatistiky; školiteľ: prof. Ing. Štefan Kozák, PhD., Bratislava, 2018, 94s.

V diplomovej práci sa zaoberáme návrhom a aplikáciou metód a algoritmov pre automatické zhlukovanie fotografií osôb, nachádzajúcich sa na obrázkoch, s využitím vybraných metód výpočtovej inteligencie využívajúcich najmä metódy učenia bez učiteľa. Na základe existujúcich súčasných trendov v predmetnej oblasti sú v diplomovej práci analyzované výhody a nevýhody existujúcich algoritmov pre zhlukovanie tvárí, pričom poukazujeme na možnosti tvorby vektorov črt s cieľom zvýšenia efektívnosti jednotlivých algoritmov. Na základe analýzy uvedených algoritmov je v diplomovej práci navrhovaný modifikovaný algoritmus zhlukovania tvárí úpravou a zavedením nových postupov a využitím dodatočných informácií skrytých v digitálnych fotografiách. Výsledky navrhovaného modifikovaného algoritmu rozpoznávania sú v diplomovej práci overené na vybraných príkladoch a porovnané s pôvodnými klasickými algoritmami. Navrhujeme spôsob, akým môžeme využiť tieto dodatočné informácie v ľubovoľnom zhlukovacom algoritme. Jadro práce tvorí naprogramovanie jednotlivých algoritmov a ich otestovanie na reálnych údajoch. V práci sme sa tiež, pre naše využitie, zamerali na odhadnutie optimálnych parametrov rôznych zhlukovacích algoritmov, nakoľko spôsob voľby týchto parametrov býva v odborných článkoch často, niekedy možno zámerne, zamlčaný. V závere diplomovej práce je uvedené celkové zhodnotenie, porovnanie jednotlivých algoritmov z hľadiska rôznych kritérií a možnosti ich ďalšieho využitia.

Kľúčové slová: Zhlukovanie, Generácia vektorov črt, Pattern recognition, Učenie bez učiteľa, Rozpoznávanie osôb

Abstract

KOVÁČ, Jakub: Application of computational intelligence techniques in clustering personal photographs [Master Thesis], Comenius University in Bratislava, Faculty of Mathematics, Physics and Informatics, Department of Applied Mathematics and Statistics; Supervisor: prof. Ing. Štefan Kozák, PhD., Bratislava, 2018, 94p.

In this master thesis, we deal with the design and application of methods and algorithms for automatic clustering of photographs of persons faces found in pictures using selected methods of computational intelligence using unsupervised learning methods. Based on existing current trends in the subject area, the thesis analyzes the advantages and disadvantages of existing face clustering algorithms while pointing out the possibilities of feature generation in order to increase the efficiency of individual algorithms. Based on the analysis of these algorithms, the thesis proposes a modified face clustering algorithm by modifying and introducing new methods and utilising additional information hidden in digital images. The results of the proposed modified clustering algorithm are verified in the master thesis on chosen examples and compared with the original classical algorithm. Additionally, we propose a way to utilise hidden additional information in arbitrary clustering algorithm. The core of the work consists in the programming of individual algorithms and their testing on real data. In the thesis, we also focused on estimation, for our purposes, of the optimal parameters of various clustering algorithms, whereas the way of choosing parameters in scientific articles is often, perhaps deliberately, omitted. In the conclusion of the master thesis, the overall evaluation and comparison of the individual recognition algorithms are presented and the possibilities of their further use are pointed out.

Key words: Clustering, Feature generation, Pattern recognition,
Unsupervised learning, Person identification

Obsah

Zoznam obrázkov	9
Zoznam tabuliek	11
Zoznam použitých symbolov	13
Úvod	17
1 Metodika rozpoznávania tvárí	19
1.1 Rozpoznávanie tvárí	20
1.1.1 Digitálne fotografie	20
1.2 Generácia vektorov črt	21
1.2.1 Exchangeable Image File Format	22
1.2.2 Redukcia dimenzie	25
1.2.3 VGG-16 a Alexnet	27
1.2.4 HOG	28
1.3 Zhlukovanie	29
1.3.1 K-means	30
1.3.2 K-medoids	32
1.3.3 Spektrálne zhlukovanie	35
1.3.4 DBSCAN	40
1.3.5 Ukážkový príklad	40
1.3.6 Zhlukovanie založené na poradí	42
1.3.7 Aglomeratívne zhlukovanie	45
1.3.8 Zhlukovanie GPS súradníc	49

1.4	Vyhodnocovanie výsledkov zhukovania	49
1.4.1	Adjusted Rand Index	50
1.4.2	Kritériá Precision, Recall, Compression Ratio	50
1.5	Kombinácia rôznych zhukovaní	51
1.5.1	Naivný prístup	52
1.5.2	Adaptácia zhukovania založeného na poradí	52
1.6	Voľba počtu zhukov	53
1.6.1	Siluety	54
1.6.2	Caliński-Harabasz kritérium	55
2	Demonštračné príklady	57
2.1	Voľba parametrov na samostatnej databáze	57
2.1.1	Parametre pre metódu DBSCAN	58
2.1.2	Parametre pre metódu zhukovania založeného na poradí	62
2.1.3	Parametre pre spektrálne zhukovanie	64
2.2	Výsledky pre jednoduchú databázu fotografií	68
2.3	Výsledky na zložitejšej databáze aj s EXIF dátami	76
2.4	Adaptované zhukovanie na poradí	85
	Záver	89
	Literatúra	93

Zoznam obrázkov

1.1	Základná schéma uvažovaného problému	19
1.2	Ukážkový príklad nájdených tvárí	20
1.3	Zobrazenie obrázku v RGB a odtieni šedej	21
1.4	Ukážkový príklad vykreslenia gps súradníc pre všetky fotografie zvolenej databázy	23
1.5	Ukážkový príklad zoradenej časovej známky	24
1.6	Laktový diagram pre ukážkový príklad PCA	26
1.7	Prvé 2 hlavné komponenty pre ukážkový príklad PCA	26
1.8	Ukážkový príklad vykreslenia hlbokých črt v škále jet pre ľudskú tvár . . .	28
1.9	Ukážkový príklad vykreslenia orientovaných gradientov pre ľudskú tvár . .	29
1.10	Ukážkový príklad špirálovitých dát a matice podobnosti 7-najbližších su- sedov váhovanej $\sigma = 20$	41
1.11	Ukážkový príklad zhlukovania pomocou K -means, K -medoids, Spektrál- neho zhlukovania a DBSCAN pre rôzne parametre ε a $\minPts$	42
1.12	Ukážkový príklad pre zhlukovanie s dvojrozmernými dátami	47
1.13	Ukážkový príklad hard aglomeratívneho zhlukovania pre 2 zhluky	48
1.14	Ukážkový príklad pílenia dendrogramu pre single linkage aglomeratívne zhlukovanie a počet zhlukov 2	49
2.1	2 fotografie z databázy Caltech Faces	58
2.2	3-rozmerný graf závislosti ARI od parametrov ε a $\minPts$	59
2.3	3-rozmerný graf závislosti ARI od parametrov ε a $\minPts$	60
2.4	3-rozmerný graf závislosti ARI od parametrov ε a $\minPts$	61
2.5	3-rozmerný graf závislosti ARI od parametrov ε a $\minPts$	62

2.6	Hodnota ARI pre rôzne hodnoty parametrov zhhlukovania na poradí	63
2.7	Hodnota ARI pre rôzne hodnoty parametra σ plného grafu podobnosti a spektrálneho zhhlukovania	65
2.8	ARI pre rôzne parametre KNN, spoločných susedov a spektrálneho zhhlukovania	66
2.9	ARI pre rôzne parametre KNN, spoločných susedov a spektrálneho zhhlukovania	67
2.10	Ukážkový príklad dvoch ľudských tvárí z jednoduchej databázy	69
2.11	Hierarchické zhhlukovanie databázy pri 2 zhlukoch, VGG16 a average linkage	75
2.12	2 fotografie zo súkromnej databázy	76
2.13	Krajinné zhluky GPS súradníc	77
2.14	Problém pri krajinných zhlukoch	78
2.15	Zhlukovanie času pre rôzne hodnoty veľkosti skupín	79
2.16	Kritériá pre voľbu počtu zhlukov pre metódu K-means na tvárach charakterizovaných pomocou siete Alexnet	80
2.17	Graf siluet pre K-means zhhlukovanie s 2 zhlukmi	80
2.18	Zhlukovanie časových údajov pre databázu zo sekcie 2.1	85
2.19	Hodnoty kritérií <i>ARI</i> a <i>Precision</i> pre adaptované, naivné a klasické rodbc .	86
2.20	Hodnota kritéria <i>Recall</i> pre adaptované, naivné a klasické rodbc	86

Zoznam tabuliek

2.1	Optimálna voľba parametrov a prislúchajúca hodnota ARI pre metódu DBSCAN	61
2.2	Optimálna voľba parametrov a prislúchajúca hodnota ARI pre metódu zhlukovania na poradi	64
2.3	Optimálna voľba parametra σ a prislúchajúca hodnota ARI pre spektrálne zhlukovanie, plný graf a L_{sym}	64
2.4	Optimálna voľba parametrov a prislúchajúca hodnota ARI pre spektrálne zhlukovanie a graf KNN	68
2.5	Hodnoty parametrov metód K-means a K-medoids a výsledná hodnota ARI	70
2.6	Hodnoty parametrov spektrálneho zhlukovania a výsledná hodnota ARI . .	72
2.7	Hodnoty parametrov zhlukovania založeného na poradi a metódy DBSCAN a ich výsledná hodnota ARI	73
2.8	Hodnoty parametrov hierarchického zhlukovania a metódy PCA a výsledná hodnota ARI	73
2.9	Hodnoty parametrov hierarchického zhlukovania, metódy HOG, Alexnet a výsledná hodnota ARI	74
2.10	Hodnoty parametrov hierarchického zhlukovania a metódy VGG16 a výsledná hodnota ARI	75
2.11	Zhlukovanie pomocou K-means a Alexnet	81
2.12	Klasické, naivné a adaptované zhlukovanie založené na poradi pre $\beta = 0.5$.	82
2.13	Výsledky pre spektrálne zhlukovanie na zvolenej databáze	84
2.14	Výsledky pre hierarchické zhlukovanie pri použití <i>Alexnet</i>	84

Zoznam použitých symbolov

$X_{n \times p}$	$n \times p$ rozmerná matica dizajnu
$EXIF$	Exchangeable Image File Format
PCA	Principal Component Analysis
HOG	Histogram of Oriented Gradients
K	počet zhlukov
x_i	i -ty objekt (i -ty riadok matice dizajnu)
$\overline{x_k}$	k -ty centroid
C_k	k -ty zhluk
$ C_k $	kardinalita množiny C_k
$WCSS$	Within Cluster Sum-of-Squares
$d(x_i, x_j)$	nepodobnosť bodu x_i a x_j
D_{ij}	nepodobnosť bodu x_i a x_j
m_k	k -ty medoid
PAM	Partitioning Around Medoids
D	matica nepodobnosti
S_e	množina zvolených bodov
U	množina nezvolených bodov
D_i	nepodobnosť medzi objektom i a najbližším objektom z S
E_i	nepodobnosť medzi objektom i a druhým najbližším objektom z S
c_{ji}	kontribúcia j do i
g_i	celkový zisk pre zaradenie i do S
T_{ih}	efekt zámeny (i, h) , kde $i \in S, h \in U$
K_{jih}	kontribúcia j po zámene (i, h)
ε	veľkosť okolia bodu

G	neocenený graf
V	množina vrcholov
E	množina hrán
$\tilde{G}$	ocenený graf
$\tilde{E}$	množina ocenených hrán
$\mathbb{R}_0^+$	množina nezáporných reálnych čísel
e_{ij}	hrana spájajúca vrcholy i a j
w_{ij}	ocenenie hrany spájajúcej vrcholy i a j
W	matica podobnosti
s_i	stupeň vrchola v_i
S	stupňová matica grafu
A^C	doplnok podgrafu A
Υ	indikátorový vektor pre graf
$\mathcal{K}$	počet uvažovaných susedov
L	nenormalizovaný Laplaceovský graf
L_{sym}	symetricky normalizovaný Laplaceovský graf
L_{rw}	normalizovaný Laplaceovský graf predstavujúci náhodnú prechádzku
$minPts$	minimálny nutný počet bodov v ε -ovom okolí pre jadrový bod
ARI	Adjusted Rand Index
O_a	poradovník tváre a
$D(a, b)$	asymetrická vzdialenosť tváre a a b založená na poradí
$O_a(b)$	poradie tváre b v poradovníku tváre a
$f_a(k)$	k -ta najbližšia tvár k tvári a
$D^R(a, b)$	vzdialenosť tváre a a b založená na poradí
$d_{SL}(C_i, C_j)$	jednoduchá (single linkage) vzdialenosť zhlukov C_i a C_j
$D^R(C_i, C_j)$	vzdialenosť zhlukov C_i a C_j založená na poradí
$D^N(C_i, C_j)$	normovaná vzdialenosť zhlukov C_i a C_j založená na poradí
$\phi(C_i, C_j)$	priemerná vzdialenosť tvárí zo zhlukov C_i a C_j
$d_{SL}(C_i, C_j)$	complete linkage vzdialenosť zhlukov C_i a C_j
$d_{AL}(C_i, C_j)$	average linkage vzdialenosť zhlukov C_i a C_j
$d_{WL}(C_i, C_j)$	Ward's linkage vzdialenosť zhlukov C_i a C_j

$D_{AC}^N(C_i, C_j)$	normovaná vzdialenosť zhlukov C_i a C_j s apriori zhlukovou informáciou
$\xi(C_i, C_j)$	proporcía objektov zo zhlukov C_i a C_j zdieľajúcich rovnaký apriori zhluk
$\tilde{\delta}_{AC_a, AC_b}$	upravené Kroneckerovo delta
$s_i(I_K)$	silueta pre dané zhlukovanie I_K
$\overline{s_K}$	priemerná šírka siluety pre dané zhlukovanie I_K
SC	koeficient siluety
VRC_K	Variance Ratio Criterion
S_B	mimo-zhluková variancia
S_W	vnútro-zhluková variancia

Úvod

I believe that at the end of the century the use of words and general educated opinion will have altered so much that one will be able to speak of machines thinking without expecting to be contradicted.

Alan Turing

V dnešnej dobe je vytváranie digitálnych fotografií úplne každodennou činnosťou. Či už sú vyhotovené zrkadlovými, bezzrkadlovými, kompaktnými fotoaparátmi alebo len pomocou mobilných telefónov, každý z nás vlastní desiatky či stovky gigabajtov fotografií z osobného života. Pri takom množstve môže byť veľmi náročné mať medzi nimi prehľad. Tu prichádza na rad pomoc počítača. Strojové učenie (*machine learning*) a disciplína s názvom počítačové videnie (*computer vision*) prešli v posledných rokoch obrovským vývojom a využiť ich na organizáciu osobných fotografií nám môže pomôcť pri obnovovaní spomienok.

Naším cieľom bude zatriediť fotografie podľa osôb, ktoré sa na nich nachádzajú. Prvým krokom je teda automatické rozpoznávanie tvárí v obrázkoch (*Face Detection*). Tieto budú musieť byť pred ďalším krokom vyňaté z jednotlivých fotografií a spracované.

Druhým krokom je generácia vektorov črt (*Feature Generation*). Jednotlivé pixely na obrázkoch predstavujú príliš obrovské množstvo informácie, častokrát aj nežiadúcej, ktoré by spomalilo rýchlosť výpočtov. Preto budeme využívať viaceré postupy ako redukovať dimenziu a dostaneme nové vektory črt, s ktorými budeme pracovať. Nie sú to však jediné informácie, ktoré budeme využívať pre posledný krok. Využívame tiež štandardizovaný formát dát *EXIF*, ktorý ukrýva dodatočné informácie o samotných fotografiách. Týmto sú napríklad čas vytvorenia fotografie, GPS súradnice, kde bola fotografia vytvorená, ale aj údaje o prístroji, ktorý tieto fotografie vyhotovil ako napríklad číslo clony, čas

uzávierky alebo ISO. My však v práci využívame len čas vytvorenia fotografie a geografické súradnice, samozrejme len vtedy, ak sú tieto informácie prístupné.

Tretím a posledným krokom bude zatriedenie tvárí do jednotlivých zhlukov (*clusterov*) a následné spätné zatriedenie fotografií. Tejto časti sme v práci venovali najviac pozornosti. Okrem tradičných zhlukovacích algoritmov ako je K -means a K -medoids, sme sa pozreli aj na pokročilejšie všeobecné metódy DBSCAN a spektrálne zhlukovanie. Výber metód sme zavřšili zhlukovaním založeným na poradí, prvýkrát predstaveným v práci [12] a tiež využitím hierarchického aglomeratívneho zhlukovania, zastaveného v nejakom konkrétnom momente pre dosiahnutie *hard* zhlukov.

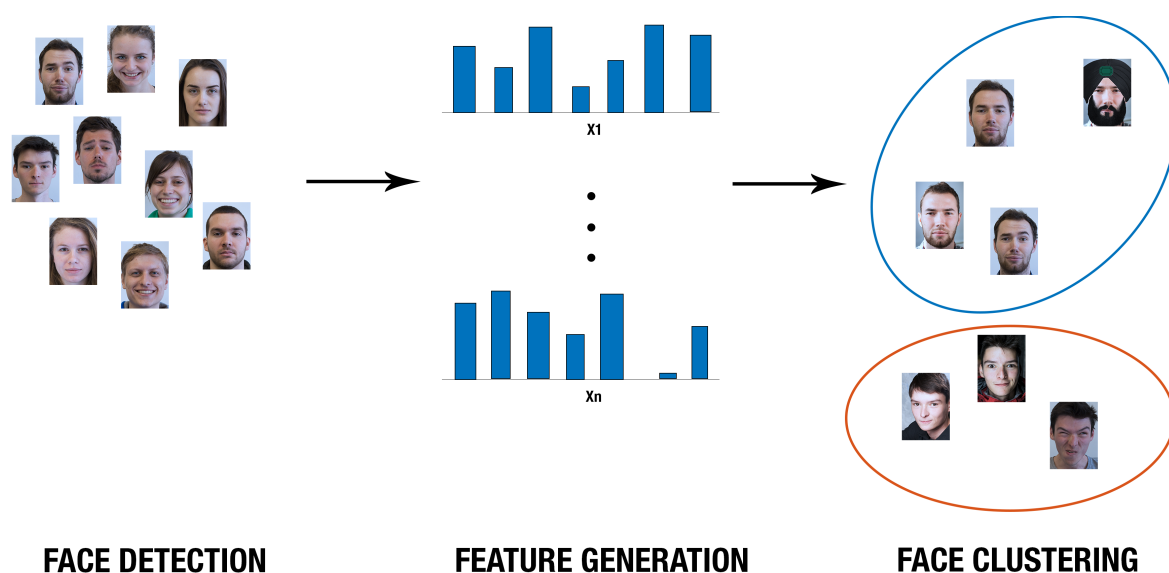
V neposlednom rade sa tiež venujeme úprave metód pre využitie *EXIF* dát v zhlukovacích algoritmoch. Rozoberáme dve varianty takéhoto použitia. Prvá, možno trochu naivná, je použiteľná pre ľubovoľnú metódu. Druhá je adaptáciou práve zhlukovania založeného na poradí a je pre túto metódu špecifická. Narozdiel od prvej však tentokrát ponúkame jeden voliteľný parameter poskytujúci dostatok voľnosti pre použitie tejto metódy aj pre reálne databázy.

Prácu uzatvára druhá kapitola venujúca sa využitiu spomínaných metód na umelých i reálnych databázach s rôznym stupňom náročnosti.

Kapitola 1

Metodika rozpoznávania tvárí

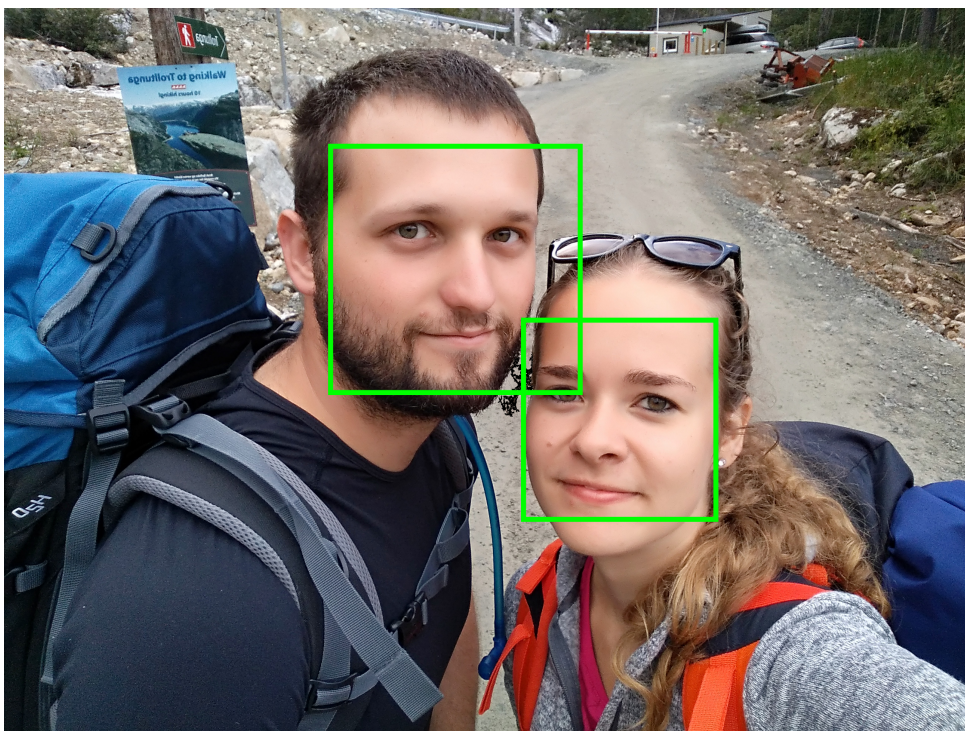
V tejto kapitole sa budeme venovať kompletnému problému zhľukovania fotografií podľa osôb v troch podkapitolách: *Rozpoznávanie tvárí*, *Generácia vektorov črt* a *Zhľukovanie*, tak ako sme to uviedli v úvode. Pre lepšiu intuíciu o čo nám pôjde, uvedieme jednoduchú schému nášho postupu na obrázku 1.1. Podrobne opíšeme niektoré algoritmy a uvedieme tiež vlastné prístupy pre rozšírenie týchto algoritmov pre náš konkrétny problém. V podkapitole *Hodnotenie metód* uvedieme prístup, akým môžeme ohodnocovať jednotlivé metódy.



Obr. 1.1: Základná schéma uvažovaného problému

1.1 Rozpoznávanie tvárí

Problém hľadania tvárí v obrázkoch je veľmi dobre rozpracovaná téma. V tejto práci nám nepôjde o zhodnotenie rôznych prístupov rozpoznávania a uspokojíme sa s jediným, s algoritmom *Viola-Jones* [13]. Ide o jeden z najpoužívanějších, takže jeho implementácie sú bežne dostupné. My využijeme implementáciu z MATLAB-ovského balíka *Computer Vision System toolbox*. Je dobré podotknúť, že síce algoritmus funguje pomerne dobre, v dnešnej dobe existujú oveľa sofistikovanejšie riešenia, hlavne na báze hlbokých sietí. Pre dosiahnutie čo najlepšieho výsledku by tak mal byť prvý krok voľba silnejšieho algoritmu, my sa však uspokojíme s *Viola-Jones* a metódy budeme vyhodnocovať až po nájdení tvárí.



Obr. 1.2: Ukážkový príklad nájdených tvárí

1.1.1 Digitálne fotografie

RGB farebný model (ale aj iné bežne používané farebné modely) bol vytvorený pre jednoduchú reprezentáciu a zobrazovanie obrázkov v počítači, respektíve na obrazovkách iných zariadení. Nevýhodou tohto modelu je, že reprezentácia obrázkov závisí na samotnom zariadení, na ktorom sa zobrazujú. To nám však vôbec nevadí, keďže pre naše účely potrebujeme len samotné číselné hodnoty zakódované v každom digitálnom obrázku. Všetky

farby sú v obrázkoch zakódované trojicou matíc, ktoré prislúchajú červenej, zelenej a modrej farbe. V prípade osem-bitovej kvantizácie sa všetky hodnoty pohybujú v rozmedzí 0 až 255 ($2^8 = 256$). Pre naše potreby nie je samotná farba až tak dôležitá, aj keď ľudia ju určite používajú pre rozpoznávanie. Aby sme zrýchlili výpočty, nebudeme teda pracovať s tenzormi, ale všetky farebné obrázky prevedieme do odtieňov šedej. Dostaneme tak jedinú maticu, ktorú budeme označovať O . Pre konvertovanie budeme využívať už hotovú funkciu v MATLABE *rgb2gray*, ktorá počíta *rec601 luma* štandard, kde váhovo spriemeruje farebné matice $O = 0.2989 * R + 0.5870 * G + 0.1140 * B$. Konštanty boli štandardizované, aby kompenzovali spôsob, akým človek vníma farby. Táto problematika je oveľa komplikovanejšia, no v tejto práci ju nebudeme ďalej rozoberať. Rozloženie na jednotlivé farby, pôvodný obrázok, aj jeho reprezentáciu v odtieňoch šedej uvádzame na obrázku 1.3.



Obr. 1.3: Zobrazenie obrázku v RGB a odtieni šedej

1.2 Generácia vektorov črt

Pod črtami budeme rozumieť sadu nejakých zvolených atribútov pre charakterizáciu konkrétneho objektu, v našom prípade konkrétnej tváre. V podkapitole 1.1.1 sme uviedli spôsob, akým počítač vníma digitálne obrázky. Pre naše potreby však nie je optimálne,

ako vektory črt využívať priamo hodnoty jednotlivých pixelov, ale najskôr vykonáme nejakú transformáciu, často do oveľa menej rozmerného priestoru, nakoľko zhlukovanie vo vysokorozmerných priestoroch býva veľmi problematické, obzvlášť pri voľbe euklidovskej normy.

Pod maticou dizajnu, respektíve dátovou maticou, budeme teda rozumieť

$$X_{n \times p} = \begin{pmatrix} \text{---} & x_1^T & \text{---} \\ \text{---} & x_2^T & \text{---} \\ & \vdots & \\ \text{---} & x_n^T & \text{---} \end{pmatrix},$$

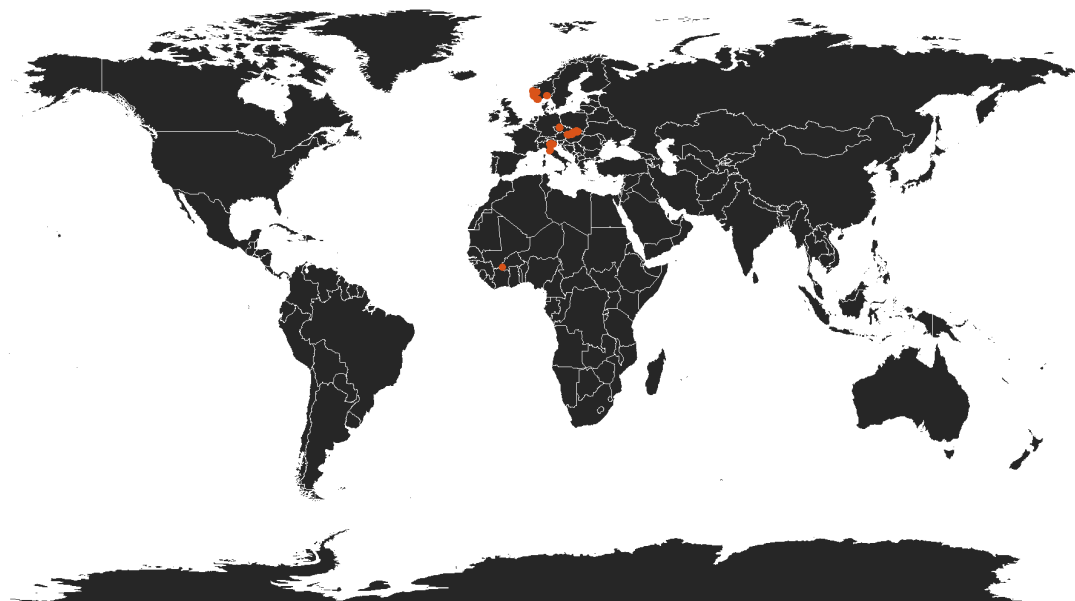
kde n je počet dát, a $x_1, \dots, x_n$ sú $p \times 1$ rozmerné vektory črt.

1.2.1 Exchangeable Image File Format

Hodnoty každého pixelu nie sú jediné informácie, ktoré digitálna fotografia zahŕňa. Jednou z ďalších informácií v obrázkoch je celá séria dát skupinovo nazývaná *EXIF* dáta (Exchangeable Image File Format). Treba podotknúť, že nie všetky formáty obsahujú *EXIF* dáta. My budeme využívať formát *.jpg*, ktorý tieto dáta poskytuje. Medzi údajmi, ktoré nás zaujímajú budú GPS súradnice fotografie a časové údaje vytvorenia fotografie. Treba si však dať pozor, že nie všetky fotoaparáty poskytujú GPS koordináty. Drvivá väčšina fotografií je však vytvorená smartfónmi, ktoré vedú takúto informáciu poskytovať. Tieto dáta sa pokúsime využiť, ak budú prítomné, no neobmedzíme sa len na obrázky, ktoré ich obsahujú. Dosiahneme tak lepšiu použiteľnosť pri možnosti zlepšenia presnosti.

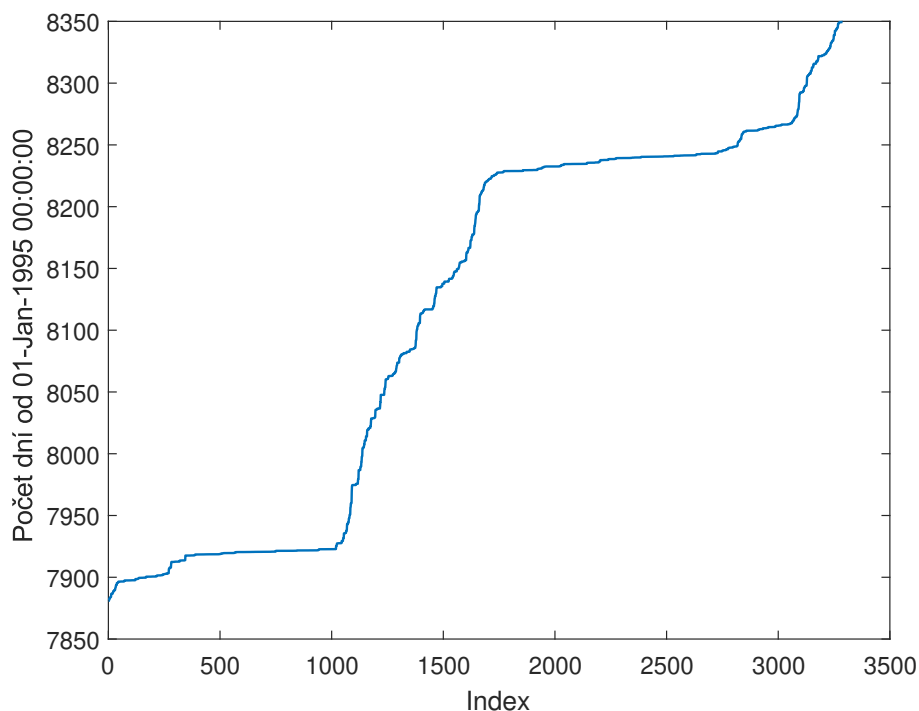
Súradnice načítame pre každú fotografiu a prevedieme ich do desatinnej podoby. Naskytá sa však jeden problém. Zo zemepisnej dĺžky a šírky nemôžeme priamo euklidovskou normou zisťovať vzdialenosti medzi dvoma bodmi, lebo zem nie je plochá. Tejto problematike sa však budeme bližšie venovať v časti 1.3.8.

Pri jednoduchých experimentoch s vlastnou databázou fotografií sme však prišli na skutočnosť, že v niektorých prípadoch môže zlyhať veľmi zvláštnym spôsobom samotný GPS modul v telefóne. Pri vykreslení lokácie všetkých fotografií sme objavili toto zlyhanie vďaka tomu, že v Afrike sa autor nikdy nenachádzal. Tieto súradnice sa dajú dodatočne manuálne prepísať v samotných EXIF dátach. Mapu s vykreslenými údajmi uvádzame na obrázku 1.4.



Obr. 1.4: Ukážkový príklad vykreslenia gps súradníc pre všetky fotografie zvolenej databázy

Časová známka fotografie je uvádzaná aj s dátumom vytvorenia. Aby sme s týmto údajom mohli jednoducho pracovať, prevedieme všetky časové údaje na neceločíselný počet dní od roku 0 a času 0, prípadne od nami zvoleného dátumu. Dostaneme tak len jednu premennú, s ktorou sa pracuje oveľa jednoduchšie. Túto transformáciu demonštrujeme na ukážkovom príklade, kde načítame 3294 fotografií, extrahujeme *EXIF* dáta a dátumy a čas prevedieme na počet dní od 1. januára 1995 a času 0h 0m 0s. Výsledky uvádzame na obrázku 1.5, kde na *y*-ovej osi je tento čas v dňoch a na *x*-ovej osi je index fotografie. Dáta sme pred vykreslením zoradili vzostupne, a teda výsledná krivka je neklesajúca.



Obr. 1.5: Ukážkový príklad zoradenej časovej známky

Referenčný čas necháme fixovaný, pretože je vhodnejšie predpokladať, aké najstaršie môžu byť digitálne fotografie, ako keby sme mali počítať minimum z časov na fotografiách v danej databáze. Budeme tak menej ovplyvňovať dĺžku výpočtového času kompletného programu.

Je dobré si všimnúť, že na krivke sa nachádza viacero takmer konštantných intervalov. Ak budeme teraz predpokladať, že všetky fotografie boli vyhotovené rovnakým prístrojom, môžeme usudzovať, že pôjde o akési významnejšie udalosti. Na týchto udalostiach je možné predpokladať opakovaný výskyt určitej skupiny osôb. Ak teda úspešne nájdeme niektoré osoby v týchto udalostiach (pri sekvenčnom spracovaní fotografií), každá ďalšia tvár by mala mať zvýšenú pravdepodobnosť, že je už jednou z rozpoznaných osôb.

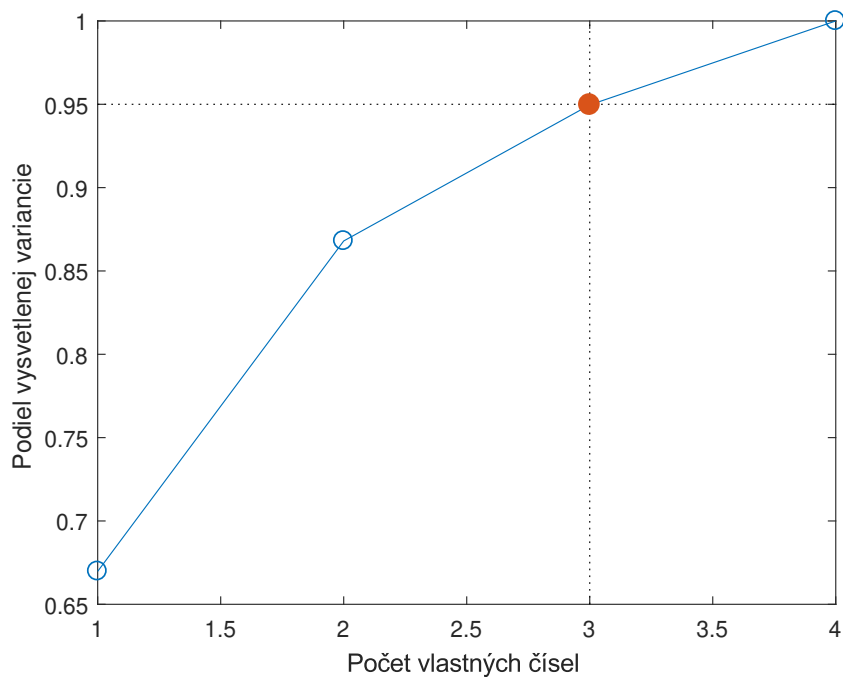
1.2.2 Redukcia dimenzie

V tejto časti uvádzame spôsoby, akými budeme redukovať dimenziu vektorov čŕt pre zrýchlenie výpočtov a tiež zahodenie veľkého množstva zbytočnej informácie, ktoré nám môže dokonca brániť v správnom zatriedení. Jedným z algoritmov, ktoré používame, no nebudeme sa im príliš venovať je *Analýza hlavných komponentov* (PCA). Keďže túto tému sme podrobne spracovali v bakalárskej práci [2], opíšeme si ju len veľmi stručne.

Pod informáciou budeme rozumieť varianciu daného atribútu. Ak by sa totiž atribút veľmi nemenil, znamená to, že jeho zahodením by sme neprišli o veľa informácie. Častokrát však nemáme prípad, že by sme mohli okamžite niektoré atribúty zahodiť. Tu vstupuje na rad metóda PCA. Poskytne nám spôsob, ako transformovať dáta do nových súradníc, ktoré budú ordinované na základe podielu celkovej variancie vysvetlenej v dátach. Treba spomenúť, aj keď pre naše potreby to nie je významné, že táto metóda má jeden nedostatok, a to ten, že zjavne vďaka svojej konštrukcii nie je škálovo invariantná. Pre nás sú však atribúty úrovné šedej na prislúchajúcich pixeloch, a teda nemusíme riešiť žiadnu štandardizáciu premenných, keďže majú rovnakú škálu. Presný popis ako sa vytvárajú nové súradnice a koľko atribútov môžeme zahodiť nájdeme, ako sme už spomínali, v kapitole PCA v práci [2].

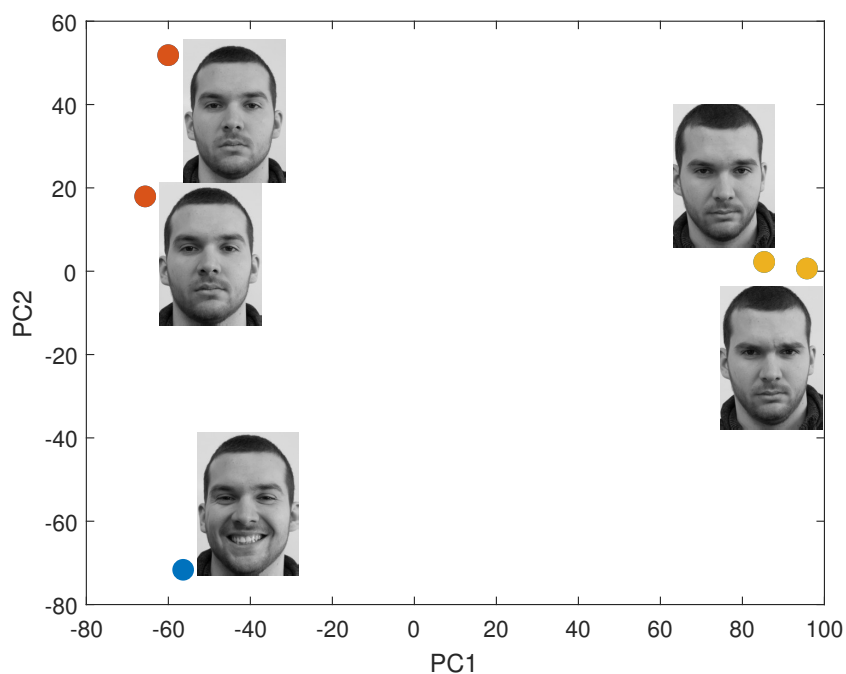
Tu sa dostávame k jednému z problémov aplikovania všetkých algoritmov na redukciu dimenzie. Pôvodné fotografie, z ktorých sme obrázky samotných tvárí vyťahovali, mali rôzne rozlíšenia, a samozrejme aj tváre boli vo fotografiách rôznej veľkosti, a teda majú všetky rôzny počet pixelov. Skôr ako aplikujeme akýkoľvek inteligentný spôsob redukcie dimenzie, potrebujeme spraviť isté predspracovanie dát. Všetky tváre interpolujeme, aby sme dostali rovnaký počet pixelov. Tento sa bude líšiť v závislosti od konkrétnych metód. V prípade PCA pracujeme s rozlíšením 80×100 pixelov. Na takto upravené dáta budeme využívať ďalšie redukcie dimenzie.

Ukážku toho, že PCA má zmysel aj pri tomto probléme uvádzame na obrázkoch 1.6 a 1.7. Ide o aplikovanie PCA na 5 obrázkov tvárí jednej osoby.



Obr. 1.6: Laktový diagram pre ukážkový príklad PCA

Na obrázku 1.6 vidíme zobrazený podiel vysvetlenej variance a počet použitých vlastných čísel. Už pre 2 vlastné čísla dostávame tento podiel väčší ako 85 %. Rozhodli sme sa teda použiť len prvé dva hlavné komponenty.



Obr. 1.7: Prvé 2 hlavné komponenty pre ukážkový príklad PCA

Na obrázku 1.7 vidíme, že už len pri použití dvoch čísel bola naša metóda vhodná pre oddelenie podobných a nepodobných tvárí. Síce ide stále o jednu a tú istú osobu, v tomto prípade sme však separovali emócie zachytené na fotografiách.

1.2.3 VGG-16 a Alexnet

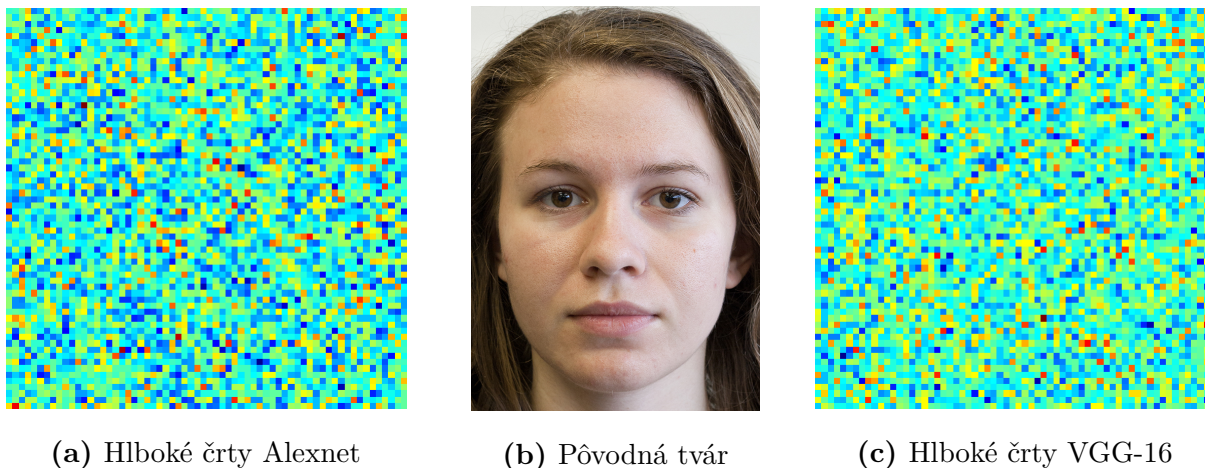
Jedným zo spôsobov generácie vektorov čŕt sú tiež rôzne špecifické druhy neurónových sietí. Keďže v našom probléme pracujeme najmä s obrázkami vo forme samotných pixelov, naskytá sa možnosť využiť práve konvolučné siete. Nepôjdeme však klasickou cestou autoenkódera, ktorý by sme trénovali na obrovskej databáze tvárí. Rozhodli sme sa využiť menej populárnu metódu transferového učenia.

Pre redukciu dimenzie používame časť už dvoch vopred natrénovaných sietí, a to konkrétne sieť *Alexnet* pochádzajúcu z práce [14] a nemenej známu *VGG-16* z práce [15]. Ide o veľmi hlboké siete obsahujúce 25, respektíve 41 vrstiev. Transferové učenie sa zvyčajne používa ako štartovací bod pre trénovanie novej siete, ktorej posledné vrstvy sa nahradia novými, a sieť sa využíva pre iný účel ako bola pôvodne natrénovaná. My sme sa rozhodli využiť z prvej siete prvých 20 vrstiev, kde 20-ta vrstva je plne prepojená.

Ďalej sme túto sieť netrénovali pre náš problém, len sme sa rozhodli využiť jej deskriptívnu silu z obrovskej dátovej množiny *Imagenet*, ktorá však neobsahuje tváre, ale len predmety. Podobne v druhej sieti využívame 36-tu opäť plne prepojenú vrstvu. Domnievame sa, že natrénovanie novej hlbokkej siete na ľudských tvárach môže mať pozitívny vplyv pre jej deskriptívnu silu, no v tejto práci ju konštruovať nebudeme.

Každá tvár, z ktorej ideme generovať vektor čŕt, je najskôr interpolovaná na 224×224 , respektíve 227×227 pixelov, vo všetkých troch farbách. Následne je posiadaná postupne cez rôzne konvolučné, poolingové, relu, dropout aj plne prepojené vrstvy až po našu zvolenú poslednú vrstvu. Výsledný vektor z oboch sietí má 4096 čŕt.

Na obrázku 1.8 uvádzame pre ilustráciu ľudskú tvár a jej reprezentáciu pomocou čŕt získaných z hlbokých sietí *Alexnet* a *VGG-16*. Tieto čŕty sú zobrazené vo farebnej škále *jet* pre jednoduchšie rozlíšenie.



Obr. 1.8: Ukážkový príklad vykreslenia hlbokých črt v škále jet pre ľudskú tvár

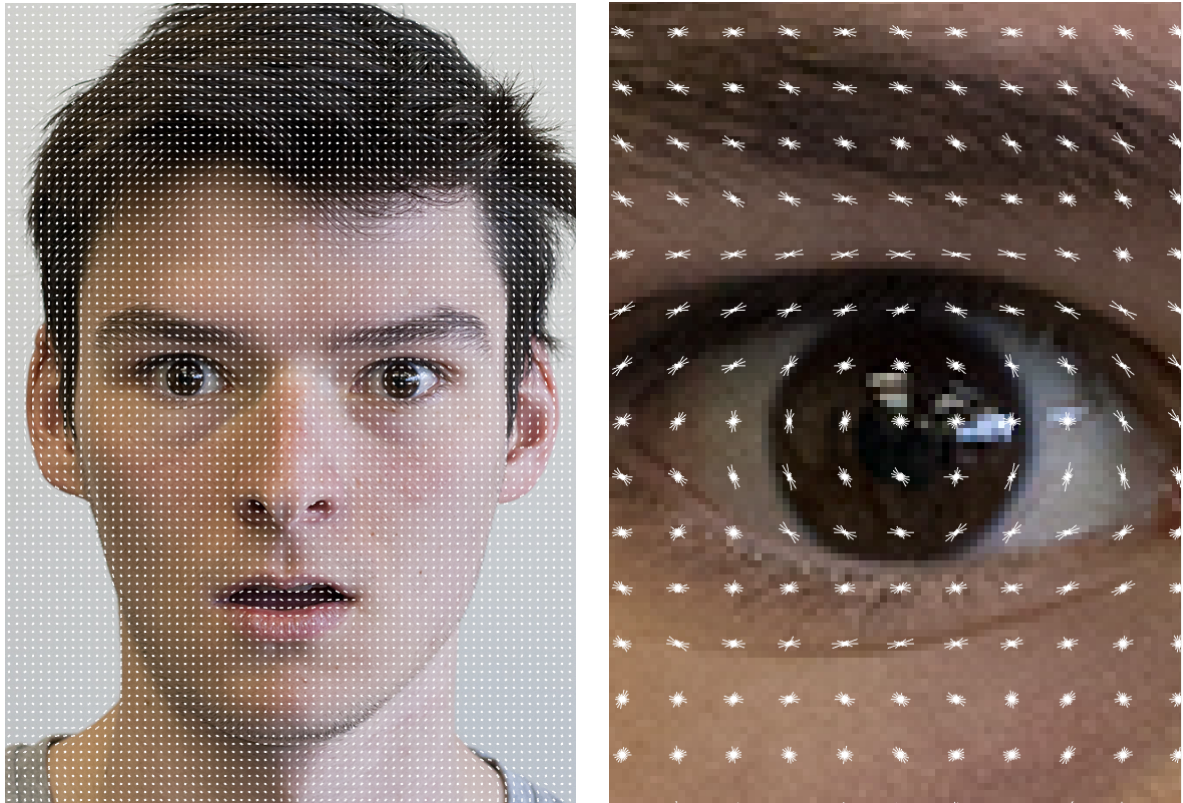
1.2.4 HOG

Prístup *HOG* (Histogram of oriented gradients) bol najviac popularizovaný v práci [16], i keď niektoré idey boli aplikované už v roku 1986. Základná myšlienka je rozdelenie obrázku do menších regiónov a vypočítanie gradientov pomocou všetkých pixelov v danom regióne.

Pôvodne bola táto metóda využívaná len pre obrázky v úrovni šedej, no populárny je aj prístup výpočtu na všetkých troch farebných vrstvách a výsledné maximum na každom regióne z týchto vrstiev. Ďalej sú tieto hodnoty rozdelené na bloky (obvykle napríklad 8×8) a v nich je vypočítaný histogram. Všetky získané frekvencie z celého obrázka sú napokon uložené ako výsledný vektor črt.

V práci [16] autori experimentovali aj s veľkosťou regiónov, kde sa pokúšali využiť aj centrálnu diferenciu (v oblasti počítačového videnia je toto označované ako Sobelov filter), no v tejto oblasti to však neviedlo k lepším výsledkom.

Na obrázku 1.9 uvádzame ľudskú tvár aj so samotnými orientovanými gradientami.



(a) Celá tvár s orientovanými gradientami

(b) Priblíženie pre lepšiu ilustráciu

Obr. 1.9: Ukážkový príklad vykreslenia orientovaných gradientov pre ľudskú tvár

1.3 Zhlukovanie

Zhlukovanie (clustering) využívame, ak chceme nájsť v dátach nejakú štruktúru, len na základe vektorov črt a podľa nej ich rozdeliť na skupiny (zhluky). Narozdiel od klasifikácie, nemáme k dispozícii tréningový vzorku so správnym zatriedením a ide teda o metódy učenia bez učiteľa. Existuje mnoho druhov zhlučovacích algoritmov a tiež rôzne delenia týchto algoritmov. Jedným základným delením je : *hierarchické* a *nehierarchické* clustrovanie. V prvom prípade nebudeme hľadať konkrétne skupiny dát, ale nájdeme určitú hierarchiu v nich. Každé dátum sa bude postupne spájať s nejakým iným na základe toho, ako ďaleko sú od seba vzdialené. Čo znamená vzdialenosť medzi nimi, sa líši pri rôznych algoritmoch. Aglomeratívne algoritmy nechajú dáta v jednotlivých skupinách samostatne a v každom kroku ich budú spájať. Divízívne algoritmy fungujú presne z opačného konca, a teda všetky dáta sú v jednom zhluke a postupne sa oddeľujú najviac nehodiace sa. Oba typy algoritmov vieme vykresliť ako binárne stromy (tie budeme nazývať dendrogramy),

kde je vidno presnú hierarchiu v dátach a tiež na základe dĺžok samotných vetiev vieme povedať ako ďaleko sú od seba jednotlivé body.

Nehierarchické clustrovanie nám len bude priradovať dáta do skupín. Existuje množstvo typov delenia takýchto algoritmov, napríklad soft a hard. Hard algoritmy udajú len prirodzené číslo zhľuku, naproti tomu soft algoritmy vracajú napríklad pravdepodobnosť priradenia k určitému zhľuku. Zaujímavým problémom môžu byť tiež outliery, ale ten vieme riešiť napríklad priradením zhľukov pozostávajúcich len z jedného prvku do spoločného zhľuku, ktorý nazveme outliery. V tejto práci budeme takýto zhľuk označovať číslom 0, čo bude dôležité v niektorých ďalších sekciách tejto kapitoly.

1.3.1 K-means

Jeden z najpopulárnejších zhľukovacích problémov je nepochybne K -means. Ide o nehierarchické hard clustrovanie. Značenie je častokrát v literatúre trochu zmätočné (napríklad v [4]), trochu ho teda zjednodušíme. Budeme sa snažiť rozdeliť dáta do vopred špecifikovaného počtu skupín - K , pričom budeme riešiť optimalizačnú úlohu:

$$\min \sum_{k=1}^K \sum_{x_i \in C_k} (x_i - \bar{x}_k)^\top (x_i - \bar{x}_k), \quad (1.1)$$

kde x_i je i -ty bod, K je počet skupín, $\bar{x}_k$ je k -ty centroid a C_k je k -ty zhľuk (t.j. množina bodov x_j priradených ku k -temu zhľuku). Pod centroidom $\bar{x}_k$ budeme rozumieť

$$\bar{x}_k = \frac{1}{|C_k|} \sum_{x_j \in C_k} x_j,$$

kde pod označením $|C_k|$ budeme rozumieť kardinalitu množiny C_k . Skôr než spomenieme riešenie takéhoto optimalizačného problému, pozrime sa na prvý viditeľný nedostatok a tou je voľba parametra K . Nie vždy (a určite nie v našom konkrétnom prípade) totiž poznáme počet zhľukov, do ktorých dáta patria.

Ak K nepoznáme, chceli by sme akési kritérium pre jeho optimálnu voľbu. Zjavne však vidíme, že hodnota účelovej funkcie klesá s rastúcim K , keďže body v zhľukoch vedia byť bližšie k ich centroidom. Môže sa však stať, že so zvyšujúcim K prestanú byť úbytky v účelovej funkcii veľké.

Podobne ako sme to ukázali v práci [2], môžeme vykresliť takzvaný laktový diagram, kde uvidíme úbytok na účelovej funkcii oproti celkovému počtu zhľukov. „Optimálne“ K

teda môžeme zvoliť napríklad ako najväčšie také k , ktoré sa nachádza pred „laktom“. Metóda K -means sa teda snaží priradiť body zhlukom, tak aby minimalizovala vnútrozhlukovú sumu štvorcov (WCSS).

Najpoužívanejšou heuristikou pre túto metódu je *Lloydov* algoritmus [3].

Algorithm 1: Lloydov algoritmus pre K -means

Input : Matica dizajnu X , prirodzené číslo K

Output: Index zhlukov idx

```

1  Zvolíme náhodné centroidy  $\overline{x}_1, \dots, \overline{x}_K$ .
2  while nastala zmena do
3      forall  $i$  do
4          forall  $j$  do
5              Vypočítame vzdialenosť  $\|x_i - \overline{x}_j\|_2$ 
6          end
7          Priradíme  $x_i$  do zhliku  $C_k$  kde  $k = \operatorname{argmin}_{j=1, \dots, K} \|x_i - \overline{x}_j\|_2$ 
8      end
9      forall  $j$  do
10         Vypočítame nový centroid  $\overline{x}_j = \frac{1}{|C_j|} \sum_{x_i \in C_j} x_i$ 
11     end
12 end
13 Upravíme indexy zhlukov  $idx$ .
```

Štandardná a veľmi jednoduchá inicializácia je napríklad zvoliť v prvom kroku centroidy ako niektoré body, alebo namiesto náhodného generovania centroidov náhodne priradíme číslo zhliku ku každému bodu a potom z nich tieto centroidy vypočítame.

Pri patologických prípadoch sa môžeme zaseknúť v lokálnom minime. Toto vieme heuristicky ošetriť viacnásobným spustením algoritmu s rôznymi inicializáciami. Dá sa tiež ukázať, že takýto algoritmus bude konvergovať v konečnom počte krokov, lebo vďaka ostro-menším hodnotám účelovej funkcie sa nevieme zacykliť.

Oveľa väčším problémom však pre nás je, že metóda K -means, vďaka svojej konštrukcii nie je vhodná pre nachádzanie nekonvexných zhlukov. Pre zhľukovanie fotografií tváří, ktoré sa odohráva vo vysoko-rozmernom priestore, však nevieme povedať, aký tvar budú mať skutočné zhľuky.

Pre zhlukovanie gps súradníc máme navyše problém, že ich nemôžeme porovnávať v euklidovskej norme.

1.3.2 K-medoids

Problém K -medoids je veľmi podobný K -means, no líši sa v tom, že centroidy nahradíme medoidmy, t.j. „najstrednejšími“ objektami pre každý zhuk, ktoré sú zároveň jedným z bodov daného zhuku, a navyše využívame namiesto euklidovskej normy maticu nepodobností. Najčastejšie používaným heuristickým algoritmom pre riešenie K -medoids je *PAM* (Partitioning Around Medoids) algoritmus.

Definícia matice nepodobností závisí od samotného užívateľa algoritmu. My budeme od matice nepodobností požadovať vlastnosti metriky (nie nutne všetky) rovnako ako v knihe [1].

Definícia 1.3.1. *Nech x_i, x_j a $x_k \in \mathbb{R}^p$. Od **nepodobnosti bodu** x_i a x_j (označ. $d(x_i, x_j)$) budeme vyžadovať:*

$$d(x_i, x_j) \geq 0, \quad (1.2)$$

$$d(x_i, x_j) = 0 \Leftrightarrow x_i = x_j, \quad (1.3)$$

$$d(x_i, x_j) = d(x_j, x_i). \quad (1.4)$$

Ak nepodobnosť navyše splňa

$$d(x_i, x_j) \leq d(x_i, x_k) + d(x_k, x_j), \quad (1.5)$$

*budeme ju nazývať **metrikou** a ak ešte splňa*

$$d(x_i, x_j) \leq \max\{d(x_i, x_k) + d(x_k, x_j)\}, \quad (1.6)$$

*budeme ju nazývať **supermetrikou**.*

Optimalizačná úloha pre K -medoids a aj pre *PAM* teda vyzerá ako

$$\min \sum_{k=1}^K \sum_{x_i \in c_k} d(x_i, m_k),$$

kde c_k je k -ty cluster, x_i je i -ty bod a m_k je medoidom k -teho zhuku. Pre algoritmus *PAM* si treba dávať obzvlášť pozor, lebo veľmi často môžeme nájsť v rôznej literatúre nesprávne opísanie tohto algoritmu. My ho budeme uvádzať podľa pôvodného článku [6].

Tento algoritmus by bol deterministický, ale ak sa nám stane, že sa optimum nejakého lokálneho problému nadobúda vo viacerých bodoch, rozhodujeme sa náhodne medzi nimi. Úplne stochastická varianta tohto algoritmu, na ktorú sa nesprávne mnohí autori odkazujú inicializuje medoidy náhodne, priradí všetky body do zhlukov podľa ich najbližších medoidov a postupuje pažravým heuristickým algoritmom pre lokálnu optimalizáciu. Takýto postup dostáva zjavne rôzne výsledky aj bez prítomnosti remíz a má zmysel ho skúšať viackrát odznova.

Pre náš problém samozrejme nemusíme do hĺbky rozoberať, ktorý spôsob je lepší a nevádi nám ani jedna implementácia riešenia problému K -medoids, ale pre teoretické účely spomenieme ako bol algoritmus definovaný v pôvodnom článku [6].

Začneme s maticou nepodobnosti D , ktorá je buď vložená do algoritmu alebo vypočítaná na základe špecifikácie nepodobnosti $D_{ij} = d(x_i, x_j)$. Táto matica, keďže bude spĺňať nami vyžadované vlastnosti, bude zjavne symetrická s nulovou diagonálou. Keďže v tomto algoritme vôbec nemusíme poznať samotné body x_i , ale len ich nepodobnosti, budeme ďalej hovoriť o zhlučovaní objektov i .

Pre jednoduchšie pochopenie vopred prezradíme, že množinu S_e sa budeme snažiť naplniť medoidmy, tak aby minimalizovali účelovú funkciu. Zadefinujeme si teraz dve veličiny.

- D_i budeme chápať ako nepodobnosť medzi objektom i a najbližším objektom z S_e .
- E_i budeme chápať ako nepodobnosť medzi objektom i a druhým najbližším objektom z S_e .

Tieto čísla sa budú samozrejme meniť s každou zmenou množín S_e a U . Autori rozdeľujú algoritmus na 2 fázy, **BUILD** a **SWAP**, kde fáza *BUILD* je sofistikovaná inicializácia, ktorú autori zvolili (a práve tá sa pri nesprávnej interpretácii neuvádza) a fáza *SWAP* sa ju už pokúša len vylepšiť. Pretože je tento algoritmus pomerne komplikovaný, nebudeme ho zapisovať vo forme pseudokódu. Autori uvádzajú (s našimi menšími úpravami, zjednodušením značenia podľa [7] a objasnením niektorých krokov) algoritmus *PAM* nasledovne:

- **BUILD**

1. Inicializujeme prázdnu množinu S_e (selected) a množinu U (unselected), do ktorej vložíme všetky body.

2. Do S_e vložíme jeden z bodov U , a to taký, ktorého suma vzdialeností od všetkých ostatných objektov je najmenšia a zároveň tento bod vylúčime z U .
3. Uvažujme o objekte i ako o kandidátovi do množiny S_e .
4. Vypočítame D_j pre $\forall j : j \in U \setminus \{i\}$.
5. Ak pre nejaké j máme $D_j > d(j, i)$, potom nám j napovedá, že pridaním i do S_e si polepšíme.
6. Zdefinujeme teda kontribúciu j do i ako $C_{ji} := \max(D_j - d(j, i), 0)$.
7. Vypočítame celkový zisk pre zaradenie kandidáta i do množiny S_e , ako

$$g_i = \sum_{j \in U} C_{ji}.$$
8. Zvolíme toho kandidáta, ktorý maximalizuje g_i cez i a zároveň priradíme $S_e := S_e \cup \{i\}$ a odoberieme $U := U \setminus \{i\}$.
9. Iterujeme (okrem prvého bodu), až kým kardinalita množiny $|S_e| = K$, teda získame všetky počiatočné medoidy.

- **SWAP** - Teraz nás bude zaujímať, aký efekt bude mať zámena všetkých dvojíc $(i, h) : i \in S_e, h \in U$, teda ako sa zmenia sumy nepodobností objektami a ich najbližšími objektami, po tom, čo presunieme i z S_e do U a h z U do S_e . Tento efekt označíme T_{ih} . Autori navrhujú počítať pre všetky objekty $j \in U \setminus \{h\}$ kontribúciu K_{jih} pre objekt j , ak zameníme objekt i a h (pre i a h platia rovnaké pravidlá ako sme spomínali pre T_{ih}). Vďaka ukončeniu iterácií v časti *BUILD* teraz zjavne platí, že $D_j \leq d(j, i)$. Poďme teda počítať K_{jih} .

1. Ak $d(j, i) > D_j$, potom K_{jih} je buď 0 ak $d(j, h) \geq D_j$, alebo $d(j, h) - D_j$ ak $d(j, h) < D_j$.

To sa dá zosumarizovať ako $K_{jih} = \min(d(j, h) - D_j, 0)$.

2. Ak $d(j, i) = D_j$, potom $K_{jih} = d(j, h) - D_j$ ak $d(j, h) < E_j$, alebo $K_{jih} = E_j - D_j$ ak $d(j, h) \geq E_j$.

To sa dá zosumarizovať ako $K_{jih} = \min(d(j, h), E_j) - D_j$.

3. Vypočítajme všetky $T_{ih} = \sum_{j \in U} K_{jih}$.
4. Nájdime dvojicu (i, h) , pre ktorú je T_{ih} je minimálne.

5. Ak je T_{ih} pre túto dvojicu < 0 , tak zámenu vykonáme a prepočítame D_j, E_j, K_{jih} a iterujeme všetky 3 inštrukcie, až pokým $T_{ih} > 0$ pre všetky prípustné (i, h) .

1.3.3 Spektrálne zhľukovanie

Skôr ako opíšeme spektrálne zhľukovanie, potrebujeme zdefinovať niekoľko pojmov z teórie grafov. Niektoré vedomosti z lineárnej algebry budeme považovať za známe. Pri grafoch budeme vychádzať najmä z [19], [20] a [4], aj keď si niektoré pojmy jemne upravíme, respektíve zmeníme značenie.

Definícia 1.3.2. Pod **grafom** G budeme rozumieť usporiadanú dvojicu vrcholov a hrán (V, E) , kde $V = \{v_1, \dots, v_n\}$ je konečná a $E \subset V \times V$ je podmnožina množiny všetkých dvoj prvkových podmnožín množiny V .

Definícia 1.3.3. Nech $G = (V, E)$ je grafom. Pod **ocenovým grafom** $\tilde{G} = (V, \tilde{E})$ budeme rozumieť usporiadanú dvojicu vrcholov a ocenovaných hrán, kde $V = \{v_1, \dots, v_n\}$ je konečná a $\tilde{E} = E \times \mathbb{R}_0^+$. Hrana e_{ij} spájajúca vrcholy v_i a v_j potom nesie ocenenie w_{ij} .

Poznámka 1: Neocenovaný graf teda len hovorí, ktoré vrcholy sú spojené, ale nenesie dodatočnú informáciu o cene spojenia týchto vrcholov. Táto cena spojenia môže napríklad znamenať vzdialenosť vrcholov alebo poplatok za cestu medzi vrcholmi.

Poznámka 2: V tejto práci nás zaujímajú len neorientované grafy, preto sme sa rozhodli orientovaný graf nedefinovať.

Definícia 1.3.4. Pod **maticou podobnosti (adjacency matrix)** W pre graf $\tilde{G} = (V, \tilde{E})$ budeme rozumieť $W = (w_{ij}), i, j = 1, \dots, n$, kde $w_{ij} := 0$ ak $v_i \notin E \vee v_j \notin E$.

Poznámka 3: Keďže, ako sme už spomenuli, nás zaujímajú len neorientované grafy, tak matica W bude zjavne symetrická.

Poznámka 4: Pre neocenovaný graf bude mať matica W na mieste $w_{ij} = 1$, ak $v_i \in E \wedge v_j \in E$.

Poznámka 5: Niekedy sa pod označením **similarity graph** myslí samotná matica podobnosti, inokedy sa tak hovorí o grafe, kde vrcholy sú naše dáta a hrany sú vážené danou maticou podobnosti. My budeme tieto označenia ľubovoľne zamieňať.

Definícia 1.3.5. Pod **stupňom vrchola (degree of vertex i)** $v_i \in V$ budeme rozumieť $s_i = \sum_{j=1}^n w_{ij}$. **Stupňovou maticou (degree matrix)** budeme nazývať diagonálnu maticu $S = \text{diag}\{s_1, \dots, s_n\}$.

Definícia 1.3.6. Nech $G = (V, E)$ je graf. Ak $V' \subseteq V$, $E' \subseteq E$ a $G' = (V', E')$ je graf, tak G' budeme nazývať **podgrafom** G , označ. $G' \subseteq G$.

Pod **sledom** rozumieme alternujúcu postupnosť vrcholov a hrán, začínajúcu a končiacu vrcholom. Sled, v ktorom sa žiaden vrchol neopakuje sa nazýva **cesta**.

Graf G sa nazýva **súvislý**, ak pre každé dva vrcholy v_i a v_j existuje sled začínajúci vo v_i a končiaci vo v_j .

G' je **súvislý komponent**, ak je maximálnym, v zmysle inklúzie, súvislým podgrafom ku grafu G . Ekvivalentne je G' súvislý komponent, ak je súvislý a neexistuje hrana medzi V' a $V^C := V \setminus V'$.

Definícia 1.3.7. Nech G' je podgrafom G , ale nie je totožné s G , t.j. $G' \subset G$. Pod **indikátorovým vektorom** Υ pre G' , budeme rozumieť $\Upsilon = (f_1, \dots, f_n)^T$, kde

$$f_i = \begin{cases} 1 & \text{ak } v_i \in V' \\ 0 & \text{inak.} \end{cases}$$

Zatiaľ sme stále nešpecifikovali ako získať maticu podobnosti W . Budeme uvažovať tri základné spôsoby ako premeniť vektory črt, respektíve maticu nepodobnosti, na maticu podobnosti. Treba poznamenať, že pre naše potreby sú vrcholy grafu vlastne dátami, takže označenia v_i a x_i môžeme ďalej ľubovoľne zamieňať.

- **Plný graf podobnosti (Full similarity graph)**

Pre i, j -te miesto (a zároveň aj j, i -te) v matici podobnosti pre tento graf budeme počítat napríklad

$$w_{i,j} = \exp\left(-\frac{\|x_i - x_j\|_2^2}{2\sigma^2}\right). \quad (1.7)$$

Vo všeobecnosti môžeme zvoliť aj iný spôsob počítania $w_{i,j}$, ale dôležité je, že body, ktoré sú identické sú si najviac podobné a akékoľvek vzdialenejšie sú si menej a menej podobné. Takáto matica má však jednu značnú nevýhodu, vo všeobecnosti nebude vôbec riedka.

- **Graf podobnosti $\mathcal{K}$ -najbližších susedov ($\mathcal{K}$ -nearest neighbor similarity graph skrátené KNN)**

V tomto grafe spojíme vrchol v_i s vrcholom v_j , ak v_j je medzi $\mathcal{K}$ -najbližšími susedmi v_i . Narážame však na problém nesymetrickosti takéhoto vzťahu, t.j. výsledný graf by bol orientovaný (digraf). Toto sa v praxi rieši dvoma spôsobmi. Prvý je zobrať spoločných susedov, t.j. $w_{i,j}$ bude nenulové ak v_i je medzi $\mathcal{K}$ -najbližšími susedmi v_j , alebo v_j je medzi $\mathcal{K}$ -najbližšími susedmi v_i . Druhý spôsob sa nazýva aj **$\mathcal{K}$ -najbližších vzájomných susedov (mutual $\mathcal{K}$ -nearest neighbors)**. Pre nenulovosť w_{ij} budeme vyžadovať, aby v_i bolo medzi $\mathcal{K}$ -najbližšími susedmi v_j a vice versa. Je dobré poznamenať, že ešte môžeme uvažovať, či budeme jednotlivé hrany váhovať, napríklad pomocou vzťahu 1.7, alebo graf necháme neocenený.

- **Graf podobnosti s epsilonovým okolím (ε -neighborhood similarity graph)**

V takomto prípade spojíme vrcholy v_i a v_j , ak sú k sebe (x_i a x_j) bližšie ako nejaké zvolené ε . To, čo znamená bližšie si opäť vieme rozumne definovať viacerými spôsobmi. V tomto prípade však graf berieme spravidla neocenený, keďže obvykle je ε malé a teda váhy hrán sú veľmi podobné.

Definícia 1.3.8. *Pod **nenormalizovaným Laplaceovským grafom (unnormalized graph Laplacian)** budeme rozumieť*

$$L = S - W, \quad (1.8)$$

kde W je matica podobnosti a S je stupňová matica.

Poznámka 6: Triviálne sa dá ukázať, že L je symetrická, pozitívne semidefinitná matica a potom sa dá priamo zo spektrálnej vety ukázať, že L je ortogonálne diagonalizovateľná a má len reálne nezáporné vlastné čísla.

Poznámka 7: [20] správne poznamenáva, že Laplaceovský graf nie je ovplyvňovaný diagonálnymi prvkami matice podobnosti W . Inak povedané slučky grafu neovplyvňujú Laplaceovský graf.

Poznámka 8: Tiež sa dá ukázať, že geometrická násobnosť vlastného čísla 0 pre L je rovná počtu súvislých komponentov pôvodného grafu (špecifikovaného dátami a W) a priestor generovaný vlastnými vektormi (eigenspace) prisluchajúcimi k vlastnému číslu

0 je generovaný tiež indikátorovými vektormi Υ pre všetky súvislé komponenty grafu. Dôkaz pre ľubovoľný počet súvislých komponentov jednoducho vyplýva z blokového tvaru L a vlastností spektra pre blokové matice.

Rôznymi spôsobmi môžeme definovať aj normalizované Laplaceovské grafy. Najčastejšie sa používajú dva:

$$L_{sym} = S^{-\frac{1}{2}} L S^{-\frac{1}{2}} \quad (1.9)$$

a

$$L_{rw} = S^{-1} L. \quad (1.10)$$

Poznámka 9: Ak rozumne definujeme $w_{ii} > 0$, tak S bude zjavne regulárna.

Poznámka 10: Pre tieto normalizované Laplaceovské grafy sa dajú ukázať veľmi podobné vlastnosti ako v prípade nenormalizovaného grafu.

Náš pôvodný problém však nebolo zostrojenie grafu, ale zhlukovanie samotných dát. Tento problém môžeme teraz prepísať v zmysle rezania grafu. Prístup bude značne zjednodušený oproti [20], nakoľko kompletný postup aj s intuíciou by bol príliš dlhý pre túto prácu. Zdefinujeme si len úlohu *RatioCut*, ale existujú aj iné prístupy, napr. *Ncut*. Taká formulácia úlohy potom vedie na problém využívajúci L_{sym} a L_{rw} . Tieto verzie spektrálneho zhlukovania však pre stručnosť tejto kapitoly spomínať nebudeme, dajú sa odvodiť veľmi podobne ako to spravíme teraz pre úlohu *RatioCut*.

Definícia 1.3.9. Pod úlohou ***RatioCut*** budeme rozumieť

$$RatioCut(A_1, \dots, A_K) = \frac{1}{2} \sum_{i=1}^K \frac{W(A_i, A_i^C)}{|A_i|}, \quad (1.11)$$

kde

$$W(A, B) = \sum_{i \in I_A, j \in I_B} w_{ij}, \quad (1.12)$$

kde $i \in I_A$ je skrátený zápis pre $\forall i : v_i$ je vrchol v podgrafe A a $|A|$ je počet vrcholov v podgrafe A .

Naším cieľom bude minimalizovať *RatioCut* cez podgrafy $A_1, \dots, A_K$. Ak teraz definujeme indikátorové vektory ako

$$h_j = (h_{1j}, \dots, h_{nj})^T, \quad (1.13)$$

kde

$$h_{ij} = \begin{cases} \frac{1}{\sqrt{|A_j|}} & \text{ak } v_i \in A_j \\ 0 & \text{inak.} \end{cases}$$

Ak teraz definujeme

$$H_{n \times k} = \begin{pmatrix} | & | & & | \\ h_1 & h_2 & \dots & h_K \\ | & | & & | \end{pmatrix}, \quad (1.14)$$

tak sa dá ukázať, že

$$RatioCut(A_1, \dots, A_K) = tr(H^T L H), \quad (1.15)$$

kde L je nenormalizovaný Laplaceovský graf. Je dôležité si všimnúť, že platí $H^T H = I$. Ďalej teda budeme hľadať riešenie úlohy

$$\begin{aligned} & \min_{H \in \mathbb{R}^{n \times K}} tr(H^T L H) \\ & s.t. \quad H \text{ definované podľa 1.14} \\ & \quad H^T H = I. \end{aligned} \quad (1.16)$$

Ak túto úlohu relaxujeme od podmienky H definované podľa 1.14, tak dostávame jednoduchšiu úlohu, ktorej riešenie dostávame priamo z vety *Rayleigh-Ritz* [21]. Optimálne H pre túto relaxovanú úlohu, teda bude matica zložená z K -vlastných vektorov prislúchajúcich K -najmenším vlastným číslam nenormalizovaného Laplaceovského grafu. Aby sme sa dostali späť do indikátorových vektorov (z relaxovanej úlohy), využijeme algoritmus K -means pre zhľukovanie riadkov matice H . Dostávame tak algoritmus Spektrálneho zhľukovania s nenormalizovanou maticou L , ktorý uvádzame v pseudokóde 2.

Algorithm 2: Nenormalizované spektrálne zhukovanie**Input** : Matica dizajnu X , prirodzené číslo K a norma**Output:** Index zhukov idx

- 1 Skonstruujeme maticu podobnosti W podľa zvolenej podobnosti.
- 2 Vypočítame nenormalizovaný Laplaceovský graf $L = S - W$.
- 3 Vypočítame K -vlastných vektorov prislúchajúcich K -najmenším vlastným číslam matice L .
- 4 Maticu $H \in \mathbb{R}^{n \times K}$ naplníme týmito vlastnými vektormi.
- 5 Označíme riadky H ako $y_i \in \mathbb{R}^K$
- 6 Pozhukujeme $y_1, \dots, y_n$ pomocou K -means.
- 7 Zhluky $C_1, \dots, C_K$ zapíšeme pomocou vektora idx .

1.3.4 DBSCAN

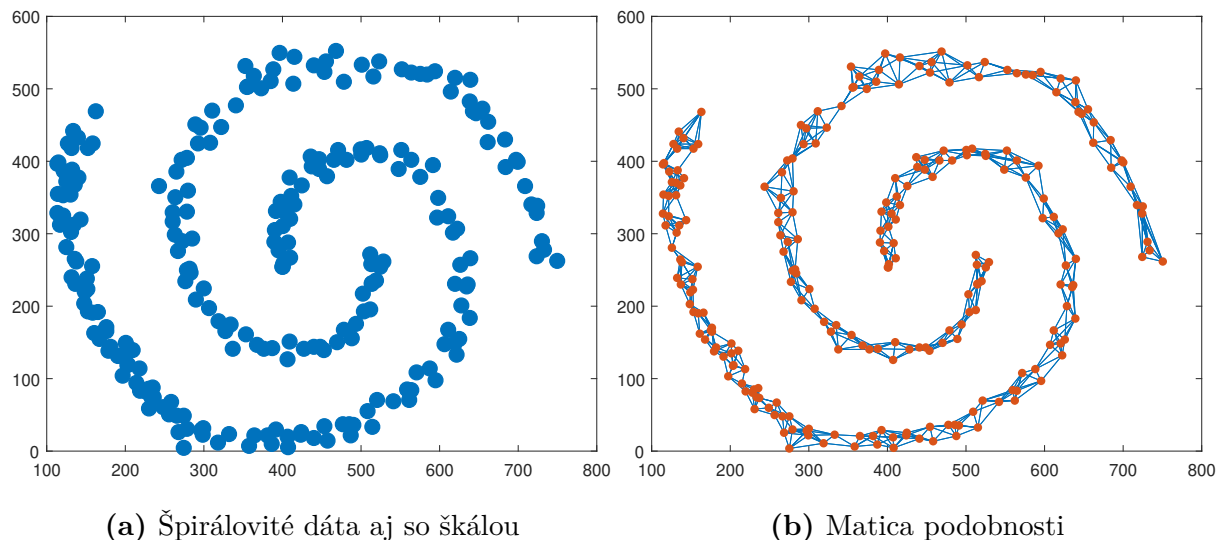
Zhlukovaciemu algoritmu DBSCAN sa nebudeme venovať tak podrobne. Uvedieme však, že pre vytváranie zhukov využíva hustotu samotných dát a zhlučuje tiež do jednej špeciálnej skupiny, ktorú nazývame šum. Do tejto skupiny sa pokúsi zaradiť všetkých outlierov z dát. Výhodou tiež je, že nemusíme špecifikovať počet clusterov. Bohužiaľ, musíme špecifikovať 2 parametre ε a **minPts**. DBSCAN klasifikuje body na jadrové, dosiahnuteľné alebo šum. Vyžadujeme nasledujúce vlastnosti podľa [9].

- Bod p je jadrovým bodom, ak aspoň **minPts** bodov je v jeho ε -ovom okolí (stačí nám okolie, nepotrebujeme prstencové okolie).
- Bod q je priamo dosiahnuteľným bodom z bodu p , ak je v ε -ovom okolí bodu p , kde p je jadrovým bodom.
- Bod q je dosiahnuteľným bodom z bodu p , pokiaľ existuje cesta $p_1, p_2, \dots, p_n$, kde $p_1 = p$ a $p_n = q$ a p_{i+1} je priamo dosiahnuteľným bodom z bodu p_i .
- Všetky ostatné body budeme nazývať šumom a označovať ich indexom 0.

1.3.5 Ukážkový príklad

Podme sa teraz pozrieť na ukážkový príklad špirálovitých dát a ako dobre sa s nimi dokážu popasovať metódy K -means, K -medoids, Spektrálne zhukovanie a samozrejme DBSCAN.

Na obrázku 1.10a uvádzame pôvodné dáta, ktoré zjavne pochádzajú z dvoch zhlukov. Na obrázku 1.10b uvádzame vizualizáciu matice podobnosti pre tieto dáta získanú pomocou 7-najbližších susedov a ocenenú Gaussovsky, t.j. pomocou vzťahu 1.7 s konkrétnou hodnotou $\sigma = 20$ a euklidovskou normou.



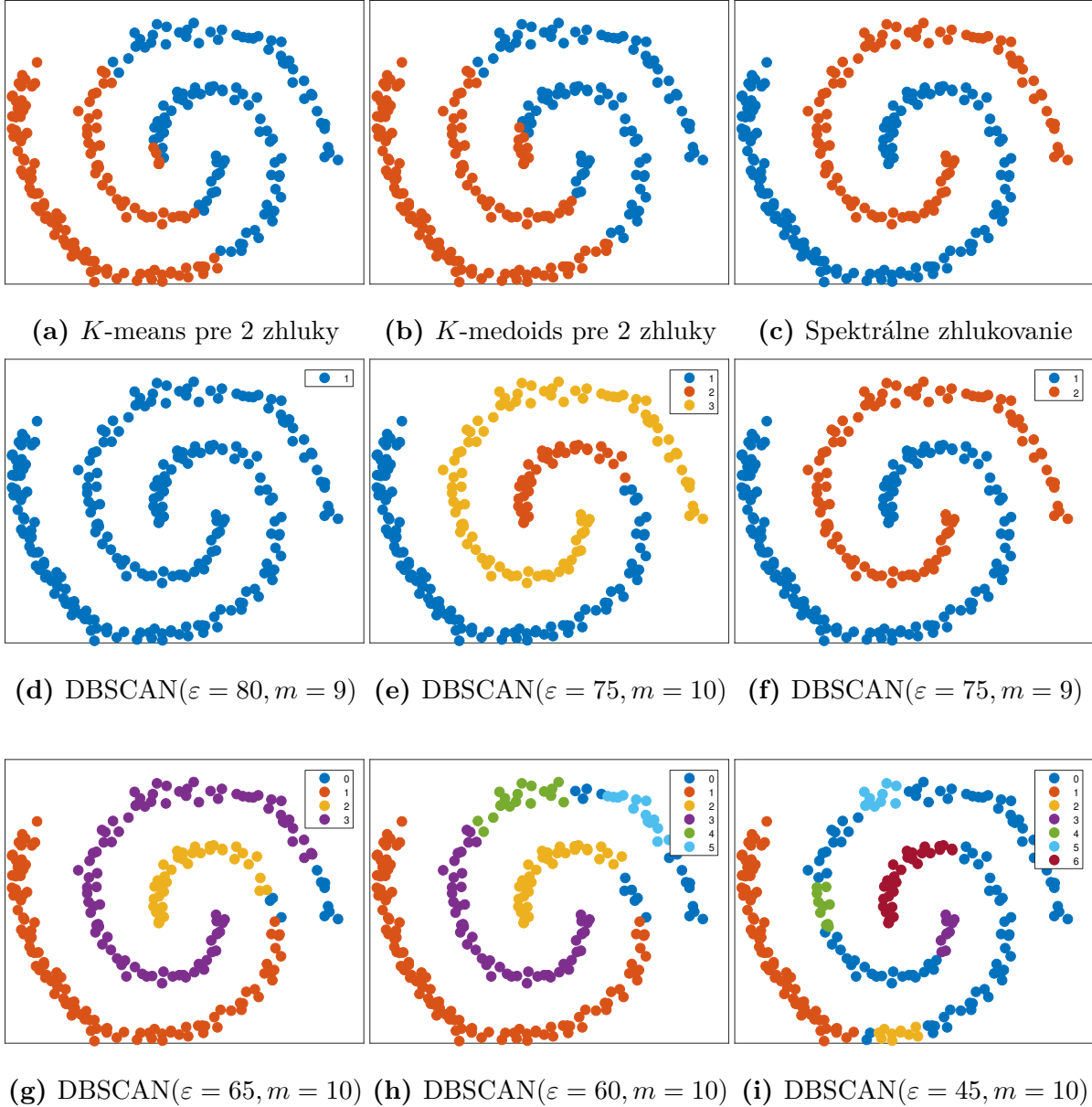
Obr. 1.10: Ukážkový príklad špirálovitých dát a matice podobnosti 7-najbližších susedov váhovanej $\sigma = 20$

Kvôli snahe zmestiť všetky grafy na túto stranu sme na obrázku 1.11 odstránili obe osi, škála však zostáva rovnaká ako na obrázku 1.10.

Na obrázku 1.11a a 1.11b môžeme vidieť, ako sme očakávali, že takéto zhľukovanie algoritmy K -means a K -medoids nezvládli, keďže tieto metódy vo všeobecnosti hľadajú guľovité zhľuky. Spektrálne zhľukovanie využívalo nenormalizovaný Laplaceovský graf a zjavne dokázalo nájsť hľadanú štruktúru v dátach, ako vidíme na obrázku 1.11c.

Na obrázkoch 1.11d až 1.11i uvádzame výsledky algoritmu DBSCAN pre rôzne voľby parametrov ε a minPts. Zjavne vidíme, že úspešnosť veľmi závisí na voľbe parametrov. Tie ovplyvňujú nielen počet nájdených zhlukov, ale aj ich typ, preto v týchto grafoch uvádzame aj legendu pre konkrétne zhľuky, keďže v niektorých prípadoch zhľuky hovoria o šume (označujeme, ako v celej práci indexom 0).

Skôr než teda budeme používať DBSCAN na zhľukovanie tvárí v kapitole 2, pokúsime sa pre rôzne druhy reprezentácie tvárí odhadnúť parametre tejto metódy, a budeme dúfať, že aj na iných databázach bude voľba týchto parametrov vhodná, inak nebudeme môcť prehlásiť metódu za vhodnú pre náš problém.



Obr. 1.11: Ukážkový príklad zhľukovania pomocou K -means, K -medoids, Spektrálneho zhľukovania a DBSCAN pre rôzne parametre ε a minPts

1.3.6 Zhľukovanie založené na poradí

V tejto časti uvedieme pôvodný algoritmus pre zhľukovanie tvárí založený na poradí (*Rank-Order Distance based clustering*), prvýkrát popísaný v [12]. Základom celého algoritmu bola voľba špeciálnej metriky pre vektory črt vychádzajúce z tvárí. Autori uvádzajú, že niekedy je jednoduchšie identifikovať 2 tváre na základe ich najbližších zdieľaných susedov. V nasledujúcej časti zavedieme všetky pojmy potrebné pre popísanie algoritmu.

Definícia 1.3.10. *Nech a predstavuje vektor črt pre jednu konkrétnu tvár. Pod **poradovníkom tváre** a O_a , budeme rozumieť zoradenie všetkých tvárí na základe vzdialeností od tváre a .*

Poznámka 11: Pre našu implementáciu sme poradovník O_a chápali ako vektor dĺžky m , t.j. počet tvárí, kde i -ta zložka predstavovala poradie vzdialenosti medzi tvárou a a i , spomedzi vzdialeností a a inej tváre. V O_a , teda na mieste prislúchajúcom tvári a , je zjavne číslo 0 a pre najvzdialenejšiu tvár od tváre a , je v tomto poradovníku číslo $m - 1$.

Definícia 1.3.11. *Pod **asymetrickou vzdialenosťou založenou na poradí** $D(a, b)$, budeme rozumieť*

$$D(a, b) = \sum_{i=0}^{O_a(b)} O_b(f_a(i)), \quad (1.17)$$

kde O_a je poradovník tváre a , $f_a(i)$ je i -ta tvár v poradovníku a , $O_a(b)$ je poradie tváre b v poradovníku tváre a .

Poznámka 12: Takáto vzdialenosť vlastne hovorí aké poradie v poradovníku O_b majú tváre najbližšie k tvári a . Prirodzene nás nezaujímajú všetky tváre, ale len tie, ktoré sú k a bližšie ako tvár b .

Definícia 1.3.12. *Pod **vzdialenosťou tvárí založenou na poradí** budeme rozumieť*

$$D^R(a, b) = \frac{D(a, b) + D(b, a)}{\min(O_a(b), O_b(a))} \quad (1.18)$$

Poznámka 13: Takáto vzdialenosť nespĺňa trojuholníkovú nerovnosť 1.5, teda nie je metrikou.

Autori ďalej uvádzajú, že zhlukovanie tvárí so vzdialenosťou menšou ako nejaký parameter t ešte nezaručí, že dostaneme dostatočne dobré zhlukovanie, pretože vďaka tomuto získame len veľmi očividné zhluky. Jedna osoba však môže mať v databáze aj výrazne sa líšiace fotografie (napríklad zobrazené iné emócie). V ďalších krokoch by sme teda chceli zhlukovať aj takéto veľmi tesné zhluky do väčších. Zovšeobecníme preto vzdialenosti pre zhluky zložené z viacerých tvárí a zavedieme aj akúsi normovanú vzdialenosť.

Definícia 1.3.13. Pod *jednoduchou vzdialenosťou zhlukov (single linkage)* budeme rozumieť

$$d_{SL}(C_i, C_j) = \min_{\forall a \in C_i, \forall b \in C_j} d(a, b), \quad (1.19)$$

kde $d(a, b)$ je ľubovoľná norma.

Definícia 1.3.14. Pod *vzdialenosťou zhlukov založenou na poradí* budeme rozumieť

$$D^R(C_i, C_j) = \frac{D(C_i, C_j) + D(C_j, C_i)}{\min(O_{C_i}(C_j), O_{C_j}(C_i))}, \quad (1.20)$$

kde $D(C_i, C_j)$ je definovaná analogicky pre zhluky podľa 1.17 a O_{C_j} je poradovník zhlukov pre zhluk C_j analogicky k definícii 1.3.10.

Definícia 1.3.15. Pod *normovanou vzdialenosťou zhlukov* budeme rozumieť

$$D^N(C_i, C_j) = \frac{1}{\phi(C_i, C_j)} d_{SL}(C_i, C_j), \quad (1.21)$$

kde

$$\phi(C_i, C_j) = \frac{1}{|C_i| + |C_j|} \sum_{a \in C_i \cup C_j} \frac{1}{\mathcal{K}} \sum_{k=1}^{\mathcal{K}} d(a, f_a(k)), \quad (1.22)$$

kde $f_a(k)$ je k -ta najbližšia tvár k tvári a , $|C_i|$ je kardinalita zhuku C_i , $\mathcal{K}$ je parameter, $\phi(C_i, C_j)$ je priemerná vzdialenosť tvárí zo zhlukov C_i a C_j s ich $\mathcal{K}$ najbližšími susedmi.

Poznámka 14: Je dôležité poznamenať, že v sume $\sum_{k=1}^{\mathcal{K}} d(a, f_a(k))$ rátame vzdialenosti medzi tvárou a a jej $\mathcal{K}$ najbližšími susedmi zo všetkých tvárí, nielen tých zo zhlukov C_i a C_j .

Detailný popis algoritmu, tak ako ho uviedli autori [12], uvádzame v pseudokóde 3. V tomto popise sme však zvýraznili číslo **1** v jednej z podmienok pre zhlučovanie. Vzhľadom na simulačné výsledky sa ukázalo, že ak na tomto mieste použijeme konštantu **2**, dostaneme signifikantne lepšie výsledky. Túto voľbu budeme neskôr demonštrovať na reálnych príkladoch v kapitole 2.

Algorithm 3: Rank-Order distance based clustering**Input** : Matica dizajnu X , prirodzené čísla t, K a norma**Output:** Index zhlukov idx

```

1 Inicializujeme všetky tváre ako samostatné zhluky  $C_1, \dots, C_m$ .
2 while v iterácii nastala nejaká zmena zhlukov do
3   forall pár  $C_i$  a  $C_j$  do
4     Vypočítame vzdialenosti  $D^R(C_i, C_j)$  a  $D^N(C_i, C_j)$ .
5     if  $D^R(C_i, C_j) < t$  a  $D^N(C_i, C_j) < 1$  then
6       Označíme  $(C_i, C_j)$  ako kandidátov pre zlúčenie zhlukov.
7     end
8   end
9   Transitívne zjednotíme všetkých kandidátov pre zlúčenie, t.j. ak  $(C_i, C_j)$  aj
       $(C_i, C_k)$  sú kandidátmi, tak zjednotíme zhluky  $C_i, C_j$  a  $C_k$ .
10  Upravíme indexy zhlukov  $idx$ .
11 end
12 Všetky jednoprvkové zhluky označíme indexom 0 ako šum.
```

1.3.7 Aglomeratívne zhlukovanie

Skôr než prejdeme k samotnému algoritmu, musíme najskôr zadať vzdialenosť medzi zhlukmi. V definícii 1.3.13 sme už spomenuli prvú možnú vzdialenosť zhlukov, *single linkage*.

Definícia 1.3.16. Pod *complete linkage vzdialenosťou zhlukov* C_i a C_j budeme rozumieť

$$d_{CL}(C_i, C_j) = \max_{a \in C_i, b \in C_j} d(a, b), \quad (1.23)$$

kde $d(a, b)$ je ľubovoľná norma.

Definícia 1.3.17. Pod *average linkage vzdialenosťou zhlukov* C_i a C_j budeme rozumieť

$$d_{AL}(C_i, C_j) = \frac{1}{|C_i| \cdot |C_j|} \sum_{a \in C_i} \sum_{b \in C_j} d(a, b), \quad (1.24)$$

kde $d(a, b)$ je ľubovoľná norma.

Často využívaná je tiež **Wardova vzdialenosť (Ward's linkage) zhlukov**. V článku [17] pôvodne metódu autor zamýšľal ako spojenie zhlukov, ktoré dávajú momentálnu optimálnu hodnotu nejakej zvolenej účelovej funkcie a tento prístup ilustroval na účelovej funkcii pre K -means, teda vnútrozhlukovej sume štvorcov popísanej v optimalizačnej úlohe 1.1. Táto účelová funkcia sa ujala a dnes je všeobecne Wardova vzdialenosť braná práve s touto funkciou.

Triviálne sa dá ukázať, že pre $K_1 < K_2$ je optimálna hodnota úlohy 1.1 menšia v prípade voľby počtu centroidov K_2 , ako v prípade K_1 .

Tiež sa dá ľahko ukázať, že ak $K = n$, t.j. ak počet hľadaných centroidov je rovný počtu bodov, tak optimálna hodnota WCSS je 0. V zásade teda v aglomeratívnom zhľukovaní s Wardovou vzdialenosťou spájame tie zhľuky, ktoré najmenej zvýšia účelovú funkciu K -means. Formálne zapísané:

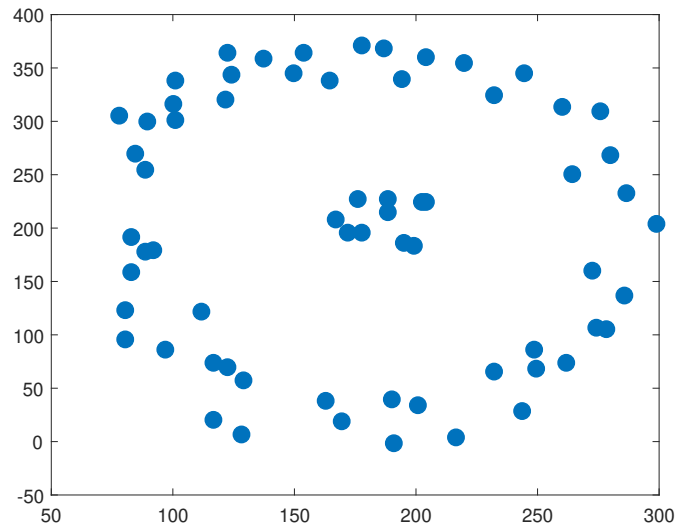
Definícia 1.3.18. Pod **Ward's linkage vzdialenosťou zhlukov** C_i a C_j budeme rozumieť

$$d_{WL}(C_i, C_j) = \sum_{a \in C_i \cup C_j} \|a - \bar{x}_{ij}\|_2^2 - \sum_{b \in C_i} \|b - \bar{x}_i\|_2^2 - \sum_{c \in C_j} \|c - \bar{x}_j\|_2^2, \quad (1.25)$$

kde $\bar{x}_{ij}$ je centroidom zhľuku $C_i \cup C_j$, $\bar{x}_i$ je centroidom zhľuku C_i a $\bar{x}_j$ je centroidom zhľuku C_j .

Ako sme už spomenuli v úvode tejto podkapitoly, aglomeratívne zhľukovanie je jeden zo spôsobov hierarchického zhľukovania. Pre potreby zhľukovania tvárí však nie je pre nás dostatočujúce nájsť hierarchiu. My potrebujeme dokonca *hard* zhľukovanie. Aglomeratívne clustrovanie v každej iterácii spojí dva najbližšie zhľuky (kde najbližšie je určené podľa linkage vzdialenosti, ktorú uvažujeme).

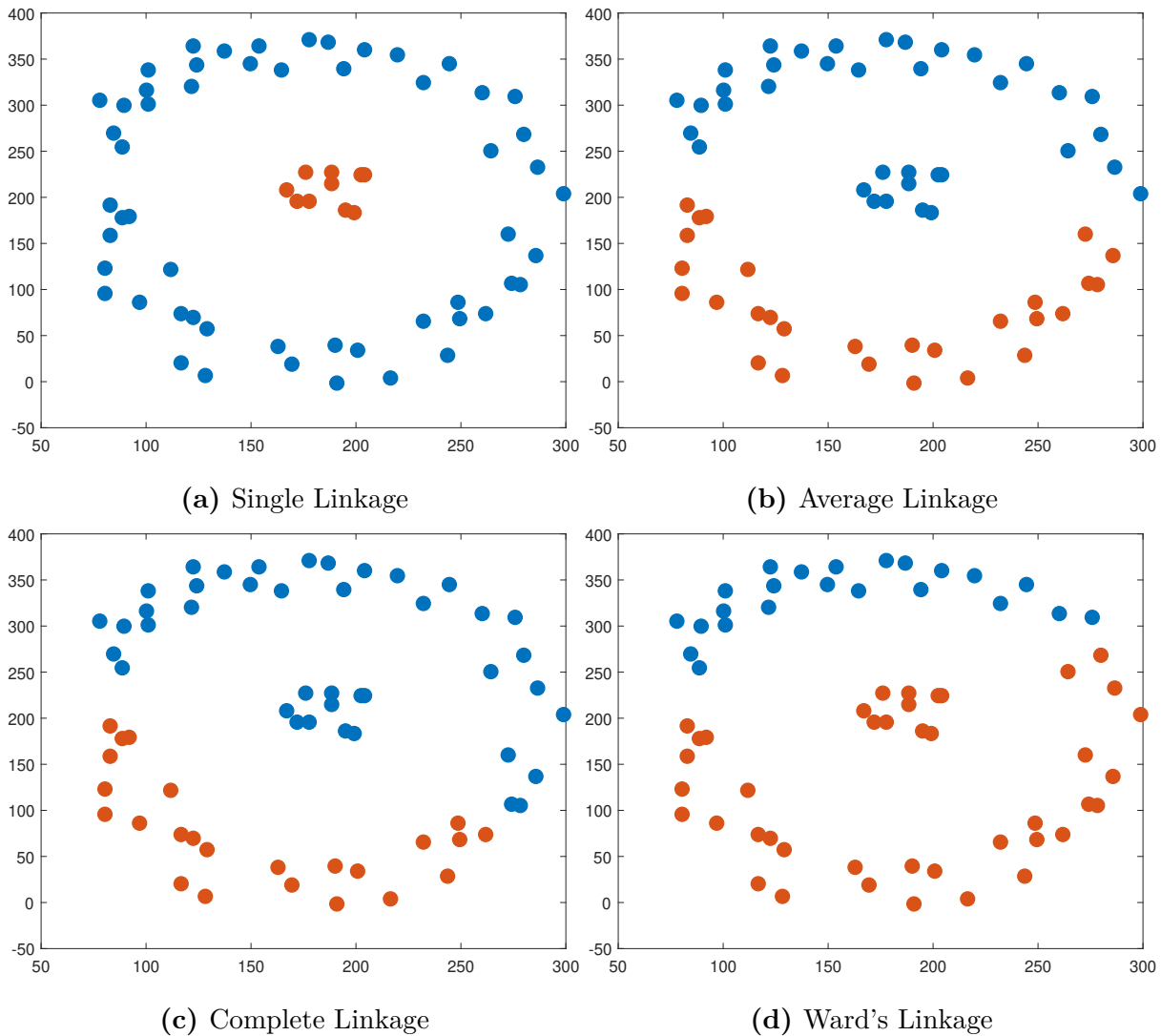
Ak dokážeme špecifikovať počet žiadaných zhlukov K , tak takéto zhľukovanie vieme zastaviť a dostať tak *hard* zhľuky. Pozrime sa na ukážkový príklad s vymyslenými dátami zobrazenými na obrázku 1.12.



Obr. 1.12: Ukážkový príklad pre zhlukovanie s dvojrozmernými dátami

Človek zjavne vidí, že sa tam nachádzajú dve skupiny, jedna vnútorná s veľmi nízkou varianciou a jedna vonkajšia rozložená približne po obode nejakej elipsy. Mohlo by nás zvádzať rozdeliť túto elipsu na dve polovice, keďže sa tam nachádzajú nejaké väčšie medzery, ale povedzme že nám ide len o nájdenie dvoch zhlukov.

Na ukážkovom príklade sme teda spravili aglomeratívne zhlukovanie so štyrmi typmi vzdialeností, ktoré sme definovali a zastavili sme ho pred posledným spojením do jedného zhluku. Vo všetkých sme využili euklidovskú normu. Výsledky uvádzame na obrázku 1.13.



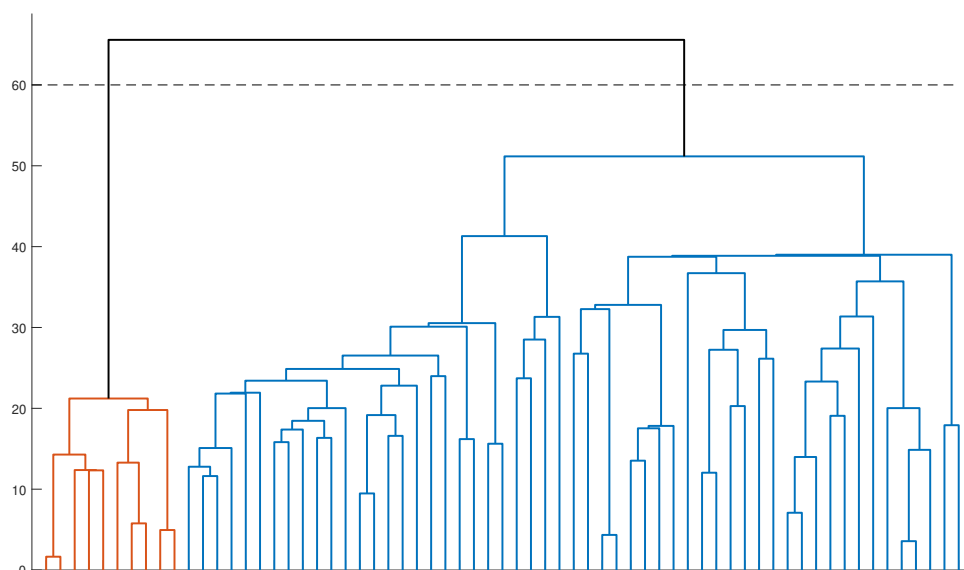
Obr. 1.13: Ukážkový príklad hard aglomeratívneho zhľukovania pre 2 zhľuky

Na obrázkoch vidíme, že túto konkrétnu úlohu zvládlo len *single linkage* zhľukovanie.

Čo graficky znamená zastavenie aglomeratívneho zhľukovania? V zásade ide len o akési horizontálne (respektíve vertikálne, v závislosti od vykreslenia samotného binárneho stromu) pílenie samotného dendrogramu. Farebný dendrogram uvádzame na obrázku 1.14 aj s pílením vyznačením prerušovanou čiernou horizontálnou čiarou.

Je nutné poznamenať, že táto čiara mohla byť nakreslená aj nižšie aj vyššie. Naša požiadavka bola len o počte zhľukov, no jedno z kritérií pre získanie *hard* zhľukovania môže byť napríklad veľkosť samotnej vzdialenosti medzi zhľukmi. Ak by sme zvolili konštantu, nad ktorú sa nechceme dostať, tak by výsledkom bolo *hard* zhľukovanie bez nutnosti udať počet zhľukov. Voľba takejto konštanty však môže byť aspoň taká náročná, ako voľba

samotného počtu zhlukov.



Obr. 1.14: Ukážkový príklad pílania dendrogramu pre single linkage aglomeratívne zhľukovanie a počet zhľukov 2

1.3.8 Zhľukovanie GPS súradníc

Pri zhľukovaní dát narážame na problém počítania vzdialeností. Kým pri črtách tváre malo zmysel uvažovať aj o euklidovskej vzdialenosti, pri gps súradniciach takto počítat určite nemôžeme. Pre výpočet vzdialeností využijeme haversinovskú rovnicu, ktorá je popísaná napríklad v [10] nasledovne:

$$d = 2R \arcsin\left(\sqrt{\sin^2\left(\frac{\varphi_1 - \varphi_2}{2}\right) + \cos(\varphi_1)\cos(\varphi_2)\sin^2\left(\frac{\lambda_1 - \lambda_2}{2}\right)}\right), \quad (1.26)$$

kde R je polomer Zeme, φ_1 a φ_2 sú geografické šírky dvoch miest a λ_1 a λ_2 sú ich zemepisné dĺžky, pričom všetky uvažujeme udané v radiánoch. Preto treba pred počítaním pri konkrétnych dátach spraviť jednoduchú konverziu. My budeme, ale používať už priamo vstavanú funkciu MATLABu `distance()`.

1.4 Vyhodnocovanie výsledkov zhľukovania

Zhľukovanie predstavuje prístup, ktorý môžeme vo všeobecnosti chápať ako učenie bez učiteľa. Pre samotný problém hľadania rovnakých ľudí vo fotografiách vo všeobecnosti

nemáme údaje o správnych odpovediach. Zhluky môžu byť rôznych veľkostí a osoby môžu byť od seba rôzne vzdialené. Stále teda chceme využívať metódy, ktoré nepotrebnú učiteľa, ale tiež ich chceme nejakým spôsobom ohodnotiť. Narážame na jeden problém, a to ten, že v prípade zhlučovania je voľba čísla pre daný zhluk ľubovoľná. Ak by sme mali dokonalé zhlučovanie a len náhodne vymeníme označenia celých zhlukov, stále to považujeme za dokonalé zhlučovanie. Pre takýto prípad teda nie je úplne intuitívne, ako by sme mali vyhodnotiť samotnú chybu.

1.4.1 Adjusted Rand Index

Jedna z metód pre vyhodnocovanie zhlučovania je **Adjusted Rand Index**, definovaný v [11] ako

$$ARI(x, y) = \frac{\sum_{i,j} \binom{n_{ij}}{2} - \frac{\sum_i \binom{a_i}{2} \sum_j \binom{b_j}{2}}{\binom{n}{2}}}{\frac{1}{2} \left(\sum_i \binom{a_i}{2} + \sum_j \binom{b_j}{2} \right) - \frac{\left(\sum_i \binom{a_i}{2} \sum_j \binom{b_j}{2} \right)}{\binom{n}{2}}}, \quad (1.27)$$

kde n_{ij} , a_i a b_j dostaneme klasicky z kontingenčnej tabuľky pre x a y , n je dĺžka vektora zhlukov (vektor dodaný zhlučovacím algoritmom a vektor skutočných zhlukov musia byť samozrejme rovnako dlhé, aby mal výpočet zmysel). V špeciálnych prípadoch rátame $\binom{m}{k}$, kde $k > m$. Pre tieto definujeme $\binom{m}{k} := 0$.

Dá sa ukázať, že pre náhodné zhlučovanie je stredná hodnota ARI rovná 0. Ak by naše výsledky dostávali negatívne ARI, znamenalo by to, že metóda sa správala vo všeobecnosti horšie ako náhoda. Naopak dokonalé zhlučovanie dostane ARI rovné presne 1. Tiež sa dá ukázať, že ARI je invariantné vzhľadom na permutácie označení zhlukov.

1.4.2 Kritériá Precision, Recall, Compression Ratio

Podobne ako v [12] definujeme kritériá *Precision*, *Recall* a *Compression Ratio*. *Precision* hovorí, aké dobré je zhlučovanie z hľadiska nezaradzovania nesprávnych tvárí do spoločných zhlukov. Takúto hodnotu očakávame veľmi vysokú od spoľahlivého zhlučovacieho algoritmu, nakoľko spätná oprava zhlukov je aj z implementačného aj užívateľského hľa-

diska veľmi nežiadúca.

$$Precision = 1 - \frac{\sum_{i=1}^Q |M_i|}{\sum_{i=1}^Q |C_i|}, \quad (1.28)$$

kde $|M_i|$ je kardinalita množiny nesprávne zaradených tvárí v zhľuku C_i a Q je počet nájdených zhľukov.

Kritérium *Recall* sa snaží zachytiť schopnosť algoritmu zhľukovať tváre, ktoré v skutočnosti nie sú šumom do zhľukov (t.j. nenechať takéto tváre v samostatných zhľukoch, lebo tie neskôr spojíme a prehlásime ich za šum).

$$Recall = \frac{\sum_{i=1}^Q |C_i| - |n_i|}{|F| - |noise|}, \quad (1.29)$$

kde $|n_i|$ je počet šumov v i -tom zhľuku, $|noise|$ je celkový počet šumu nájdený v databáze a $|F|$ je celkový počet nájdených tvárí (t.j. aj šum aj skutočné tváre).

Compression Ratio hovorí o veľkosti samotných zhľukov. Čím väčšie sú samotné zhľuky, tým nižšia je interakcia užívateľa, ktorý by musel manuálne rôzne zhľuky prislúchajúce tej istej osobe zjednotiť.

$$Compression\ Ratio = \frac{1}{Q} \sum_{i=1}^Q |C_i| \quad (1.30)$$

1.5 Kombinácia rôznych zhľukovaní

V predchádzajúcich častiach sme spomenuli rôzne zhľukovacie algoritmy aj ako ich použiť v špeciálnych prípadoch zhľukovania času a gps súradníc. Čo sme zatiaľ nespomenuli je však spôsob ako spojiť takéto prístupy do jedného a vo všeobecnosti ako spojiť akékoľvek dve zhľukovania.

V ďalších dvoch častiach budeme predpokladať teda už znalosť zatriedenia objektov do zhľukov, avšak získanú z iných dát ako ideme momentálne zhľukovať. Pre jednoduchosť môžeme uvažovať, že sa nám podarilo získať zhľukovanie samotných fotografií podľa času (respektíve GPS súradníc) a chýba nám len zaradiť tváre nájdené na týchto fotografiách do zhľukov.

1.5.1 Naivný prístup

Naivný prístup spočíva v zaradení údajov o zhľuku fotografií do samotných vektorov črt. Pôjde teda o jednu kategorickú premennú, ktorá bude mať faktory prislúchajúce rôznym zhľukom v našom vopred pripravenom zhľukovaní. Treba si dávať pozor, že na fotografii sa môžu nachádzať viacerí ľudia súčasne, a teda informácia, že niektoré fotografie pochádzajú z rovnakej časovej udalosti nám má len pomôcť v zhľukovaní tvárí, nemá nás donútiť označiť dvoch ľudí na jednej fotografii ako rovnakého človeka.

Mnoho metód nedokáže priamo pracovať s kategorickými črtami, tobôž nie s kombináciou kategorických a numerických veličín. Rozhodli sme sa preto využiť prepísanie kategorickej premennej to takzvaných dummy premenných, nazývaných aj indikátorové premenné, alebo one-hot encoding. Aj keď v prípade napríklad zovšeobecnenej lineárnej (a teda aj klasickej lineárnej) regresie musíme jeden z faktorov premennej odstrániť a jej interpretáciu zachovať v absolútnom člene pre vyhnutie sa singularite, v tomto prípade je skôr žiadúce využiť indikátor pre všetky faktory kategorickej premennej.

1.5.2 Adaptácia zhľukovania založeného na poradí

Špeciálne pre zhľukovanie pomocou 1.3.6 sme sa rozhodli upraviť metódu, aby sme dokázali pracovať s nejakou počiatočnou informáciou o časových (alebo iných zhľukoch). Konkrétne sme sa rozhodli zmeniť spôsob počítania normovanej vzdialenosti zhľukov v rovnici 1.21. Zdefinujme si teda náš spôsob.

Definícia 1.5.1. *Pod normovanou vzdialenosťou zhľukov s apriori zhľukovou informáciou budeme rozumieť*

$$D_{AC}^N(C_i, C_j) = \frac{1}{\phi(C_i, C_j)} d_{SL}(C_i, C_j) \cdot (1 - \beta \cdot \xi(C_i, C_j)), \quad (1.31)$$

kde

$$\phi(C_i, C_j) = \frac{1}{|C_i| + |C_j|} \sum_{a \in C_i \cup C_j} \frac{1}{\mathcal{K}} \sum_{k=1}^{\mathcal{K}} d(a, f_a(k)), \quad (1.32)$$

$$\xi(C_i, C_j) = \frac{\sum_{a \in C_i} \sum_{b \in C_j} \tilde{\delta}_{AC_a, AC_b}}{|C_i| \cdot |C_j|}. \quad (1.33)$$

- $f_a(k)$ je k -ta najbližšia tvár k tvári a ,

- $|C_i|$ je kardinalita zhuku C_i ,
- $\mathcal{K}$ je parameter označujúci počet najbližších susedov,
- $\phi(C_i, C_j)$ je priemerná vzdialenosť tvárí zo zhukov C_i a C_j s ich $\mathcal{K}$ najbližšími susedmi,
- $d_{SL}(C_i, C_j)$ je single linkage vzdialenosť zhukov popísaná v definícii 1.3.13,
- β je parameter upravujúci mieru dôvery dávanej apriori zhukovaniu,
- AC_a je index zhuku objektu a v apriori zhukovaní (AC sme to označili ako apriori clustering)
- $\tilde{\delta}_{AC_a, AC_b}$ definujeme ako **upravené Kroneckerovo delta** nasledovne:

$$\tilde{\delta}_{AC_a, AC_b} = \begin{cases} 1 & \text{ak } AC_a = AC_b \neq 0 \\ 0 & \text{inak.} \end{cases}$$

Poznámka 15: Upravené Kroneckerovo delta sme definovali takým spôsobom, aby sme sa neprikláňali k zhukovaniu vecí patriacich do zhuku šum. Tento zhuk označujeme číslom 0, ako sme to už spomenuli v sekcii 1.3.

Poznámka 16: Funkcia $\xi(C_i, C_j)$ vlastne hovorí o proporcii objektov zo zhukov C_i a C_j , ktoré zdieľajú rovnaký apriori zhuk. Je to teda číslo patriace intervalu $[0, 1]$.

1.6 Voľba počtu zhukov

Niektoré zhukovacie metódy, ktoré sme uviedli v tejto kapitole potrebovali vopred špecifikovaný počet zhukov. Vo všeobecnosti však takýto údaj v našom prípade zhukovania tvárí nemáme. Pozrime sa teda na 2 prístupy, ako môžeme počet zhukov odhadnúť.

Je dobré poznamenať, že v zásade budeme musieť zhukovať pre rôzne hodnoty počtu zhukov K , a preto je veľmi vhodné ak v reálnom prípade dokážeme aspoň ohraničiť K zhora aj zdola.

1.6.1 Siluety

Najskôr si uvedieme čo silueta znamená pre nejaké konkrétne zhľukovanie, a potom uvedieme prístup, akým nám pomôže nájsť optimálny počet zhľukov. V tejto časti budeme vychádzať najmä z knihy [1], ale mierne upravíme značenie.

Definícia 1.6.1. *Majme dané jedno konkrétne zhľukovanie I_K . Pod **siluetou** budeme rozumieť*

$$s_i(I_K) = s_{iK} = \frac{b_i - a_i}{\max\{a_i, b_i\}}, \quad (1.34)$$

kde a_i je priemerná nepodobnosť objektu i a ostatných objektov z jeho zhľuku a b_i je priemerná nepodobnosť objektu i a objektov zo susedného zhľuku. Pod susedným zhľukom, rozumieme taký zhľuk, v ktorom sa nenachádza objekt i , ale je najbližším z hľadiska priemernej nepodobnosti s objektom i .

Poznámka 17: I_K v tomto prípade predstavuje vektor, v ktorom sa nachádzajú indexy zhľukov pre všetky dostupné dáta.

Poznámka 18: Zjavne veľmi závisí ako definujeme nepodobnosť objektov, avšak nezávisle od toho $s_{iK} \in [-1, 1]$ a dobre zaradené objekty majú siluety blízke hodnote 1, zle zaradené ich majú blízko hodnote -1.

V tejto definícii sme strategicky využili pri označení zhľukovania I_K písmeno K . Ak totiž získame zhľukovania pre rôzne K , tak vieme napočítať akúsi **priemernú šírku siluety** pre konkrétne K :

$$\overline{s_K} = \frac{1}{n} \sum_{i=1}^n s_{iK}. \quad (1.35)$$

Definícia 1.6.2. *Pod **koeficientom siluety** potom budeme rozumieť*

$$SC = \max_K \overline{s_K}. \quad (1.36)$$

Ako odhadované optimálne $\widehat{K}$ potom teda zvolíme

$$\widehat{K} = \operatorname{argmax}_K \overline{s_K}. \quad (1.37)$$

V knihe [8] ešte autori subjektívne zhodnotili zhľukovanie na základe koeficientu siluety, nás však nebude zaujímať veľkosť tohto koeficientu, ako samotné odhadnuté K .

1.6.2 Caliński-Harabasz kritérium

Calińského-Harabaszove kritérium, niekedy nazývané aj *Variance Ratio Criterion*, je iný prístup pre odhadnutie počtu zhlukov K . My ho uvedieme s upraveným značením oproti pôvodnému článku [18]. Pre nejaké konkrétne K je definovaná ako

$$VRC_K = \frac{S_B}{S_W} \cdot \frac{(n - K)}{(K - 1)}, \quad (1.38)$$

kde

$$S_B = \sum_{i=1}^K |C_i| \cdot (\bar{x}_i - \bar{x})^T (\bar{x}_i - \bar{x}) = \sum_{i=1}^K |C_i| \cdot \|\bar{x}_i - \bar{x}\|_2^2 \quad (1.39)$$

je mimo-zhluková variancia a

$$S_W = \sum_{i=1}^K \sum_{a \in C_i} \|a - \bar{x}_i\|_2^2 \quad (1.40)$$

je vnútro-zhluková variancia. $|C_i|$ je kardinalita zhľuku C_i , $\bar{x}_i$ je centroid zhľuku C_i , $\bar{x}$ je priemer všetkých dát a n je počet všetkých dát. Pre dobré zhľukovanie chceme vysokú mimo-zhlukovú varianciu a nízku vnútro-zhlukovú varianciu. V tomto kritériu teda odhadujeme optimálne $\widehat{K}$ ako

$$\widehat{K} = \operatorname{argmax}_K VRC_K. \quad (1.41)$$

Kapitola 2

Demonštračné príklady

V tejto kapitole využijeme všetky algoritmy uvedené v predchádzajúcej kapitole a pozrieme sa, ktoré nastavenia fungujú najlepšie na rôznych reálnych fotografiách z osobnej zbierky. Súčasťou diplomovej práce nebudú všetky testované databázy, ale len niektoré vybrané pre zachovanie súkromia.

Na experimenty bola použitá verzia MATLABu R2017a 64bit a operačný systém Microsoft Windows 10. Tieto boli nainštalované na SSD Samsung 750 Evo, respektíve Samsung 840 Pro. Niektoré výpočty boli urýchlené grafickou kartou Nvidia 1070Ti. Voľba systému samozrejme veľmi ovplyvňuje výpočtové časy.

2.1 Voľba parametrov na samostatnej databáze

Viacere metódy, ktoré sme spomenuli v prvej kapitole v časti 1.3 obsahovali jeden alebo viac parametrov nesúvisiacich priamo s počtom zhlukov. Považujeme za veľmi dôležité preskúmať voľbu práve týchto parametrov, nakoľko dokážu značne ovplyvniť výsledky a niekedy nie je vôbec jasné, akým spôsobom by sme ich mali voliť.

Všetky zhlukovacie metódy boli v zásade metódy učenia bez učiteľa. My sme sa však rozhodli niektoré z týchto parametrov zvoliť na inej databáze tváří a dúfať, že takto zvolené parametre budú vhodné aj pri použití na nových fotografiách. V zásade teda budeme pracovať s nejakou trénovacou množinou.

V tejto časti je tiež dobré poznamenať, že aby sme predišli príliš nízkym hodnotám kritéria Adjusted Rand Index, budeme v metóde zhlukovania založeného na poradí brať

aj jednoprvkové zhluky ako samostatné a nebudeme ich spájať do jedného zhluku. Toto kritérium totiž nedokáže dobre vyhodnotiť tento konkrétny zhluk. Výsledky pre túto metódu presne tak ako sme ju opísali v časti 1.3.6 budeme uvádzať pri iných kritériách až v ďalšej sekcii tejto práce.

Pre tento experiment sme si zvolili časť voľne dostupnej databázy *Caltech Faces*. Pôvodná databáza je dostupná na [22]. Naša uvažovaná vzorka obsahuje 102 fotografií, na ktorých sa nachádza 25 rôznych osôb. Ako príklad uvádzame 2 fotografie z tejto databázy na obrázku 2.1.

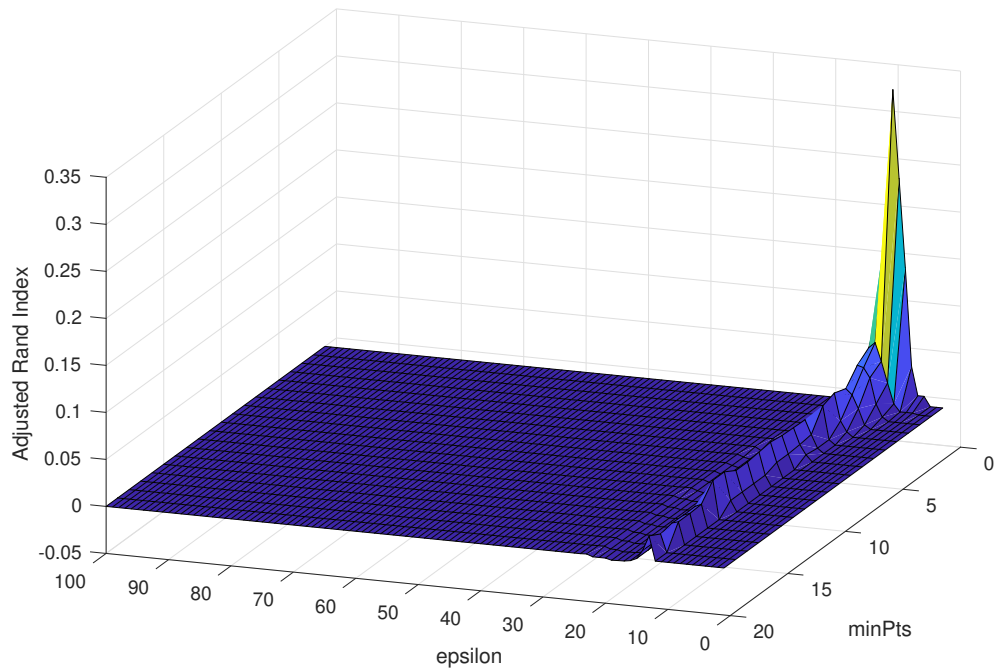


Obr. 2.1: 2 fotografie z databázy Caltech Faces

2.1.1 Parametre pre metódu DBSCAN

Najskôr začneme s algoritmom DBSCAN. Po nájdení samotných tvárí vo fotografiách pomocou metódy Viola-Jones, sme manuálne zostrojili vektor indexov priradzujúci osobám správne zhluky. Nechali sme vypočítať vektor indexov pre DBSCAN pre všetky kombinácie prirodzených čísel $\varepsilon \in [1; 100]$ a $minPts \in [1; 20]$. Výsledky Adjusted Rand Indexu pre redukciiu dimenzie pomocou PCA a 50-tich vlastných čísel (pri 50 prvýkrát prekročíme podiel vysvetlenej variancie cez 95%) uvádzame na obrázku 2.2.

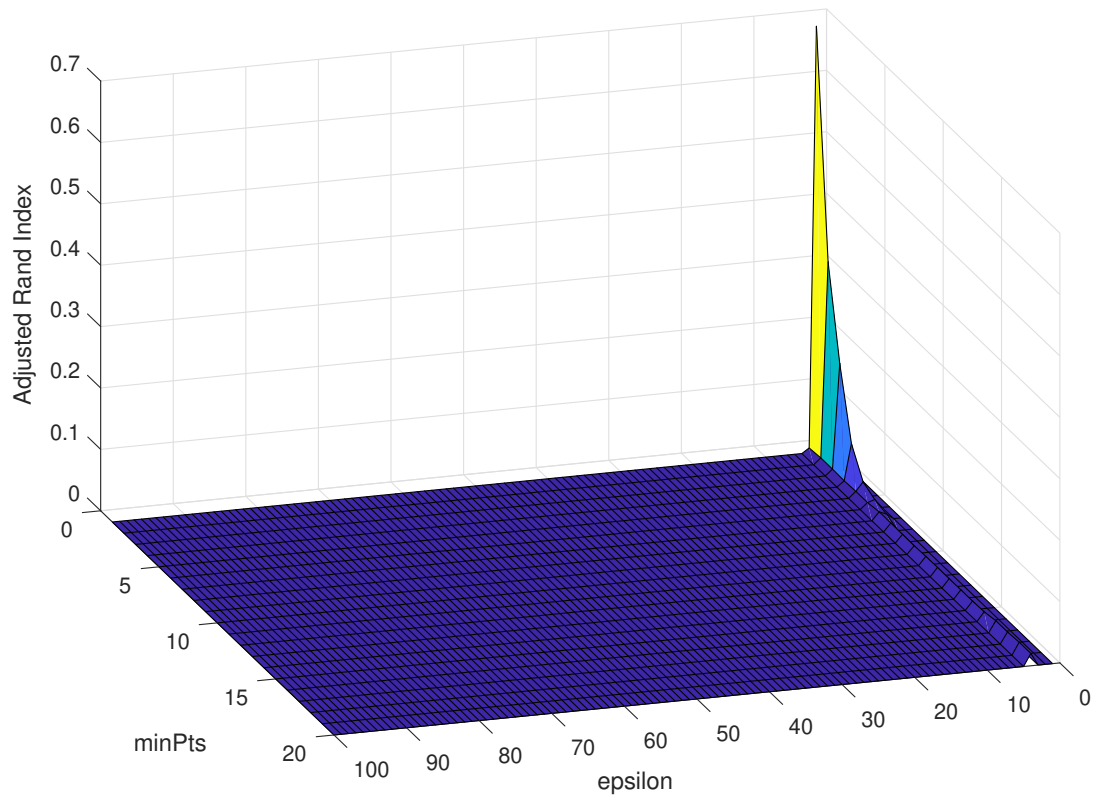
Dobré je tiež podotknúť, že v celej kapitole budeme od tohto momentu pracovať s desatinnou bodkou.



Obr. 2.2: 3-rozmerný graf závislosti ARI od parametrov ε a $minPts$ pre DBSCAN a PCA

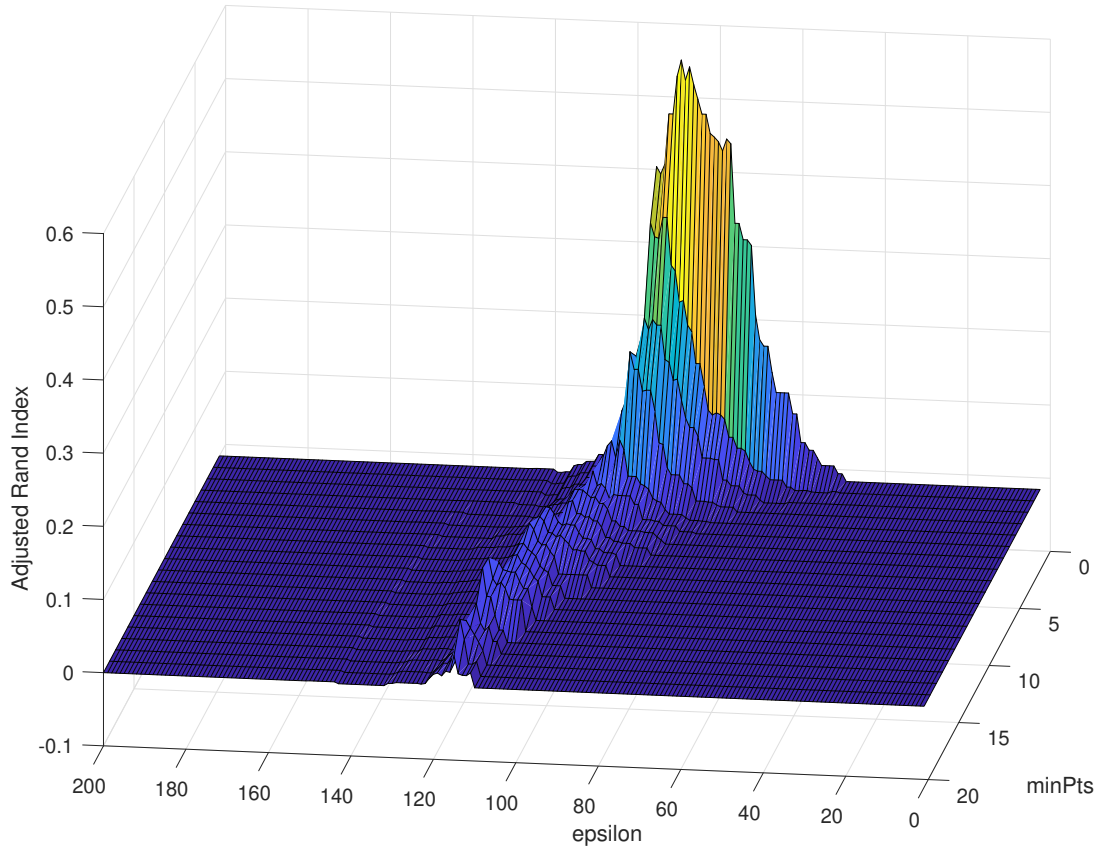
Maximum sa v tomto prípade nadobúda pre $\varepsilon = 9$ a $minPts = 1$, a je rovné **ARI = 0.3332**. Je však zjavné, že metóda dosahuje postačujúce výsledky len na veľmi malej sade parametrov.

V prípade histogramov orientovaných gradientov je voľba parametrov ešte zložitejšia, no redukcia dimenzie sa zdá byť oveľa úspešnejšia. Maximum sa nadobúda pre $\varepsilon = 3$ a $minPts = 1$, a je rovné **ARI = 0.6936**. Výsledky uvádzame na obrázku 2.3.



Obr. 2.3: 3-rozmerný graf závislosti ARI od parametrov ε a $minPts$
pre DBSCAN a HOG

Pri redukcii dimenzie pomocou hlbkej neurónovej siete Alexnet a VGG16 sme sa rozhodli vypočítať ARI pre ε z väčšieho intervalu, a to konkrétne $\varepsilon \in [1; 200]$. Maximum pri sieti Alexnet sa nadobúda pre $\varepsilon = 88$ a $minPts = 1$, a je rovné **ARI = 0.5667** a výsledky uvádzame na obrázku 2.4.

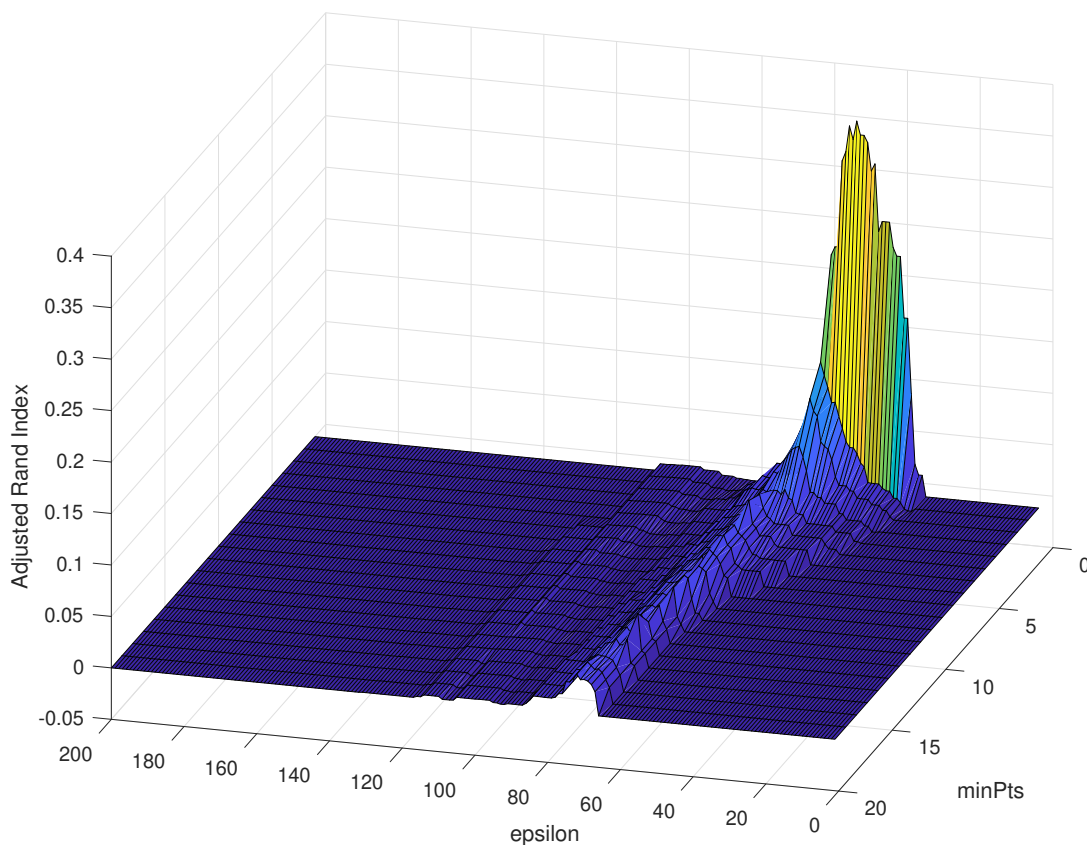


Obr. 2.4: 3-rozmerný graf závislosti ARI od parametrov ε a $minPts$ pre DBSCAN a Alexnet

Maximum pri sieti VGG16 sa nadobúda pre $\varepsilon = 51$ a $minPts = 1$, a je rovné **ARI = 0.3585** a výsledky uvádzame na obrázku 2.5. Ucelené výsledky pre všetky typy tvorby vektorov črt uvádzame v prehľadnej tabuľke 2.1.

Črty	max ARI	ε	minPts
PCA	0.3332	9	1
HOG	0.6936	3	1
Alexnet	0.5667	88	1
VGG16	0.3585	51	1

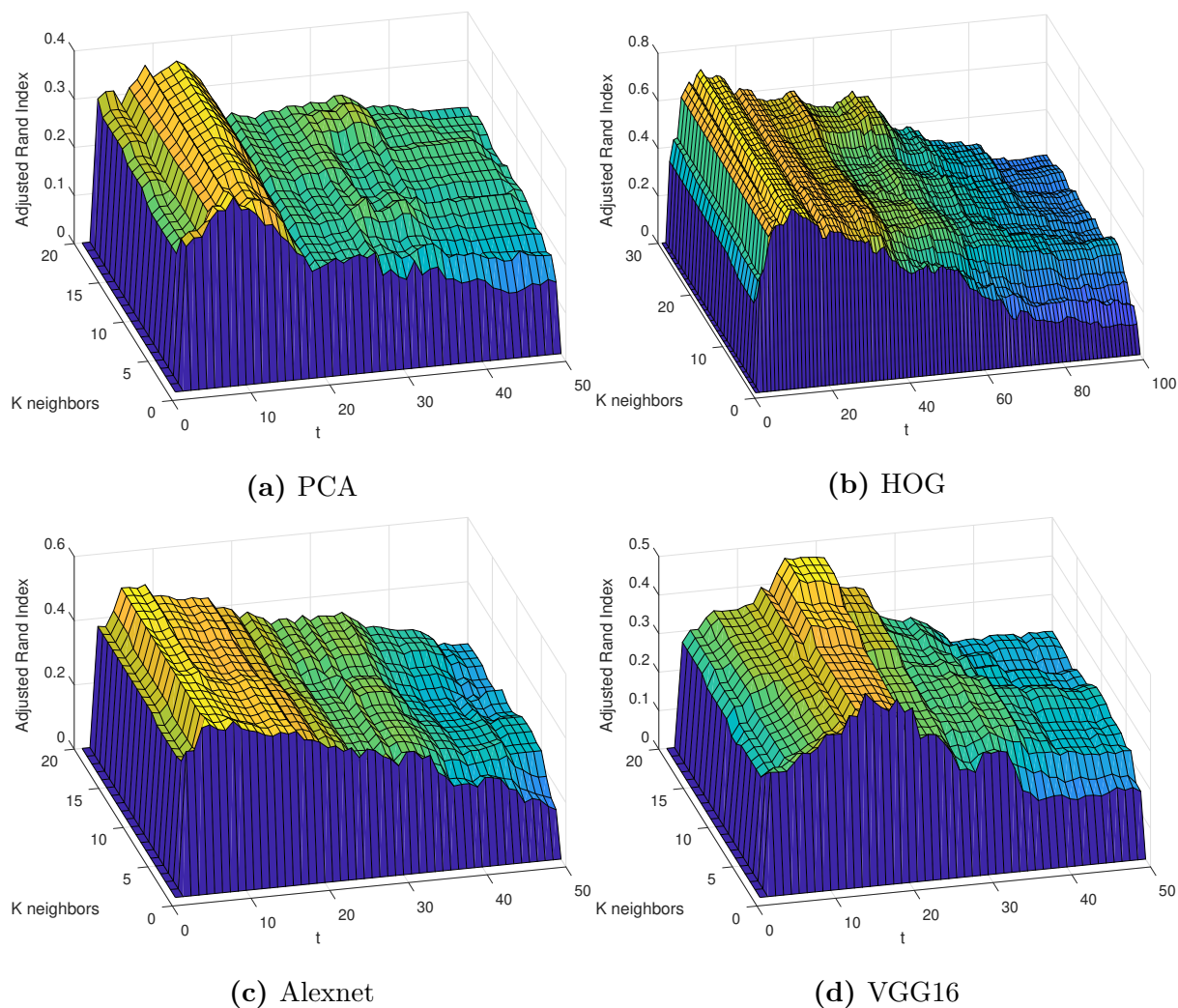
Tabuľka 2.1: Optimálna voľba parametrov a prislúchajúca hodnota ARI pre metódu DBSCAN



Obr. 2.5: 3-rozmerný graf závislosti ARI od parametrov ϵ a $minPts$ pre DBSCAN a VGG16

2.1.2 Parametre pre metódu zhlukovania založeného na poradí

Pri voľbe parametrov metódy zhlukovania založeného na poradí sme v prípade voľby vektorov črt pomocou metódy orientovaných gradientov volili najskôr $t \in [1; 100]$ a $K \in [1; 30]$, no tento výpočet trval popri iných procesoch spustených na počítači viac ako 7 hodín. V ostatných voľbách vektorov črt sme sa preto rozhodli zmenšiť skúmanú oblasť na $t \in [1; 50]$ a $K \in [1; 20]$. Výsledky uvádzame na obrázku 2.6.



Obr. 2.6: Hodnota ARI pre rôzne hodnoty parametrov zhlukovania na poradí

Ukazuje sa, že pre rozumné voľby parametrov (nie patologické prípady pri $t = 1$, alebo $K = 1$) je hodnota ARI zjavne lepšia ako pri náhodnom zhlukovaní. Zhlukovanie založené na poradí sa teda ukazuje ako pomerne robustná metóda vzhľadom na voľbu týchto parametrov. Optimálne hodnoty uvádzame v tabuľke 2.2.

Črty	max ARI	t	$\mathcal{K}$
PCA	0.3750	9	2
HOG	0.7348	11	2
Alexnet	0.5046	9	2
VGG16	0.4807	15	2

Tabuľka 2.2: Optimálna voľba parametrov a prislúchajúca hodnota ARI pre metódu zhlukovania na poradi

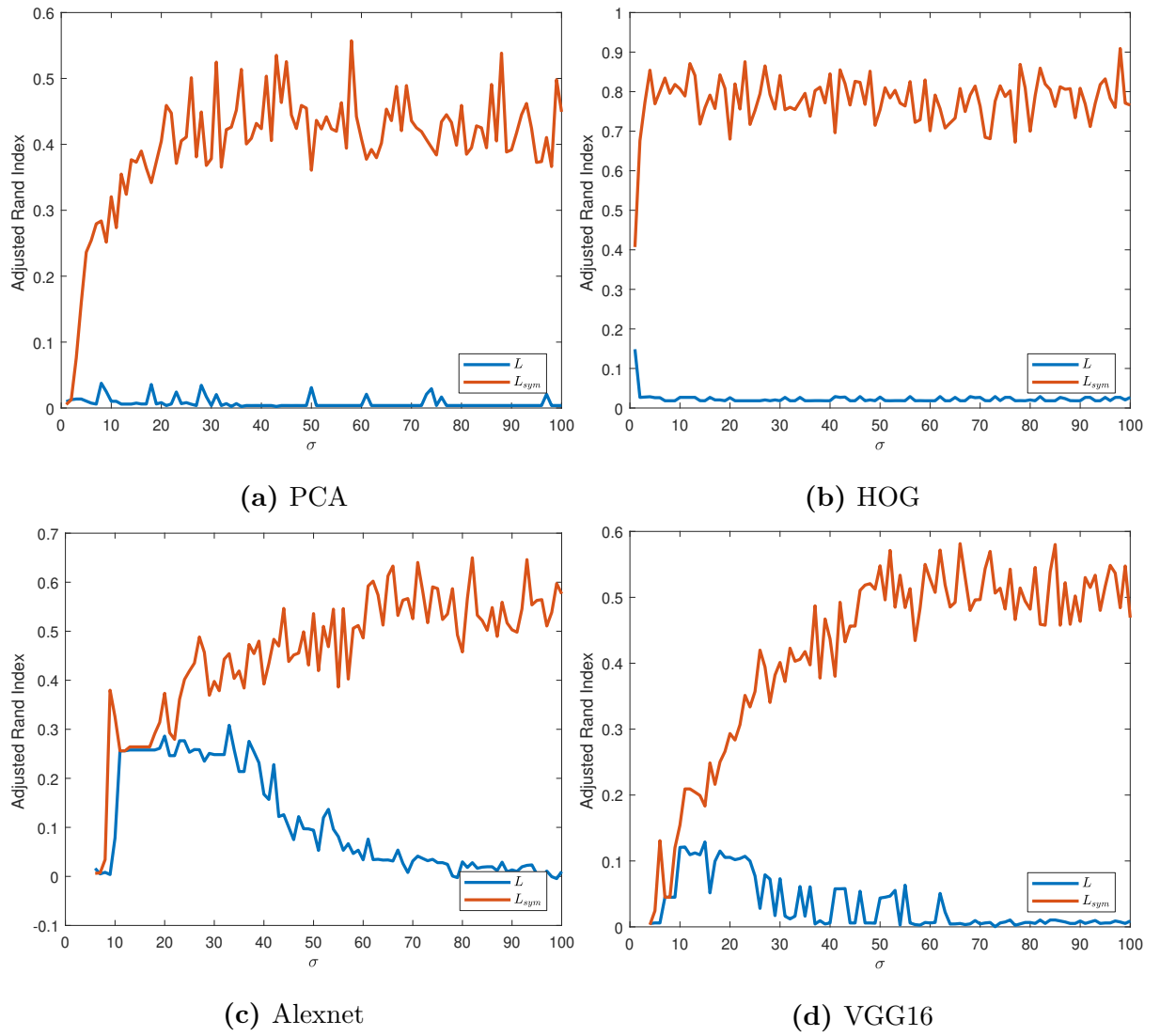
2.1.3 Parametre pre spektrálne zhlukovanie

Pri hľadaní vhodných parametrov sme narazili na závažný problém pri implementácii Laplaceovského grafu L_{rw} definovaného v rovnici 1.10. Takáto matica totiž napriek tomu, že má všetky vlastné čísla nezáporné a reálne, nie je symetrická. V praxi sa teda často stane, že pri numerickom výpočte vlastných čísel narazíme na numerické nepresnosti, ktoré spôsobia komplexné vlastné čísla. Riešenie týchto numerických problémov je netriviálne, a teda sme sa rozhodli, že od ďalšieho používania takto normalizovaného Laplaceovského grafu v tejto práci upustíme.

Na obrázku 2.7 vidíme, že dosahujeme signifikatne lepšie výsledky pri voľbe normalizovaného grafu L_{sym} . Najlepší najdený parameter pre všetky štyri prípady tak uvádzame v tabuľke 2.3, kde vynechávame informáciu o parametroch pre nenormalizovaný Laplaceovský graf L .

Črty	max ARI	σ
PCA	0.5572	58
HOG	0.9091	98
Alexnet	0.6498	82
VGG16	0.5814	66

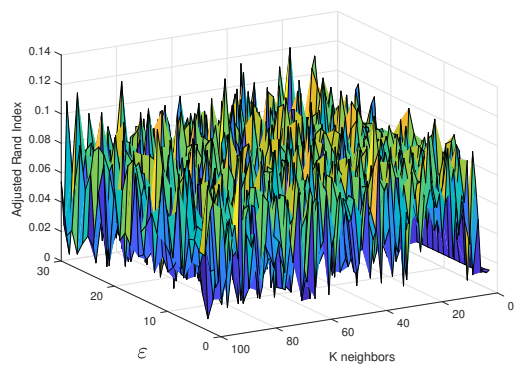
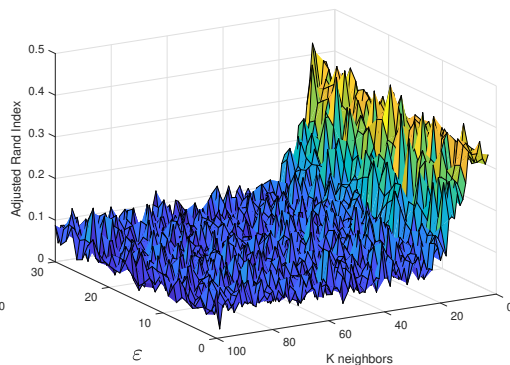
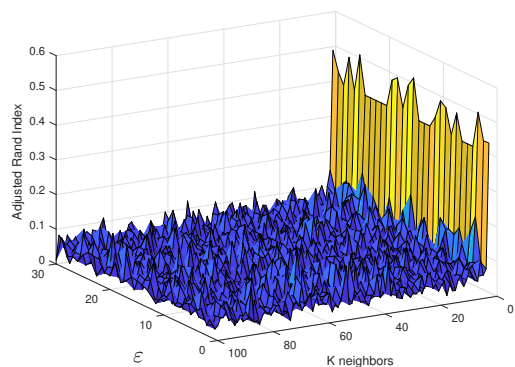
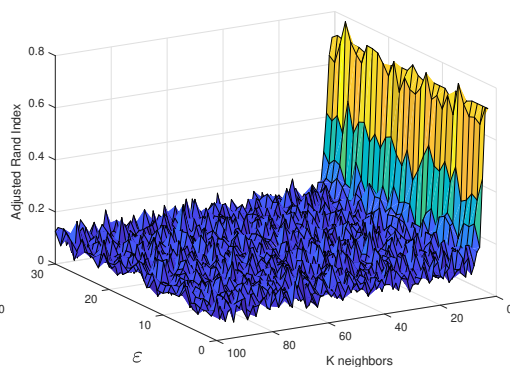
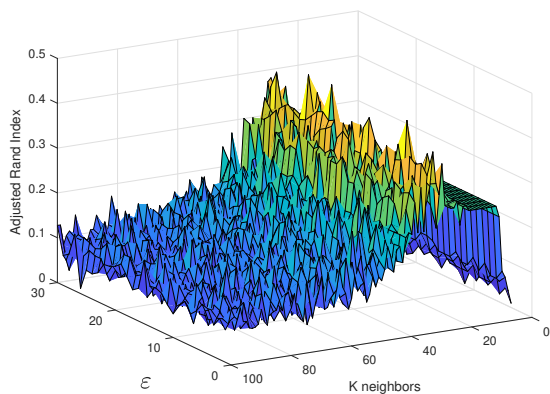
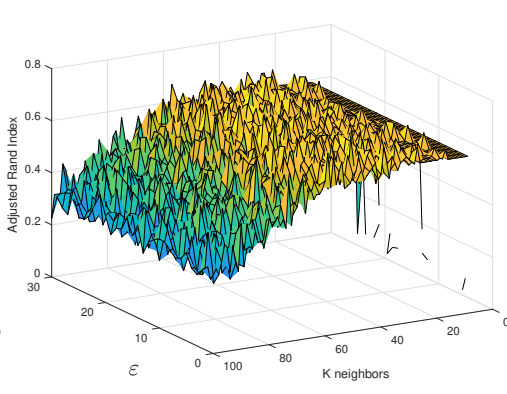
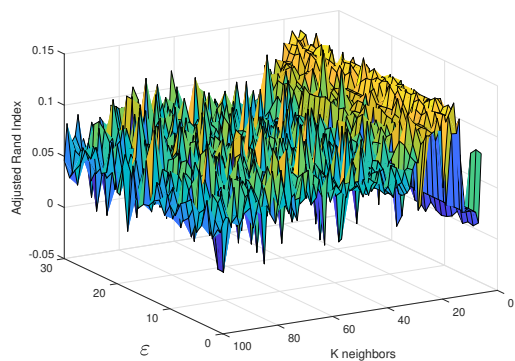
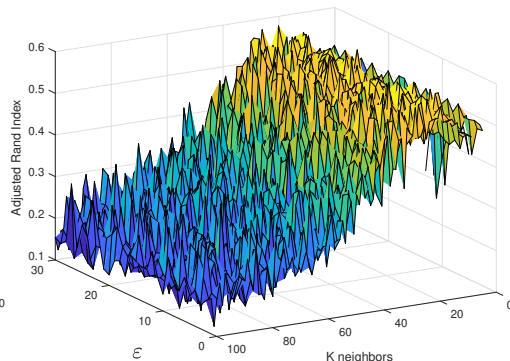
Tabuľka 2.3: Optimálna voľba parametra σ a prislúchajúca hodnota ARI pre spektrálne zhlukovanie, plný graf a L_{sym}



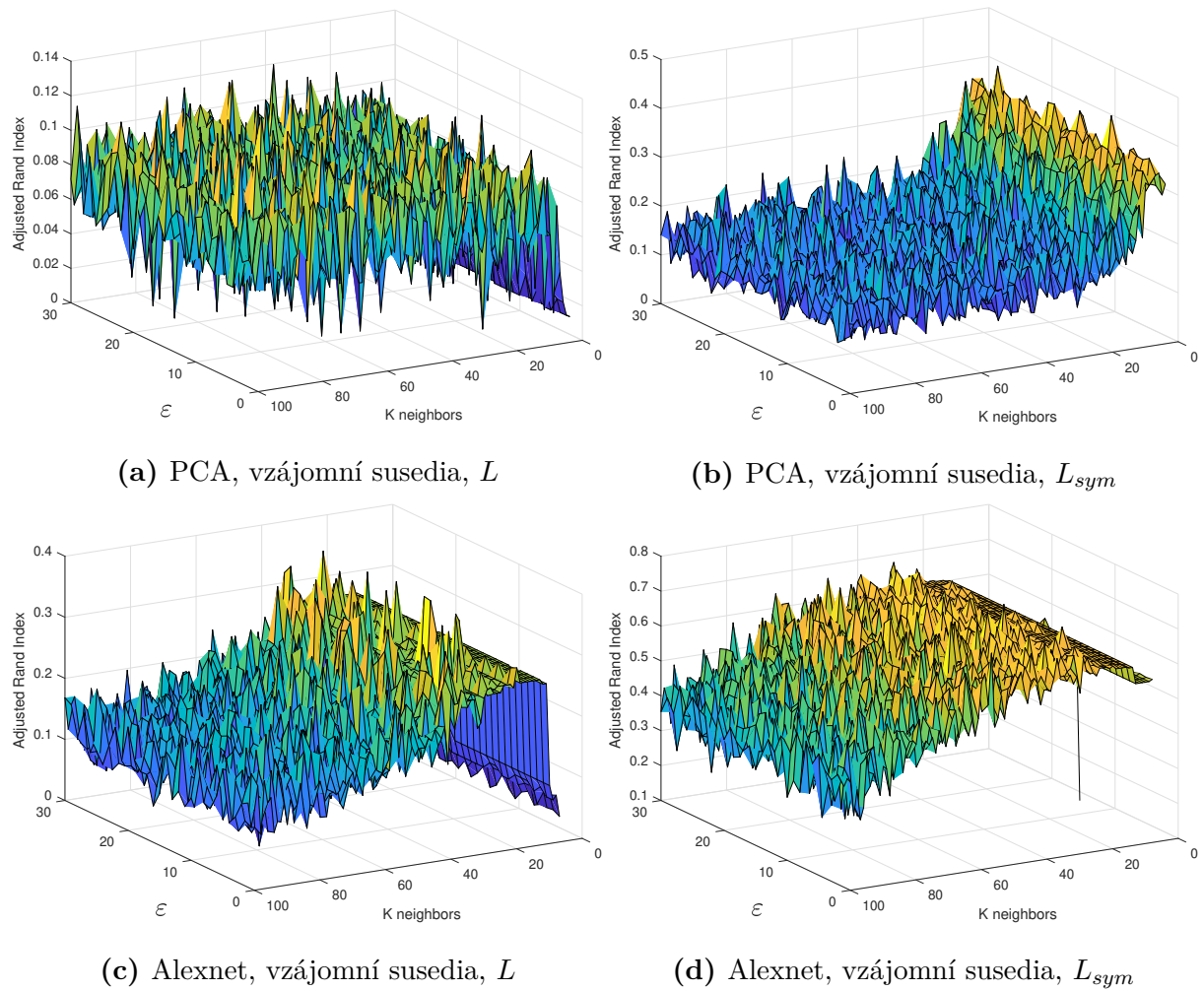
Obr. 2.7: Hodnota ARI pre rôzne hodnoty parametra σ plného grafu podobnosti a spektrálneho zhľukovania

Ďalej preskúmame už len voľbu parametrov pre graf podobnosti $\mathcal{K}$ -najbližších susedov. V praxi sa ukázalo, že graf podobnosti s ε -ovým okolím je pre náš problém nie úplne vhodný a ťažko sa hľadá také ε , aby sme spektrálne zhľukovanie vôbec vedeli spraviť. Rozhodli sme sa preto nepoužívať tento graf.

V prípade $\mathcal{K}$ -najbližších susedov bolo potrebné preskúmať vplyv samotného $\mathcal{K}$, parametra σ , typu susedov (spoloční a vzájomní ako sme to definovali v časti 1.3.3) a voľby Laplaceovského grafu. Výsledky metód uvádzame na obrázkoch 2.8 a 2.9.

(a) PCA, spoloční susedia, L (b) PCA, spoloční susedia, L_{sym} (c) HOG, spoloční susedia, L (d) HOG, spoloční susedia, L_{sym} (e) Alexnet, spoloční susedia, L (f) Alexnet, spoloční susedia, L_{sym} (g) VGG16, spoloční susedia, L (h) VGG16, spoloční susedia, L_{sym}

Obr. 2.8: ARI pre rôzne parametre KNN, spoločných susedov a spektrálneho zhľukovania



Obr. 2.9: ARI pre rôzne parametre KNN, spoločných susedov a spektrálneho zhľukovania

Problém hľadania vlastných čísel je numericky pomerne komplikovaný. V prípade zlej voľby parametrov pri tvorbe grafu dostávame častokrát samotnú maticu blízku singularite (s číslom podmienenosti danej matice blízky nule). Na oboch obrázkoch 2.8 aj 2.9 vidíme zjavné skoky v hodnote ARI. V prípade tvorby vektorov črt pomocou metódy PCA sú tieto výkyvy veľmi nežiadúce, dokonca aj v najlepších oblastiach dosahujeme veľmi nízke kritérium Adjusted Rand Index.

V prípade tvorby vektorov črt pomocou oboch neurónových sietí zase narážame na oblasti, kedy sa kritérium dokonca nedalo vypočítať, pretože sme správne nevedeli vypočítať vlastné čísla, respektíve vlastné vektory danej matice.

Na obrázku 2.9 sme vynechali grafy pre metódu *HOG* a *VGG16*, lebo sme pri rovnakej škále parametrov nedokázali vypočítať ani jedinú hodnotu *ARI*. Tieto metódy teda určite neodporúčame používať pre náš problém.

Najlepšie výsledky pre rôzne kombinácie parametrov uvádzame v tabuľke 2.4. Opäť je vidno, že normalizovaný Laplaceovský graf L_{sym} je lepšou voľbou ako jeho nenormalizovaná verzia. Najlepšie výsledky dostávame pre vektory črt tvorené pomocou histogramu orientovaných gradientov v prípade spoločných susedov a neurónovej siete *Alexnet* pri oboch typoch susedov.

Susedia	Črty	Laplace	max ARI	$\mathcal{K}$ -susedov	σ
spoloční	PCA	L	0.1289	60	20
spoloční	PCA	L_{sym}	0.4275	8	30
spoloční	HOG	L	0.5204	1	15
spoloční	HOG	L_{sym}	0.7934	2	27
spoloční	Alexnet	L	0.4343	30	6
spoloční	Alexnet	L_{sym}	0.7133	44	22
spoloční	VGG16	L	0.1464	25	20
spoloční	VGG16	L_{sym}	0.5954	27	6
vzájomní	PCA	L	0.1354	51	23
vzájomní	PCA	L_{sym}	0.4186	5	8
vzájomní	HOG	L	NA	NA	NA
vzájomní	HOG	L_{sym}	NA	NA	NA
vzájomní	Alexnet	L	0.3653	27	11
vzájomní	Alexnet	L_{sym}	0.7236	32	11
vzájomní	VGG16	L	NA	NA	NA
vzájomní	VGG16	L_{sym}	NA	NA	NA

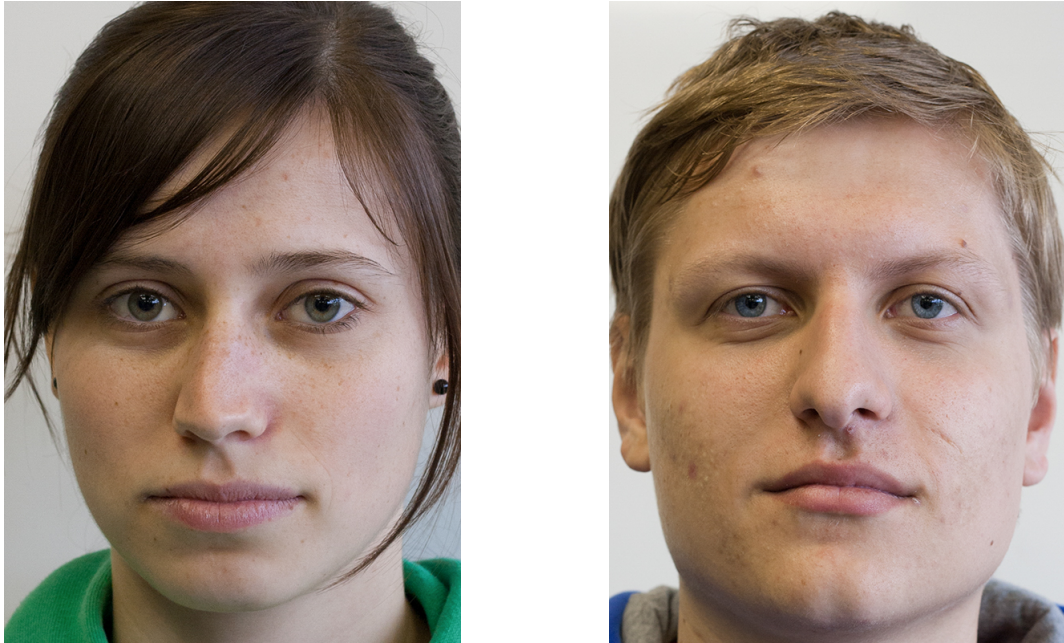
Tabuľka 2.4: Optimálna voľba parametrov a prislúchajúca hodnota ARI pre spektrálne zhľukovanie a graf KNN

2.2 Výsledky pre jednoduchú databázu fotografií

V predchádzajúcej časti sme preskúmali voľbu parametrov niektorých metód. V tejto časti využijeme aj ostatné metódy, ale budeme sa pozerať už len na výsledky pri daných parametroch. Databázu fotografií použijeme rovnakú ako v našej bakalárskej práci [2].

Nachádza sa v nej 61 fotografií 11-tich rôznych ľudí. Pre ilustráciu z nej uvádzame dve fotografie na obrázku 2.10. Tieto tváre sú už manuálne orezané a centrovane podľa očí. Ide teda o oveľa jednoduchšiu úlohu zhlukovania tvárí ako môžeme očakávať v reálnych podmienkach. Od metód vhodných pre použitie v praxi by sme teda čakali len veľmi dobré výsledky.

Pre ďalšie časti práce teda budeme pracovať už len s normalizovaným Laplaceovským grafom L_{sym} .



Obr. 2.10: Ukážkový príklad dvoch ľudských tvárí z jednoduchej databázy

V tejto časti budeme znova vyhodnocovať len kritérium ARI, nakoľko je databáza jednoduchá a uvidíme značné rozdiely v hodnote tohto kritéria. Aj napriek prvej voľbe parametrov v časti 2.1 máme stále obrovské množstvo kombinácií zhlukovania, metód na tvorbu vektorov črt a voľby počtu zhlukov. Chceli by sme teda vyselektovať niektoré najlepšie metódy a použiť ich na zložitejší problém zhlukovania osôb neskôr.

Pre potreby tabuliek s vyhodnotením metód budeme písmenom M označovať manuálnu voľbu počtu zhlukov (ktorá zodpovedá aj skutočnosti, a teda pôjde o najideálnejší prípad). Písmenom CH a S, budeme označovať spôsob voľby zhlukov na základe kritéria *Calinski-Harabasz*, respektíve *Siluetty*. V oboch prípadoch sme museli najskôr definovať hranice pre preskúmané hodnoty počtu zhlukov K , použiť dané zhlukovanie pre všetky hodnoty a následne kritériom zvoliť taký počet zhlukov, aby maximalizoval uvažované kritérium.

Ako najmenšie K sme strategicky zvolili $K = 2$ a najväčšie sme odhadli $K = 20$.

V tabuľke 2.5 uvádzame výsledky pre metódy K-means a K-medoids aj s výpočtovými časmi v sekundách. Je dobré poznamenať, že pri metóde K-means sme využili 300 reštartov.

Metóda	Črty	K-krit.	K	Čas	ARI
k-means	PCA	M	11	1.2095	0.8179
k-means	PCA	CH	18	6.9952	0.783
k-means	PCA	S	20	6.9448	0.7229
k-means	HOG	M	11	2.001	0.9134
k-means	HOG	CH	11	20.271	0.8884
k-means	HOG	S	17	22.267	0.8431
k-means	Alexnet	M	11	4.0276	1
k-means	Alexnet	CH	11	29.015	1
k-means	Alexnet	S	15	29.49	0.9028
k-means	VGG16	M	11	5.9418	0.8905
k-means	VGG16	CH	2	30.321	0.1594
k-means	VGG16	S	11	32.165	0.8905
k-medoids	PCA	M	11	1.0083	0.7702
k-medoids	PCA	CH	20	1.1993	0.7617
k-medoids	PCA	S	19	1.227	0.8137
k-medoids	HOG	M	11	0.98531	0.9134
k-medoids	HOG	CH	11	1.4275	0.9134
k-medoids	HOG	S	17	1.9128	0.8574
k-medoids	Alexnet	M	11	2.7177	1
k-medoids	Alexnet	CH	11	3.2623	1
k-medoids	Alexnet	S	13	4.0474	0.9285
k-medoids	VGG16	M	11	4.4506	0.8905
k-medoids	VGG16	CH	2	5.0248	0.1594
k-medoids	VGG16	S	16	5.9672	0.8895

Tabuľka 2.5: Hodnoty parametrov metód K-means a K-medoids a výsledná hodnota ARI

Vidíme, že vektory črt vytvárané pomocou hlbkej neurónovej siete Alexnet dosahovali dokonalé výsledky, ak sme uvažovali správny počet zhlukov pri použití oboch zhlukovacích algoritmov. Čo je veľmi prekvapivé, je naopak sieť VGG16, ktorá v prípade využitia kritéria *Caliński-Harabasz* dosahovala veľmi zlé výsledky a snažila sa všetky osoby charakterizovať len do dvoch zhlukov.

V tabuľke 2.6 uvádzame výsledky pre metódu spektrálneho zhlukovania. Vo všetkých experimentoch sme využívali len normalizovaný Laplaceovský graf L_{sym} a 300 reštartov algoritmu K-means pri zhlukovaní riadkov matice vlastných vektorov $H_{n \times K}$.

Ihneď si môžeme všimnúť, že v niektorých prípadoch metóda nedokázala vypočítať nič. Ide o problém so samotnou maticou podobnosti, kvôli ktorému sme neboli schopní vypočítať zmysluplné vlastné čísla a vektory. Naopak pri využití siete Alexnet dosahujeme, podobne ako v predchádzajúcom prípade, vynikajúce výsledky.

V tabuľke 2.7 uvádzame výsledky pre metódu DBSCAN a zhlukovanie založené na poradí (**r**ank **o**rders **d**istance **b**ased **c**lustering). Tieto výsledky nie sú veľmi dobré, čo nás prekvapuje najmä pri zhlukovaní založenom na poradí. Očakávali sme od tejto metódy lepšie výsledky. V ďalšej časti preto otestujeme túto metódu s pôvodnými parametrami, ktoré zvolili autori v článku [12].

V prípade hierarchického zhlukovania ukončeného horizontálnym pílením dendrogramu pre dosiahnutie *hard* zhlukovania dostávame príliš veľa kombinácií možných parametrov. Výsledky preto uvádzame v tabuľkách 2.8, 2.9 a 2.10. Metóda HOG dosahuje pomerne dobré výsledky, no aj pri tomto zhlukovaní zjavne najlepšie funguje práve Alexnet. Až na prípady, kedy siluety nedokázali určiť správny počet zhlukov dostávame dokonalé zhlukovanie. V prípade siete VGG16 a kritéria *Caliński-Harabasz* dostávame už z predchádzajúcich experimentov známy výsledok, že nie je vhodné využívať tieto metódy spolu, nakoľko vôbec nie sme schopný určiť rozumný počet zhlukov.

Metóda	Črty	K-krit.	K	Graf	Susedia	$\mathcal{K}$	σ	Čas	ARI
spectral	PCA	M	11	FULL	-	-	58	1.2404	0.8776
spectral	PCA	CH	18	FULL	-	-	58	6.9118	0.7563
spectral	PCA	S	15	FULL	-	-	58	6.9612	0.8286
spectral	HOG	M	11	FULL	-	-	98	1.2694	0.8971
spectral	HOG	CH	11	FULL	-	-	98	7.0128	0.8971
spectral	HOG	S	11	FULL	-	-	98	7.4775	0.8971
spectral	Alexnet	M	11	FULL	-	-	82	3.0218	1
spectral	Alexnet	CH	11	FULL	-	-	82	8.6398	1
spectral	Alexnet	S	11	FULL	-	-	82	9.5288	1
spectral	VGG16	M	11	FULL	-	-	66	4.7679	0.8905
spectral	VGG16	CH	2	FULL	-	-	66	10.408	0.1594
spectral	VGG16	S	11	FULL	-	-	66	11.173	0.8905
spectral	PCA	M	11	KNN	spol.	8	30	1.2233	0.5736
spectral	PCA	CH	2	KNN	spol.	8	30	6.869	0.06436
spectral	PCA	S	2	KNN	spol.	8	30	6.9216	0.06436
spectral	HOG	M	11	KNN	spol.	2	27	-	NA
spectral	HOG	CH	-	KNN	spol.	2	27	-	NA
spectral	HOG	S	-	KNN	spol.	2	27	-	NA
spectral	Alexnet	M	11	KNN	spol.	44	22	3.0361	1
spectral	Alexnet	CH	11	KNN	spol.	44	22	8.1133	1
spectral	Alexnet	S	13	KNN	spol.	44	22	8.9958	0.9285
spectral	VGG16	M	11	KNN	spol.	27	6	4.8045	0.8459
spectral	VGG16	CH	12	KNN	spol.	27	6	9.9208	0.9282
spectral	VGG16	S	13	KNN	spol.	27	6	10.81	0.9036
spectral	PCA	M	11	KNN	vzáj.	5	8	-	NA
spectral	PCA	CH	-	KNN	vzáj.	5	8	-	NA
spectral	PCA	S	-	KNN	vzáj.	5	8	-	NA
spectral	Alexnet	M	11	KNN	vzáj.	32	11	3.0325	1
spectral	Alexnet	CH	11	KNN	vzáj.	32	11	8.2945	1
spectral	Alexnet	S	13	KNN	vzáj.	32	11	9.0526	0.9285

Tabuľka 2.6: Hodnoty parametrov spektrálneho zhľukovania a výsledná hodnota ARI

Metóda	Črty	$\mathcal{K}$	t	minPts	ε	Čas	ARI
rodbc	PCA	2	9	-	-	1.4381	0.6512
rodbc	HOG	2	11	-	-	1.6468	0.8527
rodbc	Alexnet	2	9	-	-	3.6259	0.7671
rodbc	VGG16	2	15	-	-	5.3524	0.7791
dbscan	PCA	-	-	1	9	0.91464	0.8237
dbscan	HOG	-	-	1	3	0.94463	0.7671
dbscan	Alexnet	-	-	1	88	2.6921	0.8939
dbscan	VGG16	-	-	1	51	4.4389	0.5531

Tabuľka 2.7: Hodnoty parametrov zhľukovania založeného na poradí a metódy DBSCAN a ich výsledná hodnota ARI

Metóda	Črty	K-krit.	K	Link	Čas	ARI
hierarch.	PCA	M	11	single	1.0474	0.5583
hierarch.	PCA	M	11	average	0.89559	0.7477
hierarch.	PCA	M	11	complete	0.90302	0.7477
hierarch.	PCA	M	11	ward	0.89551	0.8131
hierarch.	PCA	CH	2	single	1.0127	0.06436
hierarch.	PCA	CH	19	average	1.0036	0.8237
hierarch.	PCA	CH	20	complete	0.99164	0.7617
hierarch.	PCA	CH	20	ward	1.0194	0.7508
hierarch.	PCA	S	20	single	1.042	0.8237
hierarch.	PCA	S	20	average	1.0278	0.8087
hierarch.	PCA	S	20	complete	1.0184	0.7617
hierarch.	PCA	S	18	ward	1.0185	0.7934

Tabuľka 2.8: Hodnoty parametrov hierarchického zhľukovania a metódy PCA a výsledná hodnota ARI

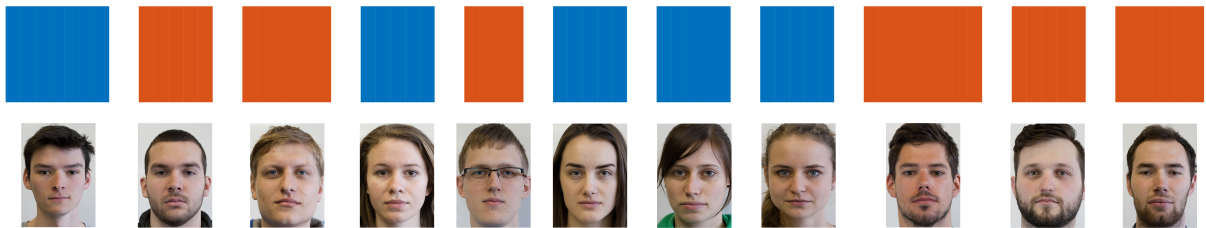
Metóda	Črty	K-krit.	K	Link	Čas	ARI
hierarch.	HOG	M	11	single	0.95012	0.825
hierarch.	HOG	M	11	average	0.958	0.9134
hierarch.	HOG	M	11	complete	0.95495	0.7964
hierarch.	HOG	M	11	ward	0.95799	0.9134
hierarch.	HOG	CH	12	single	1.0744	0.9038
hierarch.	HOG	CH	11	average	1.0935	0.9134
hierarch.	HOG	CH	12	complete	1.0715	0.8735
hierarch.	HOG	CH	11	ward	1.0586	0.9134
hierarch.	HOG	S	20	single	1.5549	0.8335
hierarch.	HOG	S	19	average	1.5623	0.8383
hierarch.	HOG	S	16	complete	1.558	0.8805
hierarch.	HOG	S	15	ward	1.5935	0.885
hierarch.	Alexnet	M	11	single	2.7009	1
hierarch.	Alexnet	M	11	average	2.6852	1
hierarch.	Alexnet	M	11	complete	2.6838	1
hierarch.	Alexnet	M	11	ward	2.6801	1
hierarch.	Alexnet	CH	11	single	2.8374	1
hierarch.	Alexnet	CH	11	average	2.8327	1
hierarch.	Alexnet	CH	11	complete	2.8225	1
hierarch.	Alexnet	CH	11	ward	2.8213	1
hierarch.	Alexnet	S	13	single	3.565	0.9693
hierarch.	Alexnet	S	14	average	3.5556	0.9201
hierarch.	Alexnet	S	13	complete	3.5613	0.9285
hierarch.	Alexnet	S	10	ward	3.5466	0.9132

Tabuľka 2.9: Hodnoty parametrov hierarchického zhukovania, metódy HOG, Alexnet a výsledná hodnota ARI

Metóda	Črty	K-krit.	K	Link	Čas	ARI
hierarch.	VGG16	M	11	single	4.4316	0.7568
hierarch.	VGG16	M	11	average	4.424	0.8905
hierarch.	VGG16	M	11	complete	4.389	0.8582
hierarch.	VGG16	M	11	ward	4.4344	0.8905
hierarch.	VGG16	CH	16	single	4.5248	0.8984
hierarch.	VGG16	CH	2	average	4.5244	0.1594
hierarch.	VGG16	CH	2	complete	4.5419	0.1594
hierarch.	VGG16	CH	2	ward	4.5364	0.1594
hierarch.	VGG16	S	13	single	5.2645	0.8548
hierarch.	VGG16	S	17	average	5.2674	0.8805
hierarch.	VGG16	S	14	complete	5.2743	0.8544
hierarch.	VGG16	S	16	ward	5.297	0.8895

Tabuľka 2.10: Hodnoty parametrov hierarchického zhľukovania a metódy VGG16 a výsledná hodnota ARI

Viacere metódy pri použití kritéria *Calínski-Harabasz* našli práve 2 zhľuky. Zistili sme, že nielenže pri nich bola rovnaká hodnota ARI, ale dokonca samotné zhľukovania boli totožné. Rozhodli sme sa preto graficky pozrieť na tento výsledok. Výsledky zhľukovania uvádzame na obrázku 2.11. Každý samostatný obdĺžnik predstavuje jeden zo skutočných zhľukov (so šírkou podľa počtu fotografií) a farby predstavujú nájdené zhľuky. Vidíme, že metódy nepokazili žiaden zo samostatných zhľukov tým, že by ich rozdelili. Dokonca až na prípad prvého subjektu (v ktorom sa metódy pomýlili), dokázali odhaliť pohlavie jedincov. Okrem prvého subjektu sú teda ženy označené modrou farbou a muži oranžovou.



Obr. 2.11: Hierarchické zhľukovanie databázy pri 2 zhľukoch, VGG16 a average linkage

2.3 Výsledky na zložitejšej databáze aj s EXIF dátami

V tejto časti aplikujeme najslubnejšie metódy na oveľa komplikovanejšiu databázu, získanú z osobných fotografií vyhotovených mobilným telefónom. Fotografie sú vyhotovené v rôznych svetelných podmienkach a na jednej fotografii sa často naraz nachádza viacero subjektov, prípadne žiaden, ako môžeme vidieť na pravej fotografii z obrázku 2.12. Databáza obsahuje 102 fotografií s nešpecifikovaným počtom ľudí na fotografiách, keďže našim cieľom nebolo otestovať samotný algoritmus *Viola-Jones* pre rozpoznávanie tvárí, ale len zhlukovacie metódy. Preto sme vyhotovili len správne odpovede pre tváre, ktoré daný algoritmus našiel. Našli sme celkovo 112 tvárí, z ktorých 7 bolo falošných (tie v skutočnosti neboli ľudskými tvármi, ale napríklad jedna z nich bola tvár detskej hračky psa zobrazenej na obrázku 2.12). Zastúpenie ľudí tiež nie je rovnaké. V nájdených tvárach majú dve osoby po 35 tvárí a celkovo je osôb 17. V zásade teda aj so šumom hovoríme o 18-tich skutočných zhlukoch.



Obr. 2.12: 2 fotografie zo súkromnej databázy

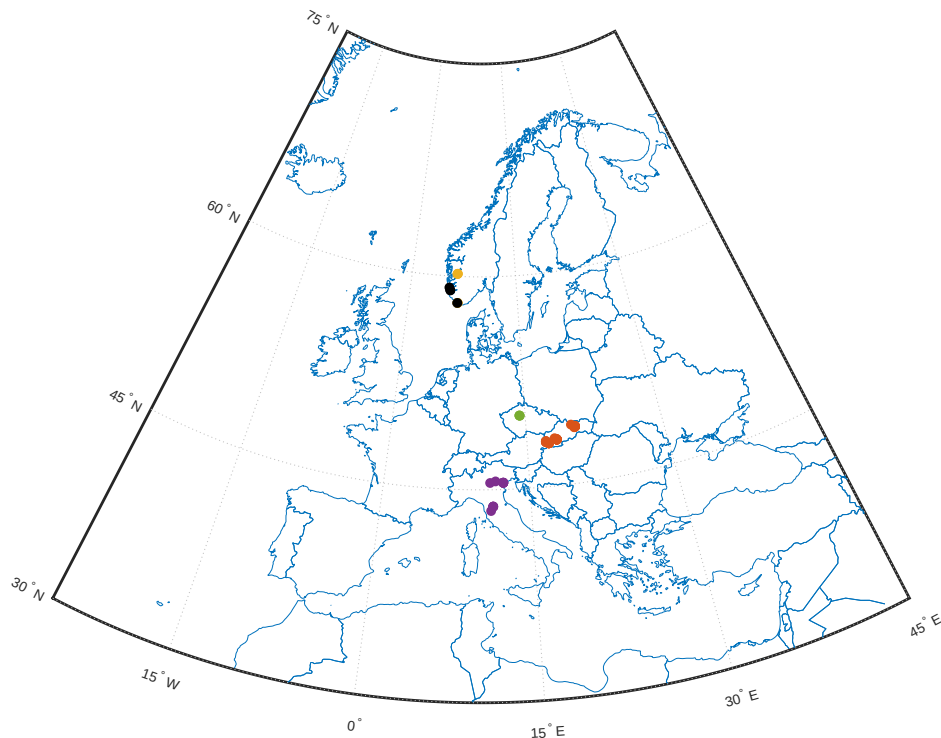
Takáto databáza predstavuje reálne podmienky a je oveľa ťažším testom pre metódy, ktoré sme definovali v prvej kapitole. Z tohto dôvodu je možno rozumnejšie nehodnotiť metódy pomocou kritéria *Adjusted Rand Index*, ale skôr pomocou kritérií *Precision*, *Recall* a *Compression Ratio* definované v časti 1.4.2. Je dobré uvedomiť si, že počítanie kritéria *Recall* nemá zmysel v prípadoch kedy zhlukovací algoritmus nepoužíva samostatný zhluk šum, nakoľko bude vtedy určite rovné 1. Každú z metód by sme teoreticky vedeli dodefi-

nováť podobným spôsobom ako zhlukovanie založené na poradí priraduje do zhluku šum všetky zhluky tvorené len jedným objektom, no rozhodli sme sa toto nespraviť.

V tejto databáze tiež prvýkrát využívame aj Exif dáta ukryté v samotných fotografiách. GPS súradnice sú však len prístupné na 68 fotografiách, zato údaje o čase vytvorenia fotografie sú prístupné na všetkých.

Pozrime sa najskôr na GPS. Nie je úplne jednoznačné, aké zhlukovanie takýchto súradníc je vhodné zvoliť. Preto je dobré najskôr vykresliť samotné dáta. Tradičným spôsobom zhlukovania takýchto dát môžu byť metódy *K*-medoids alebo hierarchické zhlukovanie, no pre korektnosť v nich použijeme meranie vzdialeností pomocou haversinovej rovnice popísanej v časti 1.3.8. Adaptácia týchto metód je pomerne jednoduchá, no zachováva si problém s voľbou počtu zhlukov *K*.

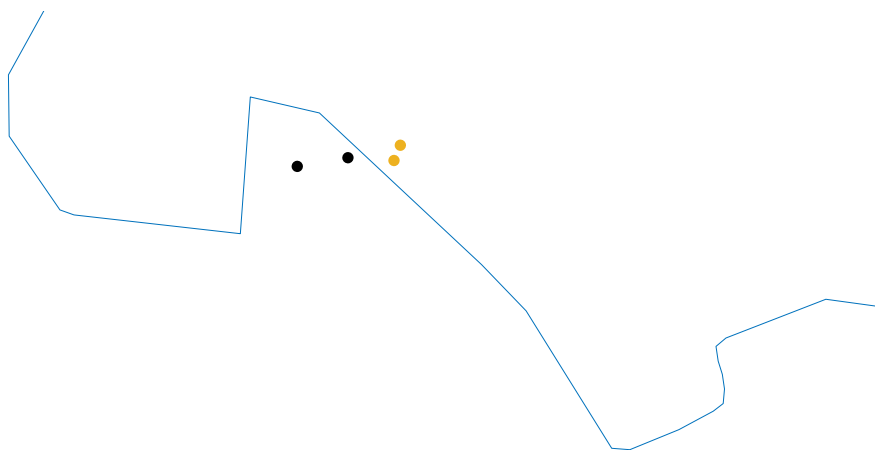
Iným spôsobom nenarážajúcim na voľbu počtu zhlukov je krajinné zhlukovanie. Súradnice vtedy porovnávame s mapou, na ktorej sú zakreslené hranice jednotlivých štátov a jednoduchou funkciou len overíme, či sa so súradnicami nachádzame vnútri nejakého konkrétného polygónu. Spomenuté krajinné zhlukovanie uvádzame na obrázku 2.13.



Obr. 2.13: Krajinné zhluky GPS súradníc

Pri pohľade na obrázok 2.13 však narážame na zaujímavý problém. V Nórsku vidíme

dáta zobrazené dvomi rôznymi farbami. Tento problém lepšie ilustrujeme na priblíženej mape na obrázku 2.14. Dve dáta sa totiž nachádzajú mimo zaznačených hraníc krajiny. Tieto dve fotografie boli konkrétne zhotovené na pobreží. Najľahšie riešenie bude v tomto prípade nájsť k nim najbližší zaradený bod a priradiť ich do rovnakého zhuku ako tento bod. Vo všeobecnosti by sme však mali zaviesť nejaký parameter, ktorý obmedzí priraďovanie v takých prípadoch, kedy na blízkom okolí nebudú žiadne zaradené body. Týmto predídeme zaraďovaniu fotografií do nesprávnych krajín v patologických prípadoch, kedy všetky fotografie budú vytvorené blízko hraníc krajiny.



Obr. 2.14: Problém pri krajinných zhlukoch

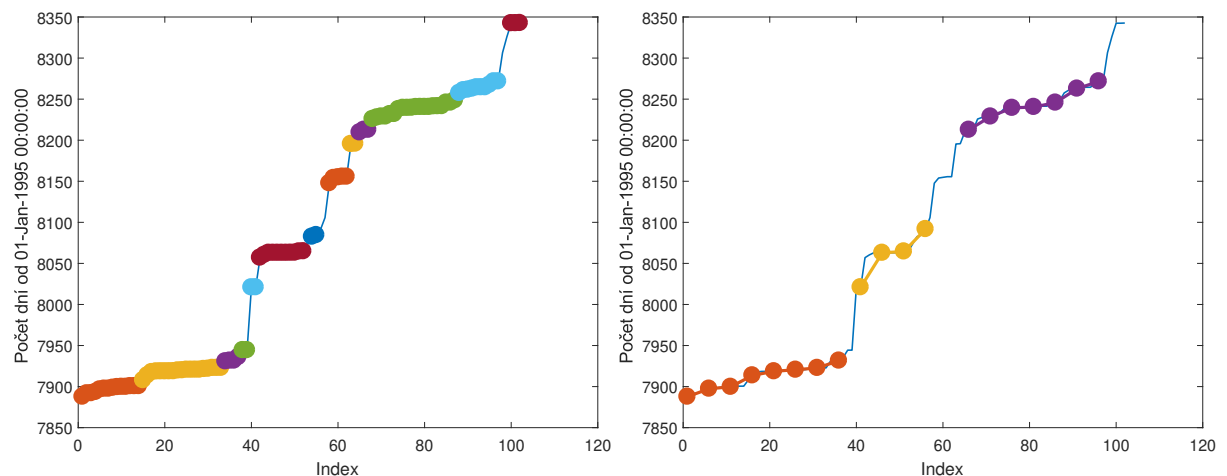
Ak sa rozhodneme problém hraničných bodov riešiť zhlukom šum, ktorý musíme používať aj v prípade chýbajúcich GPS súradníc, narážame na ďalší problém v prípade následného naivného využitia zhlukovania ako samotných črt. V časti 1.5.1 sme opísali naivný prístup, kde zhlukovanie využilo výsledky zhlukovania pomocou indikátorových premenných. Je však dobré poznamenať, že nie je vhodné enkódovať zhluk šum ako vektor črt samých núl. Dve šumové dáta by totiž mali v každej metrike vzdialenosť 0 (za predpokladu vektorov črt tvorených len enkódovaním kategorickej premennej daného zhlukovania). Taktiež z rovnakého dôvodu nie je vhodné vytvoriť samostatný indikátor pre zhluk šum. Riešenie navrhujeme jednoduché, no pamäťovo a výpočtovo náročné. Pre potreby enkódovania navrhujeme každému dátumu zo zhluk šum priradiť nový, samostatný zhluk, a teda týmto spôsobom značne rozšíriť vektor črt.

V časti 1.2.1 sme spomínali ako budeme postupovať pri extrahovaní časových údajov z fotografií, no nespomenuli sme konkrétny spôsob zhlukovania. V tomto prípade nie je na

mieste používať klasické zhukovacie algoritmy, keďže ide len o jednorozmerné dáta, a teda len akési hľadanie intervalov. Ako sme už spomenuli, dáta teda extrahujeme a prevedieme na počet dní od 1. Januára 1995 a času 0:00:00. Následne ich zoradíme o najmenšieho po najväčšie.

Pre možné veľké množstvo fotografií sme sa rozhodli zvoliť akési delenie dát na skupiny a pri následnom zhukovaní sa už len pýtame, či máme spojiť dve susedné skupiny do jedného zhuku. Voľba delenia samozrejme závisí od veľkosti databázy a môžeme zvoliť skupiny aj ako samostatné fotografie. Priradovanie skupín do zhukov robíme na základe sklonu lineárnej interpolácie dvoch susedných skupín. Z oboch skupín vždy berieme najmenší časový údaj a výsledný sklon porovnávame s vopred zvolením prahom. Takýmto spôsobom vieme interpretovať obe veličiny ako **prah $\times$ delenie = najväčší rozdiel v dňoch medzi najmenšími hodnotami z dvoch susedných skupín, ktorý ešte považujeme za rovnaký zhuk**.

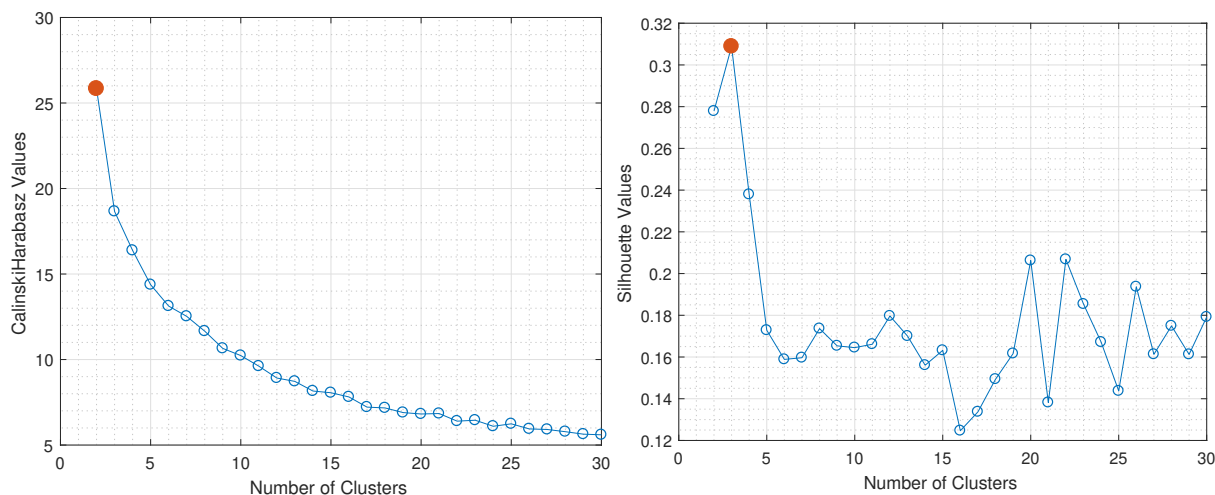
Na obrázku 2.15 uvádzame výsledky zhukovania na našej databáze pri delení 1 a 5 fotografií. Takéto zhukovanie zjavne nemá správnu odpoveď a je len na nás ako definujeme všetky parametre potrebné pre tvorbu zhukov. Pre potreby tejto databázy sme využili delenie na jednočlenné skupiny a prah rovný siedmym dňom. Graficky to uvádzame na ľavom grafe obrázku 2.15.



Obr. 2.15: Zhukovanie času pre rôzne hodnoty veľkosti skupín

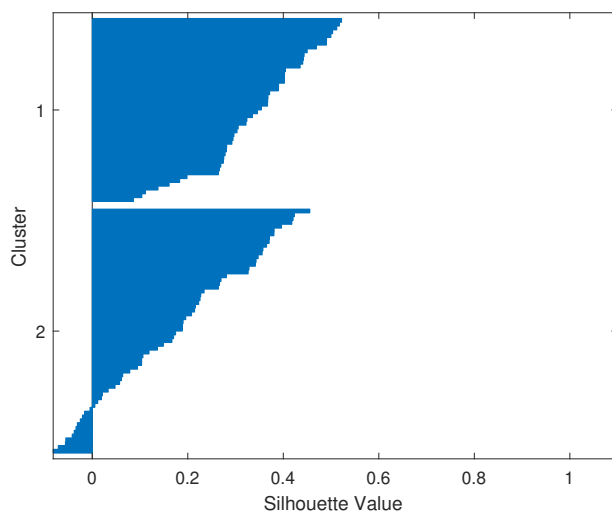
Skôr než sa pozrieme na výsledky zhukovacích algoritmov, zamerajme sa najprv na voľbu počtu zhukov tváří konkrétne pre metódu *K*-means. Na obrázku 2.16 uvádzame výsledky kritérií *Siluet* a *Calínski-Harabasz* pre rôzne hodnoty počtu zhukov *K*. Voľba

optimálneho K z hľadiska oboch parametrov je jednoduchá, no vykreslenie všetkých skúmaných hodnôt dáva zaujímavú informáciu. Ako sme už spomenuli, v databáze sme našli 17 rôznych ľudí a tiež skupinu šum. V ideálnom prípade by sme teda očakávali 18 zhlukov, prípadne 24, ak by sme chceli mať šum v samostatných zhlukoch. Bohužiaľ však kritériá naznačujú optimálny počet zhlukov veľmi jednoznačne ako dva, respektíve tri.



Obr. 2.16: Kritériá pre voľbu počtu zhlukov pre metódu K-means na tvárach charakterizovaných pomocou siete Alexnet

Pozrime sa ešte na samotné siluety v prípade dvoch zhlukov pre metódu K -means. Na obrázku 2.17 vidíme samotné hodnoty siluet jednotlivých tvárí. Zdá sa, že veľké množstvo objektov je zaradených správne v zmysle siluet, no istá časť je blízko hranice a niektoré sú zjavne zle zaradené a nadobúdajú záporné hodnoty.



Obr. 2.17: Graf siluet pre K -means zhlukovanie s 2 zhlukmi

Venujme sa najskôr na výsledkom metódy K -means, pri použití siete *Alexnet*. V tabuľke 2.11 uvádzame hodnoty kritérií *ARI*, *Precision* a *Compression Ratio*. Toto je tiež jediná tabuľka, v ktorej uvedieme hodnotu *Compression Ratio*, nakoľko toto kritérium síce hovorí o priemernej veľkosti zhlukov, ale z jeho samotnej hodnoty vieme povedať len veľmi málo o tom, ako dobre si počínala daná metóda. Závisí totiž nielen od zvolenej metódy, ale aj od samotnej databázy tváří.

Exif	K-krit.	K	Čas	ARI	Precision	CR
-	M	18	0.6573	0.1119	0.6071	6.222
Krajiny	M	18	0.657	0.1368	0.6161	6.222
Čas	M	18	0.6506	0.1139	0.6161	6.222
-	CH	2	20.06	0.1719	0.5268	56
Krajiny	CH	2	19.87	0.1586	0.5179	56
Čas	CH	2	19.15	0.1586	0.5179	56
-	S	3	24.05	0.1719	0.5268	56
Krajiny	S	3	24.76	0.1474	0.5179	37.33
Čas	S	3	25.79	0.1302	0.5089	37.33

Tabuľka 2.11: Zhlukovanie pomocou K -means a Alexnet

Prvým a hlavným problémom je, ako sme už spomínali na obrázku 2.16, voľba počtu zhlukov. Bez manuálnej voľby K sa nevieme dobre priblížiť skutočným zhlukom, čo sa prejavuje aj na veľmi nízkej hodnote kritéria *Precision*. Ako neskôr uvidíme, hodnota *ARI* je pre K -means značne vyššia ako pri iných metódach. Toto je však zavádzajúce, pretože v reálnych podmienkach by sme chceli práve vysokú hodnotu kritéria *Precision*, aby používateľ nejakej konkrétnej aplikácie nemusel manuálne opravovať chyby samotnej metódy. Preňho je oveľa jednoduchšie zjednotiť určité zhluky ako opraviť zle zaradených ľudí. Tohto pravidla sa držia aj súkromné spoločnosti poskytujúce služby zhlukovania fotografií.

Medzi využitím Exif dát prístupných vo fotografiách v podobe krajinných zhlukov, času, prípadne žiadnych však nevidíme žiaden významný rozdiel.

V tabuľke 2.12 uvádzame výsledky pre metódu založenú na poradí. Malým n budeme

označovať naivnú verziu a β budeme označovať adaptovanú verziu tohto algoritmu. Hodnotu parametra β sme zvolili 0.3 a hodnoty parametrov $\mathcal{K}$ a t sme využili najskôr optimálne v zmysle časti 2.1 podľa tabuľky 2.2 a potom rovnaké ako samotní autori metódy v článku [12].

Exif	Črty	$\mathcal{K}$	t	Čas	ARI	P	R
-	HOG	2	11	4.157	-0.02275	0.806	0.6286
Krajiny _n	HOG	2	11	6.729	-0.01614	0.7945	0.6952
Krajiny _{β}	HOG	2	11	4.269	-0.01736	0.7941	0.6381
Čas _n	HOG	2	11	4.198	-0.03368	0.8286	0.6667
Čas _{β}	HOG	2	11	4.184	-0.01736	0.7941	0.6381
-	HOG	9	14	8.617	-0.03927	0.8169	0.6667
Krajiny _n	HOG	9	14	6.638	-0.03215	0.803	0.6286
Krajiny _{β}	HOG	9	14	7.972	-0.02624	0.8219	0.6857
Čas _n	HOG	9	14	5.873	-0.03251	0.8169	0.6762
Čas _{β}	HOG	9	14	7.362	-0.031	0.8194	0.6762
-	Alexnet	2	9	7.202	0.01999	0.7703	0.6857
Krajiny _n	Alexnet	2	9	7.903	0.0215	0.7432	0.6857
Krajiny _{β}	Alexnet	2	9	8.125	0.03425	0.7632	0.7048
Čas _n	Alexnet	2	9	7.777	0.02804	0.75	0.6667
Čas _{β}	Alexnet	2	9	7.797	0.03425	0.7632	0.7048
-	Alexnet	9	14	8.476	0.001434	0.7568	0.6667
Krajiny _n	Alexnet	9	14	10.53	-0.001797	0.7432	0.6857
Krajiny _{β}	Alexnet	9	14	11.83	0.02979	0.725	0.7238
Čas _n	Alexnet	9	14	10.25	0.007814	0.7536	0.6381
Čas _{β}	Alexnet	9	14	10.95	0.03751	0.716	0.7333

Tabuľka 2.12: Klasické, naivné a adaptované zhľukovanie založené na poradí pre $\beta = 0.5$

Hodnoty kritéria *ARI* sú tentokrát negatívne ovplyvnené vytvorením zhľuku šum, ktorý však toto kritérium považuje za rovnocenný zhľuk oproti ostatným zhľukom. Čo nás však teší, je značne vyššia hodnota *Precision* oproti metóde *K-means*. Nie je však taká vysoká

ako dosiahli autori v článku [12] (tam dosahovala hodnoty blízke 0.98). Toto je zjavne ovplyvnené napríklad voľbou parametra **2** pri normovanej vzdialenosti zhlukov, ktorý sme popísali v časti 1.3.6. Pri pôvodnej voľbe tohto parametra však potom dostávame 110 rôznych zhlukov (v prípade klasického zhlukovania bez exif dát) a hodnotu kritéria *Precision* rovnú 1. Pripomeňme si však, že celkový počet tvárí bol 112. Takéto zhlukovanie takmer ničomu nepomôže. Sme síce veľmi obozretný pri zaradovaní tvárí do zhlukov, no sme až tak obozretný, že nezlúčime skoro žiadne objekty. Autori však dostali aj pomerne dobré hodnoty nami nedefinovaného kritéria *Nearest Mutual Information*. Nekonzistentnosť našich a ich výsledkov bude teda zrejmé spôsobená práve tvorbou vektorov črt a našou voľbou komplikovanejšej databázy. Autori použili neurónovú sieť vytvorenú práve pre účel charakterizácie ľudských tvárí. Táto sieť však nie je voľne dostupná a nemôžeme teda priamo porovnať naše výsledky.

Čo si však môžeme všimnúť je, že naša adaptovaná metóda dosahuje často o niečo lepšie výsledky v kritériu *ARI* a naopak občas o niečo horšie pre kritérium *Precision*. Niekedy je však aj *Precision* na porovnateľnej, alebo lepšej úrovni. Pri použití siete Alexnet pre tvorbu vektorov črt je výsledok *ARI* v prospech adaptovanej metódy najvýraznejší. Až na niektoré ojedinelé prípady sme sa v tomto kritériu však zlepšili oboma metódami využívajúcimi dodatočné informácie. Klasická metóda teda v tomto zaostáva, no drží si stabilne dobré výsledky práve v *Precision*.

V tabuľke 2.13 uvádzame výsledky pre spektrálne zhlukovanie. Opäť však vidíme, že obe kritériá pre voľbu optimálneho počtu zhlukov zlyhali. Nadôvažok je aj hodnota *Precision* značne nižšia než v prípade predchádzajúcej metódy. Zdá sa teda, že spektrálne zhlukovanie v tom prípade nie je veľmi vhodným kandidátom pre zhlukovanie tvárí z reálnej databázy pri použití zvolených metód na vytváranie vektorov črt (v tomto prípade sme opäť použili zatiaľ našu najosvedčenejšiu metódu *Alexnet*).

Poslednou metódou, ktorú sme overili na tejto databáze bolo hierarchické zhlukovanie. Opäť sme siahli len po črtách získaných zo siete *Alexnet*, pretože tie vykazovali najlepšie výsledky v predchádzajúcich častiach. Výsledky uvádzame v tabuľke 2.14. Je dobré poznamenať, že tentokrát uvádzame výsledky len pre manuálnu voľbu počtu zhlukov, nakoľko obe kritériá neboli schopné nájsť skutočné K . Z uvedených výsledkov vyplýva, že práve single linkage vzdialenosť zhlukov dopadla spomedzi testovaných najhoršie pri kritériu

ARI, ako aj pri kritériu *Precision*.

Exif	K-krit.	K	Graf	Susedia	$\mathcal{K}$	σ	Čas	ARI	Precision
-	M	18	FULL	spol.	44	82	0.5602	0.08991	0.5893
Krajiny	M	18	FULL	spol.	44	82	0.5313	0.07754	0.6161
Čas	M	18	FULL	spol.	44	82	0.5374	0.07563	0.6071
-	CH	2	FULL	spol.	44	82	14.31	0.1727	0.5179
Krajiny	CH	2	FULL	spol.	44	82	14.18	0.158	0.5179
Čas	CH	2	FULL	spol.	44	82	13.92	0.158	0.5179
-	S	2	FULL	spol.	44	82	19.34	0.00674	0.3214
Krajiny	S	2	FULL	spol.	44	82	18.68	0.158	0.5179
Čas	S	2	FULL	spol.	44	82	18.8	0.158	0.5179

Tabuľka 2.13: Výsledky pre spektrálne zhľukovanie na zvolenej databáze

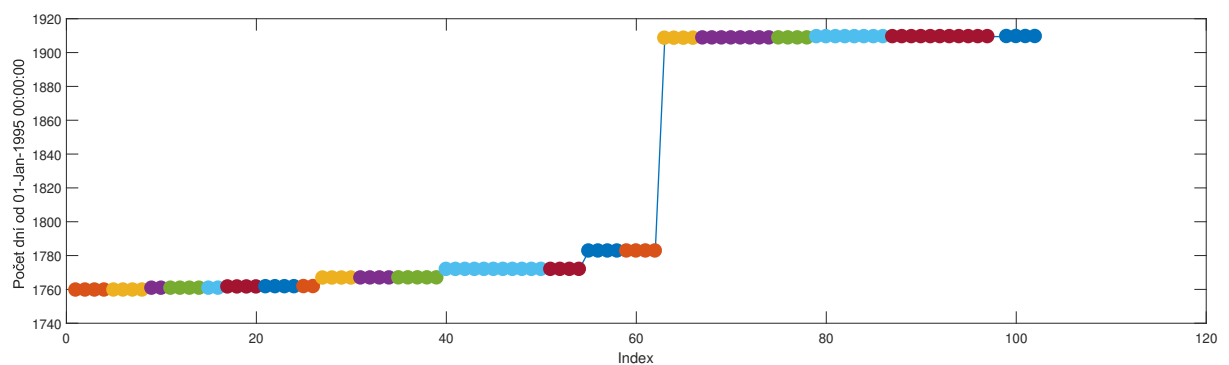
Exif	K-krit.	K	Link.	Čas	ARI	Precision
-	M	18	single	0.01748	0.0303	0.4375
Krajiny	M	18	single	0.02702	0.03507	0.4286
Čas	M	18	single	0.02499	0.03507	0.4286
-	M	18	average	0.01406	0.1444	0.6071
Krajiny	M	18	average	0.02257	0.1774	0.6429
Čas	M	18	average	0.02139	0.1449	0.6071
-	M	18	complete	0.01585	0.1133	0.6339
Krajiny	M	18	complete	0.02545	0.098	0.625
Čas	M	18	complete	0.02246	0.09832	0.625
-	M	18	ward	0.01704	0.1171	0.6161
Krajiny	M	18	ward	0.02546	0.1373	0.625
Čas	M	18	ward	0.02931	0.1246	0.6161

Tabuľka 2.14: Výsledky pre hierarchické zhľukovanie pri použití *Alexnet*

2.4 Adaptované zhlukovanie na poradí

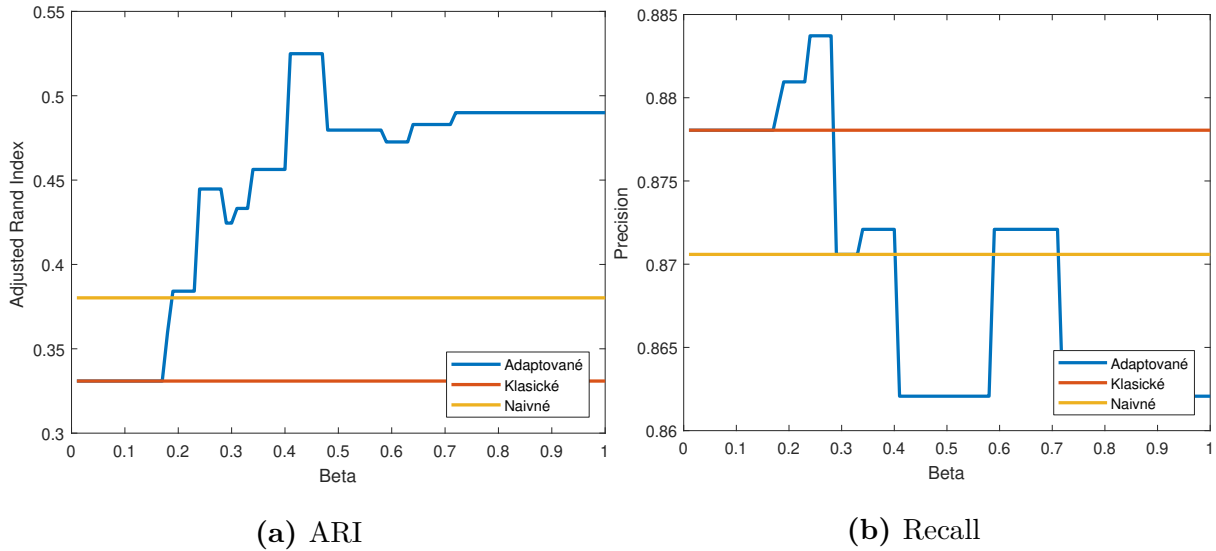
V tejto časti sa vrátíme opäť k databáze zo sekcie 2.1, aby sme lepšie porovnali klasickú metódu zhlukovania založeného na poradí, jej adaptovanú verziu, ktorú sme vymysleli v časti 1.5.2 a jej naivnú verziu, ktorá využíva apriori zhlukovanie priamo vo vektoroch črt. Fotografie z tejto databázy nedisponujú GPS súradnicami, a teda využijeme časy vytvorenia fotografií.

Zvolili sme delenie na samostatné fotografie a prah rovný 0.0033 dňa, čo zodpovedá niečo menej ako 5-tim minútam. Po zhlukovaní (ktorého postup sme opísali v predchádzajúcej časti) sme dostali zhľuky zobrazené na obrázku 2.18. Rovnaká farba zhľukov, ktoré nie sú bezprostredne vedľa seba samozrejme nezodpovedá rovnakému zhľuku, takto sú zobrazené len pre nedostatok rozumne rozlíšiteľných farieb. Zhľukovanie fotografií sme previedli na jednotlivé tváre (tak, že sme priradili každú tvár do zhľuku podľa pôvodnej fotografie, z ktorej bola vyrezaná) a dostali sme hodnotu *ARI* rovnú približne **0.75819**, čo zodpovedá pomerne slušnému zhlukovaniu, no vo všeobecnosti nemožno očakávať (dokonca čakáme, že v reálnych podmienkach to tak nebude skoro nikdy), že zhľuky získané len z exif dát budú tak veľmi podobné skutočnému priradeniu tvárí. V skutočnosti totiž netvoríme fotografie tak, že by sme za niekým prišli, spravili mu 5 fotografií a následne ho už nikdy viac nefotili.

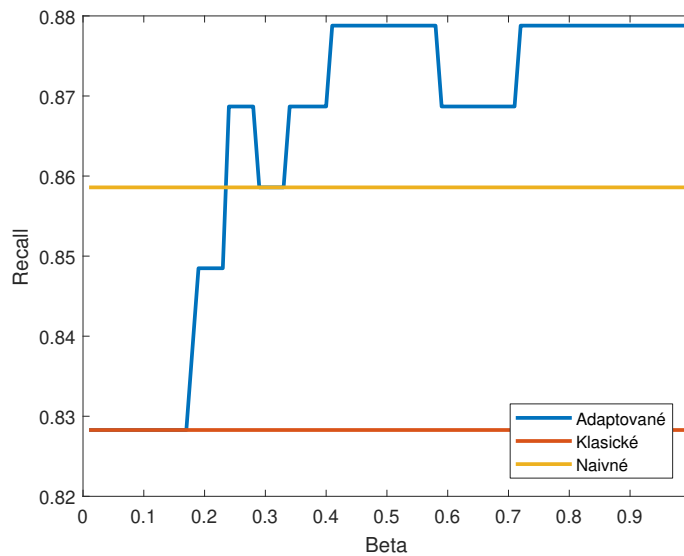


Obr. 2.18: Zhľukovanie časových údajov pre databázu zo sekcie 2.1

Pre porovnanie zhlukovania založeného na poradí sme teda vypočítali hodnotu kritérií *ARI*, *Precision* a *Recall* pre hodnoty β v intervale $[0.01; 1]$. Výsledky uvádzame na obrázkoch 2.19 a 2.20.



Obr. 2.19: Hodnoty kritérií *ARI* a *Precision* pre adaptované, naivné a klasické rodbe



Obr. 2.20: Hodnota kritéria *Recall* pre adaptované, naivné a klasické rodbe

Pri všetkých sme využili reprezentáciu tvárí pomocou siete Alexnet a parametre t a $\mathcal{K}$ sme zvolili 9, respektíve 2. Nekonzistentnosť výsledkov klasickej metódy na obrázku 2.19 a tabuľky 2.2 je spôsobená tým, že v tomto prípade sme zaradili jednoprvkové zhľuky do zhľuku šum a tým znížili hodnotu kritéria *ARI*, ktoré neberie do úvahy tento špeciálny zhľuk.

Adaptovaná metóda dopadla najlepšie najmä z hľadiska *ARI*, kde poskytuje značne lepšie výsledky pre mnohé hodnoty parametra β . Podobne je tomu aj pri kritériu *Recall*,

kde sú však rozdiely v hodnotách značne menšie. Najmenšie rozdiely sú napokon v kritériu *Precision*, kde všetky tri metódy dopadli pomerne dobre, no adaptované aj naivné zhlukovanie niekde zaostávajú za klasickou metódou. Toto sa dalo očakávať, keďže apriori zhlukovanie času neposkytuje (a ani nemá poskytovať) dokonalú informáciu o skutočných zhlukoch. Využívanie takejto informácie je potom akýsi kompromis medzi vyššou hodnotou *ARI* a nižšou hodnotou *Precision*, preto považujeme za značnú výhodu schopnosť adaptovať parameter β pri adaptovanej verzii zhlukovania založeného na poradí.

Záver

Cieľom tejto diplomovej práce bol návrh a aplikácia efektívnych metód výpočtovej inteligencie pre zhľukovanie tvárí pre následné zhľukovanie osobných fotografií, podľa ľudí nachádzajúcich sa na nich. V prvej kapitole sme najskôr uviedli algoritmus *Viola-Jones*, ktorým sme hľadali jednotlivé tváre na fotografiách. Rozpoznávanie ľudí je pomerne obširná a dobre spracovaná téma, preto sme sa zamerali skôr na tvorbu vektorov črt zo samotných tvárí a tiež zhľukovacím algoritmom, ktoré poskytujú finálne zatriedenie tvárí, a teda aj pôvodných fotografií.

V druhej časti tejto práce sme sa venovali práve spomínanej tvorbe vektorov črt. Využili sme štandardný algoritmus pre redukciiu dimenzie PCA s voľbou počtu najväčších vlastných čísel podľa podielu vysvetlenej variancie (podrobne sme ho rozobrali už v predchádzajúcej práci [2]). Okrem tohto algoritmu sme používali tiež jeden z najrozšírenejších algoritmov *HOG*. Tento dosahoval výborné výsledky pri testovaní na umelých dátach a bol časovo i výpočtovo nenáročný. Tiež sme využili jeden netradičný spôsob redukcie dimenzie pomocou transferového učenia. Išlo o využitie hlbokých predtrénovaných neurónových sietí *Alexnet* [14] a *VGG-16* [15]. Týmto sieťam sme odstránili niektoré koncové vrstvy (v zásade je dôležité odstrániť práve poslednú, klasifikačnú vrstvu) a využili sme ich deskriptívnu silu pre tvorbu vektorov črt z interpolovaných obrázkov tvárí. Najmä menšia zo sietí *Alexnet* dosahovala vynikajúce výsledky pri „umelých“ databázach.

Tretia časť prvej kapitoly sa zaoberala jednotlivými zhľukovacími algoritmami. Využili sme najznámejšie algoritmy K -means a K -medoids, kde pri druhom zo spomínaných sme sa zamerali na korektné použitie heuristického prístupu *PAM* popísaného v pôvodnom článku [6] a tiež so zjednodušeným značením v [7]. Ďalej sme sa venovali spektrálnemu zhľukovaniu, kde sme najskôr zaviedli a definovali niektoré dôležité pojmy z teórie grafov a pri využití poznatkov z lineárnej algebry načrtli riešenie relaxovaného problému

„krájania“ grafu pre dosiahnutie zhľukovania v pôvodných dátach. Otestovali a verifikovali sme tiež metódy K -means, K -medoids, spektrálne zhľukovanie a DBSCAN na umelo vytvorenom príklade dvojrozmerných špirálovitých dát, kde sme zistili nedostatky jednotlivých algoritmov. Metódy K -means a K -medoids umožnili hľadať len konvexné zhľuky a v prípade metódy DBSCAN bolo veľmi problematické voľiť parametre ε a $minPts$. Jedine spektrálne zhľukovanie bolo dostatočne jednoduché na voľbu parametrov.

Posledné dve zhľukovacie metódy, ktoré sme spomenuli, boli zhľukovanie založené na poradí podľa článku [12] a hierarchické zhľukovanie. Zhľukovanie založené na poradí sme podrobne analyzovali a opísali všetky potrebné kroky k jeho implementácii. Vyjadrili sme tiež isté znepokojenie pri voľbe parametra pri normalizovanej vzdialenosti zhľukov, ktoré sa v druhej kapitole ukázali ako opodstatnené. V prípade hierarchického zhľukovania sme najskôr zaviedli niekoľko vhodných spôsobov pre meranie vzdialenosti medzi zhľukmi a ukázali sme na príkladoch, že získanie *hard* zhľukovania (teda zhľukovania, kde výsledné zhľuky sú pevne dané) je jednoduché a stačí zastaviť aglomeratívne zhľukovanie pri požadovanom počte zhľukov.

Okrem algoritmov zhľukovania sme sa v tejto kapitole venovali ešte špecifiku merania vzdialenosti na guli, keďže meranie vzdialenosti medzi GPS súradnicami nie je úplne korektné pri použití euklidovskej normy. Tiež sme uviedli 4 spôsoby merania kvality zhľukovania, ktoré sme napokon využívali pre demonštračné príklady v druhej kapitole.

Pre potreby kombinácie črt tváří a výsledkov zhľukovania fotografií pomocou skrytých *EXIF* dát, sme najskôr navrhli vlastný naivný prístup, použiteľný pre ľubovoľné zhľukovanie a následne sme vyvinuli spôsob, ako modifikovať zhľukovanie založené na poradí pre využitie týchto informácií. V oboch prípadoch bolo potrebné ošetriť, aby sme neprispievali k zhľukovaniu objektov zaradených do zhľuku šum (v prípade naivnej metódy sme toto rozobrali v druhej kapitole v časti 2.3).

Poslednou časťou riešenia v prvej kapitole bola definícia kritéria siluet a Caliínského-Harabaszovho kritéria pre optimálnu voľbu počtu zhľukov pre metódy, ktoré vyžadujú určiť počet zhľukov K .

V prvej časti druhej kapitoly (2.1) sa venujeme často opomenutej téme, voľbe parametrov jednotlivých metód. Autori rôznych článkov síce uvádzajú, aké hodnoty pre danú metódu zvolili, nikde však nehovoria, ako na to prišli. Pre korektnosť (t.j. neovplyvňovanie

výsledkov samotných metód) sme sa rozhodli na umelo vytvorenej databáze nájsť vhodnú kombináciu všetkých parametrov pre jednotlivé metódy aj v závislosti od spôsobu tvorby vektorov črt. Metóda DBSCAN, ako sme ukázali v časti 2.1.1, bola veľmi komplikovaná z hľadiska voľby parametrov. Tento výsledok sa zhodoval so zisteniami na špirálovitých dátach uvádzaných už v predchádzajúcej kapitole na obrázku 1.11. Naopak veľmi robustnou a dobrou metódou sa ukázalo zhlukovanie založené na poradí. Na celej skúmanej množine parametrov t a $\mathcal{K}$ sme dosahovali dostačujúce až výborné výsledky, kde sa kritérium ARI pohybovalo v rozmedzí 0.2 až 0.8 (okrem prípadu kde t alebo $\mathcal{K}$ bolo rovné 1).

Spektrálne zhlukovanie pri voľbe plného grafu ukázalo, že voľba symetrického Laplaceovského grafu bola signifikantne lepšia (pre všetky skúmané hodnoty parametra σ) ako voľba nenormalizovaného Laplaceovského grafu. V zásade teda odporúčame preferovať symetricky normalizovaný Laplaceovský graf. Graf podobnosti s epsilonovým okolím sa ukázal ako numericky značne nestabilný a podobne dopadol aj normalizovaný Laplaceovský graf predstavujúci náhodnú prechádzku (s nízkou pravdepodobnosťou prechodu medzi zhlukmi vyplývajúcimi z tohto grafu). Napriek reálnosti a nezápornosti vlastných čísel tejto matice sme mali problém s ich numerickým vyčíslením vďaka jej nesymetricke. Črty *HOG* a *VGG16* sa v prípade vzájomných susedov pri $\mathcal{K}$ -najbližších susedoch ukázali tiež ako nevyhovujúce. Nedokázali sme nájsť žiadne vhodné parametre.

V časti 2.2 sme využili hlavnú databázu fotografií spolužiakov z našej predchádzajúcej práce [2]. Išlo o veľmi jednoduchú umelú databázu s centrovanými očami a homogénnym pozadím, na ktorej sme očakávali dobré výsledky. V tejto podkapitole sme tiež prvýkrát využívali, pri metódach, ktoré to vyžadovali, voľbu počtu zhlukov podľa kritéria siluet a Caliňského-Harabaszovho kritéria. Práve tu sa ukázalo, že tvorba vektorov črt podľa hlbkej siete *VGG16* a súčasne C-H kritérium nie je vhodná dvojica. Takmer vždy nám totiž „odporúčala“ hľadať len 2 zhluky. Po podrobnejšej analýze sme však prišli na to, že tieto 2 zhluky, až na jeden nesprávne zaradený subjekt, reprezentovali mužské a ženské pohlavie. Naopak sieť *Alexnet* dosahovala stabilne dobré výsledky naprieč širokým spektrom zhlukovacích metód.

Medzi metódami vynikalo najmä hierarchické zhlukovanie pri voľbe črt pomocou siete *Alexnet*. Metóda založená na poradí doplatila práve na opatrnosť pri zjednocovaní menej podobných objektov do zhlukov.

V poslednej časti práce 2.3 sme metódy testovali na reálnej databáze, ktorú sme vytvorili zo súkromných fotografií. Táto databáza predstavovala oveľa zložitejší problém ako predchádzajúce. Skutočné zhľuky osôb boli rôznej veľkosti a aj v rámci jedného zhľuku boli veľmi odlišné fotografie. Osoby tu na niektorých fotografiách dokonca mali slnečné okuliare a prvýkrát sme pracovali s tvármi, ktoré v skutočnosti boli len šumom. Tento šum bol nesprávne identifikovaný ako tvár rozpoznávacím algoritmom *Viola-Jones*. V tejto časti sme tiež poukázali na problém pri zhľukovaní GPS súradníc pomocou máp jednotlivých krajín. Fotografie vyhotovené na pobreží niektorých krajín tak boli síce veľmi blízko hraníc, ale nepatrili do žiadneho z definovaných polygónov, a teda museli byť najskôr prehlásené za šum. Tento problém sme však vyriešili pomocou priradenia k najbližšie definovaným bodom a definovaním parametra, ktorý by mal zamedziť patologickým prípadom, kedy žiadna z fotografií v blízkom okolí nie je zaradená do skutočného zhľuku. Tiež sme sa podrobnejšie venovali zhľukovaniu časových údajov, na základe zoradenia dát a interpretácie sklonu medzi skupinami časových známk.

Ďalej sme v tejto sekcii graficky ilustrovali už spomínané kritériá pre voľbu optimálneho počtu zhľukov aj s vykreslením konkrétnej siluety pre 2 zhľuky.

Pri porovnaní metód sme v tejto kapitole narazili na problém oboch kritérií pre kvalitu zhľukovania, *ARI* a *Precision*. Pri vysokej hodnote *Precision* sme dostávali veľmi nízke hodnoty *ARI* a naopak. Z celkového porovnania najlepšie vychádzala práve metóda zhľukovania založeného na poradí, ktorá síce mala nižšie hodnoty *ARI*, no mala aj signifikantne vyššie výsledky v kritériu *Precision*.

Výsledky diplomovej práce sú tak v oblasti analýzy, ako aj v návrhu konvenčných a nových efektívnych metód a algoritmov rozpoznávania tvárí osôb. Možno ich využiť v pedagogickej, ale aj vo výskumnej oblasti.

Literatúra

- [1] Izenman A. J.: *Modern Multivariate Statistical Techniques*, Springer Science + Business Media, New York, 2008
- [2] Kováč J.: *Rozpoznávanie ľudských tvárí pomocou PCA a LDA*: bakalárska práca, Bratislava: UK, 2016
- [3] Lloyd S.: *Least squares quantization in PCM*, IEEE transactions on information theory, Vol. 28, 1982
- [4] Hastie T., Tibshirani R., Friedman J.: *The Elements of Statistical Learning*, 2nd ed., Springer, Stanford, 2017
- [5] Haykin S.O.: *Neural Networks and Learning Machines*, 3rd ed., Pearson Education, Hamilton, 2011
- [6] Kaufman L., Rousseeuw P.J.: *Clustering by Means of Medoids*, red. Y. Dodge., 1987
- [7] Simovici D. A.: *The PAM Clustering Algorithm*, [cit. 6.5.2018], dostupné na internete: <https://www.cs.umb.edu/cs738/pam1.pdf>
- [8] Kaufman L., Rousseeuw P.J.: *Finding groups in data: an introduction to cluster analysis*, John Wiley & Sons, 2009
- [9] Ester M., Kriegel H.P., Sander J., Xu X.: *A density-based algorithm for discovering clusters in large spatial databases with noise*, InKdd, Vol. 96, No. 34, pp. 226-231, 1996
- [10] Omatu S., Neves J., Rodriguez J. M. C., Santana J. F. P., Gonzalez S. R. *Distributed Computing and Artificial Intelligence: 10th International Conference*, Springer Science & Business Media, 2013

- [11] Hubert L., Arabie P.: *Comparing partitions*, Journal of Classification 2 (1), pp. 193–218, 1985
- [12] Zhu C., Wen F., Sun J.: *A Rank-Order Distance based Clustering Algorithm for Face Tagging*, In Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference, pp. 481-488, 2011
- [13] Viola P., Jones M.: *Rapid object detection using a boosted cascade of simple features*, In Computer Vision and Pattern Recognition (CVPR), 2001 IEEE Computer Society Conference, Vol. 1, pp. I-I , 2001
- [14] Krizhevsky A., Sutskever I., Hinton G.: *Imagenet classification with deep convolutional neural networks*, In Advances in neural information processing systems, pp. 1097-1105, 2012
- [15] Simonyan K., Zisserman A.: *Very deep convolutional networks for large-scale image recognition*, arXiv preprint arXiv:1409.1556, 2014
- [16] Dalal N., Triggs B.: *Histograms of oriented gradients for human detection*, In Computer Vision and Pattern Recognition (CVPR), 2005 IEEE Computer Society Conference (Vol. 1, pp. 886-893), 2005
- [17] Ward Jr J. H.: *Hierarchical grouping to optimize an objective function*, Journal of the American statistical association, 58(301), pp. 236-244, 1963
- [18] Caliński T., Harabasz J.: *A dendrite method for cluster analysis*, Communications in Statistics, Vol. 3, No. 1, pp. 1–27, 1974
- [19] Plesník J.: *Algoritmy na sieťach*, učebné texty, KAMŠ, FMFI UK, Bratislava, 2016
- [20] von Luxburg U.: *A Tutorial on Spectral Clustering*, Statistics and Computing 17 (4), 2007
- [21] Lutkepohl H.: *Handbook of matrices*, Computational Statistics and Data Analysis, Sect. 5.2.2(6), 1997
- [22] Databases, [cit. 6.5.2018], dostupné na internete:
<http://www.face-rec.org/databases/>