Rozpoznávač samohlások („Vowel recognizer“)

Anton Urbaník

1. Abstrakt

Cieľom tohto projektu je overenie predpokladu, že samohlásky sú dostatočne špecificky modelované hlasivkami človeka. Aby ich bolo možné rozlíšiť je potrebné prevedenie zvuku do správneho tvaru, ktorý najcharakteristickejšie popisuje práve zvuk z obsiahnutým hlasivkovým tónom a následné aplikovanie klasifikačného algoritmu.
Dobrou reprezentáciou týchto zvukových dát sú koeficienty charakterizujúce frekvenčné spektrum zvuku akoby v komprimovanej forme. Takouto reprezentáciou sú napríklad LPC (pomenované po algoritme „Linear Predictive Coding“) alebo MFCC („Mel-Frequency Cepstrum Coeficients“) koeficienty. Ja som sa na základe pokusu rozhodol pre MFCC koeficienty . Na ich získanie som využil voľne dostupný softvér (HTK toolkit).

Na klasifikáciu som použil hľadanie centroidov zhlukov pre jednotlivé samohlásky Ako uvidíme neskôr, každá samohláska reprezentovaná koeficientmi je v priestore pekne oddelená od ostatných a tvorí zhluk. Po natrénovaní (nájdení centroidov pre jednotlivé samohlásky), klasifikácia funguje na rovnakom princípe hľadania centroidu rozpoznávanej hlásky a následnom zaklasifikovaní k najbližšiemu centroidu („a“, „e“, „i“, „o“, „u“).
Výsledkom projektu bude aplikácia, ktorá po natrénovaní rozpozná ľubovoľnú samohlásku.

2. Úvod

Na základe základných znalostí o zvuku ako takom, ľahko možno ukázať, že najcharakteristickejšie hlásky ľudskej reči sú samohlásky („a“, „e“, „i“, „o“, „u“) a to z dôvodu ich vzniku. Práve samohlásky sú modelované hlasivkami s vyskytujúcim sa výrazným hlasivkovým tónom a sú jemne domodelované nosovou a ústnou dutinou. Ostatné hlásky (spoluhlásky) sú modelované prechodom vzduchu cez hrtan a hlavne špecifickým vytvarovaním úst. Preto je predpoklad, že práve samohlásky sú špecifické a ľahko rozpoznateľné na základe ich charakteristiky.

Ako som už uviedol základom pri analýze zvuku je jeho **vhodná reprezentácia**. V našom prípade pôjde o reprezentovanie trojicami koeficientov MFCC, ktoré sa nám budú

dobre vizualizovať v troj-rozmernej sústave súradníc. Následne budeme hľadať centroidy vzniknutých zhlukov podľa, ktorých budeme klasifikovať testovacie dáta.

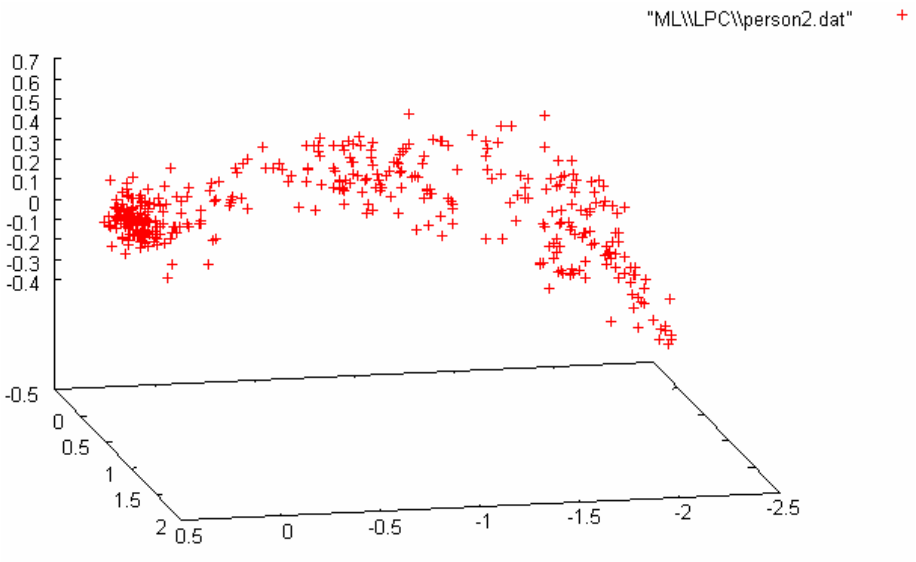
Ako **trénovaciu množinu** som použil nahrávky od šiestich rôznych ľudí vo veku od 20 do 25 rokov. Ide o troch chlapcov a tri dievčatá, ktorí mi cez rovnaký mikrofón nahrali jednu zvukovú stopu, kde po sebe povedali samohlásky („a“, „e“, „i“, „o“, „u“). Pre testovacie účely mi sporadicky nahrali aj jednotlivé hlásky od každého rôzne a každú do zvlášť súboru. Tie som použil ako testovacie dáta. Okrem nich mi ešte jeden iný chlapec a jedno iné dievča nahralo tiež zvlášť samohlásky, aby som dostal variabilnú testovaciu množinu dát. Trénovacie dáta som ešte podľa potreby orezal v programe WaveSurfer, kde som zo zvuku vystrihol veľké medzery medzi slovami. Ide o šum, ktorý vo veľkom množstve kazí trénovací proces ako vysvetlím neskôr. Potom sa dáta pri testovaní, či trénovaní prevedú zo štandardného formátu *.wav* do textového formátu *.dat* kde sú reprezentované ako MFCC koeficienty, s ktorými sa ďalej pracuje.

3. Použité metódy (riešenie projektu)

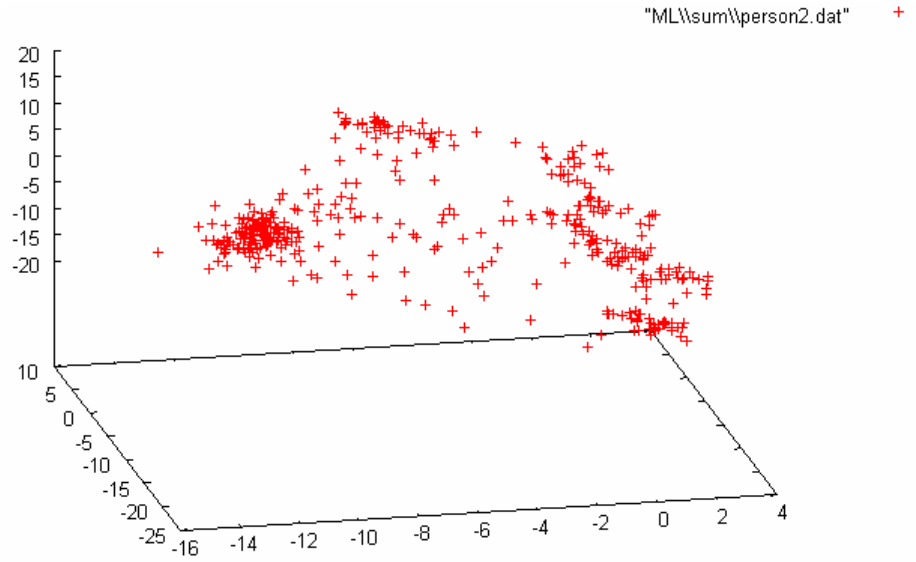
V prvom rade šlo o zvolenie metódy, alebo algoritmu pomocou, ktorého by som dostal zvuk z formátu *.wav* do formátu vhodného na analýzu, ktorý najviac vystihuje samotné dáta.

Výber medzi **LPC a MFCC** koeficientmi:

U oboch koeficientov sa signál rozdelí do políčok (vzoriek) určitej dĺžky, kde každú vzorku signálu možno popísať kombináciou predchádzajúcich vzoriek. To sa deje pomocou počítania energie jednotlivých frekvencií za použití určitých komplikovaných vzorcov. Napríklad pri výpočte MFCC sa počíta fouriérova transformácie a inverzná fouriérova transformácie a vypočítajú sa kepstrálne koeficienty a upravia podľa mel škály... Keďže ide o komplikované počítanie s potrebnými hlbšími vedomosťami o zvuku tak som použil voľne dostupný softvér „HTK toolkit“ a to konkrétne jednu jeho knižnicu s názvom „HList“, starajúcu sa o generovanie práve týchto koeficientov. Aby som sa rozhodol, ktoré koeficienty budú lepšie (z literatúry som zistil že MFCC lepšie popisujú reč) rozložil som trénovacie dáta od rovnakej osoby do súborov z LPC a MFCC koeficientmi a porovnal rozdiel v grafickej aplikácií „Gnuplot“ :



Obrázok 1. - LPC koeficienty samohlások „a“, „e“, „i“, „o“, „u“



Obrázok 2. - MFCC koeficienty samohlások „a“, „e“, „i“, „o“, „u“

Z testu jednoznačne vyplynulo, že **MFCC koeficienty pre naše účely budú vhodnejšie** na základe väčšej priestorovej separovateľnosti jednotlivých samohlások.

Hľadanie centroidov:

Pre každý vygenerovaný súbor s MFCC koeficientmi prislúchajúci jednej osobe, nájdeme centroidy jednotlivých hlások, za pomoci metódy zhlukovania a hľadania centroidu. Okrem hľadania **piatich centroidov** samohlások musíme zahrnúť aj šiesty **centroid pre šum**. Šum môžeme vidieť napríklad na obrázku 1 i 2 a jedná sa o najväčší zhluk na ľavo. Ako som spomenul v abstrakte, kvôli zvýšenému šumu v trénovacej množine som musel jednotlivé zvukové signály upraviť, aby v nich bolo rovnaké zastúpenie trvania jednotlivých hlások

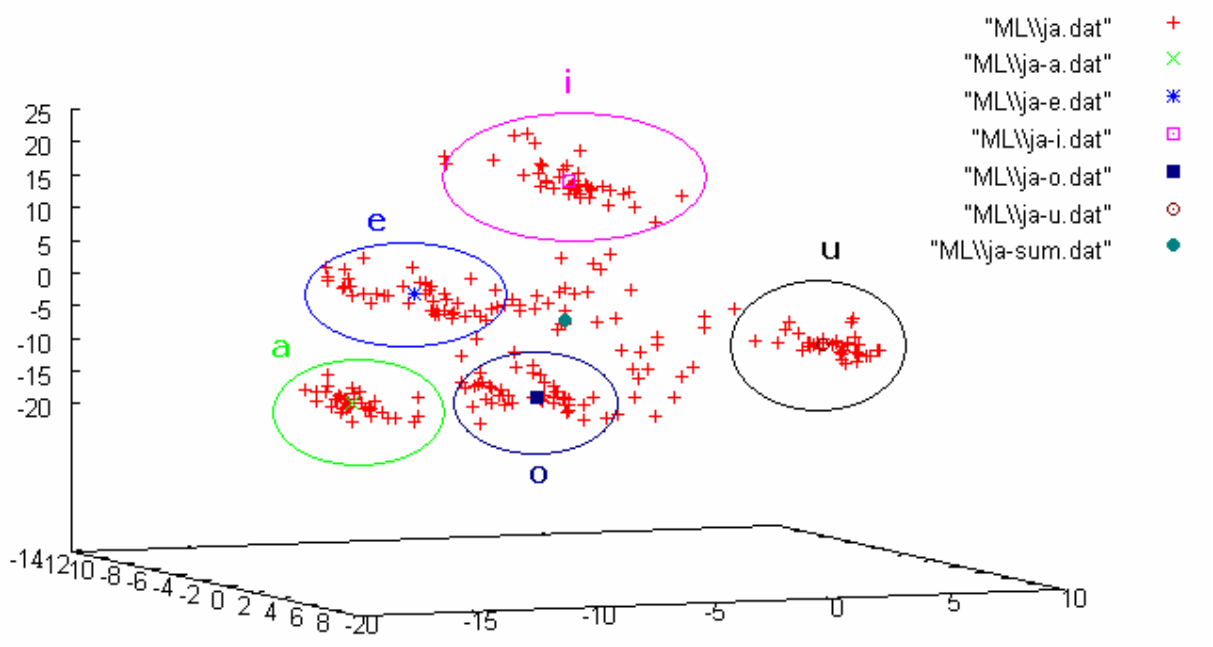
a šumu. V prípade že trvanie šumu bolo väčšie, vo výsledku sa vyskytli napríklad dva centroidy z oblasti jedného zhluku, alebo z oblasti šumu, v závislosti na povahe dát.

Algoritmus na hľadanie centroidov zhlukov počíta nasledovne:

- načíta do zoznamu vektorov MFCC koeficienty zvuku
- vytvorí štartovacie defaultné centroidy
- vytvorí šesť prázdnych množín zhlukov
- v cykle, ktorému definujeme počet interáci na pevno vykoná:
    - zmaže množinu zhlukov
    - rozdelí množinu vektorov do zhlukov podľa najbližšieho centroidu
    - prepočíta nové centroidy zhlukov
- pred ukončením pridelí zhluku samohlásku, ktorá má v zhluku najpočetnejšie zastúpenie

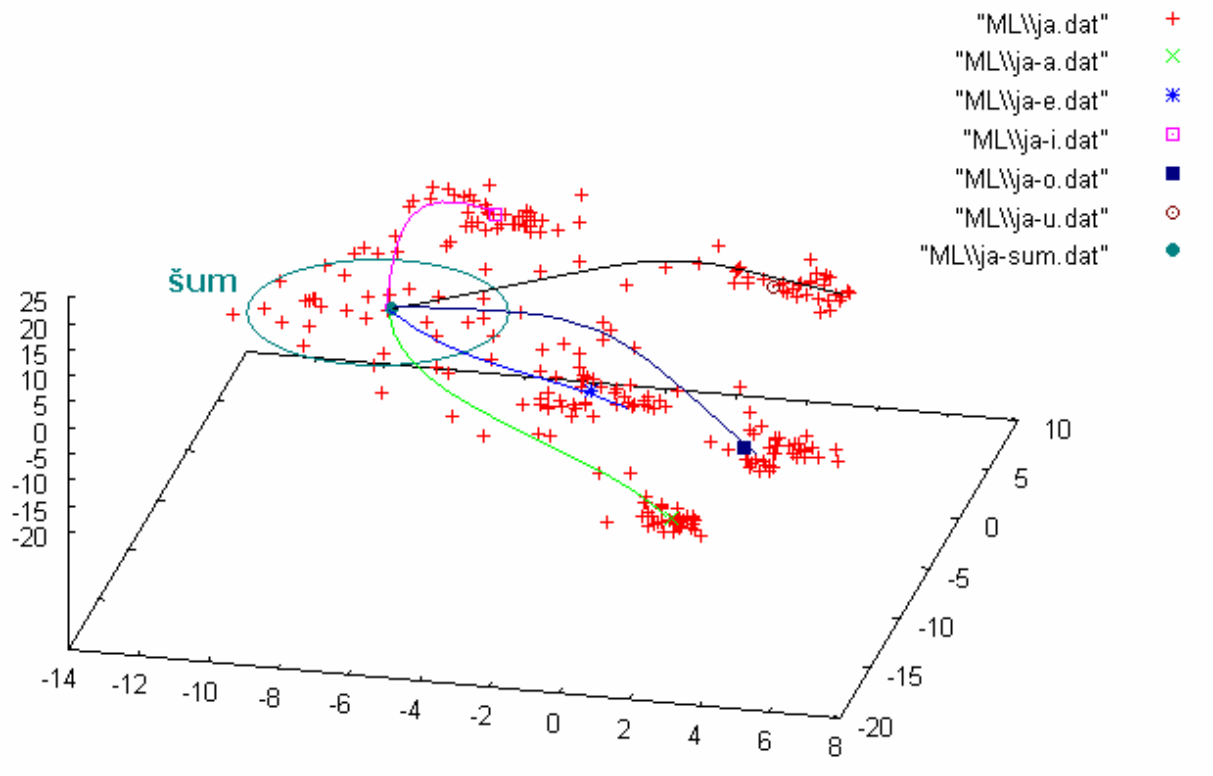
Týmto spôsobom dostaneme centroidy zhlukov pre zvuk jednej osoby.

Graficky to vyzerá nasledovne:



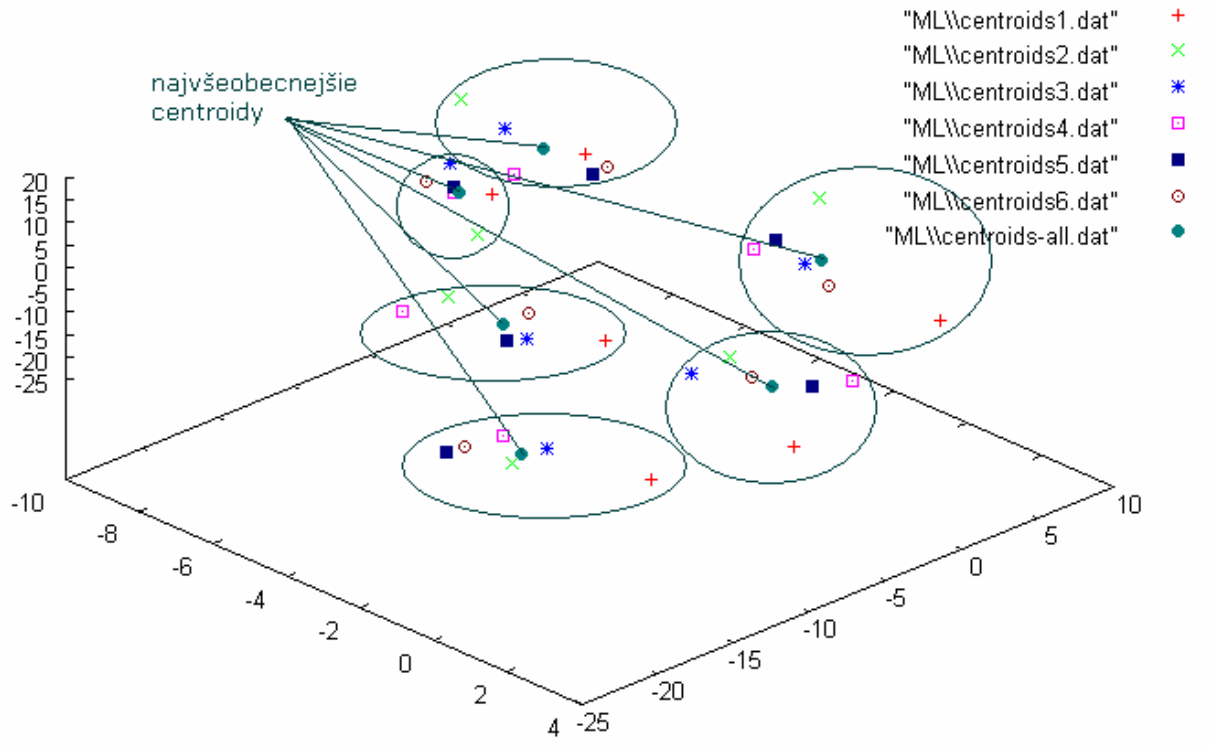
Obrázok 3. - zvuková stopa samohlások „a“, „e“, „i“, „o“, „u“ spolu s centroidmi

V strede obrázka 3. si môžeme všimnúť šum. Každá samohláska je reprezentovaná v troj-rozmernej súradnicovej sústave ako „slíž“ vychádzajúci z miesta šumu. To môžeme vidieť na pohľade z iného uhla na tieto dáta :



Obrázok 4. - zvuková stopa samohlások „a“, „e“, „i“, „o“, „u“ spolu s centroidmi a šumom

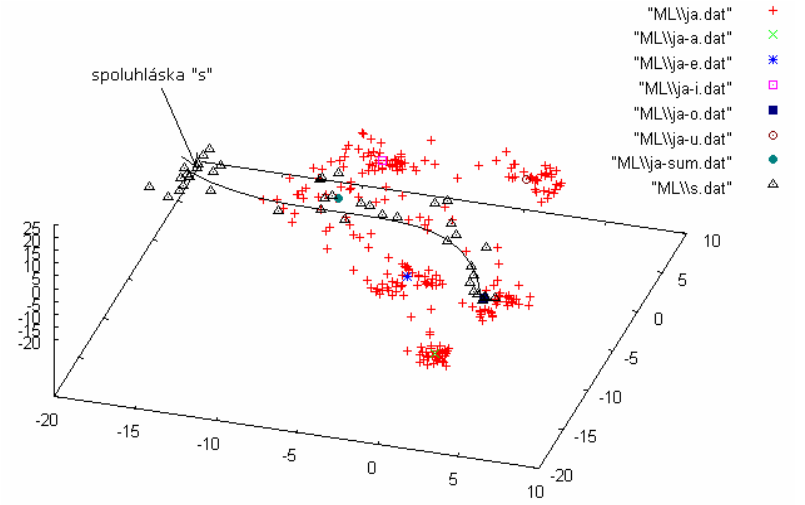
Po spočítaní centroidov pre celú trénovaciu množinu určíme **„najvšeobecnejšie“ centroidy**, ktoré budeme používať pri klasifikácii. Tie vypočítame rovnakým spôsobom ako centroidy zhlukov. Jednoducho povedané, ide o **centroidy** už spočítaných **centroidov**.



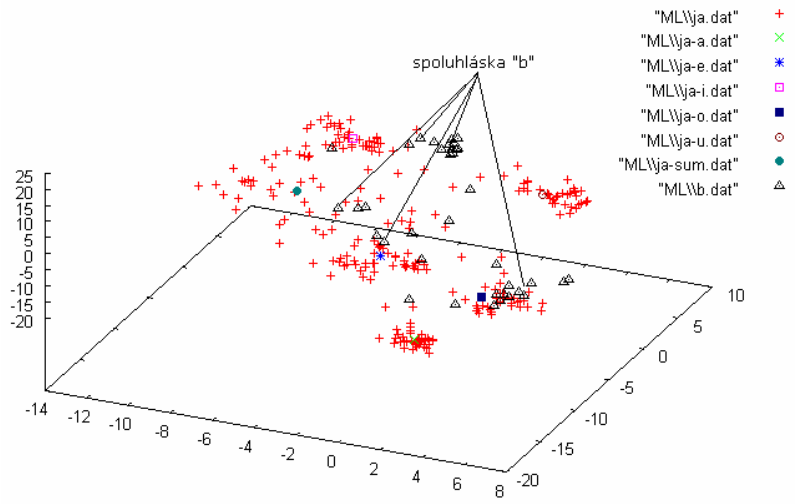
Obrázok 5. – centroidy centroidov samohlások „a“, „e“, „i“, „o“, „u“ a šumu

Klasifikácia:

Klasifikovanie testovacích zvukov je z dobre natrénovaným modelom už pomerne jednoduché. Používame už dostupnú funkcionalitu. Neznámi zvuk najskôr prevedieme z formatu .wav do textového súboru .dat z MFCC koeficientmi a tým následne nájdeme **dva centroidy**. Dva z dôvodu toho, že jeden je pre šum a druhý pre hľadanú hlásku. Následne nájdeme **najbližšie najvšeobecnejšie centroidy**. V ideálnom prípade (ako testovací zvuk je samohláska) ako výsledok dostaneme jeden najvšeobecnejší centroid šumu a druhý hlásky, ktorá je vlastne riešením a teda rozpoznanou hláskou. Ak sa jedná o rozpoznávanie šumu tak oba nájdené centroidy budú najbližšie k najvšeobecnejšiemu centroidu šumu. V prípade, že sa snažíme klasifikovať nenatrénovanú hlásku (napr. spoluhlásku „b“, „c“, „d“ ...) algoritmus ju zaklasifikuje k jednej zo samohlások, ku ktorej je centroid najbližšie.



Obrázok 6. - zvuková stopa spoluhlásky „s“ a samohlások „a“, „e“, „i“, „o“, „u“



Obrázok 7. - zvuková stopa spoluhlásky „b“ a samohlások „a“, „e“, „i“, „o“, „u“

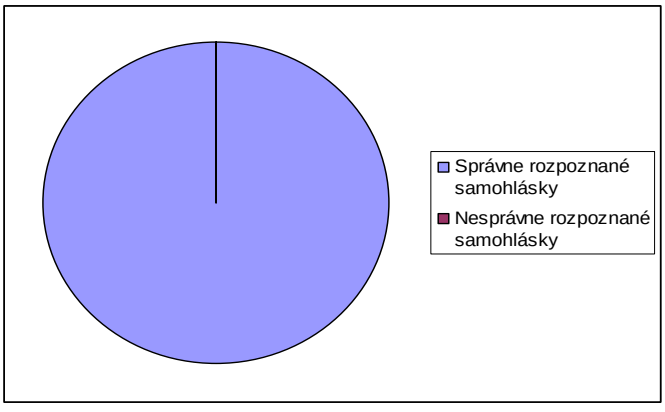
Ako vidno z obrázkov 6 a 7 spoluhlásky majú svoje koeficienty roztrúsené po celom priestore a preto sa nedajú jednoznačne natrénovať a klasifikovať týmto opisovaným spôsobom.

4. Implementácia

Projekt som vypracoval vo vývojovom štúdiu Visual Studio 2008 v jazyku C# na platforme .NET Framework 3.5. Ide o štandardnú WinForm aplikáciu, ku ktorej som pripojil knižnicu *VowelRecognizer.Library.dll*, ktorá obsahuje základnú funkcionalitu. K triedam knižnice som vygeneroval class diagram pripojený v prílohe. Okrem štandardných knižníc frameworku som využil knižnicu HTK toolkitu konkrétne *HList.exe*. Pri generovaní koeficientov sa používa nastavenie z konfiguračného súboru *HTKToolkit/hlist.conf*. Kvôli problémom predania parametrov z .NET aplikácie do *HList.exe*, za behu dynamicky vytváram *.bat* súbory, ktoré zavolajú externý program. Ide o súbory *HTKToolkit/TestData.bat* a *HTKToolkit/TrainData.bat*.

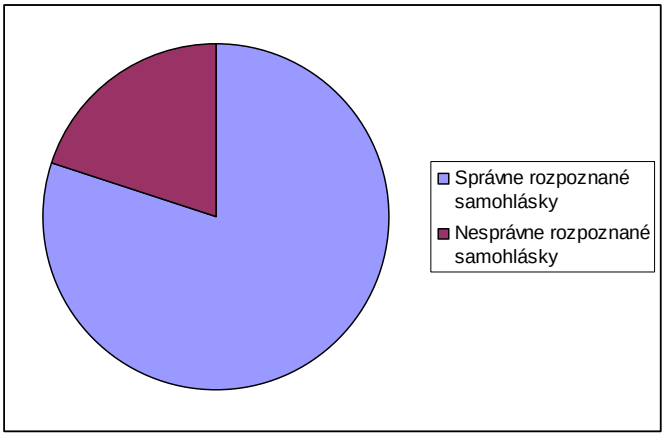
5. Vyhodnotenie

Do prvej testovacej množiny som si vybral jednu päticu samohlások nahratých náhodne od chlapcov, a jednu päticu samohlások nahratých náhodne od dievčat. Išlo o rovnaké osoby ako v prípade nahrávania trénovacích dát – celkovo teda 10 testovacích pokusov. Pri ich rozpoznávaní algoritmus nesklamal a rozpoznal všetky samohlásky správne. Výsledok je znázornený na grafe 1.



Graf 1. – správne a nesprávne rozpoznané samohlásky prvej testovacej množiny

Do druhej testovacej množiny som vzal všetky hlásky od inej osoby (nezávislej od trénovacích dát) – celkovo teda 5 testovacích pokusov. Algoritmus rozpoznal 4 z 5 samohlások („o“ rozpoznal ako „u“). Výsledok je znázornený na grafe 2.



Graf 2. – správne a nesprávne rozpoznané samohlásky druhej testovacej množiny

Ďalšie zistenie bolo, že dôležitou súčasťou získavania dát je mikrofón. Pri použití iného menej kvalitného mikrofónu pri nahrávaní testovacích dát som zistil, že obsahujú väčšie množstvo šumu a zvuk bol troška skreslený oproti defaultnému mikrofónu. Takže takto získane dáta boli veľmi neúspešne rozpoznané vo väčšine prípadoch ako šum.

Posledným zistením bolo, že pri vizuálnom porovnaní centroidov samohlások chlapcov a dievčat som nenašiel žiadne markantné odchýlky. Jednoducho každého hlas je špecifický, a každý mal centoridy posunuté do rôznych smerov bez pozorovaných závislostí medzi chlapcami a dievčatami.

6. Záver

Myslím si že tento projekt (rozpoznávač samohlások) dopadol úspešne. Podarilo sa navrhnúť klasifikovanie samohlások, ktoré na základe zistení a za pomoci naprogramovanej aplikácie, úspešne rozpoznáva samohlásky. Pri opätovnom riešení tohto problému by som skúsil použiť metódu SVM („Support vector machines“). Natrénoval by som SVM centroidmi hlások jednotlivých osôb a pri klasifikácii by som nehľadal najbližší centroid , ale klasifikoval za pomoci SVM.

7. Zdroje

[1] The Hidden Markov Model Toolkit (HTK) - http://htk.eng.cam.ac.uk/

konkrétne jedna jeho súčasť „HList.exe“ na generovanie koeficientov popisujúcich zvuk

[2] Teoretické informácie o MFCC koeficientoch - http://en.wikipedia.org/wiki/Mel_frequency_cepstral_coefficient

[3] Teoretické informácie o LPC koeficientoch - http://cs.wikipedia.org/wiki/LPAC

[4] Vizuálne zobrazenie dát pomocou aplikácia Gnuplot - http://www.gnuplot.info/

8. Prílohy a kometáre

Obrázok 1. - class diagram základných tried („ClassDiagram.cd“):