دانشکده برق و رباتیک

گروه الکترونیک

پایان نامه کارشناسی ارشد

طراحی و پیاده سازی سیستم شناسایی زبان گفتاری به صورت خودکار

پریا مهارلویی

اساتید راهنما:

دکتر امیدرضا معروضی

دکتر حسین مروی

استاد مشاور :

دکتر هادی گرایلو

بهمن ۸۹


بسم الله الرحمن الرحيم

دانشکده برق و رباتیک

گروه الکترونیک

طراحی و پیاده سازی سیستم شناسایی زبان گفتاری به صورت خودکار

دانشجو: پریا مهارلویی

اساتید راهنما:

دکتر امیدرضا معروضی

دکتر حسین مروی

استاد مشاور :

دکتر هادی گرایلو

پایان نامه ارشد جهت اخذ درجه کارشناسی ارشد

بهمن ۸۹

تقدیم به پدر و مادر عزیزم که گرانبهاترین درسهای زندگی را به من آموخته اند.

با تشکر از

دکتر معروضی، به پاس راهنمایی های بی دریغشان

دکتر مروی

دکتر گرایلو

و خانم مهندس قاهری که در طی این مسیر با من همراه بودند

و همچنین با سپاس فراوان از مسئولین پژوهشکده پردازش هوشمند علائم که در تهیه نمونه های صوتی چند زبانه ما را یاری نمودند.

چکیده

تشخیص اتوماتیک زبان[1] در واقع مسأله تشخیص زبان برای یک نمونه گفتار صحبت شده توسط سخنگوی نامعـلوم است. تشخیص خودکار زبان می‌تواند به ارتباط بین مردم نواحی گوناگون کمک کند و کاربردهای مختلفی در توسعه گردشگری ، تجارت آزاد، تقویت امنیت ملی از طریق پیش‌پردازش و فیلتر نمودن مکالمات مشکوک ، خدمات اورژانس، ترجمه همزمان در همایش‌ها و مکالمات بین المللی دارد.

در این پایان‌نامه با کمک کلاسه‌بندی ویژگی های مختلف، سیستم تشخیص خودکار زبان، طراحی و پیاده سازی شده است. برای این منظور، ویژگی‌های مناسب هر زبان را یافته و با دسته بندی آن برای زبان‌های مختلف، الگوریتم کلاس‌بندی و گسسته سازی چند بازه ای را آموزش داده و پس از دسته‌بندی آنها، قواعد تصمیم گیری برای هر زبان تعیین شده و از این دسته‌بندی برای تشخیص زبان‌های تست استفاده می‌کنیم.

برای آزمایش روش پیشنهادی، از نمونه‌های صوتی ۱۰ ثانیه‌ای و ۴۵ ثانیه‌ای پایگاه اطلاعاتی OGI-TS استفاده گردیده‌است. در OGI-TS نمونه‌های صوتی از ۱۱ زبان انگلیسی، فارسی، آلمانی، اسپانیایی، کره‌ای، ماندارین، ژاپنی، تامیل، ویتنامی، فرانسوی و هندی با زمان‌بندی‌های گوناگون موجود است. اما در سیستم‌های تشخیص زبان، بیشتر از ۹ زبان اول استفاده شده است. به همین

[1] Language Identification Automatic

منظور ما نیز آزمایش‌ها را بر روی این ۹ زبان انجام داده و با روش‌های پیشین مقایسه نمودیم. آزمایش‌ها بر روی ضرایب مختلف موجک[۲]، MFCC، PLP و LPC انجام شده اند.

تا کنون روش‌های مختلفی برای شناسایی زبان گفتاری به صورت خودکار پیشنهاد شده است، که بیشتر آنها وابسته به اطلاعات واج‌آرایی بوده و استفاده از آنها دشوار می‌باشد. ما در این پژوهش روشی مستقل از واج‌آرایی ارائه دادیم که در عین سهولت، با درصد خوبی قادر به تشخیص زبان‌ها است. در این روش از تبدیل موجک و تبدیل کپسترال[۳] نمونه‌های صوتی استفاده گردیده که بدون نیاز به اطلاعات زبان‌شناسی، بر روی زبان‌های گوناگون قابل استفاده می‌باشند. مشاهده گردید که ضرایب کپسترال به درصد صحت بالاتری نسبت به ضریب موجک می‌رسند. همچنین برای هر دو ضریب کپسترال و موجک، نمونه‌های صوتی ۴۵ ثانیه‌ای به دلیل مدت زمان بیشتر، درصد تشخیص بهتری دارند. روش‌های پیشین بیشتر به تشخیص دوبه دوی زبان‌ها می‌پرداختند، در حالیکه روش پیشنهادی قادر به تشخیص نوع زبان، از میان ۹ زبان موجود در OGI-TS نیز می‌باشد.

[2]Wavelet transformation
[3] Cepstral transformation

فهرست

فهرست شکل‌ها:

فهرست جدول‌ها:

فهرست علائم و اختصارات

AR	Autoregressive
ASR	Automatic Speech Recognition
DARPA	the Defense Advanced Research Project Agency
DCT	Discrete Cosine Transformation
ELDER/ELDA	European Language Resource Distribution Agency
GMM	Gaussian Mixture Model
HMM	Hidden Markov Model
LDC	Linguistic Data Consortium
LID	Language Identification
LM	Language Model
LPC	Linear Prediction cepstral
LPCC	Linear Prediction cepstral coefficient
LVCSR	Large Vocabulary Continuous Speech Recognition System
MFCC	Mel-Frequency Cepstral Coefficients
OGI-TS	OGI Multi-language Telephone Speech Corpus
PLP	Perceptual Linear Predictive
PPR	Parallel Phone Recognition
PRLM	Phone Recognition followed by Language Modeling
PRLM-P	Phoneme Recognition followed by Language Modeling performed in Parallel
PSWR	Parallel Sub-Word Recognition
SDC	Shifted Delta Cepstrum
SNR	Signal Noise Ratio
SVM	Support Vector Machine
SWR	Sub-Word Recognition
VOP	Vowel Onset Points
VQ	Vector Quantization

فصل اول

مقدمه

فصل اول: مقدمه

۱.۱) بیان مسئله

شش میلیارد نفر در سطح جهان زندگی می‌کنند که ۶۴ درصد آنها به ۱۴ زبان از حدود ۳۰۰۰ زبان شناخته شده جهان سخن می‌گویند[۱]. دانستن زبانی که به آن سخن می‌گوییم برای ارتباط با یکدیگر لازم است.

تشخیص اتوماتیک زبان در واقع مسأله تشخیص زبان برای یک نمونه گفتارصحبت شده توسط سخنگوی نامعلوم است. انسان‌ها، تا به امروز، دقیق‌ترین سیستم تشخیص زبان[۴] در جهان هستند. معمولاً اشخاص پس از شنیدن یک یا دو جمله، قادر به تشخیص زبانی که می‌شناسند، هستند و اگر زبانی باشد که با آن آشنایی نداشته باشند، لااقل می‌توانند به طور ذهنی تشخیص دهند که زبان مورد نظر به کدامیک از زبان‌هایی که می‌شناسند، نزدیک‌تر است. مثلاً می‌گویند:" شبیه به زبان آلمانی است." این قابلیتی است که مایلیم آن را به کمک سیستم‌های خودکار پیاده‌سازی نماییم.

[4] Language Identification (LID)

قابلیت‌های انسانی بر مکالمات کوتاه‌مدت تأثیرگزارند، اما تعداد زبان‌های بومی و آشناتر در مقایسه با تعداد زبان‌های گفتاری متمایز در جهان بسیار کم است و بنابراین درصد خطای تشخیص بالایی خواهیم داشت. همچنین تشخیص زبان غیرسلیس، برای انسان دشوار است و در ترجمه همزمان گفتگوها، گاهی تأخیرهای طولانی در حدود چندین دقیقه، با توجه به زبان یک گفتار، رخ می‌دهد [۲][۳][۴].

سؤال‌های اساسی در زمینه تشخیص خودکار زبان شامل این موارد می‌باشد:

کدام الگوریتم‌ها، روش‌ها و یا راهکارها برای تشخیص خودکار زبان مفیدترند؟ کدام منابع زبان برای مدلسازی زبان گفتاری مورد نیازند؟ ارتباط بین طول مدت سیگنال گفتاری و درصد صحت تشخیص چه میزان است؟ کدام پارامترها برای به دست آوردن اطلاعات زبان‌های مجزا بهترین هستند؟ کدام مرحله اطلاعات، بیشترین تأثیر را بر روی درصد تشخیص دارد؟ و همچنین سؤالات وابسته به نحوه عملکرد انسانی به هنگام تشخیص زبان، از قبیل اینکه انسان‌ها چگونه اقدام به تشخیص زبان می‌کنند؟ و آیا دانستن نحوه تشخیص انسان‌ها می‌تواند برای عملکرد تشخیص خودکار مفید باشد؟ و سؤالات مفید دیگر مربوط به تعریف زبان.

۱.۲) کاربردهای تشخیص زبان و ضرورت آن

کاربردهای تشخیص زبان به دو دسته مهم تقسیم می‌شوند:

۱) پیش‌پردازش برای ماشین‌ها ۲) پیش‌پردازش برای شنونده انسانی

سیستم بازیابی اطلاعات یک کنترل‌کننده چند زبانی، مثالی از دسته اول می‌باشد. سیستم تشخیص زبان برای مسیر یک تلفن ورودی به اپراتور انسانی و انتخاب مترجم زبان متناظر جهت پاسخگویی، مثالی از دسته دوم می‌باشند[۵][۳]. برای چنین کاربردهای چند زبانی، ماشین باید قادر به تشخیص زبان باشد.

تشخیص خودکار زبان می‌تواند به ارتباط بین مردم نواحی گوناگون کمک کند. در ذیل چندین کاربرد مهم برای تشخیص خودکار زبان را مرور می‌کنیم:

با گسترش اقتصاد جهانی و تجارت آزاد، نیاز به خدمات تشخیص خودکار زبان افزایش یافته است. برای مثال: رزرو هتل، تنظیم یک نشست کاری، و یا ایجاد تمهیدات لازم جهت مسافرت می‌تواند برای گوینده غیر بومی دشوار باشد[۲].

آژانس‌های مسافرتی برای تسریع در تشخیص زبان ارباب رجوع، به تشخیص خودکار زبان نیاز دارند تا سرویس‌های مناسبتری ارائه نمایند. به دلیل حملات تروریستی در سال‌های اخیر، حفظ امنیت ملی در همه کشورها اهمیت بسیار یافته است. تکنیک تشخیص خودکار زبان می‌تواند برای پیش‌پردازش و فیلتر نمودن مکالمات مشکوک مورد استفاده قرار گیرد.

در یک کشور چند زبانه همچون سنگاپور، که چهار زبان اصلی را در زندگی روزمره خود به کار می-گیرند، سیستم‌های تشخیص خودکار زبان اهمیت بسیار دارند. برای مثال یک سیستم ثبت خودکار قرار ملاقات‌های بیمارستانی، برای تشخیص زبان تماس گیرنده، به سیستم تشخیص خودکار زبان نیاز دارد تا بتواند به صورت خودکار، مکالمه را به اپراتوری که تماس گیرنده به آن زبان سخن می‌گوید منتقل نماید. این امر می‌تواند بهبود اساسی را در سیستم ثبت ایجاد کند.

همچنین شرکت‌های تلفنی که در زمینه مکالمات بین‌المللی فعالیت دارند، می‌توانند با استفاده از سیستم تشخیص زبان که بتواند مکالمات خارجی را به صورتی سلیس ترجمه کرده و به زبان مورد نظر برگرداند، خدمات مطلوبتری به مشترکین ارائه نمایند. گزارشاتی وجود دارد که اپراتور پاسخگوی اورژانس قادر به فهمیدن زبان تماس گیرنده مضطرب نبوده است. برای پاسخ به این نیازها AT&T اخیراً سرویس مترجم خودکار زبان را استفاده می‌کند[۲].

تشخیص زبان به صورت خودکار، یکی از قابلیت‌های مورد انتظار سیستم‌های هوشمند، در ترجمه مکالمه به متن و مکالمه به مکالمه در شرایط چند زبانه می‌باشد و تحقیقات چالش برانگیزی را در زمینه پردازش فناوری گفتار چند زبانه ایجاد نموده است[۶][۷][۸].

۱.۳) روش‌های تشخیص خودکار زبان

در دهه گذشته روش‌های متعددی برای تشخیص خودکار زبان پیشنهاد شده است. این روش‌ها، برای استخراج ویژگی‌های خاص هر زبان استفاده می‌شدند. ویژگی‌ها به طور کلی به سه دسته تقسیم‌بندی می‌شوند: مجموعه آواشناسی[۵]، واج‌آرایی[۶] و علم عروضی و بدیعی[۷][۹].

دو راهکار برای تشخیص زبان وجود دارد:

مورد عام‌تر محتوای واج‌آرایی هر زبان را استفاده می‌کند که این بر پایه قسمت‌بندی سیگنال گفتار در واج‌ها و بر استفاده مدل‌های زبان می‌باشد که همه ترکیبات ممکن واج‌ها از زبان ویژه را محاسبه می‌کند.

راهکار دیگر هیچ اطلاعات واج‌آرایی را در نظر نمی‌گیرد و منحصراً از خواص صوتی که از سیگنال گفتار استخراج می‌شوند، بهره می‌گیرد، همچون خواص عروضی، وزن[۸] و برخی خواص ادراکی دیگر[۱۰].

اخیراً در تحقیقات پردازش گفتار، برای ایجاد تنوع منابع در مدلسازی زبان‌ها، بیشتر به زبان‌های اصلی جهان توجه می‌کنند که به صورت استاندارد می‌باشند. اما از آنجا که کمبود منابع، دامنه انتخاب را برای مدلسازی زبان محدود می‌کند، توسعه بدون مرز ابزارهای تشخیص زبان با استفاده از سایر

[5] Phonetics
[6] Phonotactics
[7] Prosodic
[8] Rrhythm

زبان‌های گفتاری متداول پیشنهاد شده است. نتیجه مثبت این راهکار بر پایه صوت شناسی[9] ، استفاده از ابزارهای ساده‌تر تشخیص زبان و کاهش هزینه‌های مربوط به بخش انسانی می‌باشد.

پژوهش‌های انجام شده در تشخیص خودکار زبان باعث ایجاد زمینه‌های جدید همچون تشخیص لهجه، در پردازش گفتار شده است[۱۱][۱۲]. به عنوان مثال، مطالعاتی بر روی ساختار لهجه ماندارین، اسپانیایی، عربی، زبان‌های گوناگون آفریقای جنوبی و همچنین زبان‌های ایالتی انگلیس و فرانسه انجام گرفته است[۷][۱۳][۱۵].

درسیستم‌های تشخیص زبان، نیازی نیست که گویندگان به زبان بومی خود سخن بگویند، زیرا سیستم تنها باید به تشخیص نوع زبان صحبت شده بپردازد. اما در تشخیص لهجه، گویندگان بهتر است بومی همان زبان باشند تا سیستم قادر به تفکیک و تشخیص لهجه‌ها باشد. امروزه برای پرهیز از چالش‌های ایجاد شده توسط گوینده غیربومی و افزایش قابلیت استفاده از سیستم‌های گفتاری بومی‌های چندزبانه ، برای تهیه نمونه‌های صوتی، بیشتر از گویندگان بومی همان زبان استفاده می‌کنند[۱۵][۱۶][۱۷][۱۸][۱۹].

تعداد زبان‌های گفتاری متمایز، بسیار زیاد و حتی انتشار تعداد دقیقشان دشوار است. بر اساس منابع، تعداد زبان‌های متنوع متمایز بین ۴۰۰۰ تا ۸۰۰۰ می‌باشد[۲۰][۲۱][۲۲][۲۳]. دقیق نبودن این تعداد، بیشتر به دلیل تنوع تعریفی است که از زبان ارائه می‌شود. برخی زبان‌ها همچون لاتین و یونانی باستان، طی گذر زمان، از گفتار روزمره ساکنین منسوخ شده‌اند و تنها در کتاب‌های آموزشی ادبیات و یا کتب تاریخی یافت می‌شوند. اما تعداد زیادی از زبان‌های اقلیت، همچنان باقی مانده‌اند، هرچند که برخی از آنها دچار آسیب‌های جدی شده‌اند.

[9] Acoustic

۱.۴) تمایز بین زبان‌ها:

مسئله اصلی در تشخیص زبان، یافتن راهی برای کاهش پیچیدگی‌های زبان انسانی است تا یک الگوریتم خودکار بتواند نوع زبان را از میان نمونه‌های صوتی تشخیص دهد. برای حل مسأله تشخیص زبان می‌توانیم به استفاده از برخی ویژگی‌هایی که انسان‌ها برای تمایز بین زبان‌ها به کار می‌برند، بپردازیم. برخی از این ویژگی‌ها به قرار زیر می‌باشند:

۱.۴.۱) صوت شناسی:

صوت شناسی یک مشخصه فیزیکی سیگنال گفتار است که با فرکانس، زمان و شدت[۱۰]، تعریف می‌شود. اطلاعات صوتی یک عبارت گفته شده، به عنوان یک سلسله بردار ویژگی بیان می‌شوند که هر بردار، اطلاعات صوتی برای یک پنجره زمانی خاص را بیان می‌کند. این بردارهای ویژگی ممکن است شامل اطلاعات اضافه فاقد ارزش برای تشخیص زبان همچون نویز مرتبط با کانال و گوینده باشند. چگونگی استخراج ویژگی‌های صوتی مفید، یک چالش همیشگی برای سیستم تشخیص زبان بر پایه ویژگی صوت شناسی می‌باشد.

۱.۴.۲) عروض:

عروض به ساختار صوتی باز می‌گردد که بر روی چندین قسمت گسترش یافته است که شامل تکیه[۱۱]، آهنگ[۱۲]، وزن، طول مدت آواها و نرخ گفتار می‌شود. سه المان اول، مهمترین المان‌های مورد استفاده در ساختار عروضی یک متن گفتاری هستند. همه این المان‌ها در ساختار عروضی تشکیل‌دهنده متن زبان‌های مختلف، متحد می‌شوند. تفاوت‌های میان زبان‌ها، اغلب می‌تواند با درک ویژگی‌های عروضی

[10] Intensity
[11] Stress
[12] Intonation

آنها مشاهده شود که توسط نوا[۱۳] و تکیه در متن بیان می‌شود. برای مثال زبان‌های نواختی[۱۴](آهنگین) همچون ماندارین، مشخصه‌های آهنگین متفاوتی را در مقایسه با زبان‌های تکیه‌ای همچون انگلیسی دارند.

۱.۴.۳) واج آرایی:

واج‌آرایی به قوانینی باز می‌گردد که ترکیبات متفاوت آواها را در یک زبان تعیین می‌کند. دامنه تنوع زیادی در قوانین واج‌آرایی زبان‌های مختلف وجود دارد. هر زبان ممکن است قوانینی خاص برای ایجاد یک رشته آوا، در کنار هم داشته باشد. این قوانین واج‌آرایی ممکن است سبب شود آواهای اصلی در برخی زبان‌ها، بسیار شبیه به هم و بسیار متفاوت از زبان‌های دیگر شود. به عنوان مثال ژاپنی، قواعد واج‌آرایی بسیار دقیقی دارد که اجازه نمی‌دهد حروف بی صدا، پشت سر هم بیایند. از سوی دیگر انگلیسی، قواعد ناپایداری دارد که می‌تواند چندین حرف بی‌صدا را پشت سر هم بپذیرد.

بنابراین قواعد واج‌آرایی می‌توانند برای به دست آوردن برخی طبیعت‌های دینامیکی گفتار از دست رفته طی استخراج ویژگی‌ها استفاده شود. امروزه بسیاری از سیستم‌های تشخیص زبان موفق از نتایج اطلاعات واج‌آرایی استفاده می‌کنند.

۱.۴.۴) مجموعه لغت‌ها:

مهمترین تفاوت میان زبان‌ها آن است که مجموعه‌های متفاوتی از لغات را مورد استفاده قرار می‌دهند که ناشی از تفاوت واژگان آنها است. بنابراین یک گوینده غیر بومی انگلیسی، با وجود استفاده از الگوهای عروضی زبان بومی‌اش، به دلیل استفاده از لغت‌های انگلیسی، به زبان انگلیسی سخن می‌گوید.

---

[13] Tone
[14] Tonal

۱.۴.۵)گرامر(دستور زبان):

روش‌هایی که لغت‌ها می‌توانند قانوناً به یکدیگر متصل شوند و حاوی اطلاعات گوناگونی باشند، در دستور زبان گنجانده می‌شود. حتی هنگامیکه دو زبان در یک لغت مشترک باشند، مجموعه لغاتی که می‌توانند قبل و بعد از آن لغت قرار گیرند، متفاوت است.

ویژگی‌های سطح پایین‌تر همچون ویژگی‌های عروضی و واج‌آرایی به راحتی به دست می‌آیند، اما به دلیل تنوع گوینده و کانال‌های انتقال صوت، جزء متغیرهای نیرومند در تشخیص زبان نمی‌باشند. طی اولین سال‌های زندگی، کودکان قادر به درک تفاوت بین زبان مادریشان و دیگر زبان‌ها هستند که به ویژگی‌های عروضی زبان گفتار باز می‌گردد[۲۴]. این پدیده بیانگر آن است که شنونده در غیاب دانش سطح بالای زبان، به ملاک‌های سطح پایین‌تر اکتفا می‌کند. با این حال ویژگی‌های عروضی به طور کامل در تشخیص خودکار زبان مورد استفاده قرار نمی‌گیرند[۲۵].

واج‌آرایی به قواعدی مراجعه می‌کند که ترکیبات آوایی گوناگون در یک زبان را تحت تأثیر قرار می‌دهد. ترکیب واج‌ها و صداهای قابل قبول، اطلاعات متمایز زبانی زیادی را با خود دارند. آنها از یک تشخیص‌دهنده آوایی استخراج شده‌اند که فرض شده است نسبت به تغییرات گوینده و کانال انتقال مقاوم باشد.

ویژگی‌های سطح بالاتر همچون لغت و دستور زبان، بر پایه مجموعه وسیع واژگان است که تحت تأثیر زبان و منطقه می‌باشند و برای تعیین منحصر به فرد زبان یک گفتار، مورد استفاده است. استفاده از این اطلاعات برای مجموعه بزرگ زبان‌های گوناگون، ممکن است محاسباتی سنگین همراه داشته باشد و در پیاده‌سازی واقعی مشکل باشد.

تحقیقات اخیر بر ویژگی‌های واج‌آرایی و عروضی متمرکز شده اند[۲۵][۲۶][۲۷]. ویژگی‌های عروضی برای مکالمات کوتاه تر مفیدند در حالیکه واج‌آرایی بر روی مکالمات طولانی، بهتر عمل می‌کند.

۱.۵) دسته گفتار تلفنی چند زبانه[15]

انگیزه اصلی برای ایجاد دسته گفتار تلفنی چند زبانه، داشتن منابع متنوع داده‌ها در میان زبان‌های مختلف است، منابعی که همه تغییرات احتمالی را شامل شود. این تغییرات می‌تواند به علت تفاوت‌های گوینده‌ها، همچون سن، جنس و لهجه باشد و یا به دلیل تفاوت در میکروفن‌ها، دستگاه تلفن، خطوط ارتباط، نویز پس-زمینه و زبانی که به آن صحبت می‌شود، ایجاد شود. در واقع پایگاهی برای دسترسی به داده‌های زبان‌های مختلف، با شرایط گوناگون ایجاد شده است. بنابراین مهم است که دسته‌ها، گوناگونی گفتار را برای هر گوینده شامل شوند[۲].

۱.۵.۱) OGI-TS:

مجموعه گفتار تلفنی چند زبانه OGI-TS به طور خاص برای تحقیقات شناسایی زبان طراحی شده‌اند که شامل گفتارهای لغت-ثابت و خود به خودی در ۱۱ زبان به شرح زیر می‌باشد:

(Ko) کره‌ای، (Ja) ژاپنی، (Ge) آلمانی، (Fr) فرانسوی، (Fa) فارسی، (En) انگلیسی، (Vi) ویتنامی، (Ta) تامیل، (Sp) اسپانیایی، (Ma) ماندارین.

(Hi) هندی اخیراً به گروه اضافه شده است.

این بیانات توسط ۹۰ گوینده بومی در هر زبان، بر روی خطوط تلفن واقعی تولید شده‌اند که مدت زمان هریک از ۱ تا ۵۰ ثانیه می‌باشند، با متوسط حدود۱۳/۴ ثانیه[۲][۲۸] .

[15] OGI Multi-Language Telephone Speech Corpus

فصل دوم

تاریخچه تشخیص زبان

فصل دوم: تاریخچه تشخیص زبان

تشخیص خودکار زبان، زمینه ای است که طی چندین سال اخیر مورد توجه قرار گرفته و از حدود ۳۰ سال پیش تحقیقات فعالی در این مورد انجام و توسط کمیسون اروپایی حمایت بسیار شده است. چنانچه گوناگونی زبان با بیش از ۲۰ زبان رسمی فعلی، یکی از چالش‌های اروپا است[۲۹][۳۰]. تحقیقات توسط آژانس توزیع منابع زبان اروپایی[۱۶]، جمع آوری و توزیع شده است[۳۱]. در ایالات متحده، آژانس پژوهشی، تحقیقاتی پیشرفته حامی[۱۷]، به طور وسیع تحقیقات چند زبانه را انجام داده و شرکت داده‌های زبانی[۱۸] فهرستی شگفت انگیز از منابع چند زبانه شامل گفتار، متن و لغت را برای حداقل ۸۰ زبان گردآوری نموده اند[۳۲]. در سال ۱۹۷۴ *لئونارد*[۱۹] و *دودینگتون*[۲۰] با کار بر روی

[16] European Language Resource Distribution Agency (ELDER/ELDA)
[17] the Defense Advanced Research Project Agency (DARPA)
[18] Linguistic Data Consortium(LDC)
[19] Leonard
[20] Doddington

روش فیلترسازی صوتی[21]شروع به تشخیص خودکار زبان نمودند[۳۳]. *هوس*[22] و *نئوبورگ*[23] در سال ۱۹۷۷ با استفاده از قواعد واج‌آرایی کمک شایانی به تشخیص خودکار زبان کردند[۳۴]. در این قسمت برخی از مقالات برجسته تر آورده شده است. در این مقالات به بعضی روش‌های متفاوت که می‌توانند برای تشخیص خودکار زبان به کار روند، اشاره شده است:

دهه ۱۹۸۰-۱۹۷۰:

۱۹۸۰-۱۹۷۴:

ابزار کار در تگزاس بر پایه فرکانس رخداد صداهای مرجع اصلی، در زبان‌های مختلف بوده است. قسمت‌بندی خودکار این صداهای مرجع[۳۳]، نتایجی با ۶۴٪ درستی را بر روی مجموعه تست، شامل ۷ زبان موجب شدند. در مطالعات بعدی، راهکار برهم کنش انسانی برای تعیین صداهای مرجع، باعث ایجاد بهبود ضرایب گردید که بهترین نتایج منتشر شده[۳۵]، ضریب صحت ۸۰٪ را بر روی ۵ زبان نشان می‌دهد. ضعف تعیین دستی صداهای مرجع، ایراد اصلی دو زبان دیگر بود. در مقاله بعدی[۳۵] نشان داده شد که عملکرد برای مجموعه ۷ زبانه از ۷۲٪ به ۶۲٪ کاهش یافته است. این مطالعات، نظریه اختلاف زبان بر پایه آواشناسی را بیان می‌کنند.

۱۹۷۷:

کار *هوس* و *نئوبورگ* بر پایه رونویسی داده‌های واج‌آرایی به صورت دستی بوده است که شامل مدل مخفی مارکوف[24] تعلیم یافته بر روی برچسب‌های آواشناسی عریض، مشتق شده از نسخه برداری آوایی می‌باشد. آنها از ویژگی‌های صوت شناسی استفاده نکردند. در مقاله خود[۳۴]، تشخیص ۸ زبان

[21] Acoustic filter bank
[22] House
[23] Neuberg
[24] Hidden Markov model (HMM)

را با درصد بالا بیان نمودند و نشان دادند که تشخیص زبان توسط اطلاعات واج‌آرایی می‌تواند عملکرد بسیار خوبی داشته باشد.

:۱۹۸۰

لی[۲۵] و *ادواردس*[۲۶] [۳۶]، تکنیک‌های مارکوف پیشنهادی توسط *هوس* و *نئوبورگ* را بر روی داده‌های گفتار واقعی به کار بستند. آنها از دسته‌های آواشناسی عریض، برای محاسبه دو مدل آماری استفاده کردند: یکی بر پایه تک آوا[۲۷]، و دیگری بر پایه هجا[۲۸].

دهه ۱۹۹۰-۱۹۸۰ میلادی:

:۱۹۸۲

سیماروسی[۲۹] و *آیوس*[۳۰] یک کلاسه ساز چند جمله‌ای بر روی ۱۰۰ المان بردار ویژگی مشتق شده از LPC[۳۱] طراحی کردند که شامل ضرایب خود همبستگی، ضرایب کپسترال، ضرایب فیلتر سازی، بهره ناحیه لگاریتمی و فرکانس‌های فرمنت[۳۲] بودند[۳۷]. این راهکار بر پایه تک‌آواها نبوده و تنها بر پایه ویژگی‌های صوتی انجام شده است. روی هم رفته ضریب صحت ۸۴٪ بر روی ۸ زبان نشان می‌دهد که تشخیص زبان می‌تواند تنها بر پایه ویژگی‌های صوتی باشد.

:۱۹۸۶

[25] Li
[26] Edwards
[27] Segments
[28] Syllables
[29] Cimarusti
[30] Ives
[31] Linear Prediction cepstral
[32] Formant

فویل[۳۳] [۳۸]، دو نوع سیستم تشخیص زبان را امتحان نمود. در اولین روش از هفت ویژگی عروضی بر پایه وزن و آهنگ مشتق شده از منحنی آهنگ انرژی و گام (دانگ)[۳۴]، و در روش دوم از فرکانس‌های فرمنت استفاده کرد. او برای بیان خصوصیات الگوهای صوتی زبان، بردار پله‌ای[۳۵] و الگوریتم کلاسه بندی K-means را مورد استفاده قرار داد. عملکرد تشخیص زبان، ضریب صحت ۶۴٪ را همراه با ۱۱٪ عدم پذیرش، بر روی ۳ زبان از داده‌های جمع آوری شده از رادیو، با SNR[۳۶] ۵ دسی بل نشان می‌دهد.

:۱۹۸۹

گودمن[۳۷] [۳۹]، کار *فویل* را با کمی تغییر و افزایش پارامترهایی به بردار ویژگی و همچنین بهبود دسته‌بندی زبان‌ها گسترش داد.

دهه ۲۰۰۰-۱۹۹۰ میلادی:

:۱۹۹۱

سوجیاما[۳۸] [۴۰]، کلاسه بندی بردار پله‌ای بر روی ویژگی‌های مشتق شده از LPC را به پایان رسانید. او استفاده از کتاب رمز[۳۹]، بر مبنای زبان را برای بردار پله‌ای بررسی نمود. یک کتاب رمز معمول برای زبان‌ها، بر اساس الگوهای نمودار ستونی احتمال رخدادشان دسته بندی شده است. بهترین ضریب صحت تشخیص کلی، ۸۰٪ برای گفتار ناشناخته ۶۴ ثانیه‌ای به دست آمده است.

---

[33] Foil
[34] Pitch and energy contour
[35] Vector quantization
[36] Signal Noise Ratio
[37] Goodman
[38] Sugiyama
[39] Cod book

:۱۹۹۲

ناکاگاوا[۴۰] [۴۱]، چهار روش بردار پله‌ای، مدل مخفی مارکوف گسسته، مدل مخفی مارکوف پیوسته و مدل توزیع مخلوط گوسین[۴۱]، را مقایسه نمود. آزمایشات برای این روش‌ها، بر روی چهار زبان انجام شد. نتایج حاصل از استفاده مدل مخفی مارکوف پیوسته و مدل مخلوط گوسین(۸۱/۱٪)، بهتر از بردار پله‌ای(۷۷/۴٪) و مدل مخفی مارکوف گسسته(۴۷/۶٪) بودند.

:۱۹۹۳

موسوسامی[۴۲] [۴۲]، در رساله‌اش، بر روی کارهای زنجیره‌ای برای تشخیص زبان بحث نمود. او بیان کرد که صوت‌شناسی، آواشناسی و اطلاعات عروضی برای دستیابی به تشخیص خودکار زبان مورد نیازند. نخستین آزمایشاتش را بر روی ۴ زبان با گفتار کیفیت بالا انجام داد. بر اساس نتایج امیدوارکننده به دست آمده از این آزمایشات، او این روش‌ها را با مجموعه گفتار فصیح تلفنی بر روی ۱۰ زبان مورد بررسی قرار داد[۴۳]. آزمایشات با استفاده از ویژگی‌های بر پایه دسته‌های آواشناسی دوگانه و سه‌گانه، ویژگی‌های طیفی PLP و ویژگی‌های بر پایه گام(دانگ)، بر روی دو زبان انگلیسی-ژاپنی و سپس مجموعه ۱۰ زبانه انجام شد.

در رساله *موسوسامی* گسترش فرکانس رخداد و همچنین نسبت و طول مدت تک‌آوا، مورد بررسی قرار گرفته است. او با یک سیستم حاصل از ادغام همه ویژگی‌های ذکر شده، به درصد صحت ۴۸/۵٪ بر روی اظهارات کوتاه مدت(با متوسط ۱۳/۴ ثانیه) و ۶۵/۶٪ بر روی اظهارات بلندمدت(با متوسط ۵۰ ثانیه) بر روی ۱۰ زبان رسید.

[40] Nakagawa
[41] Gaussian Mixture distribution Model (GMM)
[42] Muthusamy

همچنین با این نتیجه‌گیری که اطلاعات آواشناسی به جای واج‌آرایی ممکن است مورد نیاز باشد، دقت تشخیص زبان‌ها را بالا برد. یک بروشور کامل مرور و توسعه دسته گفتار چند زبانه[۲۸]، بخش بزرگی از این کار است. آزمایشات ادراکی به گونه‌ای انجام شده‌اند که شنوندگان را برای تشخیص زبان‌ها، با استفاده از نمونه‌های صوتی ۴،۲،۱و ۶ ثانیه‌ای، بر روی هریک از ۱۰ زبان، توسط خبرگان گفتار، آموزش می‌دهند. متوسط عملکرد بر روی همه زبان‌ها، با افزایش طول مدت گفتار، از ۳۷٪ به۴۳٪، ۵۱/۲٪ و ۵۴/۶٪ افزایش یافته است.

:۱۹۹۵

یان[۴۳] [۴۴]، با مطالعه نقش صوت‌شناسی، واج‌آرایی و اطلاعات عروضی، یک اتحاد جزئی ایجاد نمود. او همچنین دو منبع اطلاعاتی جدید شامل LM وارونه و مدل درنگ[۴۴] وابسته به متن را معرفی نمود. بهترین ضریب صحت منتشر شده توسط او، ۹۱٪ برای نمونه‌های ۴۵ ثانیه‌ای و۷۷٪ برای نمونه‌های ۱۰ ثانیه‌ای بر روی ۹ زبان بوده است. او مجموعه‌ای از شش تشخیص دهنده وابسته به آوا که بر پایه مدل مخفی مارکوف، مطابق با مدلسازی زبان دنباله صوتی، برای هریک از زبان‌ها استفاده می‌شدند، به عنوان بهترین سیستم معرفی نمود.

:۱۹۹۶

اسکولتز[۴۵] و دوستانش[۴۵]، از سیستم تشخیص گفتار پیوسته، بر پایه مجموعه وسیع لغات[۴۶] استفاده نمودند. آنها، هر دو سیستم تشخیص زبان بر پایه سطح آوا[۴۷] و سطح لغت[۴۸] را با مدل زبانی[۴۹] مقایسه کردند. در نخستین تلاش مدل زبانی یگانه پیاده سازی شد، اما در مرحله بعدی دریافتند با استفاده از

[43] Yan
[44] Duration
[45] Schultz
[46] large vocabulary continuous speech recognition system (LVCSR)
[47] Phone Level
[48] Word Level
[49] Language Model (LM)

سه‌گانه‌ها می‌توان نتایج بهتری به دست آورد. سیستم بر پایه لغت با مدلسازی یگانه، به درصد صحت ۸۴٪ و سیستم بر پایه آوا با مدلسازی سه گانه به درصد صحت ۸۲/۶٪ بر روی چهار زبان رسیده است. آنها ادعا نمودند که در سیستم تشخیص زبان بر پایه لغت، دانش بیشتری موجود بوده و عملکرد بهتری خواهد داشت.

:۱۹۹۹

برکلینگ[50] [۴۶]، راه‌های گوناگونی برای ارزیابی درصد صحت سیستم تشخیص زبان، امتحان نمود. سه روش پیشنهادی توسط او به شرح زیر می‌باشد:

روش اول: امتیازبندی و شمارش بر مبنای برنده: مجموعه هدف، شامل امتیازات گفتارهای به درستی تشخیص داده شده می‌باشد.

روش دوم: امتیازبندی و شمارش بر مبنای ورودی: مجموعه هدف شامل همه امتیازاتی می‌باشد که زبان ورودی و مدل زبانی مطابقت داشته باشند.

روش سوم: در این روش هدف و پس زمینه جدا نمی‌شوند، بلکه همه امتیازات برنده به یک مجموعه تنها داده می‌شود، بدون توجه به اینکه گفتار ورودی درست و یا غلط دسته‌بندی شده است.

این روش از تشخیص‌دهنده آوا بر اساس مدل‌های زبانی[51]، برای ارزیابی استفاده می‌کند.

روش اول عملکرد سیستم را بهتر پیگیری می‌کند. *برکلینگ* همچنین برای بهبود معیار ارزیابی، تأثیر افزایش ویژگی‌های جدید همچون درنگ صوتی، فرکانس رخداد واج و غیره را مورد بررسی قرار داد. آزمایشات بر روی داده‌های پایگاه داده NIST1996 انجام شده است.

[50] Berkling
[51] Phone Recognition followed by Language Modeling (PRLM)

:۱۹۹۹

هومبرت[۵۲] و *مادیسون*[۵۳][۴۷]، استفاده از قسمت‌های کم[۵۴](نادر) را برای سیستم‌های تشخیص زبان پیشنهاد نمودند زیرا قسمت‌هایی که برای تشخیص ساده و کمیابند، در یک سیستم تشخیص زبان به شدت ارزشمندند. همچنین در این مقاله، توصیف جزئی کلاس‌های آواشناسی و ارائه در گروه‌های زبان گوناگون فراهم شده است.

:۲۰۰۰-۲۰۰۳

:۲۰۰۱

ناوراتیل[۵۵][۴۸]، با یک راهکار موفق بر پایه ویژگی‌های صوتی واج‌آرایی، سیستم‌هایی برای تشخیص زبان و همچنین عدم پذیرش زبان‌های ناشناخته ارائه داد. او نشان داد ساختاری با رمزگشایی چند مسیره، مدل‌های واج‌آرایی با استفاده از ساختارهای دوگانه-سه‌گانه را بهبود می‌دهد.

:۲۰۰۲

جایرام[۵۶] و دوستان[۴۹]، یک سیستم تشخیص زبان به صورت تشخیص زیر-لغت موازی[۵۷]، را به عنوان جایگزینی برای سیستم تشخیص صوت موازی[۵۸] متداول، پیشنهاد دادند. تشخیص‌دهنده زیر-لغت[۵۹]، بر پایه قطعه‌سازی خودکار مطابق با دسته‌بندی هر بخش و مدلسازی مدل مخفی مارکوف می‌باشد. PSWR نسبت به PPR تنها بر روی مجموعه آموزشی، با ضریب صحت (۹۰/۲٪)

[52] Hombert
[53] Maddieson
[54] Rare
[55] Navrátil
[56] Jayram
[57] Parallel Sub-Word Recognition(PSWR)
[58] Parallel Phone Recognition (PPR)
[59] Sub-Word Recognition (SWR)

حدود ۴٪ برتری دارد، اما بر روی مجموعه تست به ضریب صحت(۶۲/۳٪) رسیده و حدود ۱٪ بدتر است.

۲۰۰۳:

آدامی[60][۵۰]، پیشنهادهایی از قبیل استفاده از مسیرهای زودگذر برای فرکانس بنیادی، انرژی کوتاه مدت برای قسمت‌بندی و برچسب‌گذاری سیگنال گفتار به یک مجموعه کوچک واحدهای مجزا که برای تشخیص زبان و گوینده مورد استفاده‌اند را مطرح نمود. او همچنین با استفاده از قسمت‌بندی‌هایی که انجام داد، به ویژگی‌های جدیدی دست یافت.

آدامی با ارزیابی سیستم تشخیص زبان پیشنهادی‌اش بر روی NIST2003 به نرخ خطای ۳۵٪ بر روی نمونه‌های گفتاری ۳۰ ثانیه‌ای از ۱۲ زبان رسید. با افزایش مدت زمان، نرخ خطا به ۳۰٪ کاهش یافت. همچنین اطلاعات مکمل را با سیستم بر پایه آوا امتحان نمود که نرخ خطا به ۲۴٪ رسید و با ترکیب این دو سیستم توانست خطا را به ۲۱/۷٪ کاهش دهد.

۲۰۰۳:

گروهMIT[۵۱]، سه راهکار مختلف را بررسی نمودند که شامل تشخیص صوت، مدل ترکیبی گوسین و کلاسه ساز SVM[61] بودند. آنها تفاوت‌ها را خلاصه نموده و از معیار ارزیابی NIST1996 به NIST2003 پیشرفت نمودند. تفاوت اصلی در روش گوسین، استفاده از مدل ترکیبی گوسین وابسته به جنسیت و تکنیک‌های نگاشت ویژگی‌ها به شبکه مستقل از فضای ویژگی بوده است. در سیستم تشخیص زبان بر پایه صوت، از مجموعه واج‌های جدید استفاده شده است.

[60] Adami
[61] Support vector machine

روش‌های بسیاری در کنفرانس‌های بین‌المللی گوناگون برای تشخیص خودکار زبان ارائه شده است. یکی از مشکلات در مقایسه این روش‌ها، نبود یک پایگاه اطلاعات داده همچون(TIMIT) در گذشته بوده است. امروزه با ایجاد سازمان جهانی استاندارد و تکنولوژی(NIST) یک معیار تشخیص زبان برای مقایسه نتایج همه زبان‌ها ایجاد شده و با تلاش گسترده در این زمینه حمایت می‌شود.

در این پایان‌نامه، ضمن بررسی برخی از راهکارهای مورد استفاده و روش‌های قبلی، از روش پیشنهادی در مقاله *ریس-هررا*[۶۲] *و همکاران*[۱۰] که مبتنی بر استفاده از ضرایب موجک است، بهره می‌بریم و آن را به ضرایب کپسترال نیز تعمیم داده و نتایج حاصل را با روش‌های قبلی مقایسه می‌کنیم. مزیت این روش آن است که نیازی به اطلاعات دستور زبان ندارند و تنها با ضرایب موجک و یا ضرایب کپسترال نمونه‌ها سروکار خواهیم داشت.

در ادامه مطالب در فصل سوم مروری بر برخی روش‌های استفاده شده در تشخیص زبان به صورت خودکار خواهیم داشت. در فصل چهارم استخراج ویژگی‌ها را بیان نموده و در فصل پنجم روش مورد استفاده در این پایان نامه را بررسی کرده و مروری بر برنامه نوشته شده داریم، همچنین در فصل آخر نتایج آزمایش‌های انجام شده ومقایسه‌ی بین آنها بیان می‌گردد.

[62] Reyes-Herrera

فصل سوم

مروری بر کارهای انجام شده

فصل سوم:مروری بر کارهای انجام شده

در این فصل برخی روش‌های مورد استفاده برای تشخیص خودکار زبان را بررسی می‌کنیم:

۳.۱) تشخیص زبان با استفاده از خواص عروضی:

عروض، نقش زیادی در تشخیص زبان انسانی به طور صحیح دارد. به طور کلی عروض به معنای "ساختاری است که صوت را منظم می‌سازد". لحن، بلندی و آهنگ ساختارهایی هستند که اجزای اصلی عروض می‌باشند که برای توصیف کمّی آنها باید مشخصه‌های فیزیکی مناسب را پیدا کرد. به طور نوعی این مشخصه‌ها عبارتند از: گام، شدت و مدت زمان نرمال شده هجاها. این خواص باید در طی قاب‌های زمانی ناشی از فرآیند قاب‌بندی زمانی صحبت ارزیابی شوند[۵۳].

برخی هجاها یا کلمات، برجسته‌تر از بقیه به نظر می‌رسند که ناشی از تکیه[۶۳] می‌باشند. اطلاعات جمع‌آوری شده از وزن، زمان و تکیه در صحبت، قابلیت فهم پیغام‌های گفته شده را افزایش می‌دهد و بنابراین اطلاعاتی همچون نواخت لغات[۶۴]، لهجه[۶۵] و هیجانات[۶۶] رامنتقل می‌نماید. خصوصیاتی که ما را قادر به درک این نتایج می‌سازند، همگی به خواص عروضی باز می‌گردند.

ویژگی‌های عروضی در میان زبان‌ها به طور قابل توجهی متفاوتند. سیستم‌های تشخیص زبان مبتنی بر عروض، خواصی نظیر مدت زمان، الگوی گام[۶۷] و الگوی تکیه را در یک زبان استفاده می‌کنند. سیستم‌های تشخیص زبانی که بر پایه ویژگی‌های عروضی عمل می‌کنند نسبت به آنهایی که بر پایه واج‌آرایی و یا مجموعه آوا شناسی می باشند، عملکرد چندان مناسبی ندارند که علت آن عدم توانایی خواص عروضی در مدلسازی زبان‌ها، به تنهایی می‌باشد[۵].

خواص عروضی، برای متون کوتاه، مفیدترند در حالیکه خواص واج‌آرایی برای متن‌های طولانی، بیشتر مورد استفاده قرار می‌گیرند. عروض، نقش کلیدی در ادراک گفتار انسانی بازی می‌کند، بیشتر تحقیقات نشان می‌دهند که خواص عروضی فقط برای زبان‌های نواختی[۶۸] مفیدند[۵۳].

می‌توان بر روی یک گفتار دو متغیر تعریف نمود: نسبت فواصل صوتی صدادار و انحراف استاندارد مدت زمان فواصل بی صدا که به عنوان وابسته‌های وزن شناخته شده‌اند. هر دو این اندازه گیری‌ها به طور مستقیم بر فهرست قسمت‌بندی شده و قواعد واج‌آرایی یک زبان خاص تأثیر دارد.

زبان‌ها می‌توانند بر اساس خواصشان به چندین دسته تقسیم‌بندی شوند:

---

[63] Stress
[64] Lexical tone
[65] Accent
[66] Emotion
[67] Pitch
[68] Tonal

انگلیسی و آلمانی زبان‌های تکیه بند[۶۹] نامیده می‌شوند یعنی مدت زمان هجاها به طور اساسی توسط حضور هجاهای تکیه بر، کنترل می‌شوند که می‌توانند به طور تصادفی رخ دهند و قسمت‌های ثابت (در حوزه زمان) بین دو قسمت هجاهای تکیه بر قرار می‌گیرند. هجاهایی که بین دو هجاهای تکیه بر رخ می‌دهد، کوتاهترند[۵]. فرانسوی و اسپانیایی زبان‌های هجابند[۷۰] می‌باشند که فاصله بین هجاها در طول صحبت همیشه ثابت باقی می‌ماند[۵].

ماندارین و ویتنامی در دسته زبان‌های نواختی دسته بندی می‌شوند. در زبان‌های نواختی نحوه تلفظ یک لغت، معنای آن را عوض می‌کند. زبان‌های نواختی خصوصیات تلفظ بسیار متفاوتی نسبت به زبان‌های تکیه بند(همچون انگلیسی) دارند[۲].

زبان‌ها همچنین می‌توانند به دو دسته لهجه برتکیه[۷۱] و لهجه برگام[۷۲] دسته‌بندی شوند. در زبان‌های لهجه برگام همچون ژاپنی، تنوع هجایی از گوناگونی گام به دست می‌آید، در حالیکه در زبان‌های لهجه برتکیه ،گوناگونی گام، تنها یک فاکتور است که به ایجاد تنوع هجایی کمک می‌کند[۲].

ما می‌توانیم به راحتی زبان‌هایی را که نواخت لغوی را به کار می‌گیرند همچون چینی قدیم و یا اهل ناتال جنوب آفریقا[۷۳] که دارای زبان نواختی هستند و زبان‌هایی شبیه سوئدی و ژاپنی که زبان‌های لهجه بر گام می‌باشند را از زبان‌های لهجه بر تکیه همچون انگلیسی و آلمانی تشخیص دهیم. تعدادی زبان دیگر وجود دارد که از قواعد یک کلاس خاص تبعیت نمی‌کنند.

[69] Stress-Timed
[70] Syllable-Ttimed
[71] Stress-Accent
[72] Pitch-Accent
[73] Zulu

۳.۱.۱) استخراج خودکار خواص عروضی:

راه‌های استخراج خواص عروضی می‌تواند به طور وسیع بر پایه استفاده از تشخیص دهنده خودکار صحبت(ASR)[۷۴] به دو دسته تفکیک گردد[۵]:

۱) راه مبنی بر استفاده از تشخیص دهنده خودکار صحبت[۷۵]

۲) راه مستقل از استفاده تشخیص دهنده خودکار صحبت[۷۶]

راه اول، از مرزبندی‌های به دست آمده از تشخیص دهنده خودکار صحبت، برای استخراج خواص عروضی استفاده می‌کند، اما برای کاربردهایی همچون تشخیص زبان، نیازی به استفاده از آن نیست. در راه دوم از نقاط عطف، شروع و پایان منحنی گام صوت، برای جداسازی استفاده شده است.

روش پیشنهادی توسط *ماری*[۷۷] و همکاران[۵] از مکان نقاط آغاز واکه(VOP)[۷۸]، برای تشخیص نواحی شبه‌هجا در گفتار پیوسته استفاده می‌کند. در این روش، از راهکارهای ارتباط با الگوی هجایی موجود در روش اول و استخراج خواص بدون استفاده از تشخیص دهنده خودکار صحبت، ارائه شده در روش دوم استفاده شده است.

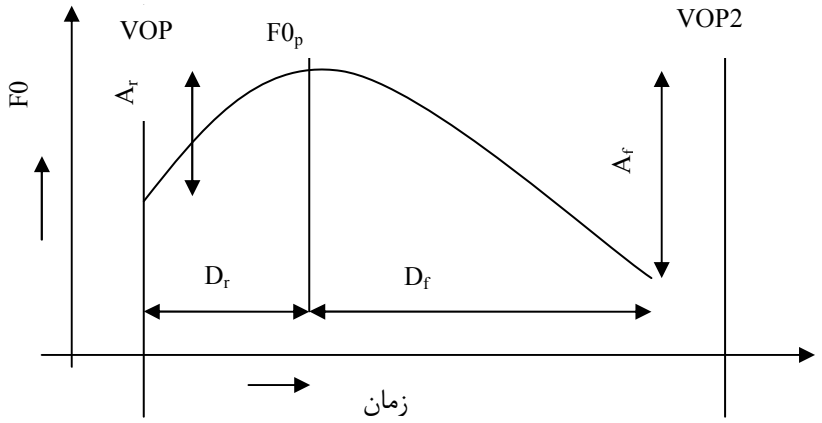
۳.۱.۲) نمایش خواص عروضی:

در این قسمت نحوه نمایش برخی از خواص عروضی را بیان می‌نماییم:

[74] Automatic Speech Recognizer (ASR)
[75] ASR-based
[76] ASR-free
[77] Mary
[78] Vowel Onset Points

۳.۱.۲.۱) نمایش آهنگ:

منحنی فرکانس F_0، بین دو نقطه آغاز واکه متوالی چنانچه در(شکل۳-۱) نشان داده شده متناظر با جابجایی فرکانس در یک ناحیه شبه‌هجا است و همچون یک قسمت از منحنی F_0 رفتار می‌کند. تغییرات F_0 برای چنین قسمتی می‌تواند یک صعود یا سقوط و یا در بیشتر موارد یک صعود همراه با سقوط داشته باشد. حدس می‌زنیم که تغییرات پیچیده تر F_0 در حدود قطعه بعید است.



(شکل ۳-۱) قسمتی از منحنی فرکانس F0.[۵]

با مراجعه به (شکل۳-۱)، شیب پارامترها، به ویژه شیب دامنه (A_t) و شیب طول زمان (D_t) برای یک قسمت شمارنده F_0 به صورت زیر تعریف شده اند:

$$A_t = \frac{A_r - A_f}{A_r + A_f} \qquad (۱-۳)$$

$$D_t = \frac{D_r - D_f}{D_r + D_f} \qquad (۲-۳)$$

که A_r وA_f به ترتیب صعود و نزول در دامنه F_0 را با توجه به مقادیر قله[۷۹] فرکانس بنیادی F_{0p} بیان می‌کنند. به طور مشابه D_f, D_r به ترتیب مدت زمان صرف شده برای صعود و نزول را بیان می‌نمایند. مطالعات نشان می‌دهد که در زبان‌های لهجه بر گام، گوینده می‌تواند از تغییر ارتفاع قله فرکانس بنیادی، برای بیان درجات مختلف صدا استفاده کند[۵].

برای بیان ارتفاع قله F_0، از اختلاف بین قله و دره فرکانس بنیادی استفاده می‌شود. بنابراین طول قله F_0، با استفاده از(فرمول۳-۳) به دست می‌آید[۵]:

$$\Delta F_0 = F_{0P} - F_{0V} \qquad (۳-۳)$$

به طور کلی خواص طرز بیان برای این روش تشخیص زبان به صورت زیر مورد استفاده است:

الف) تغییر در F_0 (ΔF_0) ب) فاصله نقطه F_0 نسبت به VOP (Dp)

ج) شیب دامنه (At) د) شیب فاصله زمانی (Dt)

۳.۱.۲.۲) نمایش وزن:

وزن موجود در گفتار، به دلیل باز و بسته شدن تارهای صوتی در هجاهای متوالی می‌باشد. نسبت فواصل صوتی در طی هر منطقه هجا، مقداری از این انتقال را می‌دهد. گفتار پیوسته تفکیک شده به واحدهای شبه‌هجا، قادر به بیان خصوصیات وزنی می‌باشند. برای بیان وزن، از طول مدت هجا (Ds)، تخمین زده شده برای فواصل بین نقاط آغاز واکه‌های متوالی و طول مدت ناحیه صوتی (Dv) استفاده می‌کنیم.

[79] Peak

پس به طور کلی خصوصیات زیر برای بیان وزن به کار می‌روند:

الف) طول مدت سیلاب (Ds)

ب) طول مدت ناحیه صوتی(Dv) طی هر هجا.

۳.۱.۲.۳) نمایش تکیه:

هجای حامل تکیه نسبت به هجاهای اطراف به علت رسایی(بلندی‌اش)، طول مدت بلندترش و همچنین جابجایی بیشتر F_0 مشخص است. بنابراین برای بیان تکیه می‌توان به همراه فرکانس F_0 ومدت زمان، تغییرات لگاریتم انرژی را نیز در حدود ناحیه صوتی به کار برد.

۳.۱.۲.۴) نمایش عروضی:

نواخت[۸۰] هجاهای مجاور، بر روی شکل و بلندای منحنی فرکانس F_0 هجای خاص، تأثیر دارد و برتری یک هجا، بر پایه ویژگی‌های گام و منحنی‌های اطراف آن تخمین زده می‌شود. به طور مشابه وزن، توسط دسته‌ای از هجاها ساخته شده است و یک هجا به تنهایی نمی‌تواند به وزن وابسته باشد. بنابراین دینامیک وابسته به این پارامترها، هنگام بیان تنوع‌های عروضی میان زبان‌ها مهم اند.

محتوای یک هجا، علاوه بر خصوصیات تقدم و تأخر هجاها، برای بیان ویژگی‌های عروضی یک زبان مورد استفاده است. از آنجا که اثر متقابل بین ضرب‌های[۸۱] گام، انرژی و مدت زمان، نقش مهمی را در تعیین عروض ایفا می‌نماید، این پارامترها با هم برای شکل‌دهی یک بردار خاصیت عروضی استفاده می‌شوند[۵].

[80] Tone
[81] Movements

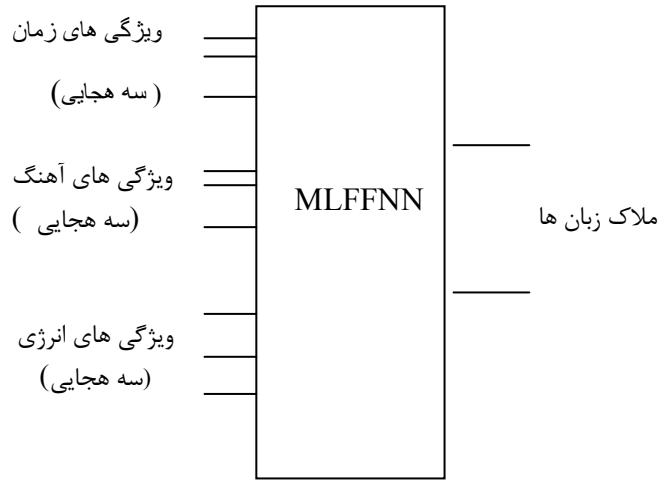
۳.۱.۳)نتیجه مطالعات آزمایشی خواص عروضی:

برای نمونه‌های آموزشی خواص عروضی، *ماری*[۸۲] و همکاران از ۴۰ گوینده مختلف به صورت آنی(بدون پیش زمینه) نمونه صوتی گرفتند که طول هر نمونه حدوداً ۴۵ ثانیه است[۵]. همچنین برای ارزیابی این روش از ۲۰ نمونه گفتاری گویندگان مختلف استفاده شده است.

در طی آزمایش برای هر بردار عروضی در گفتار تست، ملاک تفاوت زبان در خروجی، کلاسه‌کننده MLFFNN است. بنابراین ملاک‌ها برای همه بردارهای خواص عروضی در گفتار تست به دست می‌آیند که برای به دست آوردن درصد صحت برای هر زبان متوسط گیری شده‌اند.

ساختار MLFFNN به این صورت می‌باشد:

21L 64N 64N 2N که L واحدهای با تابع فعالیت خطی را بیان می‌کند و N بیانگر توابع فعال غیر خطی و بیان عددی تعداد واحد در لایه‌ها می‌باشد.



(شکل ۳-۲) دسته بندی شبکه عصبی برای زبان با استفاده از عروضی. [۵]

[82] Mary

با مقایسه نتایج حاصل از روش عروضی با روش‌های مورد استفاده توسط *رواس*[۸۳][۵۴] و *لین*[۸۴][۹] مشاهده می‌شود که خواص عروضی برای تفکیک زبان‌هایی مؤثرترند که بر پایه خواص نواخت یا وزن می‌باشند[۵]. برای مثال ژاپنی و ماندارین به خوبی از زبان‌های دیگر متمایزند در حالیکه تمایز بین خود آنها بسیار دشوار است.

به هنگام استفاده از خواص عروضی، سکوت که هیچ اطلاعات دیگری جز نویز نمی‌دهد، مشکل‌ساز است. چرا که در قسمت‌های سکوت، شمارش تعداد عبور از صفرها، گام تخمین زده شده بر اساس *مدل مخلوط گوسی* GMM[۸۵] را تحریف خواهد کرد و یک مدل غیر صحیح تولید می‌نماید. یک جداساز گام بر پایه سکوت، راه حل‌های ارائه شده را بهبود خواهد داد[۵۵].

۳.۲) نقش ترکیب خواص عروضی و کپسترال در شناسایی زبان:

با ترکیب خواص عروضی و مشخصه‌های کپسترال راهی جدید ارائه شده است که برای رفع مشکلات ناشی از سکوت و اطلاعات وابسته به تکنیک‌های استخراج، استفاده می‌شود. مزیت این روش آن است که بهبود مؤثر را در شناسایی هر دو نوع زبان نواختی و غیر نواختی نشان می‌دهد[۵۳].

مشخصه‌های کپسترال، که به طور نوعی مقدار خواص طیف صوت را بیان می‌نمایند، به طور وسیع در پردازش صوت مورد استفاده‌اند[۵۳]. انتخاب مشخصه‌های مؤثر برای دستیابی به عملکرد بالاتر، مهم است، زیرا مشخصه‌های متفاوت کپسترال و ضرایب مختلف منجر به عملکرد متفاوت خواهند شد. مشخصه‌هایی همچون ضرایب MFCC[۸۶] و PLP[۸۷]، مشخصه‌هایی هستند که بیشتر در سیستم‌های تشخیص خودکار زبان مدرن استفاده می‌شوند. البته هنوز تحقیقات در زمینه چگونگی تأثیر تعداد

[83] Rouas
[84] Lin
[85] Gaussian Mixture Model
[86] Mel-Frequency Cepstral Coefficients
[87] Perceptual Linear Predictive

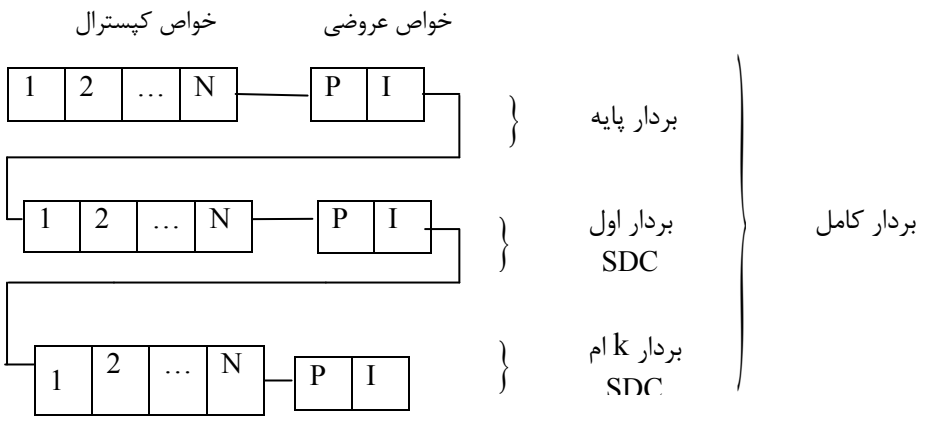
ضرایب بر روی عملکرد تشخیص خودکار زبان ادامه دارد. برخی از تحقیقات نشان می‌دهند که به طور معمول از ۱۲ ضریب MFCC و یا ۹ ضریب PLP استفاده می‌شود[۵۳].

جستجو نشان می‌دهد که سیستم‌ها با MFCC به عملکرد بهتری نسبت به PLP می‌رسند[۵۶]. از آنجا که پردازش PLP و MFCC مشابه اند، اطلاعات به دست آمده از آنها نیز مشابه خواهد بود و به همین دلیل ترکیب این دو خاصیت، بهبود مؤثری ایجاد نمی‌کند، اما اطلاعات موجود در خواص عروضی نسبتاً متفاوت از اطلاعات موجود در خواص کپسترال می‌باشد و بنابراین می‌توان با ترکیب آنها به بهبودی مؤثر برسیم.

۳.۲.۱) خاصیت ترکیب:

تکنولوژی ترکیب که نتایج شبه‌سیستم‌های مختلف را با هم ترکیب می‌نماید به طور گسترده در سیستم‌های تشخیص زبان مدرن استفاده می‌شود[۵۷]. در یک سیستم نوعی بر پایه ترکیب، هر مجموعه خواص به طور مجزا برای ایجاد یک مدل مستقل استفاده می‌شود. این مدل‌ها در شبه‌سیستم‌های مختلف به کار می‌روند. با مشاهده تئوری آماری، مشخص است که خواص با تعداد عضو کمتر، به یک مدل آزمایشی با تعداد عضو کمتر منجر می‌شوند. این بدان معناست که بیشتر نمونه‌ها برای آموزش مدل‌های جداگانه، نیازمند رسیدن به پایداری قابل قبول می‌باشند. متاسفانه خواص عروضی تنها دو عضو را در این مورد شامل می‌شوند. خاصیت الحاق، یک راه حل ساده برای اجتناب از این مشکل است.

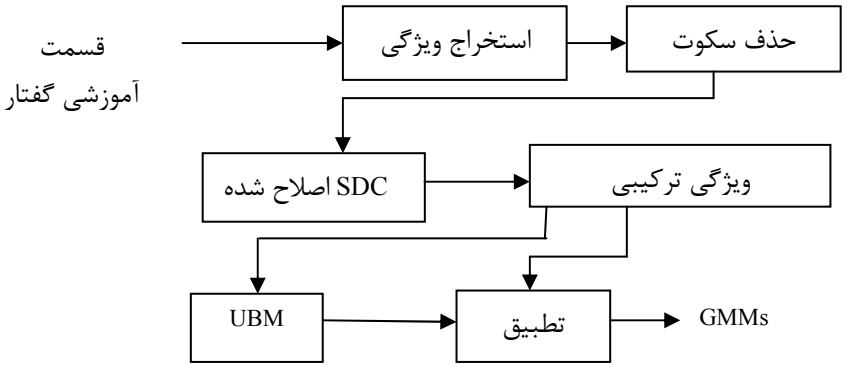
(شکل۳-۳) ساختار برداری خاصیت ترکیب پیشنهادی را نشان می‌دهد:



(شکل ۳-۳) ساختار بردار خواص ترکیبی [۵۳]

با این روش، خواص کپسترال و عروضی مستقیماً به صورت یک بردار تک خاصیته مدل می‌شوند، که همه اطلاعات را از هر خاصیت ارائه می‌دهد. بردار کامل توسط الحاق این بردارهای خاصیت توسط پردازش دلتا کپستروم شیفت‌یافته[۸۸]تولید شده‌اند.

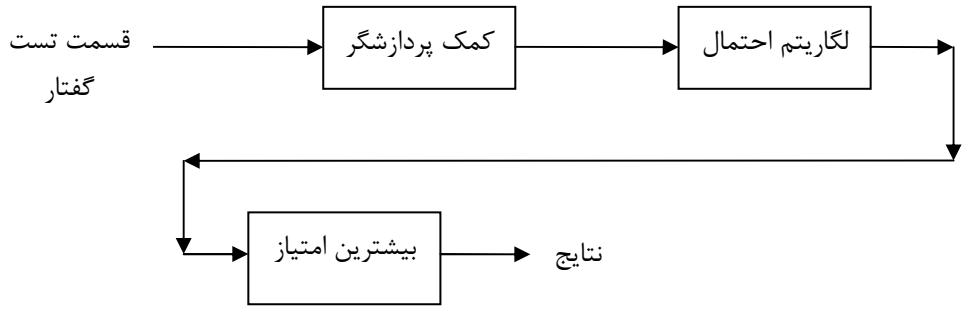
نحوه پردازش داده های آموزشی در (شکل۳-۴) نشان داده شده است. در شکل مشخص است که داده‌های آموزشی از داده گفتار استخراج شده و سپس برای افزودن اطلاعات وابسته توسط SDC اصلاح شده، گسترش یافته اند.



(شکل ۳-۴) پردازش داده های آموزشی [۵۳]

[88] Shifted Delta Cepstrum(SDC)

همچنین پردازش داده‌های تست در (شکل۳-۵) مشخص شده است:



(شکل ۳-۵) پردازش داده های تست[۵۳]

به هنگام آزمایش، خواص گفتار هدف استخراج شده‌اند و با هر GMM مقایسه می‌شوند. زبان با بیشترین احتمال به عنوان نتیجه قطعی است.

چنانچه در (جدول۳-۱) نشان داده شده‌است، تغییر در تعداد ضرایب، دقت تشخیص نهایی را برای هر دو MFCC و PLP تغییر می‌دهد.

(جدول ۳-۱) نتایج از ویژگی های کپسترال گوناگون [۵۳]

درصد صحت	ویژگی‌ها	درصد صحت	ویژگی‌ها
۷۷/۴	ترکیب ۵ PLP	۷۳/۴	ترکیب ۵ MFCC
۷۸/۲	ترکیب PLP۷	۷۷/۸	ترکیب ۷ MFCC
۷۷/۰	ترکیب ۹ PLP	۷۶/۶	ترکیب ۹ MFCC
۷۷/۴	ترکیب ۱۱ PLP	۷۵/۴	ترکیب ۱۲ MFCC
۷۲/۶	ترکیب ۱۳ PLP		

نتایج (جدول۳-۱) می‌تواند به این صورت مطرح شود:

با افزایش اعداد ضریب، اطلاعات گفتار بیشتری استخراج می‌شود، هرچند نویز بیشتری نیز به وجود می‌آید، بنابراین یک موازنه بین این دو اثر وجود دارد. همچنین یک متوسط‌گیری بر روی MFCC و PLP عملکرد بسیار مشابهی را در تشخیص زبان نشان می‌دهد.

از آنجا که خواص PLP و MFCC بادقت مشابه به دست می‌آیند، آزمایشات بر روی ترکیب خواص عروضی با هر دو خاصیت PLP و MFCC انجام شده است. (جدول۳-۲)، درصد صحت را برای هر دو حالت مجموعه خواص کپسترال تنها و مجموعه خواص عروضی ترکیبی با کپسترال، با تعداد ضرایب مختلف نشان می‌دهد.

(جدول ۳-۲) ترکیب خواص عروضی با خواص کپسترال [۵۳]

ترکیب کپسترال با عروضی	ضرایب کپسترال تنها	ضرایب
۸۴/۷	۷۶/۶	ترکیب MFCC۹
۸۰/۲	۷۵/۴	ترکیب MFCC ۱۲
۸۱/۹	۷۷/۴	ترکیب PLP ۵
۸۵/۵	۷۸/۲	ترکیب PLP ۷
۸۰/۶	۷۷/۰	ترکیب PLP ۹

از (جدول ۳-۲) مشخص است که بالاترین عملکرد توسط ترکیب ۷ ضریب MFCC با خواص عروضی به دست می‌آید که ۱/۲% بهبود را ایجاد نموده است[۵۳]. اما هنگامیکه عدد ضرایب MFCC از ۷ کاهش می‌یابد نرخ تصحیح به شدت افت می‌کند، در حالیکه نرخ تصحیح ترکیبی، به آرامی افت می‌نماید. این توضیح می‌دهد که هنگامیکه اعداد ضریب کلی کوچکتر می‌شوند، خواص عروضی

مشارکت بیشتری دارند، زیرا وزن خواص عروضی بزرگتر می‌شود، در حدود ۲/۷، هنگام ترکیب با ۵ ضریب MFCC و ۲/۹ به هنگام ترکیب با ۷ ضریب MFCC[۵۳]. این نتیجه آشکار می‌سازد که ترکیب خواص عروضی با خواص کپسترال، بهبودی قابل توجه در عملکرد کلی ایجاد می‌کند. درصد صحت در سیستم تشخیص زبان، برای مجموعه ۱۰ زبانه، با استفاده از ترکیب ۷ ضریب MFCC ، از ۷۷/۸% به ۸۷/۱% افزایش یافته است.

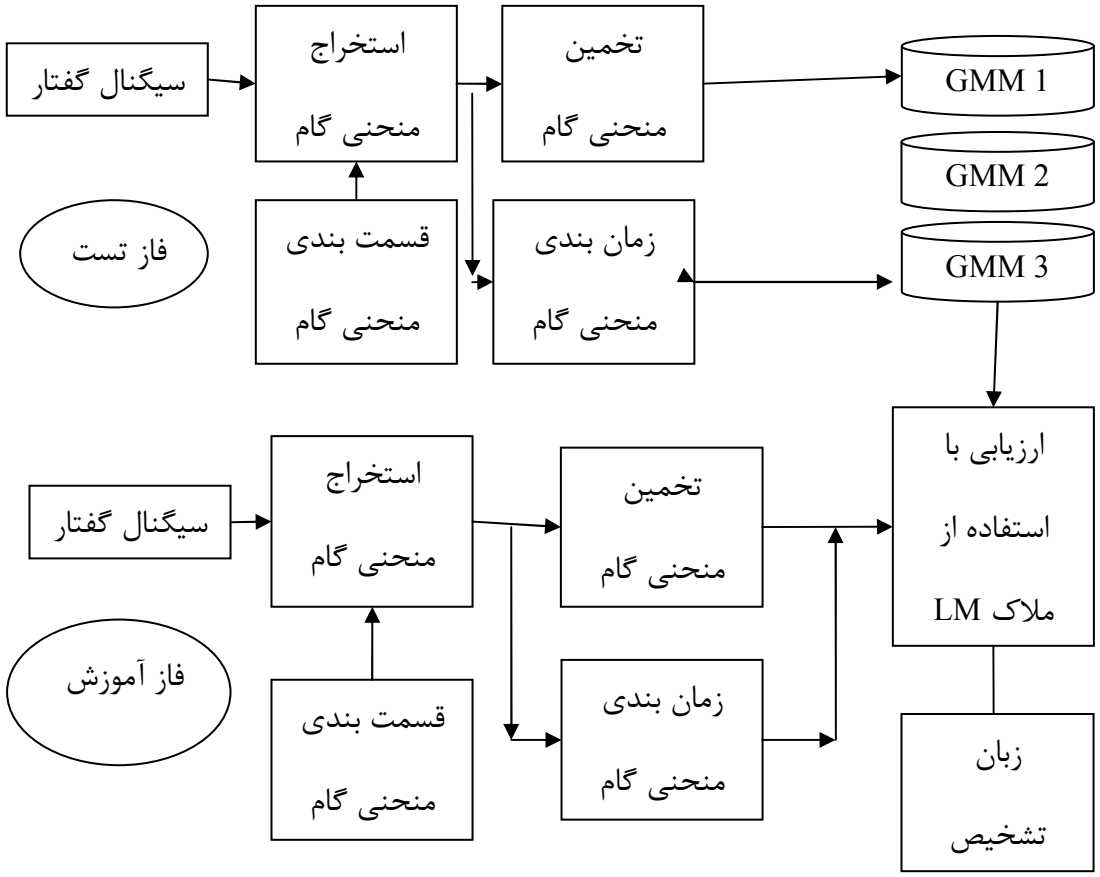
۳.۳) شناسایی زبان با استفاده اطلاعات منحنی گام:

در این قسمت راهی برای تشخیص خودکار زبان، با استفاده از اطلاعات منحنی گام، توسط مجموعه‌ای از علائم چند جمله ای لژاندر تخمین زده شده‌است که ضرایب چند جمله ای، یک بردار خاصیت را برای بیان منحنی گام تشکیل می‌دهند. در این روش مدل مخلوط گوسین[۸۹] بر پایه بردارهای خواص استخراج شده از منحنی گام، پایه‌گذاری شده‌اند. آزمایشات نشان می‌دهد که تنها ۲ یا ۳ ضریب برای به دست آوردن نرخ شناسایی مناسب لازم است. ما همچنین می‌فهمیم که طول منحنی گام تکه‌ای، یک خاصیت مهم دیگر برای تشخیص زبان است و بنابراین باعث بهبود بیشتر در عملکرد تشخیص زبان می‌شود[۹].

۳.۳.۱) توضیحات سیستم:

بلوک دیاگرام سیستم تشخیص زبان پیشنهادی توسط *لین* در(شکل ۳-۶) نشان داده شده است:

[89] GMM



(شکل ۳-۶) ساختار سیستم تشخیص زبان عروضی[۵۸]

در فاز آموزشی، روش استخراج گام مورد استفاده، روشی است که توسط *بورسما*[۹۰] [۵۸]پیشنهاد شده که برای پیدا کردن حد فاصل گام به کار رفته است. پس از استخراج، حد فاصل گام با طول مدت طولانی، آن را به قسمت‌های کوچکتر تقسیم‌بندی نموده، سپس هر قسمت، توسط یک چندجمله‌ای لژاندر تخمین زده می‌شود.

[90] Boersma

این ضرایب همراه با طول منحنی گام تکه‌ای، برای ساخت یک بردار خاصیت استفاده شده‌اند، سپس این بردار خاصیت برای آزمایش یک مدل مخلوط گوسین برای هر زبان مورد استفاده است. در طی شناسایی زبان‌های دوبه دو[۹۱]، امتیاز لگاریتم احتمال برای هر مدل زبان محاسبه می‌شود، سپس زبان با بالاترین درجه، زبان فرضی است. اما همه خواص استخراجی مفید نیستند و برخی خواص، عملکرد کلی را تنزل می‌دهند.

در قسمت زیر هر بلوک با جزئیات شرح داده شده است:

۳.۳.۱.۱) استخراج منحنی گام:

استخراج منحنی گام، برای کمک به برنامه *پرات*[۹۲] [۵۸] مهم است. روش استخراج گام مورد استفاده در این قسمت، روشی است که توسط *بورسما* [۵۸]پیشنهاد شده که برای پیدا کردن منحنی گام به کار رفته است. این روش تابع خودهمبستگی[۹۳] را برای آشکارسازی قسمت‌های صوتی و یافتن کاندیداهای گام استفاده می‌کند. سپس از الگوریتم *ویتربای*[۹۴], برای یافتن مناسب‌ترین منحنی استفاده شده است. مقدار برخی پارامترهای مورد استفاده، در (جدول۳-۳) لیست شده‌اند:

[91] Pair-Wise
[92] Praat
[93] Autocorrelation
[94] Viterbi

(جدول ۳-۳) پارامترهای استخراج منحنی گام موجود در برنامه Praat. [۵۸]

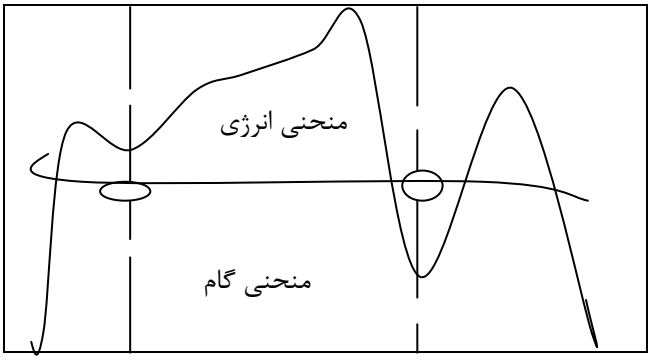
Pitch extraction parameter settings	
Analysis window length	30 ms
Analysis window time step	10 ms
Pitch floor(Hz)	50
Pitch Ceiling(Hz)	500
Max number of candidates	5
Silence threshold	0.03
Voicing threshold	0.6
Octave cost	0.01
Octave-jump cost	0.6
Voiced/Unvoiced cost	0.14

۳.۳.۱.۲) قسمت‌بندی منحنی گام:

پس از استخراج، حد فاصل گام با طول مدت طولانی، آن را به قسمت‌های کوچکتر تقسیم‌بندی نموده، سپس هر قسمت، توسط یک چندجمله‌ای لژاندر تخمین زده می‌شود. در گفتار آنی، بخش صدادار سیگنال صوت ممکن است در میان هجا و یا مرزهای لغت باشد.

برخی قسمت‌های منحنی گام استخراج شده، تا حد زیادی بلندند. بر اساس قسمت‌بندی منحنی گام بلند به بخش های کوتاه‌تر، نخست مرز منحنی گام را با مرز انرژی مطابق با آنچه در(شکل ۲-۷) نشان داده شده، تنظیم می‌کنیم. کاندیدای نقطه انتهایی یک قسمت، نقاط دره مرزهای انرژی آن هستند. البته باید توجه شود که محدودیت مدت باید بر اساس اجتناب از ایجاد یک قسمت بسیار کوتاه تنظیم شود. محدودیت مدت در اینجا ۵۰ میلی ثانیه می‌باشد.

چنانچه در(شکل۳-۷) نشان داده شده، دو نقطه به عنوان کاندیدا انتخاب شده‌اند. تنها از کاندیدای دوم برای قطعه بندی استفاده می‌شود زیرا اولین کاندیدا فاصله بسیار کوتاهی ایجاد می‌کند که کمتر از ۵۰ میلی‌ثانیه می‌باشد، بنابراین از اولین کاندیدا صرف نظر می‌کنیم.



(شکل ۳-۷) قسمت بندی منحنی گام[۵۸]

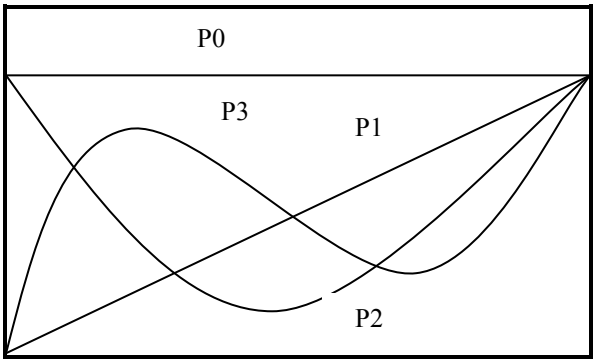
۳.۳.۱.۳) تخمین منحنی گام:

برای هر قسمت مرز گام ،f_k ، می‌توان آن را با چند جمله‌ای لژاندر درجه Mام در شاخه کمترین خطای میانگین مربع تخمین زد:

$$\hat{f}_k = \sum_{i=0}^{M} a_{ik} P_i \qquad (۴-۳)$$

که K شاخص مرز گام، M بالاترین درجه چند جمله‌ای، a_{ik} ،i امین درجه ضریب و p_i ، i امین درجه چند جمله ای لژاندر می‌باشد. در بیشتر موارد، مقادیر کوچک M کافی است، در اینجا۳ = M قرار داده شده است.

چند جمله ای لژاندر p_i در (شکل ۳-۸) نشان داده شده است:



(شکل ۳-۸) نمایش چندجمله ای لژاندر. [۵۸]

p_0 برای بالاترین مرز گام، p_1 برای شیب مرز گام، p_2 برای انحنای مرز گام و p_3 برای انحنا-S مرز گام به کار می‌رود. با این بیان یک بردار خاصیت $\overrightarrow{v_k}$ ساخته شده است که شامل طول مرز گام، D_k و ۴ ضریب $a_{0k}, a_{1k}, a_{2k}, a_{3k}$ است. اما خواهیم دید که همه این خواص مفید نیستند.

۳.۳.۲)مدل مخلوط گوسی و ارزیابی

برای هر زبانL، یک GMM، λ_ℓ ایجاد شده است. تحت فرض GMM، احتمال یک بردار خاصیت $\overrightarrow{v_k}$ استخراجی از مدل λ_ℓ به نمایندگی از یک جمع وزنی چگالی گوسین چند متغیره می‌باشد:

$$P\left(\vec{v}_K \mid \lambda_\ell\right) = \sum_{i=1}^{N} w_i . b_i\left(\vec{v}_K\right) \qquad (۵-۳)$$

که $b_i\left(\overrightarrow{v_k}\right)$، چگالی ترکیب مولفه ها و w_i وزن‌های ترکیبی می‌باشند.

مدل زبان λ_ℓ به این صورت بیان شده است:

$$\lambda_\ell = \{w_i, \mu_i, \Sigma_i\} \quad (۶-۳)$$

که i ، شاخص ترکیب است.

مدت زمان شناسایی یک متن گفتار ناشناخته توسط یک رشته بردارهای خاصیت بیان شده است. سپس لگاریتم احتمال، L_ℓ به این صورت تعریف می‌شود:

$$L_\ell = \sum_{K=1}^{K} \log\ p\left(\vec{\nu}_K \mid \lambda_\ell\right) \quad (۷-۳)$$

که k شاخص مرز گام و K تعداد کل مرزهای گام دریک گفتار است. نهایتاً یک کلاسه کننده، زبانی که بیشترین احتمال ، $\hat{\ell}$ را داشته باشد به عنوان زبان گفتار تعیین می‌کند در حالیکه:

$$\hat{\ell} = \arg \max_{1 \le \ell < 2} L_\ell \quad (۸-۳)$$

این ضرایب همراه با طول منحنی گام تکه‌ای، برای ساخت یک بردار خاصیت استفاده شده‌اند، سپس این بردار خاصیت برای آزمایش یک مدل مخلوط گوسین برای هر زبان مورد استفاده است. در طی شناسایی زبان‌های دوبه دو، امتیاز لگاریتم احتمال برای هر مدل زبان محاسبه می‌شود، سپس زبان با بالاترین درجه، زبان فرضی است.

آزمایشات *لین* برای تشخیص دوبه دوی زبان‌ها بر روی OGI-TS انجام شده است. برای هر زبان ۵۰ گوینده در مجموعه آموزشی و ۲۰ گوینده در مجموعه تست برای ارزیابی عملکرد سیستم مورد استفاده قرار گرفته است. در اولین آزمایش سعی شده است خواص مؤثر تفکیک شوند. آزمایش اولیه بر روی متن‌های ۳ ثانیه‌ای با تعداد ترکیبات متفاوت انجام شده که نتایج در (جدول۳- ۴) نشان داده شده است.

(جدول ۳-۴) نتایج آزمایشات اولیه بر روی گفتارهای ۳ ثانیه ای متوسط گیری شده بر روی ۴۵ جفت آزمایشی [۵۳].

تعداد ترکیب / ویژگی‌ها	۴	۸	۱۶	۳۲	۶۴	۱۲۸
$a_0a_1a_2a_3$	۴۴/۳	۴۴/۶	۴۲/۹	۴۲/۵	۴۲/۹	۴۳/۱
$a_1a_2a_3$	۴۵/۲	۵۶/۶	۵۶/۹	۵۶/۲	۵۳/۱	۵۴/۱
a_1a_2	۵۱/۷	۵۸/۳	۵۹/۸	۵۸/۳	۵۸/۵	۵۶/۸
Δa_1a_2	۵۲/۴	۵۹/۱	۵۸/۰	۶۱/۰	۶۲/۹	۶۲/۰
$Da_1a_2\Delta E$	۵۷/۵	۵۹/۴	۵۶/۱	۵۹/۸	۵۹/۱	۶۰/۱

ΔE در (جدول۳-۵)، میانگین اختلاف لگاریتم انرژی محاسبه شده بر روی قسمت‌های مشابه استخراج شده توسط منحنی گام می‌باشد. از جدول مشخص است که همه خواص استخراجی مفید نیستند و برخی خواص، عملکرد کلی را تنزل می‌دهند. خواص مفید a_{1k}, a_{2k} و D_K هستند.

نتایج تشخیص زبان بر روی ۵ زبان مورد استفاده در مقاله *گومینز* در (جدول۳-۵) و (جدول ۳-۶) نشان داده شده است. در ارزیابی از گفتارهای ۱۰ ثانیه‌ای و ۴۵ ثانیه‌ای استفاده شده است:

(جدول ۳-۵) ماتریس مقایسه برای تشخیص زبان نمونه های ۱۰ثانیه ای برای ۵ زبان. نتایج آزمایشات *گومینز* برای مقایسه در کروشه آورده شده است. [۵۸]

۱۰ثانیه‌ای	آلمانی	اسپانیایی	ژاپنی	ماندارین
انگلیسی	۶۱ [۵۶]	۴۷ [۵۰]	۸۵ [۶۳]	۷۴ [۶۳]
آلمانی	-	۴۷ [۵۴]	۸۵ [۶۹]	۷۱ [۶۹]
اسپانیایی	-	-	۷۷ [۶۰]	۶۶ [۶۲]
ژاپنی	-	-	-	۷۲ [۵۰]

(جدول ۳-۶) ماتریس مقایسه برای تشخیص زبان نمونه های ۴۵ثانیه ای برای ۵ زبان. .نتایج آزمایشات *گومینز* برای مقایسه در کروشه آورده شده است. [۵۸]

۴۵ثانیه‌ای	آلمانی	اسپانیایی	ژاپنی	ماندارین
انگلیسی	۵۶ [۵۵]	۵۳ [۵۲]	۸۴ [۶۲]	۷۶ [۶۲]
آلمانی	-	۴۹ [۵۴]	۷۷ [۷۲]	۸۴ [۷۰]
اسپانیایی	-	-	۸۱ [۷۱]	۷۱ [۶۳]
ژاپنی	-	-	-	۷۸ [۴۴]

همچنین در (جدول ۳-۷) نتایج تشخیص زبان بر روی ۹ زبان با نمونه‌های ۴۵ ثانیه‌ای نشان داده شده است.

نتایج به ترتیب از گفتارهای ۳ ثانیه‌ای، ۱۰ ثانیه‌ای و ۴۵ ثانیه‌ای ساخته شده اند. همچنین نتایج آزمایشات *رواس* بر روی گفتارهای ۴۵ ثانیه‌ای برای مقایسه داخل کروشه آورده شده است.

با مقایسه روش *لین*، با روش‌های ارائه شده توسط *گومینز*[۹۵] [۵۹] و *رواس*[۵۴] مشاهده می‌شود روش فوق عملکرد بهتری را بر روی زبان‌های نواختی و زبان‌های لهجه بر گام دارد اما خطای کوچکی بر روی سایر موارد ایجاد می‌کند[۹].

زبان‌های نواختی در این سیستم خوب عمل می‌نماید که دلیل آن تنوع گام این نوع زبان‌ها است که می‌تواند به خوبی با روش فوق تحلیل شود. عملکرد بالای ژاپنی و فارسی به دلیل الگوی لهجه برگام آنها می‌باشد، در مقایسه زبان‌های تکیه بند و هجابند، به نسبت در سیستم فوق ضعیف عمل می‌کنند. این بدان معناست که اطلاعات وابسته به زبان این نوع از زبان‌ها، با مرزهای منحنی گام، بررسی نمی‌شوند.

[95] Cummins

(جدول ۳-۷) ماتریس مقایسه برای ۱۰ زبان برای گفتارهای ۳ ثانیه ای، ۱۰ ثانیه ای و ۴۵ ثانیه ای.(نتایج آزمایشات *رواس* برای مقایسه داخل کروشه آورده شده است). [۵۸]

۱۰،۳،۴۵ ثانیه‌ای	آلمانی	فرانسوی	اسپانیایی	ماندارین	ویتنامی	ژاپنی	کره ای	تامیل	فارسی
انگلیسی	۵۶-۶۱-۴۹ [۶۰]	۵۴-۴۴-۴۸ [۵۲]	۵۳-۴۷-۴۷ [۶۸]	۷۶-۷۴-۵۸ [۷۵]	۸۰-۵۸-۶۲ [۶۸]	۸۴-۸۵-۶۲ [۶۸]	۷۵-۵۹-۵۸ [۷۹]	۶۴-۵۳-۵۳ [۷۷]	۶۲-۵۴-۳۹ [۷۶]
آلمانی	-	۴۲-۴۴-۴۴ [۵۶]	۴۹-۴۷-۵۳ [۵۹]	۸۴-۷۲-۵۸ [۶۲]	۶۹-۵۹-۶۲ [۶۶]	۷۷-۸۵-۶۳ [۶۶]	۶۵-۶۱-۵۹ [۷۱]	۵۹-۵۸-۵۳ [۷۰]	۷۳-۶۴-۴۰ [۷۲]
فرانسوی			۵۷-۵۷-۵۱ [۶۴]	۶۹-۶۹-۶۴ [۶۱]	۷۶-۶۴-۵۸ [۵۸]	۶۵-۸۱-۶۲ [۵۶]	۵۴-۶۰-۶۹ [۵۵]	۴۴-۴۵-۵۹ [۶۰]	۸۷-۷۴-۵۴ [۶۹]
اسپانیایی	-	-	-	۷۱-۶۶-۶۶ [۸۱]	۶۱-۶۵-۶۵ [۶۲]	۸۱-۷۷-۶۴ [۶۳]	۵۹-۵۷-۶۳ [۷۶]	۴۸-۴۸-۵۰ [۶۵]	۷۳-۶۲-۳۶ [۶۷]
ماندارین	-	-	-	-	۷۹-۷۵-۶۴ [۵۰]	۷۸-۷۲-۷۱ [۵۰]	۸۰-۶۷-۶۶ [۷۴]	۶۹ -۷۱-۴۷ [۷۴]	۸۲-۷۷-۵۲ [۷۶]
ویتنامی	-	-	-	-	-	۸۹-۸۵-۷۷ [۶۹]	۷۳-۷۰-۶۵ [۵۶]	۷۷-۶۹-۵۷ [۷۱]	۶۹-۷۱-۴۷ [۶۷]
ژاپنی	-	-	-	-	-	-	۷۵-۸۰-۶۱ [۶۶]	۷۹-۸۴-۶۸ [۵۹]	۸۵-۸۵-۵۵ [۶۷]
کره ای	-	-	-	-	-	-	-	۵۸-۶۱-۵۶ [۶۲]	۷۰-۶۵-۴۷ [۷۵]
تامیل	-	-	-	-	-	-	-	-	۷۱-۶۱-۴۶ [۷۰]

۳.۴) شناسایی زبان با استفاده از شناسایی واج و مدل‌سازی واج‌آرایی زبان:

یک تکنیک شناسایی زبان، استفاده از چندین تشخیص دهنده واج تک زبانه، همچون مدل زبانی N-gram می‌باشد. یک مجموعه تشخیص دهنده‌های آوایی تک زبانه مطابق واج آرایی، مدل‌های زبانی N-gram را شامل می‌شود. اساس سیستم تشخیص زبان در این قسمت یک مجموعه موازی تشخیص‌دهنده‌های واجی[۹۶] همچون مدل‌های واج‌آرایی زبان N-gram می‌باشد[۶۰].

توجه کنید که ما تنها می‌توانیم از کمک‌پردازشگرهای تشخیص واج [۹۷] در زبان‌هایی استفاده کنیم که گفتار نقل شده بر اساس قواعد صدا باشد اما سیستم تشخیص زبان با استفاده از یک مجموعه موازی تشخیص‌دهنده‌های واجی، برای سیستم‌هایی که از قواعد صدا پیروی نمی‌کنند نیز کاربرد دارد. سیستم ارائه شده توسط *زیسمان*[۹۸] [۳] متفاوت از روش‌های به کار رفته توسط *هازن*[۹۹] [۶۱] و *توکر*[۱۰۰] [۶۲] است، زیرا آنها از کمک‌پردازشگرهای تشخیص واج تکی استفاده می‌نمودند اما *زیسمان* از کمک‌پردازشگرهای تشخیص واج موازی استفاده‌کرده است. همچنین این سیستم متفاوت از روش *لامل*[۱۰۱] [۶۳] می‌باشد چراکه *زیسمان* برای تصمیم‌گیری تشخیص زبان عموماً از امتیازات واج‌آرایی استفاده کرده است، نه امتیازات صوتی که *لامل* استفاده می‌نمود.

بررسی بیشتر بر روی تأثیرکاهش تعداد کمک‌پردازشگرهای تشخیص واج نشان می‌دهد که کاهش تعداد کانال‌ها به طور کلی عملکرد سریعتری را برای گفتارهای ۱۰ ثانیه‌ای نسبت به گفتارهای

[96] Phoneme Recognition followed by Language Modeling performed in Parallel (PRLM-P)
[97] front-end
[98] Zissman
[99] Hazen
[100] Tucker
[101] Lamel

۴۵ ثانیه‌ای موجب می‌شود، همچنین استفاده از یک کانال تنها (هر کدام که باشد) عملکرد سیستم را بسیار کاهش خواهد داد.

زیسمان همچنین بررسی دیگری بر روی مدل‌های صوتی وابسته به جنسیت انجام داد که نشان می‌دهد که استفاده از کمک‌پردازشگرهای تشخیص واج وابسته به جنسیت همراه با کمک‌پردازشگرهای تشخیص واج مستقل از جنسیت، عملکرد تشخیص زبان را بهبود داده است. برای عملکرد بهتر سیستم باید از صداهای واضح استفاده نمود تا تشخیص با دقت بهتری انجام گیرد. صداهای "بد" به صداهایی اطلاق می‌شود که یا نمی‌توان آنها را به درستی تشخیص داد و یا با نرخ‌های متفاوت در زبان‌های مختلف روی می‌دهند و فقط عملکرد تشخیص زبان را مغشوش می‌کنند.

فصل چهارم

استخراج ویژگی

فصل چهارم: استخراج ویژگی

فرآیند ایجاد بردارهای ویژگی از روی سیگنال گفتاری را، استخراج ویژگی می‌نامند. هدف از استخراج ویژگی در تشخیص گفتار، ایجاد یک نمایش فشرده از شکل موج گفتار می‌باشد که دارای بیشترین اطلاعات مربوط به گفتار است. این نمایش فشرده باید بیشترین جداسازی را بین صداهای مختلف ایجاد نماید. مرحله استخراج ویژگی یکی از مهمترین مراحل موجود در سیستم‌های تشخیص گفتار می‌باشد که بسیار تأثیرگذار بوده و دقت سیستم، تا حد زیادی وابسته به آن است. اگر ویژگی‌های استخراج شده بتوانند نمایش خوبی از سیگنال داشته باشند و اطلاعات اساسی موجود در سیگنال گفتار را که برای جداسازی بین صداهای مختلف ضروری می‌باشند، حذف نکنند، دقت سیستم تشخیص تا حد زیادی بالا خواهد رفت. ویژگی‌های استخراج شده باید تا حد امکان نسبت به کانال مخابراتی، نویز و تغییرات ناشی از گوینده‌های متفاوت، مقاوم باشند.

در این فصل ابتدا پردازش کپسترال را توضیح داده و دو روش متداول کپسترال شامل MFCC و PLP و همچنین LPC را بررسی می‌کنیم. سپس ضرایب موجک را توضیح داده و نحوه استخراج برخی ضرایب مفیدتر آن، برای استفاده در برنامه اصلی را بیان می‌کنیم:

۴.۱) پردازش کپسترال

برای استخراج ویژگی، معمولاً از پاسخ فرکانسی مربوط به مجرای صوتی استفاده می‌شود و اطلاعات مربوط به سیگنال تحریک که برای اصوات صدادار، متناوب و برای اصوات بی‌صدا، نویز مانند است نادیده گرفته می‌شوند. دلیل این امر آن است که پاسخ فرکانسی مربوط به مجرای صوتی، بهترین جداسازی بین صداهای مختلف مربوط به گفتار را ایجاد می‌کند[۶۴].

برای جداسازی سیگنال تحریک و پاسخ فرکانسی مجرای صوتی، از پردازش کپسترال استفاده می‌کنیم. کپسترال یکی از انواع تبدیل‌های هوموروفیک[۱۰۲] است که قادر به جداسازی اطلاعات مربوط به منبع از فیلتر می‌باشد. برای تولید گفتار از رابطه(۴-۱) استفاده می‌کنیم. داریم:

$$s(n) = Gu(n) \otimes h(n) \qquad (۱-۴)$$

که u(n) سیگنال تحریک، h(n) پاسخ فرکانسی مجرای صوتی، s(n) سیگنال گفتار و G بهره می‌باشد. اگر از رابطه(۴-۱) تبدیل فوریه بگیریم خواهیم داشت:

$$S(f) = GU(f).H(f) \qquad (۲-۴)$$

حال از دو طرف رابطه لگاریتم مختلط می‌گیریم. داریم:

$$\log(S(f)) = \log(GU(f)) + \log(H(f)) \qquad (۳-۴)$$

[102] Homomorphic Transformation

با توجه به رابطه (۴-۳)، در حوزه لگاریتم، اطلاعات مربوط به سیگنال تحریک و مجرای صوتی بر روی هم افتاده و با تکنیک‌های موجود در پردازش سیگنال، قابل جداسازی می‌باشند[۶۵].

باگرفتن عکس تبدیل از رابطه (۴-۳) کپستروم مختلط مربوط به سیگنال s(n) بدست می‌آید. در کپسترال، ضرایب مرتبه پایین‌تر، مربوط به مجرای صوتی بوده و ضرایب مرتبه بالاتر مربوط به تحریک می‌باشند. بنابراین با حذف ضرایب مرتبه بالاتر و نگه داشتن ۱۲ یا ۱۳ ضریب مرتبه پایین‌تر، اطلاعات مربوط به مجرای صوتی از سیگنال تحریک جدا می‌شوند.

۴.۱.۱) ضرایب کپسترال پیشگویی خطی(LPCC[۱۰۳])

یکی از قویترین تکنیک‌های آنالیز صوت، آنالیز پیشگویی خطی است[۶۶]. این روش به یکی از روش‌های پرکاربرد در محاسبه پارامترهای مربوط به سیگنال صوت، مانند دوره تناوب اصلی و فرکانس‌های فرمنت تبدیل شده و کاربردهای زیادی در زمینه‌های مختلف پردازش صوت پیدا کرده است. آنالیز پیشگویی خطی به دلیل توانایی‌اش در محاسبه نسبتاً دقیق پارامترهای مربوط به صوت و همچنین سرعت بالا، اهمیت زیادی در پردازش صوت دارد. ایده اولیه آنالیز پیشگویی خطی آن است که می‌توان یک نمونه سیگنال را توسط ترکیبی از نمونه‌های قبلی تخمین زد[۶۷].

مشکل اساسی در آنالیز پیشگویی خطی، تعیین مجموعه ضرایب پیشگویی a_k به طور مستقیم از روی سیگنال گفتار است به گونه‌ای که تخمینی خوب از خواص طیفی سیگنال گفتار بدست آید. به دلیل خاصیت تغییرپذیری با زمان سیگنال گفتار، ضرایب پیشگویی باید از بخش‌های کوتاه سیگنال گفتار تخمین زده شوند. راهکار ابتدایی برای حل این مسأله، پیدا کردن مجموعه‌ای از ضرایب پیشگویی است که میانگین مربع خطای پیشگویی را در طول بخش کوچکی از شکل موج گفتار، مینیمم

[103] Linear Prediction Cepstral Coefficient

کند[۶۸]. پارامترهای بدست آمده از این روش به عنوان پارامترهای تابع سیستم، در مدل تولید گفتار فرض می‌شوند[۶۷].

بعد از بدست آمدن ضرایب پیشگویی خطی، ضرایب کپسترال پیشگویی خطی به کمک رابطه(۴-۴) محاسبه می‌شوند. همچنین از این رابطه دیده می‌شود که با تعداد محدودی ضریب پیشگویی خطی، ضرایب LPCC بدست آمده نامحدود می‌باشند. به طور تجربی نشان داده شده است که معمولاً استفاده از ۱۲ تا ۲۰ ضریب اول، برای بدست آوردن نتایج خوب، در تشخیص گفتار مفید است[۶۹].

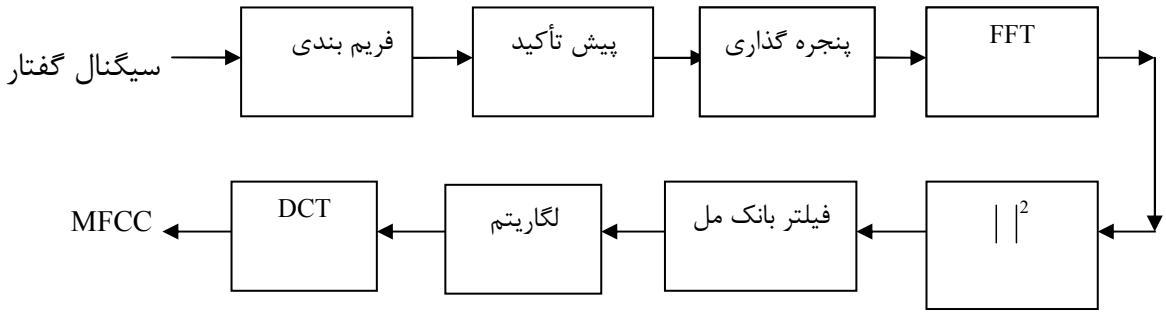
$$c(n)=\begin{cases} 0 & n<0 \\ \ln G & n=0 \\ a+\sum_{k=1}^{n-1}\left(\frac{k}{n}\right) c(k)\, a_{n-k} & 0<n\le p \\ \sum_{k=p}^{n-1}\left(\frac{k}{n}\right) c(k)\, a_{n-k} & n>p \end{cases} \qquad (۴-۴)$$

در رابطه (۴-۴) G نشان‌دهنده بهره بوده و a_k مجموعه ضرایب پیشگویی را بیان می‌کند.

۴.۱.۲) ضرایب کپسترال فرکانس مل(MFCC[۱۰۴])

ضرایب کپسترال فرکانس مل، ابتدا در سال ۱۹۸۰ توسط *دیویس*[۱۰۵] و *مرملستین*[۱۰۶] به عنوان ویژگی، برای استفاده در تشخیص گفتار پیشنهاد شد[۷۰] و امروزه به عنوان پرکاربردترین روش استخراج ویژگی در سیستم‌های بازشناخت گفتار مورد استفاده قرار می‌گیرد. روش MFCC، برای مدل کردن مکانیزم شنوایی موجود در انسان، مقیاس مل را به طیف توان سیگنال اعمال می‌کند[۷۱]. بلوک دیاگرام مربوط به محاسبه ضرایب کپسترال فرکانس مل، در (شکل ۴-۱) نشان داده شده است:

[104] Mel Frequency Cepstral Coefficients
[105] Davis
[106] Mermelstein



(شکل ۴-۱) بلوک دیاگرام مربوط به استخراج ضرایب کپسترال فرکانس مل

در ادامه به طور مختصر، مراحل مختلف روش استخراج ضرایب MFCC بررسی می‌شود:

۴.۱.۲.۱) فریم بندی[۱۰۷]:

سیگنال گفتار، سیگنالی ناایستان است و مشخصه‌های آن در طول زمان تغییر می‌کند. بنابراین برای بدست آوردن ویژگی‌های معتبر از سیگنال، باید آن را به فریم‌های ۲۰ تا ۳۰ میلی ثانیه‌ای تقسیم‌بندی کرد، به گونه‌ای که فریم‌ها ۱۰ تا ۱۵ ثانیه همپوشانی داشته باشند.

۴.۱.۲.۲) پیش تأکید[۱۰۸]:

برای کاهش اثر حنجره و لب‌ها بر روی مدل مجرای صوتی، از فیلتر دیجیتال مرتبه اول با پارامتر پیش تأکید αاستفاده می‌شود، مطابق بارابطه(۴-۵)، که مقدار α بین ۰/۹ و ۱ می‌باشد.

$$H(z) = 1 - \alpha z^{-1} \qquad (۵-۴)$$

[107] Frame Blocking
[108] Pre-emphasis

۴.۱.۲.۳) پنجره گذاری[۱۰۹]:

برای از بین بردن ناپیوستگی موجود در مرز بین فریم‌ها، باید هر فریم را در یک تابع پنجره ضرب کنیم. برای این منظور، معمولاً از پنجره همینگ استفاده می‌شود که رابطه آن به صورت زیر است:

$$W(n) = 0.54 - 0.46(\cos\frac{2\pi n}{N-1}) \qquad (۶-۴)$$

در رابطه(۴-۶) N نشان دهنده طول هر فریم می‌باشد.

۴.۱.۲.۴) آنالیز طیفی:

حال، از تبدیل فوریه گسسته زمان کوتاه، برای بردن هر فریم گفتار به حوزه فرکانس استفاده می‌شود. در این مرحله به دلیل آنکه اطلاعات مربوط به فاز، از نظر شنوایی حاوی اطلاعات مفیدی نیستند، حذف می‌شوند.

۴.۱.۲.۵) فیلتر بانک مقیاس مل:

فیلتر بانک، مجموعه‌ای از فیلترهای میان‌گذر با فرکانس‌های متفاوت است که محدوده مورد نظر سیگنال به لحاظ طیفی را می‌پوشاند. با روش‌های متفاوتی می‌توان فرکانس مرکزی این فیلترها را انتخاب نمود. معمولاً فرکانس‌ها به گونه‌ای انتخاب می‌شوند که حساسیت گوش انسان نسبت به حوزه‌های مختلف فرکانسی، مدل شود. یکی از تخمین‌هایی که برای این منظور استفاده می‌شود، مقیاس مل است[۷۲].

مل واحد ارزیابی صدای درک شده است. فرکانس مل به صورت زیر تعریف می‌شود:

$$f_{mel} = 2595\log(1+\frac{f}{700}) \qquad (۷-۴)$$

در این رابطه، f فرکانس واقعی بوده و برحسب هرتز می‌باشد.

[109] Windowing

تعداد فیلترهای موجود در بانک مقیاس مل برای کاربردهای مختلف، متفاوت است، اما معمولاً تعدادی بین ۱۹ تا ۲۴ دارند.

۴.۱.۲.۶)تبدیل کسینوسی گسسته[۱۱۰]:

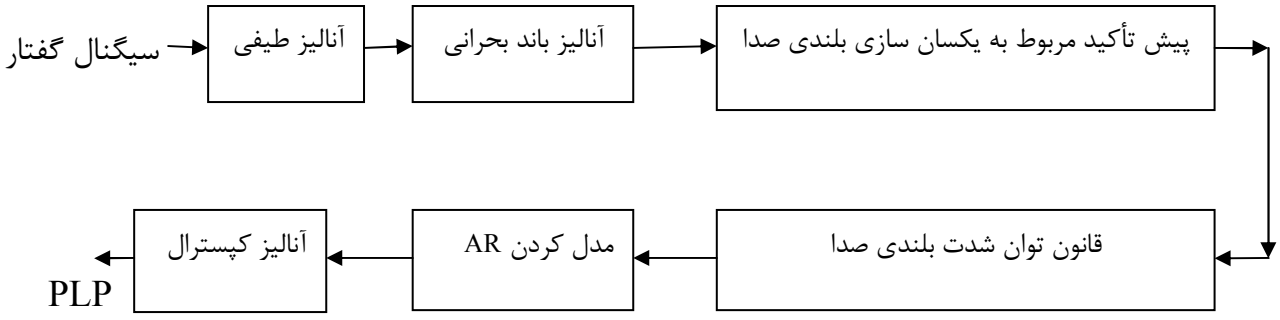
بعد از اعمال فیلتر بانک به طیف سیگنال، لگاریتم انرژی‌های فیلتر بانک محاسبه می‌شود. در نهایت با اعمال تبدیل کسینوسی گسسته به لگاریتم انرژی‌های فیلتر بانک، ضرایب MFCC بدست می‌آیند. از آنجا که ضرایب پایین‌تر حاوی اطلاعات مربوط به مجرای صوتی هستند، ضرایب ۱ تا ۱۲ را نگه داشته و باقی ضرایب را حذف می‌کنند[۷۳]. معمولاً لگاریتم انرژی هر فریم، به عنوان ضریب صفرم به مجموعه ضرایب کپسترال فرکانس مل افزوده می‌شود.

۴.۱.۳) پیشگویی خطی مبتنی بر درک انسان(PLP[۱۱۱])

روش پیشگویی خطی برای بدست آوردن ضرایب کپسترال، در قسمت قبل بیان شد. یکی از معایب LPCC آن است که مکانیزم شنوایی انسان را در محاسبه ویژگی‌ها منظور نمی‌کند. به عبارت دیگر پیشگویی خطی در تمام فرکانس‌ها، سیگنال گفتار را به یک صورت تخمین می‌زند که این مطابق با سیستم شنوایی انسان نیست. برای هماهنگ کردن روش پیشگویی خطی با سیستم شنوایی انسان، ابتدا *هرمانسکی*[۱۱۲] در سال ۱۹۸۹ روش پیشگویی خطی مبتنی بر درک انسان را پیشنهاد کرد[۷۴]. *هرمانسکی* برای هماهنگی ویژگی‌ها با سیستم شنوایی انسان، آنالیز طیفی را به گونه‌ای انجام داد که بعضی از نواحی حساستر از بقیه قسمت‌ها شوند. برای این منظور از مقیاسی شبیه مقیاس مل با نام مقیاس *بارک*[۱۱۳] استفاده نمود. بلوک دیاگرام نشان داده شده در (شکل۴-۲) مراحل مختلف روش PLP را نشان می‌دهد:

---

[110] Discrete Cosine Transformation(DCT)
[111] Perceptual Linear Prediction
[112] Hermansky
[113] Bark Scale



(شکل ۴-۲) بلوک دیاگرام مربوط به بدست آوردن ضرایبPLP [۷۴]

۴.۱.۳.۱) آنالیز طیفی:

بعد از فریم بندی و اعمال پنجره، طیف توان زمان کوتاه مربوط به هر فریم با استفاده از تبدیل فوریه سریع محاسبه می‌شود.

۴.۱.۳.۲) آنالیز باند بحرانی:

بعد از محاسبه طیف توان، محور فرکانس با توجه به رابطه (۴-۸) به فرکانس بارک نگاشت می‌شود.

$$\Omega(\omega) = 6\ln\left\{\frac{\omega}{1200\pi} + \left[\left(\frac{\omega}{1200\pi}\right)^2 + 1\right]^{0.5}\right\} \qquad (۸-۴)$$

که در این رابطه ω فرکانس زاویه‌ای برحسب رادیان بر ثانیه است. سپس طیف توان نگاشت داده شده، با نمودار باند بحرانی پوشاننده[۱۱۴] ، $\psi(\Omega)$ کانوالو می‌شود. این مرحله تقریباً شبیه به آنالیز کپسترال فرکانس مل است که در قسمت قبلی بیان شد. رابطه مربوط به نمودار باند بحرانی به صورت زیر داده شده است[۷۴].

[114] Critical Band Masking Curve

$$\psi(\Omega)=\begin{cases}0 & \Omega<-1.3\\ 10^{2.5(\Omega+0.5)} & -1.3\le\Omega<-0.5\\ 1 & -0.5\le\Omega<0.5\\ 10^{-1.0(\Omega-0.5)} & 0.5\le\Omega<2.5\\ 0 & \Omega\ge 2.5\end{cases} \qquad (۹-۴)$$

۴.۱.۳.۳) پیش تأکید مربوط به یکسان سازی بلندی صدا:

در این مرحله نمودار مربوط به یکسان سازی بلندی صدا، به رشته بدست آمده از مرحله قبلی اعمال می‌شود. تابع استفاده شده در این مرحله، تخمینی از حساسیت غیر یکنواخت شنوایی انسان در فرکانس‌های مختلف می‌باشد و حساسیت شنوایی انسان را در حدود ۴۰ دسی بل شبیه سازی می‌کند[۷۴].

۴.۱.۳.۴) قانون توان شدت بلندی صدا:

آخرین مرحله قبل از مدل کردن AR ، اعمال ریشه سوم به طیف برای تخمین قانون توان مربوط به سیستم شنوایی انسان می‌باشد. به عبارت دیگر در گوش انسان، میزان احساس بلندی صدا، با ریشه سوم انرژی آن متناسب است.

۴.۱.۳.۵) مدل کردن ***AR***[۱۱۵]:

در این مرحله برای بدست آوردن ضرایب کپسترال حاصل از پیشگویی خطی مبتنی بر درک انسان، ابتدا تبدیل فوریه معکوس گرفته می‌شود تا رشته خودهمبستگی[۱۱۶] بدست آید. سپس توسط معادله‌های *یول-واکر*، که یکی از روش‌های محاسبه ضرایب پیشگویی خطی هستند، فیلتر تمام قطب مدل می‌شود و ضرایب AR بدست می‌آید.

---

[115] Autoregressive Modeling
[116] Autocorrelation

۴.۱.۳.۶) آنالیز کپسترال:

بعد از به دست آمدن ضرایب AR ، ضرایب کپسترال مانند حالت LPCC با استفاده از معادله(۴-۴) به دست می‌آید.

آنالیز باند بحرانی انجام شده در PLP، شباهت زیادی به آنالیز بانک فیلتر مل استفاده شده در روش MFCC دارد. برخی آزمایش‌های انجام شده نشان داده اند که روش MFCC نسبت به PLP دارای نتایج بهتری در تشخیص گفتار است[۵۶]. اما روش PLP نسبت به تغییرات تعداد ضرایب و تعداد فیلترهای مورد استفاده در محاسبه ضرایب، دارای نتایج پایدارتری نسبت به MFCC می‌باشد[۷۵]. علاوه بر این PLP مقاومت بیشتری نسبت به نویز دارد.

۴.۲) ضرایب موجک

همانطور که در فصل قبل دیدیم راهکارهای سنتی از اطلاعات واج‌آرایی در تشخیص زبان‌ها استفاده می کردند، اما برای زبان‌های حاشیه‌ای[۱۱۷] یعنی زبان‌هایی با تعداد گوینده کم، یا زبان‌های شفاهی بدون استاندارد نوشتاری ثابت، اطلاعات واج‌آرایی عملاً در دسترس نیستند. بنابراین راهکارهای قبلی قابل استفاده نبوده و به روشی جدید مستقل از اطلاعات دستور زبان، نیازمندیم. یکی از این روش‌ها استفاده از تبدیل موجک برای استخراج خواص صوتی سیگنال گفتار است. این روش توسط کاربردهای تبدیل موجک در تشخیص گفتار و تشخیص گوینده حمایت می گردد[۷۶][۷۷][۷۸] .

۴.۲.۱) علت استفاده از تبدیل موجک:

تبدیل موجک از توابع پایه محلی استفاده می‌کند، بنابراین قابلیت تخمین سیگنال به نحوی مطلوب با تعداد کمی از جملات را دارا می‌باشد. به ویژه تبدیل‌های موجک *دابیچی*[۱۱۸]، فرم فشرده‌ای در حوزه

[117] Marginalized Language
[118] Daubechies

زمان دارند، یعنی فقط تعداد محدودی از جملات برای ایجاد موجک مورد نیاز است. تبدیل موجکی میر[۱۱۹] دارای فرم فشرده در حوزه فرکانس بوده و موجک *باتل-لماری*[۱۲۰]، هم در حوزه زمان و هم در حوزه فرکانس فرم فشرده را دارا می‌باشد.

به دلیل آنکه تبدیل‌های موجک محلی هستند، تفکیک پذیری در حوزه‌های زمان و فرکانس قابلیت جابجایی دارند. این ویژگی آنها را برای مشاهده سریع و مؤثر سیگنال در بازه زمانی خاص، موجه می‌سازد [۷۹].

تبدیل موجک، یک سیگنال را به سطوح فرکانسی پایین به بالا[۱۲۱]، در درجات تفکیک مختلف تجزیه می‌نماید که به عنوان تبدیل چند تفکیکه[۱۲۲] شناخته شده است. این خاصیت سبب تمایز روشنی بین مؤلفه‌های فرکانس‌های پایین و بالای سیگنال می‌شود که برای عملکردمان مفید خواهد بود. از آنجا که مؤلفه‌های فرکانس پایین گفتار، در برگیرنده ویژگی‌های صوتی نظیر ضرب‌آهنگ است[۱۰] که نقشی اساسی در تشخیص زبان بازی می‌کند، استفاده از تبدیلات چند تفکیکه نظیر تبدیل موجک، کمک شایانی در این امر می‌باشد.

فیلتر موجک نیاز به تقسیم نمونه سیگنال در مقیاس کوچک ندارد و به عبارتی یک فیلتر بالاگذر[۱۲۳] می‌باشد. برای فیلتر در هر مرحله می‌توان قسمت‌های تخمین (خروجی فیلتر پایین‌گذر) و جزئیات (خروجی فیلتر بالاگذر) را مشخص نمود.

شایستگی فیلتر موجک در فشرده‌سازی گفتار، به دلیل قابلیتش در متمرکز نمودن اطلاعات می‌باشد هنگامیکه گفتار، صدادار[۱۲۴] ویا ترکیبی[۱۲۵] است. یک فیلتر موجک مناسب باید بیش از ۹۰٪ انرژی

---

[119] Meyer
[120] Battle-Lemarie
[121] Low-to-high
[122] Multi resolution
[123] High-pass
[124] Voiced
[125] Mixed

گفتار صدادار را در نخستین $N/2$ ضرایب و حدود ۹۰٪ را در نخستین $N/4$ ضرایب متمرکز نماید. اما برای گفتار بی‌صدا[۱۲۶] تنها حدود ۳۵٪ ضرایب در نخستین $N/2$ و حدود ۸٪ در نخستین $N/4$ ضرایب و برای گفتار ترکیبی ۲۴٪ ضرایب در نخستین $N/2$ و ۱۹٪ در نخستین $N/4$ ضرایب موجود است[۷۹].

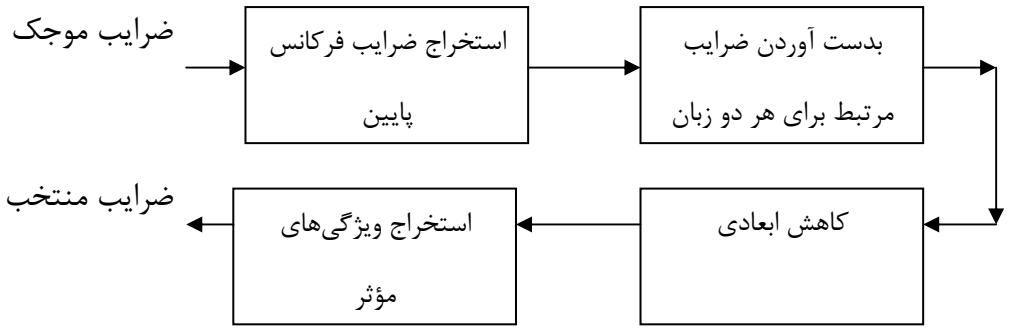
فریم‌های گفتار صدادار، بی‌صدا و ترکیبی بیشتر به چگونگی تقسیم انرژی در باندها توسط ضرایب موجک بستگی دارند. اگر درصد انرژی متمرکز شده کمتر از ۴۰٪ باشد، گفتار بی‌صدا است، اگر بین ٪۴۰ تا ۹۰٪ باشد، گفتار ترکیبی بوده و اگر بالای ۹۰٪ باشد، صدادار است. در فریم‌های صدادار، انرژی بیشتر در ۲ باند متمرکز شده است، در فریم‌های بی‌صدا در میان همه عرض باندها پخش شده و در فریم‌های ترکیبی، دربین ۳ تا ۵ باند محدود شده است[۷۹].

بیشتر مواقع، تبدیل موجک، انرژی را در قسمت تخمینی ضرایب که خروجی فیلتر پایین گذر می‌باشد، متمرکز می‌نماید. به همین دلیل ما نیز از قسمت تخمین این ضرایب بهره می‌بریم.

۴.۲.۲) نحوه استخراج ضرایب موجک:

پیش از آنکه ضرایب موجک را به برنامه تشخیص زبان دهیم، جهت کاهش پیچیدگی محاسباتی، تعداد محدودی از آنها را که در تعیین قواعد بهینه دسته‌بندی زبان گفتاری نقش مؤثری دارند، انتخاب می‌کنیم. برای این منظور می‌توان مطابق (شکل ۴-۳) به استخراج ضرایب مفیدتر پرداخت:

[126] Unvoiced



(شکل ۴-۳) بلوک دیاگرام مربوط به استخراج ضرایب موجک

۴.۲.۳) استخراج ضرایب فرکانس پایین:

از آنجا که برای هر نمونه، تعداد بسیاری ضریب به دست می‌آید، بنابراین باید تعداد این ضرایب را کاهش دهیم. چنانچه می‌دانیم ضرایب با دامنه بیشتر، بیانگر فرکانس پایین و ضرایب با دامنه کمتر بیانگر فرکانس بالا می‌باشند[۸۰]. ما به ضرایب فرکانس پایین نیاز داریم، بنابراین ضرایبمان را از میان ضرایب با دامنه بالاتر بر می‌داریم. برای اینکار از یک مقدار آستانه[۱۲۷] استفاده می‌شود. برای هر نمونه صوتی، آستانه بنحوی انتخاب می‌شود که یک درصد از ضرایب موجکی که بزرگترین قدر مطلق را دارند نگه داشته و بقیه را صفر می‌کنیم.

۴.۲.۳.۱) بدست آوردن ضرایب مرتبط برای هر دو زبان:

پس از آنکه برای هر نمونه صوتی، آستانه تعیین گردیده و ۹۹٪ ضرایب با قدر مطلق کوچکتر را صفر نمودیم، باید ویژگی‌هایی که به ازای آنها تمامی ضرایب نمونه‌های مختلف صفر شده‌اند را مشخص نموده و حذف کنیم تا ضرایب مرتبط برای هردو زبان به دست آیند.

[127] Threshold

۴.۲.۳.۲) کاهش ابعادی:

از آنجا که تعداد ضرایب موجک همچنان زیاد است، در این مرحله نیاز به کاهش ابعادی ضرایب داریم. پس با استفاده از بهره اطلاعات [۸۱]، ویژگی‌های مفیدتر را برای تمایز بین زبان‌ها به دست می‌آوریم.

۴.۲.۳.۳) استخراج ویژگی‌های مؤثر:

با محاسبه اطلاعات بهره برای ویژگی‌ها، با اعمال یک حد آستانه، تعدادی از ویژگی‌ها را که بهره بالاتری دارند به برنامه اصلی می‌دهیم تا از میان آنها، بهترین ویژگی جهت تمایز بین زبان‌ها انتخاب شود.

پس از طی این مراحل می‌توان این ضرایب را به برنامه داده تا با انتخاب بهترین ضریب، اقدام به تشخیص زبان‌ها نموده و درصد صحت حاصل از تشخیص را باز گرداند. روند اصلی برنامه و چگونگی انتخاب ضرایب برتر در فصل بعدی شرح داده شده است.

فصل پنجم

روش پیشنهادی

فصل پنجم: روش پیشنهادی

در این فصل به معرفی روش پیشنهادی پرداخته و عملکرد سیستم را بیان می‌کنیم. برنامه اصلی مورد استفاده برای تشخیص زبان، برای ضرایب موجک و کپسترال یکی است و تنها نحوه استخراج این ضرایب اندکی متفاوتند. برای استخراج ضرایب موجک، ابتدا این ضرایب را محاسبه کرده و پس از انجام کاهش ابعادی، که با کمک بهره اطلاعات [۸۱] انجام می‌شود، ویژگی‌های مفیدتر را استخراج نموده و آنها را به برنامه اصلی می‌دهیم. اما از آنجا که ضرایب کپسترال تعداد محدودتری می‌باشند، نیازی به کاهش ابعادی و بهره اطلاعات نداشته و تنها با تنظیم اندازه پنجره‌های مورد استفاده می‌توان به تعداد ضریب مطلوب دست یافت. در برنامه اصلی که برای ضرایب موجک و کپسترال یکسان است، روش کار بدین صورت می‌باشد:

ابتدا ضرایب مربوط به هر ستون ویژگی را به صورت مجموعه‌ای مانند S در نظر گرفته و جهت پیدا کردن نقاط قطع از الگوریتم بازگشتی بر روی مجموعه استفاده می‌شود. برای اینکار از تکنیک گسسته‌سازی چند بازه‌ای[۵۲] بهره برده ایم. در هر مجموعه S، مقدار میانگین هر دو ضریب متوالی نابرابر، از دو کلاس متفاوت را به عنوان یک کاندیدای نقطه قطع T در نظر گرفته و با استفاده از آن، مجموعه S را به دو زیر مجموعه S_1 وS_2 تفکیک می‌کنیم که در ادامه هر یک از این S_1 و S_2 خود یک مجموعه جدید S در نظر گرفته شده و بر روی آن به دنبال نقاط قطع جدید می‌گردیم.

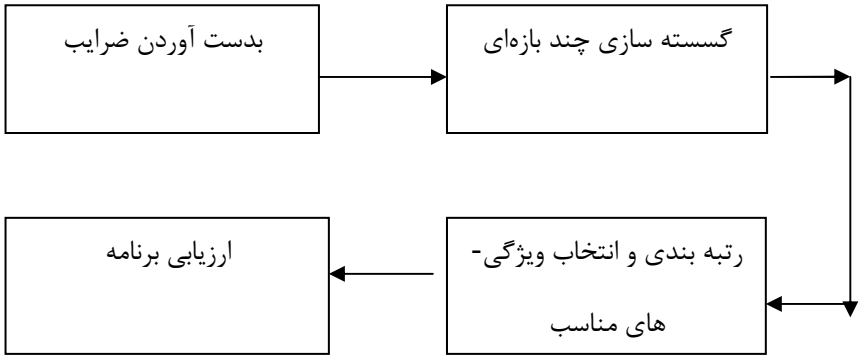
نقاط قطع، هر ستون ویژگی را به تعدادی دسته، تقسیم‌بندی کرده و بر اساس احتمال وجود هر زبان در دسته‌ها، می‌توان آنها را به نام کلاسی که بیشترین احتمال وقوع در آن دسته را دارد، نامگذاری نمود. این عمل را نامگذاری دسته می‌نامیم.

برای پیدا نمودن ویژگی‌های مؤثر، باید دید با استفاده از کدام ویژگی، به درصد صحت بالاتری در تشخیص زبان‌ها می‌رسیم. برای اینکار باید قاعده دسته بندی زبان را برای هر ویژگی به دست آوریم. با مقایسه زبان تعیین شده توسط برنامه، با زبان واقعی نمونه‌ها، می‌توان ضریب درستی تشخیص را برای هریک از ستون‌های ویژگی به دست آورد.

همچنین با استفاده از تکنیک رتبه بندی ویژگی‌ها[۸۱]، می‌توانیم برای بهبود درصد صحت تشخیص زبان، از سایر ویژگی‌ها به صورت ترکیبی با ویژگی انتخابی اولیه استفاده نماییم. جواب حاصل از تلفیق ویژگی ها را با درصد صحت قبلی مقایسه کرده، چنانچه بیشتر باشد ویژگی جدید را می‌پذیریم و در غیر اینصورت به ویژگی‌های قبلی اکتفا می‌کنیم. در نهایت بهترین مجموعه ویژگی‌ها برای داشتن بیشترین درصد صحت یافت می‌شوند.

در ادامه راهکارهای به کار رفته در برنامه نویسی به تفصیل بیان می‌شود:

. به طور کلی برای تشخیص خودکار زبان ، مراحل زیر باید طی شود:



(شکل ۵-۱) مراحل مختلف تشخیص خودکار زبان گفتاری

که بدست آوردن ضرایب و گسسته سازی در قالب پیش‌پردازش ضرایب گنجانده شده‌اند.

۵.۱) پیش پردازش ضرایب

پیش از آنکه ضرایب را به برنامه تشخیص زبان دهیم، جهت کاهش پیچیدگی محاسباتی، تعداد محدودی از آنها را که در تعیین قواعد بهینه دسته‌بندی زبان گفتاری نقش مؤثری دارند، انتخاب می‌کنیم. بدین منظور از روش گسسته سازی چند بازه‌ای ویژگی‌های پیوسته [۵۲] کمک می‌گیریم. حال نحوه به‌دست آوردن ضرایب موجک، ضرایب کپسترال و گسسته سازی چند بازه‌ای آنها را شرح می‌دهیم:

۵.۱.۱) به دست آوردن ضرایب:

در این روش از ضرایب موجک و کپسترال بهره برده‌ایم که نحوه استخراج آنها تا حدودی متفاوت است. از آنجا که ضرایب حاصل از تبدیل موجک معمولاً تعداد زیادی دارند، نیاز به روش‌هایی جهت

کاهش ابعادی آنها و استخراج ضرایب مؤثرتر می‌باشد. همچنین در ضرایب کپسترال اندازه پنجره انتخابی و مقدار همپوشانی پنجره‌ها و همچنین تعداد ترکیب برای هر ضریب اهمیت بسیار دارد. حال با نحوه به دست آوردن هر یک از این ضرایب آشنا می‌شویم:

۵.۱.۱.۱) بدست آوردن ضرایب کپسترال:

برای پردازش ضرایب کپسترال، ما از متداول‌ترین این ضرایب استفاده کرده‌ایم که شامل ضرایب PLP, MFCC و LPC می‌باشد[۸۲]. برای هر یک از این ضرایب، ترکیبات متفاوت ۵،۷،۹،۱۲،۱۳ تایی را استفاده نموده و هر بار اندازه پنجره را بنابر طول مدت نمونه صوتی تغییر داده‌ایم تا بهترین اندازه پنجره را در هر حالت بیابیم. همچنین برای تمام ترکیبات مختلف از همپوشانی ۵۰٪ استفاده شده است..

۵.۱.۱.۲) بدست آوردن ضرایب موجک:

در این قسمت، از ضرایب موجک نوع دابیچی استفاده می‌شود، پس برای هر نمونه صوتی ضرایب دابیچی را به دست می‌آوریم. بدین منظور ابتدا به ترتیب هر یک از نمونه‌های صوتی داده شده را با دستور"wavread" که در "Matlab" موجود می‌باشد، خوانده و ضرایبی به دست می‌آید. از آنجا که معمولاً "Matlab" این ضرایب را به صورت ستونی می‌دهد و ما می‌خواهیم آنها را به صورت سطری داشته باشیم، به یک عملگر ترانهاده جهت تبدیل بردار ستونی به سطری نیاز خواهیم داشت. پس از آن، تعداد این ضرایب را محاسبه نموده و برای هر نمونه داده، این تعداد را ذخیره می‌نماییم.

پس از آنکه بر روی نمونه‌های آموزشی(train) داده شده، عملیات بالا انجام شد، کمترین تعداد ذخیره شده بین ضرایب را به عنوان سایز عمومی انتخاب نموده و در مرحله بعد، همه ضرایب حاصل از نمونه‌های مختلف را بر اساس کمترین اندازه[۱۲۸]، هم اندازه می‌کنیم، یعنی از ضریب اول تا شماره

[128] Size

کمترین اندازه را نگه داشته و باقی ضرایب را حذف می‌کنیم. بنابراین به تعداد نمونه‌های صوتی، بردار افقی ضرایب داریم و در هر بردار به تعداد کمترین اندازه انتخابی، ضریب خواهیم داشت.

حال باید ضرایب موجک دابیچی را به دست آوریم، بدین منظور مجموعه ضرایب مربوط به هر نمونه صوتی را خوانده و با استفاده از دستور"dwt" که در " Matlab " موجود است، با انتخاب فیلتر دابیچی و شماره مورد نظر برای فیلتر، قسمت‌های تخمینی و جزئیات سیگنال را تفکیک می‌کنیم. در این راهکار می‌خواهیم تأثیر درجه‌های گوناگون فیلتر دابیچی، به ویژه درجه‌های ۱۰،۲ و ۲۰ را مشاهده نماییم، بنابراین درجه فیلتر را در مراحل مختلف تغییر داده و نتایج را ثبت می‌کنیم. همچنین آزمایشات نشان دادند که پردازش قسمت تخمینی سیگنال، پاسخ بهتری نسبت به جزئیات دارد. بنابراین ما بر روی قسمت تخمینی سیگنال کار می‌کنیم و جزئیات را کنار می‌گذاریم. تعداد ضرایب حاصل از قسمت تخمینی سیگنال را برای هر نمونه محاسبه نموده و ذخیره می‌کنیم.

از آنجا که برای هر نمونه تعداد بسیاری ضریب به دست می‌آید، بنابراین باید تعداد این ضرایب را کاهش دهیم. چنانچه می‌دانیم ضرایب با دامنه بیشتر، بیانگر فرکانس پایین و ضرایب با دامنه کمتر، بیانگر فرکانس بالا می‌باشند[۸۰]. ما به ضرایب فرکانس پایین نیاز داریم، بنابراین ضرایبمان را از میان ضرایب با دامنه بالاتر بر می‌داریم. برای اینکار از یک مقدار آستانه استفاده می‌شود. برای هر نمونه صوتی، آستانه بنحوی انتخاب می‌شود که یک درصد از ضرایب موجکی که بزرگترین قدر مطلق را دارند نگه داشته و بقیه را صفر می‌کنیم یعنی برای هر نمونه صوتی، ضرایب را با داشتن شماره ستون(شماره ویژگی) از بیشترین به کمترین مرتب نموده و سپس یک درصد ضرایب با قدرمطلق بزرگتر را برداشته و آنها را در ماتریسی جدید می‌ریزیم(در سطر مربوط به نمونه صوتی داده شده و در ستون شماره ویژگی ذخیره شده) و باقی ضرایب مربوط به آن سطر را صفر می‌نماییم.

این روند برای نمونه‌های آموزشی(train) انجام شده، ماتریسی با m سطر و n ستون به دست می‌آید که m تعداد نمونه‌های صوتی بوده و n تعداد ویژگی‌ها تا این مرحله می‌باشد که برابر تعداد ویژگی‌های

حاصل از فیلتر دابیچی است. حال ویژگی‌هایی که به ازای آنها تمامی ضرایب نمونه‌های مختلف صفر شده‌اند را مشخص نموده و حذف می‌کنیم تا ضرایب مرتبط برای هردو زبان به دست آیند.

در این مرحله نیاز به کاهش ابعادی ضرایب داریم، پس با استفاده از بهره اطلاعات [۸۱]، به صورت زیر ویژگی‌های مفیدتر را برای تمایز بین زبان‌ها به دست می‌آوریم:

$$IG_i = H(C) - H(C|A_i) \quad (۱-۵)$$

$$H(C) = -\sum_{c \in C} p(c) \log_2 p(c) \quad (۲-۵)$$

$$H(C|A) = -\sum_{a \in A} p(a_i) \sum_{c \in C} p(c|a_i) \log_2 p(c|a_i) \quad (۳-۵)$$

که در این فرمول‌ها $p(c)$ بیانگر احتمال وجود نمونه از هر کلاس (زبان) بوده، $p(a_i)$ احتمال هریک از ویژگی‌ها و $p(c|a_i)$ احتمال وجود ضرایب از کلاس c را به شرط ویژگی a_i بیان می‌کند.

با محاسبه اطلاعات بهره برای ویژگی‌ها، با اعمال یک حد آستانه، تعدادی از ویژگی‌ها را که بهره بالاتری دارند به برنامه اصلی می‌دهیم تا از میان آنها، بهترین ویژگی‌ها جهت تمایز بین زبان‌ها انتخاب شوند.

در مرحله بعدی به ازای ستون‌های(ویژگی‌های) باقی مانده، از روی شماره ویژگی‌ها، مقادیر اصلی جایگزین صفرهای ایجاد شده در پردازش می‌شوند و در نتیجه ماتریس آموزشی(train) آماده خواهد بود. پس از این پردازش‌ها، ماتریس به دست آمده، ماتریس ضرایب موجک مورد نیاز برای پردازش‌های بعدی را در اختیار ما قرار می‌دهد که همراه با بردار ستونی(n*۱) است که نام و در واقع شماره مربوط به زبان هریک از نمونه‌های صوتی را در خود ذخیره دارد.

۵.۱.۲) گسسته سازی چند بازه‌ای

برای یافتن بهترین ویژگی‌ها، از تکنیک گسسته‌سازی چندبازه‌ای، کمک می‌گیریم [۵۲]. بدین منظور ابتدا ماتریس ضرایب را به همراه نوع زبان هر نمونه دریافت می‌کنیم، برای هر ویژگی، ضرایب را به ترتیب صعودی مرتب کرده و بردار نام زبان‌ها را نیز با توجه به محل قرارگیری هر نمونه تغییر می‌دهیم. حال باید در هر ستون ویژگی، به دنبال مرزهایی برای جداسازی نمونه‌های مربوط به هر یک از زبان‌ها بگردیم، که این روش با استفاده از راهکار پیشنهادی توسط *فیاد*[۱۲۹] و *ایرانی*[۱۳۰] [۵۲] انجام گرفته است. اگر هر ستون ویژگی را با مجموعه S نشان دهیم ونقطه قطع را با T بیان نماییم، در هر مجموعه S، مقدار میانگین هر دو ضریب متوالی نابرابر، از دو کلاس متفاوت را به عنوان یک کاندیدای نقطه قطع Tدر نظر گرفته و با استفاده از آن، مجموعه S را به دو زیر مجموعه S_1 وS_2 تفکیک می‌کنیم که S_1 از اولین ضریب تا نقطه قطع T بوده و S_2 از ادامه T ، تا آخرین ضریب می‌باشد.

سپس به محاسبه آنتروپی حاصل از هر نقطه قطع می‌پردازیم[۵۲]:

$$Ent(S) = -\sum_{i=1}^{k} P(C_i, S)\log_2(P(C_i, S)) \qquad (۴-۵)$$

در (فرمول۵-۴) k نشان دهنده تعداد کلاس‌های گوناگون(در اینجا تعداد زبان‌های مختلف) بوده و $P(C_i,S)$ احتمال وجود نمونه‌های کلاس C_i در مجموعه S را بیان می‌کند.

به همین ترتیب با محاسبه $P(C_i,S_1)$ و $P(C_i,S_2)$ و تغییر فرمول می‌توان به صورت زیر $Ent(S_1)$ و $Ent(S_2)$ را نیز محاسبه نمود:

$$Ent(S_1) = \sum_{i=1}^{k} P(C_i, S_1)Log_2(P(C_i, S_1)) \qquad (۵-۵)$$

[129] Fayyad
[130] Irani

$$Ent(S_2) = \sum_{i=1}^{k} P(C_i, S_2) Log_2(P(C_i, S_2)) \qquad (۶-۵)$$

حال باید با استفاده از این آنتروپی‌ها به محاسبه آنتروپی اطلاعات کلاس E(A,T;S) بپردازیم. برای نمونه‌های مجموعه S، برای یک ویژگی A و با یک نقطه قطع T، آنتروپی اطلاعات کلاس از رابطه (۵-۷) محاسبه می‌گردد[۵۲]:

$$E(A,T;S) = \frac{|S_1|}{|S|} Ent(S_1) + \frac{|S_2|}{|S|} Ent(S_2) \qquad (۷-۵)$$

پس از آنکه به ازای هریک از کاندیداهای نقطه قطع T، E(A,T;S) محاسبه گردید، نقطه قطعی که آنتروپی اطلاعات کلاسی حاصل از آن کمترین باشد را در نظر گرفته و زیر مجموعه‌های حاصل از آن، S_1 و S_2 و همچنین آنتروپی‌های حاصل از آن، Ent(S) و Ent(S1) و Ent(S2) ذخیره خواهد شد. اما تنها هنگامی می‌توان آن را به عنوان نقطه قطع حقیقی پذیرفت که در شرط MDLPC [۵۲] صدق نماید:

$$Gain(A,T;S) > \frac{\log_2(N-1)}{N} + \frac{\Delta(A,T;S)}{N} \qquad (۸-۵)$$

$$Gain(A,T;S) = Ent(S) - \frac{|S1|}{N} Ent(S1) - \frac{|S2|}{N} Ent(S2) \qquad (۹-۵)$$

$$\Delta(A,T;S) = \log_2(3^k - 2) - [kEnt(S) - k1Ent(S1) - k2Ent(S2)] \qquad (۱۰-۵)$$

N تعداد کل نمونه‌های موجود در S می‌باشد که برابر با تعداد نمونه‌های صوتی داده شده است و k و k_1 و k_2 به ترتیب بیانگر تعداد کلاس‌های متمایز موجود در S و S_1 و S_2 می‌باشند.

اگر شرط معادله برقرار گردد، نقطه قطع T پذیرفته شده و S به زیر مجموعه‌های S_1 و S_2 تفکیک می‌گردد. در ادامه هریک از این S_1 و S_2 ، خود یک مجموعه جدید S درنظر گرفته می‌شوند و روال فوق بصورت بازگشتی بر روی S_1 و S_2 اعمال می‌شود تا جایی که دیگر نتوانیم نقطه قطع جدیدی برای ستون ویژگی بیابیم، پس نقاط قطع (T) های به دست آمده را ذخیره می‌کنیم و به سراغ ستون ویژگی بعدی خواهیم رفت. به همین ترتیب برای هر یک از ویژگی‌ها (ستون‌های ماتریس train) تعدادی نقاط قطع به دست می‌آید.

نقاط قطع، هر ستون ویژگی را به تعدادی دسته، تقسیم‌بندی کرده و بر اساس احتمال وجود هر زبان در دسته‌ها، می‌توان آنها را به نام کلاسی که بیشترین احتمال وقوع در آن دسته را دارد، نامگذاری نمود. این عمل را نامگذاری دسته می‌نامیم.

۵.۲)روند کلی برنامه

از ۲/۳ نمونه‌های صوتی داده شده، به عنوان داده‌های آموزشی(train) و از ۱/۳ دیگر به عنوان داده تست(test) استفاده می‌کنیم. نخست با استفاده از داده‌های آموزشی و پردازش آنها بصورت ذکر شده در فصل قبل، قواعد دسته‌بندی(پیدا کردن نقاط قطع و دسته‌بندی ویژگی‌ها و همچنین نامگذاری هر دسته) را بدست می‌آوریم.

در بخش قبل نحوه به دست آوردن ضرایب موجک و گسسته‌سازی چند بازه‌ای را مشاهده نمودیم. حال ادامه مسیر را بررسی می‌کنیم:

۵.۲.۱) تکنیک انتخاب ویژگی:

برای پیدا نمودن ویژگی‌های مؤثر، باید دید با استفاده از کدام ویژگی، به درصد صحت بالاتری در تشخیص زبان‌ها می‌رسیم. برای اینکار باید قاعده دسته‌بندی زبان را برای هر ویژگی به دست آوریم.

حال اگر ماتریس ضرایب را به برنامه بدهیم، با مقایسه کلاس(زبان) تعیین شده توسط برنامه، با کلاس واقعی نمونه می‌توان ضریب درستی تشخیص را برای هریک از ستون‌های ویژگی به دست آورده و در نتیجه بهترین ویژگی را بر اساس بالاترین درصد صحت تشخیص زبان انتخاب نماییم.

همچنین با استفاده از تکنیک رتبه بندی ویژگی‌ها[۸۱]، می‌توانیم برای بهبود درصد صحت تشخیص زبان، از سایر ویژگی‌ها به صورت ترکیبی با ویژگی انتخابی اولیه استفاده نماییم. برای اینکار ترکیب دوتایی هر یک از جفت ویژگی‌ها را در نظر گرفته و درصد صحت تشخیص زبان را برای هر جفت بدست می‌آوریم. جفتی که بیشترین درصد صحت تشخیص زبان را ایجاد می‌کند، به عنوان جواب انتخاب می‌کنیم. جواب حاصل از تلفیق جفت ویژگی‌ها را با درصد صحت قبلی مقایسه کرده، چنانچه بیشتر باشد ویژگی جدید را می‌پذیریم و در غیر اینصورت به ویژگی‌های قبلی اکتفا می‌کنیم. این کار با تلفیق سه تایی و بیشتر ویژگی‌ها تا زمانی که درصد صحت تشخیص زبان بهبود می‌یابد ادامه خواهد داشت و در نهایت بهترین مجموعه ویژگی‌ها برای داشتن بیشترین درصد صحت یافت می‌شوند.

۵.۲.۲) ارزیابی برنامه:

برای مشاهده نتایج برنامه، از ۵۰ نمونه صوتی برای هر زبان استفاده شده است، که نزدیک به ۳۲ نمونه برای قسمت آموزش و ۱۸ نمونه برای قسمت تست به کار می‌رود. در نتیجه برای مقایسه دوبه دوی زبان‌ها، هربار ۶۴ نمونه برای آموزش و ۳۶ نمونه برای تست داریم.

از نمونه‌های قسمت آموزشی برای یافتن بهترین ویژگی‌ها استفاده نموده و با استفاده از ویژگی‌های انتخابی، به تعیین نوع زبان هر نمونه تست می‌پردازیم. سپس جواب‌های برنامه را با کلاس(زبان) اصلی نمونه‌ها که به همراه هر نمونه صوتی داده شده است، مقایسه می‌کنیم. تعداد جواب‌های درست و نادرست را تعیین نموده، درصد صحت تشخیص برنامه که بیانگر عملکرد سیستم خواهد بود، به دست می‌آید.

فصل ششم

نتایج آزمایش‌های انجام شده

فصل ششم: نتایج آزمایش‌های انجام شده

برای آزمایش روش پیشنهادی، از نمونه‌های صوتی ۱۰ ثانیه‌ای و ۴۵ ثانیه‌ای پایگاه اطلاعاتی OGI-TS [۴۳][۲۸][۲] استفاده گردیده است. در OGI-TS نمونه‌های صوتی از ۱۱ زبان انگلیسی، فارسی، آلمانی، اسپانیایی، کره‌ای، ماندارین، ژاپنی، تامیل، ویتنامی، فرانسوی و هندی با زمان‌بندی‌های گوناگون موجود است. اما در سیستم‌های تشخیص زبان بیشتر از ۹ زبان اول استفاده شده است. به همین منظور ما نیز آزمایش‌ها را بر روی این ۹ زبان انجام داده و با روش‌های پیشین مقایسه نموده‌ایم.

همچنین به دلیل آنکه نتایج بیشتر بر روی نمونه‌های ۱۰ ثانیه‌ای و ۴۵ ثانیه‌ای منتشر شده اند، ابتدا نمونه‌های OGI-TS، را دسته بندی کرده و نمونه‌های ۱۰ ثانیه‌ای و ۴۵ ثانیه‌ای آن را جدا نمودیم.

تعداد این نمونه‌ها برای زبان‌های مختلف متفاوت بود. بنابراین از یک مقدار میانگین برابر با ۵۰ نمونه برای هر زبان استفاده شده است، که نزدیک به ۳۲ نمونه برای قسمت آموزش و ۱۸ نمونه برای قسمت تست به کار می‌رود. در نتیجه برای مقایسه دوبه دوی نمونه‌ها، هربار ۶۴ نمونه برای آموزش و ۳۶ نمونه برای تست داریم.

آزمایش‌ها بر روی ضرایب مختلف موجک، MFCC، PLP و LPC انجام شده که نتایج حاصل از هر یک از این ضرایب در ادامه نشان داده خواهد شد.

۶.۱) نتایج آزمایش‌های انجام شده بر روی ضرایب موجک:

ابتدا آزمایش‌ها بر روی ضرایب موجک انجام می‌شود. برای این منظور از ضرایب موجک، بر مبنای فیلتر دابیچی از درجه ۲، ۱۰و ۲۰ استفاده نموده و آنها را برای نمونه‌های ۱۰ ثانیه‌ای و ۴۵ ثانیه‌ای بدست می‌آوریم. باید توجه داشت که ما تنها قسمت‌های تخمینی حاصل از فیلتر دابیچی را مورد استفاده قرار می‌دهیم و جزئیات را در نظر نمی‌گیریم.

در (جدول ۶-۱) نتایج حاصل از ارزیابی برنامه با استفاده از ضرایب دابیچی درجه ۱۰ و ۲۰ بر روی نمونه‌های ۱۰ ثانیه‌ای آورده شده است. در این جدول اعداد سمت راست، مربوط به نتایج حاصل از دابیچی از درجه ۱۰ و اعداد سمت چپ، مربوط به نتایج حاصل از دابیچی درجه ۲۰ می‌باشند. لازم به ذکر است که اعداد در تمامی این جداول بر حسب درصد می‌باشند.

(جدول ۶-۱): نتایج حاصل از دابینچی درجه ۱۰ و۲۰ برای نمونه های ۱۰ ثانیه ای

۱۰ ثانیه‌ای	آلمانی	اسپانیایی	ژاپنی	ماندارین
انگلیسی	۶۲-۵۹	۶۷-_	۶۷-۶۱	۶۷-۴۲
آلمانی	-	۶۴-۶۴	۵۵-۵۸	۶۲-۵۶
اسپانیایی	-	-	۶۱-۴۹	۵۳-۴۵
ژاپنی	-	-	-	۵۸-۴۶

علامت (-) در ارزیابی مجموعه دوزبانه انگلیسی-اسپانیایی با دابیچی درجه ۲۰، نشان دهنده این است که برنامه در کلاسه بندی، قادر به یافتن نقطه قطعی نبوده و در نتیجه کلاسه بندی انجام نشده است. همانطور که از (جدول۶-۱) مشاهده می شود، به طور کلی ضرایب دابیچی درجه ۲۰ نتایج بهتری دارند زیرا هرچه درجه ضرایب دابیچی افزایش یابد، توانایی تفکیک ضرایب بیشتر شده و به فیلتر ایده‌آل نزدیکتر می‌شوند. در (جدول۶-۲) بهترین نتایج حاصل از دابیچی های از درجه ۱۰،۲ و۲۰ برای نمونه‌های۱۰ ثانیه‌ای نشان داده می‌شود. همچنین نتایج به دست آمده توسط گومینز[۵۹] برای مقایسه داخل پرانتز آورده شده است:

(جدول ۶-۲) بهترین نتایج به دست آمده از دابیچی های درجه ۲و ۱۰و۲۰ برای نمونه های ۱۰ ثانیه ای(نتایج به دست آمده توسط گومینز داخل پرانتز آورده شده است)

۱۰ثانیه‌ای	آلمانی	اسپانیایی	ژاپنی	ماندارین
انگلیسی	۶۲(۵۶)	۶۷(۵۰)	۶۷(۶۳)	۶۷(۶۳)
آلمانی	-	۶۴(۵۴)	۵۸(۶۹)	۶۴(۶۹)
اسپانیایی	-	-	۶۷(۶۰)	۶۷(۶۲)
ژاپنی	-	-	-	۵۸(۵۰)

مشاهده می‌شود که روش پیشنهادی در بیشتر موارد درصد صحت بالاتری نسبت به روش گومینز[۵۹] دارد، به جز دو دسته آلمانی-ژاپنی و آلمانی-ماندارین. زیرا ما از ضرایب در حوزه فرکانس استفاده می‌کنیم و اطلاعات زمانی را در قسمت آموزش دخالت نمی‌دهیم بنابراین زبان‌هایی که خواص آهنگین دارند، در روش ما برجستگی خاصی ندارند.

همچنین(جدول۶-۳) نتایج حاصل از ضرایب دابیچی درجه ۲۰ را برای نمونه‌های ۴۵ ثانیه‌ای نشان می‌دهد. نتایج به دست آمده توسط *گومینز* [۵۹] برای مقایسه داخل پرانتز و نتایج به دست آمده توسط *رواس* [۵۴] داخل کروشه آورده شده است:

(جدول ۶-۳) نتایج حاصل از دابیچی درجه ۲۰ برای نمونه های ۴۵ ثانیه ای(نتایج به دست آمده توسط گومینز و روآس به ترتیب داخل پرانتز و کروشه آورده شده است)

۴۵ثانیه‌ای	آلمانی	اسپانیایی	ژاپنی	ماندارین
انگلیسی	۶۲ [۶۰](۵۵)	۷۳ [۶۸](۵۲)	۷۰ [۶۸](۶۲)	۶۴ [۷۵](۶۲)
آلمانی	-	۶۲ [۵۹](۵۴)	۴۸ [۶۶](۷۲)	۶۲ [۶۲](۷۰)
اسپانیایی	-	-	۶۲ [۶۳](۷۱)	۵۹ [۸۱](۶۳)
ژاپنی	-	-	-	۵۹ [۵۰](۴۴)

از (جدول۶-۳) مشخص است که نتایج نسبت به روش *گومینز*[۵۹] بهبود خوبی داشته و در بسیاری حالت‌ها نسبت به روش *رواس*[۵۴] نیز بهبود دارد.

در (جدول ۶-۴) نتایج مقایسه دوبه دو نمونه‌های صوتی ۱۰ ثانیه ای، برای همه ۹ زبان موجود در OGI-TS به ازای ضرایب دابینچی ۱۰،۲ و ۲۰ آورده شده است. در این قسمت نیز ضرایب دابینچی ۲۰ نتایج بهتری را نشان می‌دهند:

(جدول ۶-۴) نتایج به دست آمده از دابیچی های درجه۲ و۱۰و ۲۰ برای نمونه های ۱۰ ثانیه ای

۱۰ثانیه‌ای	آلمانی	اسپانیایی	ماندارین	ویتنامی	ژاپنی	کره ای	تامیل	فارسی
انگلیسی	۶۲-۵۹-۴۸	۶۷-۶۱-۵۶	۴۲-۵۶-۵۳	۵۶-۵۶-۵۳	۶۷-۶۱-۵۵	۴۵-۵۶-۵۹	۵۶-۶۲-۴۲	_-۴۸-۵۲
آلمانی	-	۶۴-۵۳-۵۳	۵۶-۵۹-۴۸	۶۲-۴۸-۶۲	۵۵-۵۸-۵۵	۵۹-۷۰-۵۰	۵۰-۷۳-۶۴	۳۹-۴۸-۴۸
اسپانیایی	-	-	۴۵-۵۳-۶۴	۵۳-۵۰-۶۲	۶۱-۴۹-۶۷	۵۶-۶۲-۵۶	۵۳-۶۴-۵۰	۵۶-۵۶-۵۳
ماندارین	-	-	-	۵۰-۴۵-۵۶	۵۸-۴۶-۵۵	۶۴-۴۲-۵۰	۵۹-۵۹-۶۲	۵۹-۴۵-۵۹
ویتنامی	-	-	-	-	۷۰-۷۰-۷۶	۶۴-۴۲-۵۰	۶۲-۴۸-۶۴	۶۴-۷۰-۵۹
ژاپنی	-	-	-	-	-	۴۹-۵۸-۵۵	۶۴-۵۸-۷۰	۵۲-۵۲-_
کره ای	-	-	-	-	-	-	۴۸-۵۰-۵۶	۵۶-۴۵-۵۶
تامیل	-	-	-	-	-	-	-	۶۲-۵۳-۵۶

۶.۲) نتایج آزمایش‌های انجام شده بر روی ضرایب کپسترال:

پس از ضرایب موجک نوبت به ضرایب کپسترال می‌رسد که باید مورد آزمایش قرار گیرند. اما می‌دانیم ضرایب کپسترال بیشتر به صورت ترکیبی استفاده می‌شوند، مثلاً ترکیب ۷ ضریب MFCC و هر کدام از این ضرایب ترکیب، به درصد صحت متفاوتی می‌رسند. ما در اینجا از ترکیب‌های ۵،۷،۹،۱۲،۱۳ تایی ضرایب کپسترال استفاده نموده‌ایم. با توجه به اینکه ضرایب کپسترال در برنامه پنجره‌بندی می‌شوند، برای هر ترکیب پنجره‌های مختلف را آزموده و بهترین پنجره‌بندی را انتخاب نمودیم. همچنین هر پنجره نسبت به پنجره قبلی ۵۰٪ همپوشانی دارد.

۶.۲.۱) ضرایب MFCC:

ابتدا از ضرایب MFCC استفاده کرده و نتایج تشخیص دوبه دوی زبان را با استفاده از این ضرایب بدست آوردیم. برای این منظور ضرایب MFCC به دست آمده از قسمت استخراج ویژگی را به ازای ترکیب‌های مختلف بررسی کرده و پس از تعیین پنجره‌بندی مناسب برای هر ضریب، آنها را به برنامه اصلی می‌دهیم و نتایج را ثبت می‌کنیم. در(جدول۶-۵) و(جدول ۶-۶) می‌توان این نتایج را به ترتیب برای نمونه‌های ۱۰ثانیه‌ای و ۴۵ ثانیه‌ای مشاهده نمود:

(جدول ۶-۵): نتایج به دست آمده ازضرایب گوناگون MFCC برای نمونه های ۱۰ ثانیه ای

MFCC ۱۰ثانیه‌ای	آلمانی					اسپانیایی					ژاپنی					ماندارین				
تعداد ترکیبات	۵	۷	۹	۱۲	۱۳	۵	۷	۹	۱۲	۱۳	۵	۷	۹	۱۲	۱۳	۵	۷	۹	۱۲	۱۳
انگلیسی	۶۲	۶۲	۵۹	۵۹	۵۹	۵۶	۵۹	۴۸	۵۳	۵۳	۶۷	۶۷	۶۷	۶۷	۶۷	۴۲	۵۶	۵۶	۴۲	۴۲
آلمانی	-					۶۷	۶۷	۴۸	۶۴	۶۴	۶۷	۶۷	۶۷	۶۷	۶۷	۵۹	۵۹	۵۹	۵۹	۵۰
اسپانیایی	-					-					۴۹	۵۵	۵۵	۵۵	۵۵	۴۸	۵۳	۴۸	۵۳	۵۳
ژاپنی	-					-					-					۵۵	۵۵	۵۵	۵۵	۵۵

در(جدول ۶-۵) نتایج حاصل از بهترین پنجره‌بندی ضرایب MFCC، به ازای ترکیبات متفاوت، برای نمونه‌های ۱۰ ثانیه‌ای و در (جدول۶-۶) نتایج برای نمونه‌های ۴۵ ثانیه‌ای آورده شده است. باید توجه داشت که پنجره‌بندی برای نمونه‌های ۱۰ ثانیه‌ای و ۴۵ ثانیه‌ای نیز متفاوت خواهد بود.

(جدول ۶-۶) نتایج به دست آمده ازضرایب MFCC برای نمونه های ۴۵ ثانیه ای

MFCC ۴۵ثانیه‌ای	آلمانی					اسپانیایی					ژاپنی					ماندارین				
تعداد ترکیبات	۵	۷	۹	۱۲	۱۳	۵	۷	۹	۱۲	۱۳	۵	۷	۹	۱۲	۱۳	۵	۷	۹	۱۲	۱۳
انگلیسی	۶۴	۷۳	۷۳	۷۳	۷۳	۵۰	۶۴	۵۶	۵۶	۵۶	۵۰	۵۰	۵۰	۵۰	۵۰	۵۹	۵۹	۵۰	۵۳	۵۳
آلمانی	-					۴۵	۴۵	۵۰	۵۰	۵۰	۶۲	۶۲	۶۲	۶۲	۶۲	۵۹	۵۹	۵۹	۶۲	۶۲
اسپانیایی	-					-					۵۹	۵۰	۵۰	۵۰	۵۶	۶۷	۶۷	۶۷	۵۹	۵۹
ژاپنی	-					-					-					۵۳	۵۳	۵۳	۵۳	۵۳

با مقایسه (جدول ۶-۵) و (جدول ۶-۶) می‌توان دید که در کل، نتایج بر روی نمونه‌های ۴۵ ثانیه‌ای بهتر از نمونه‌های ۱۰ ثانیه‌ای است. همچنین ترکیب ۷ ضریب MFCC در بیشتر نمونه‌ها جواب بهتری داشته است.

۶.۲.۲) ضرایب PLP:

پس از به دست آوردن درصد صحت تشخیص حاصل از ضرایب MFCC نوبت به ضرایب PLP می‌رسد. در اینجا نیز باید ابتدا ترکیبات مختلف را بررسی کرده و سایز پنجره‌بندی مناسب را برای هر یک از ترکیب‌ها به‌دست آوریم. نتایج حاصل از نمونه‌های صوتی ۱۰ثانیه‌ای و ۴۵ ثانیه‌ای را می‌توان به ترتیب در (جدول۶- ۷) و (جدول ۶-۸) مشاهده نمود:

(جدول ۶-۷) نتایج به دست آمده ازضرایب PLP برای نمونه های ۱۰ ثانیه ای

PLP ۱۰ثانیه‌ای	آلمانی					اسپانیایی					ژاپنی					ماندارین				
تعداد ترکیبات	۵	۷	۹	۱۲	۱۳	۵	۷	۹	۱۲	۱۳	۵	۷	۹	۱۲	۱۳	۵	۷	۹	۱۲	۱۳
انگلیسی	۵۹	۵۶	۵۶	۵۹	۶۲	۶۴	۴۸	۵۶	۵۶	۷۰	۷۰	۷۹	۵۵	۶۷	۵۴	۵۰	۴۸	۵۹	۶۲	۵۰
آلمانی	-					۵۳	۵۳	۶۲	۵۶	۶۲	۵۸	۵۲	۵۵	۵۹	۵۸	۵۹	۶۷	۵۹	۵۶	۶۲
اسپانیایی	-					-					۴۹	۶۴	۵۵	۵۲	۵۲	۵۰	۴۸	۶۲	۵۹	۵۰
ژاپنی	-					-					-					۵۸	۶۱	۶۱	۵۸	۶۷

از(جدول۶-۷) مشخص است که ترکیب ۱۳ تایی و پس از آن ترکیب ۱۲ تایی برای بیشتر نمونه‌ها جواب‌های بهتری دارند. همچنین ترکیب ۷تایی برای دسته دوزبانه انگلیسی-ژاپنی به ضریب صحت ۷۹٪ رسیده است که بالاترین ضریب است.

(جدول ۶-۸) نتایج به دست آمده ازضرایب PLP برای نمونه های ۴۵ ثانیه ای

PLP ۴۵ثانیه‌ای	آلمانی					اسپانیایی					ژاپنی					ماندارین				
تعداد ترکیبات	۵	۷	۹	۱۲	۱۳	۵	۷	۹	۱۲	۱۳	۵	۷	۹	۱۲	۱۳	۵	۷	۹	۱۲	۱۳
انگلیسی	۵۹	۷۳	۶۷	۵۹	۷۵	۶۲	۷۵	۵۰	۵۹	۷۰	۴۸	۶۴	۵۶	۵۹	۵۹	۴۸	۵۹	۶۴	۵۹	۵۶
آلمانی	-					۶۷	۵۹	۵۳	۵۹	۵۶	۵۰	۵۶	۵۰	۵۶	۵۹	۵۶	۵۳	۵۶	۶۲	۶۷
اسپانیایی	-					-					۵۹	۶۲	۵۹	۶۴	۵۳	۶۴	۶۴	۶۲	۵۳	۶۲
ژاپنی	-					-					-					۵۳	۵۶	۶۴	۵۹	۶۲

در(جدول ۶-۸) نیز ترکیب ۷تایی و ۱۳ تایی جواب‌های بهتری دارند. اما این روند نظم خاصی نداشته و در برخی موارد، ضرایب بدست آمده توسط سایر ترکیبات بهتر می‌باشند. در کل، نمونه‌های ۴۵ ثانیه‌ای نسبت به نمونه‌های ۱۰ثانیه‌ای به جواب‌های بهتری رسیده‌اند.

۶.۲.۳) ضرایب LPC:

به عنوان آخرین آزمایش، از ضرایب LPC بهره برده‌ایم و نتایج تشخیص زبان را با کمک برنامه پیشنهادی و با استفاده از این ضرایب، بر روی نمونه‌های ۱۰ ثانیه‌ای و ۴۵ ثانیه‌ای، به ازای پنجره‌بندی مناسب برای هر ترکیب بدست آوردیم که در (جدول ۶-۹) و (جدول ۶-۱۰) مشخص شده است:

(جدول ۶-۹) نتایج به دست آمده ازضرایب LPC برای نمونه های ۱۰ ثانیه ای

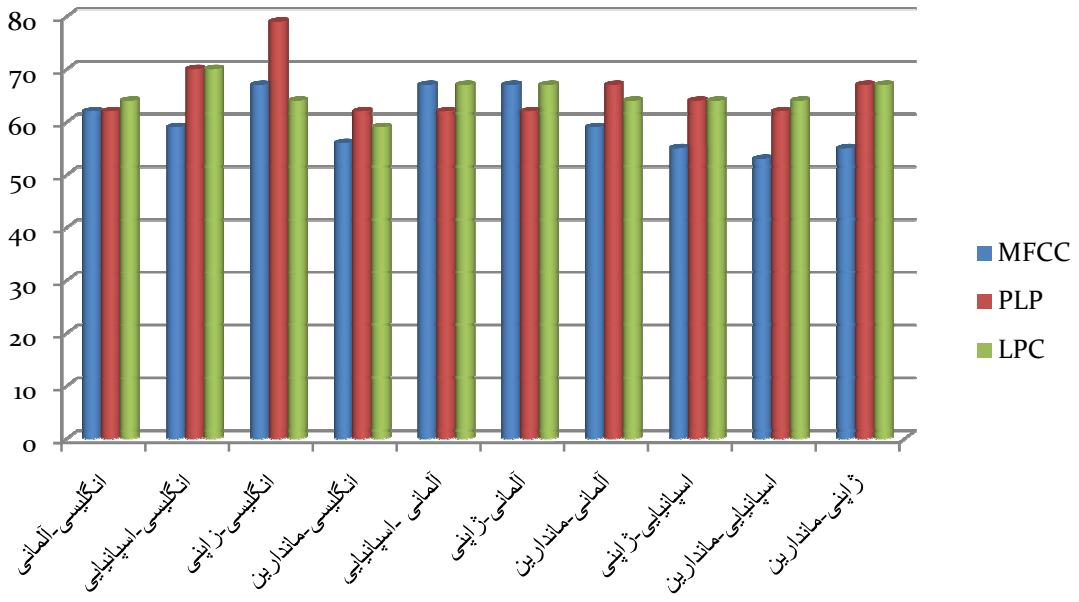
LPC ۱۰ثانیه‌ای	آلمانی					اسپانیایی					ژاپنی					ماندارین				
تعداد ترکیبات	۵	۷	۹	۱۲	۱۳	۵	۷	۹	۱۲	۱۳	۵	۷	۹	۱۲	۱۳	۵	۷	۹	۱۲	۱۳
انگلیسی	۶۱	۵۹	۵۶	۶۴	۶۴	۴۵	۵۳	۶۲	۷۰	۵۶	۵۵	۶۴	۴۹	۴۳	۶۱	۵۰	۵۹	۵۶	۵۹	۵۶
آلمانی	-					۵۳	۵۹	۶۲	۶۴	۶۷	۴۳	۶۷	۴۹	۴۳	۵۸	۵۰	۵۰	۶۴	۵۰	۶۲
اسپانیایی	-					-					۵۵	۶۴	۵۲	۵۲	۶۴	۵۰	۶۲	۵۰	۵۹	۶۴
ژاپنی	-					-					-					۵۸	۶۷	۶۴	۶۱	۶۷

(جدول ۶-۱۰) نتایج به دست آمده ازضرایب LPC برای نمونه های ۴۵ ثانیه ای

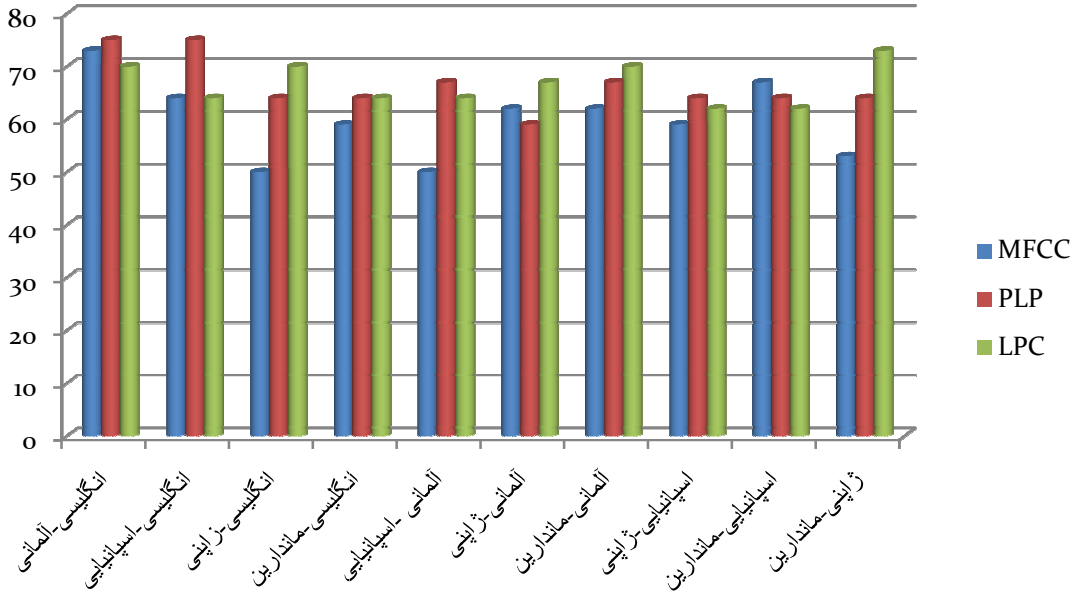
LPC ۴۵ثانیه‌ای	آلمانی					اسپانیایی					ژاپنی					ماندارین				
تعداد ترکیبات	۵	۷	۹	۱۲	۱۳	۵	۷	۹	۱۲	۱۳	۵	۷	۹	۱۲	۱۳	۵	۷	۹	۱۲	۱۳
انگلیسی	۵۶	۶۷	۶۷	۷۰	۶۴	۵۰	۶۴	۵۶	۵۰	۵۶	۵۶	۵۶	۵۰	۷۰	۶۲	۵۰	۵۹	۵۳	۵۶	۶۴
آلمانی	-					۵۹	۴۸	۶۴	۵۹	۵۶	۶۲	۵۶	۵۳	۶۷	۶۲	۵۳	۵۳	۷۰	۶۷	۴۸
اسپانیایی	-					-					۵۶	۵۳	۵۹	۵۳	۶۲	۶۲	۵۳	۶۲	۴۵	۶۲
ژاپنی	-					-					-					۵۹	۶۲	۷۳	۶۲	۶۴

در(جدول۶- ۹) ترکیب ۱۳ تایی و در (جدول ۶- ۱۰) ترکیب ۱۲ تایی بهترین درصد صحت را در بیشتر موارد نشان می‌دهند. باز هم نمونه‌های ۴۵ ثانیه‌ای در مقایسه با نمونه‌های ۱۰ ثانیه‌ای جواب‌های نسبتاً بهتری داشته است.

حال برای مقایسه ضرایب مختلف کپسترال و یافتن بهترین ضریب از میان آنها، نتایج حاصل از این ضرایب را در کنار هم گذاشته و بررسی می‌کنیم. با مشاهده (شکل۶-۱) و (شکل۶-۲) درمی‌یابیم که به ترتیب ضرایب PLP، LPC و MFCC بهترین نتایج را در میان کپسترال دارند:

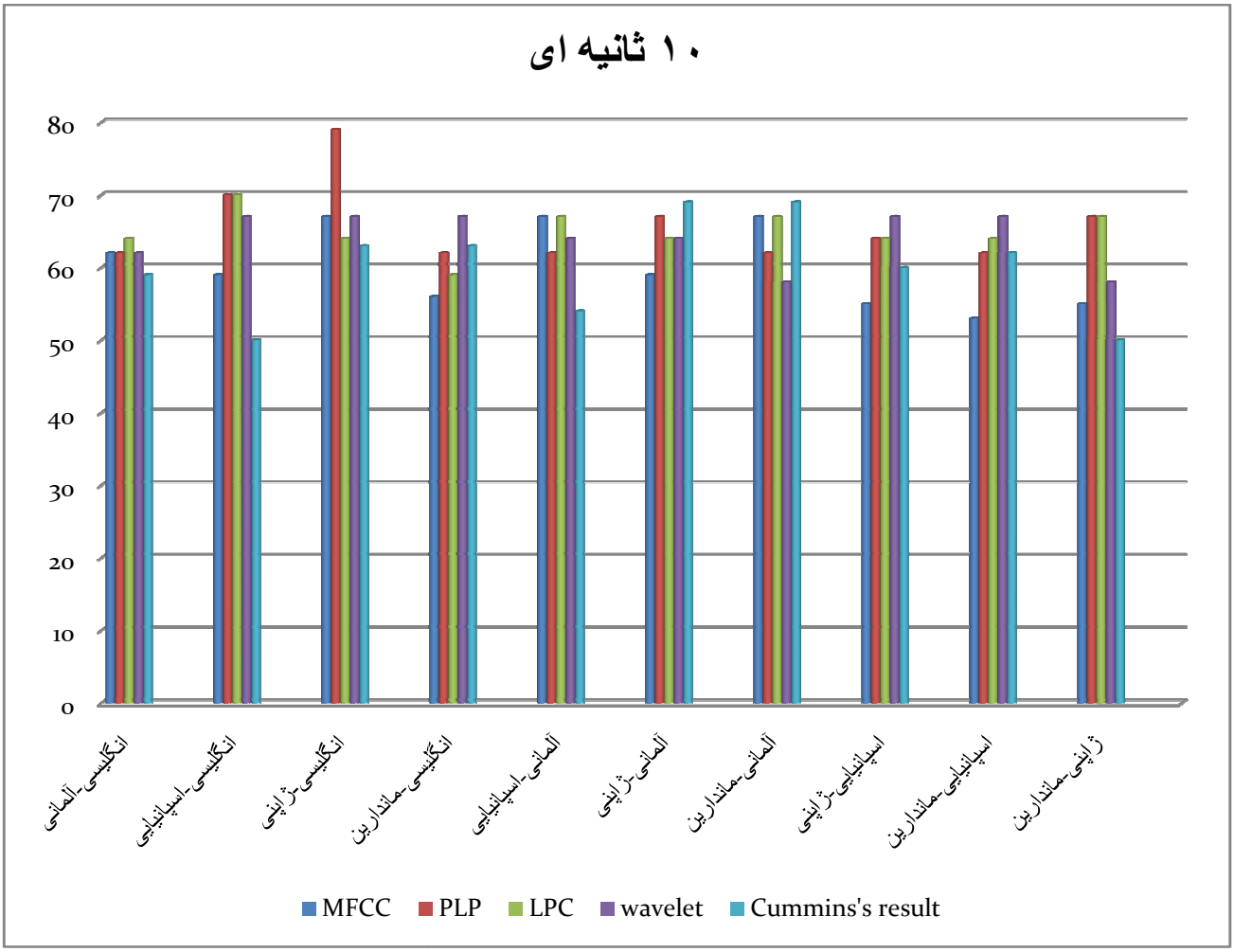


(شکل۶-۱) مقایسه بهترین نتایج حاصل از MFCC,PLP,LPC برای نمونه های ۱۰ ثانیه ای



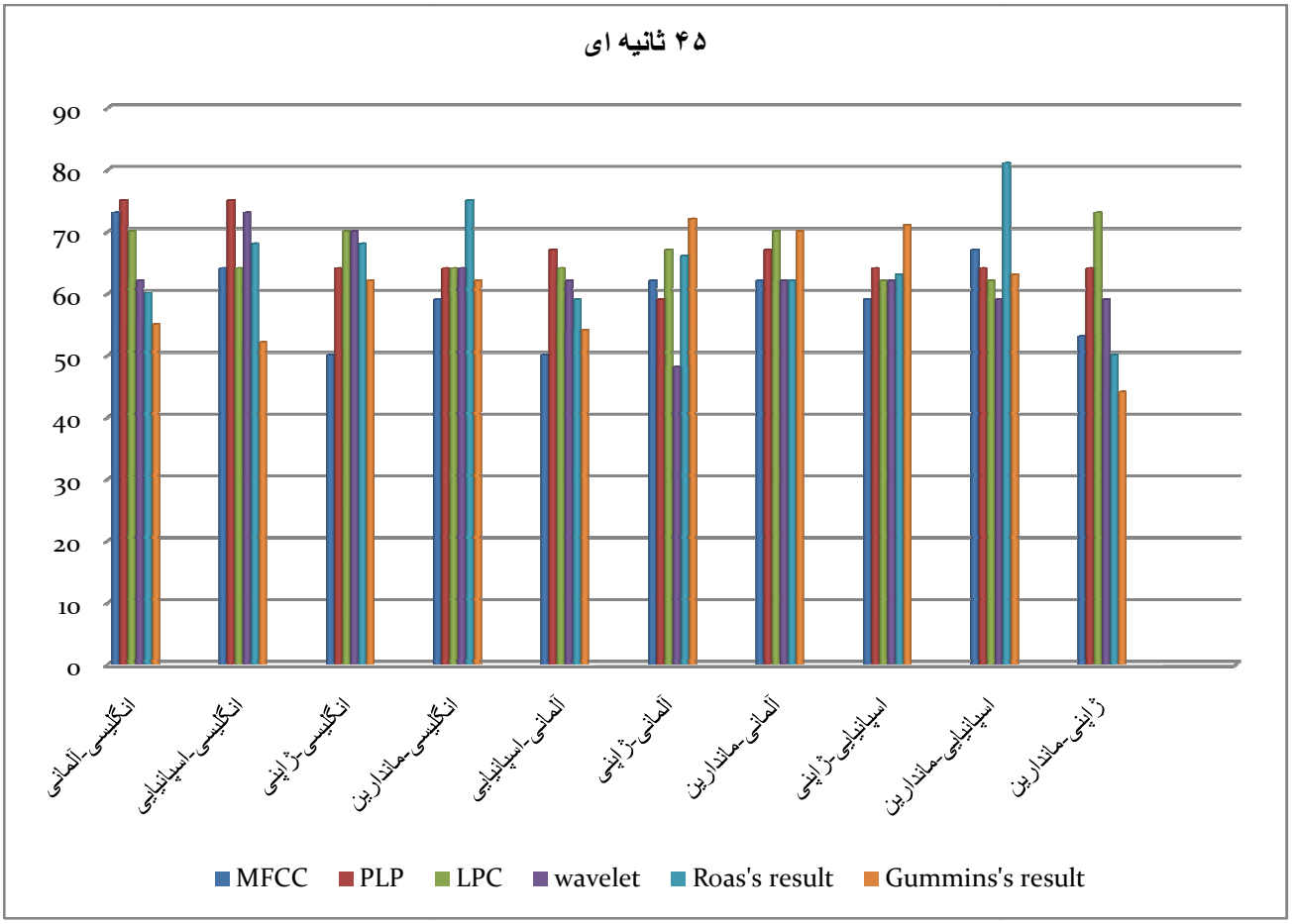
(شکل۶-۲) مقایسه بهترین نتایج حاصل از MFCC,PLP,LPC برای نمونه های ۴۵ ثانیه ای

در (شکل ۶-۳) برای نمونه‌های صوتی ۱۰ ثانیه‌ای، نتایج روش پیشنهادی، شامل نتایج ضرایب موجک، ضرایب PLP، LPC و MFCC ، با نتایج به دست آمده توسط گومینز مقایسه شده است. مشاهده می‌شود PLP بالاترین درصد صحت را نشان می‌دهد که حدود ۷۹٪ است. پس از آن در اکثر موارد به ترتیب، LPC ، موجک و در نهایت MFCC درصد صحت بهتری دارند. همچنین از (شکل ۶-۳) مشخص است در بیشتر موارد نتایج بدست آمده بهبود قابل توجهی نسبت به نتایج گومینز[۵۹] دارد.



(شکل ۶-۳) مقایسه نتایج حاصل از ضرایب موجک، ضرایب PLP، LPC و MFCC با نتایج به دست آمده توسط گومینز برای نمونه های صوتی ۱۰ ثانیه ای

همچنین در (شکل ۶-۴) برای نمونه‌های صوتی ۴۵ ثانیه‌ای، نتایج حاصل از ضرایب موجک، ضرایب PLP، LPC و MFCC با نتایج به دست آمده توسط *رواس* [۵۴] و*گومینز*[۵۹] مقایسه شده است. در اینجا نیز PLP با ضریب صحت ۷۵٪ بالاترین ضریب صحت را دارد و پس از آن در اکثر موارد به ترتیب، LPC ، موجک و در نهایت MFCC ضریب صحت بهتری دارند.



(شکل ۶-۴) مقایسه نتایج حاصل از ضرایب موجک، ضرایب PLP، LPC و MFCC با نتایج به دست آمده توسط رواس و گومینز برای نمونه های صوتی ۴۵ ثانیه ای

در (شکل ۶-۴) که مربوط به نمونه‌های ۴۵ ثانیه‌ای می‌باشد، نتایج به دست آمده توسط *رواس* [۵۴] و گومینز [۵۹] بهبود قابل توجهی نسبت به نمونه‌های ۱۰ ثانیه‌ای(جدول ۶-۳) دارد که به دلیل افزایش طول نمونه‌های صوتی می‌باشد. زیرا برنامه استفاده شده توسط آنها، به زمان وابسته بوده و با افزایش

طول نمونه‌ها، درستی تصمیم‌گیری افزایش می‌یابد. اما برنامه پیشنهادی در این پایان نامه وابسته به فرکانس نمونه‌ها است و زمان تأثیر چندانی بر آن ندارد.

۶.۳) نتیجه‌گیری و پیشنهادات

تا کنون روش‌های مختلفی برای شناسایی زبان گفتاری به صورت خودکار پیشنهاد شده است، که بیشتر آنها وابسته به اطلاعات واج‌آرایی بوده و استفاده از آنها دشوار می‌باشد. ما در این پژوهش روشی مستقل از واج‌آرایی ارائه دادیم که در عین سهولت، با درصد خوبی قادر به تشخیص زبان‌ها است، زیرا روش ما مستقل از پیچیدگی‌های زبانی بوده و همه نمونه‌های صوتی را به حوزه زمان می‌بریم و بر روی آنها پردازش یکسانی انجام می‌دهیم. در این روش از تبدیل موجک و تبدیل کپسترال نمونه‌های صوتی استفاده گردیده که بدون نیاز به اطلاعات زبان‌شناسی، بر روی زبان‌های گوناگون قابل استفاده می‌باشند. مشاهده گردید که ضرایب کپسترال به درصد صحت بالاتری نسبت به ضریب موجک می‌رسند. همچنین برای هر دو ضریب کپسترال و موجک، نمونه‌های صوتی ۴۵ ثانیه‌ای در بیشتر موارد، درصد تشخیص بهتری نسبت به نمونه‌های ۱۰ثانیه‌ای دارند. اما در برخی از نمونه‌ها، این افزایش درصد تشخیص مشاهده نمی‌شود و این می‌تواند به دلیل محدودیت تعداد نمونه‌های آموزشی ومساوی شدن امتیازهای چندین ویژگی باشد که در این موارد، برنامه اولین ویژگی را به عنوان برترین ویژگی می‌پذیرد. روش‌های پیشین بیشتر به تشخیص دوبه دوی زبان‌ها می‌پرداختند، در حالیکه روش پیشنهادی قادر به تشخیص نوع زبان، از میان ۹ زبان موجود در OGI-TS نیز می‌باشد.

برای ایجاد بهبود در نتایج می‌توان تعداد نمونه‌های آموزشی و تست را افزایش داد. استفاده از چندین فیلتر پایین‌گذر پشت سرهم، در قسمت استخراج ویژگی ممکن است باعث گردد ضرایب مفیدتری استخراج شده و در پردازش‌های بعدی به ضرایب بهتری برسیم و در نتیجه، درصد تشخیص بهبود

یابد. همچنین استفاده از کلاسه بندهای پیچیده‌تر غیر خطی، مانند شبکه های عصبی برای الگوریتم تشخیص زبان می‌تواند مفید باشد و نتایج بهتر به همراه داشته باشد.

به عنوان کارهای بعدی میتوان روش‌های تشخیص خودکار زبان گفتاری را بر روی لهجه‌ها و گویش‌های متفاوت زبان فارسی اعمال نمود.

مراجع:

[1] Pavel Mat·ejka, Jan ·Cernocky, Milan Sigmund**"Introduction to Automatic Language Identification",** Faculty of Electrical Engineering and Communication, BUT,2004

[2] Y.K. Muthusamy, E. Barnard, and R.A. Cole, "**Reviewing Automatic Language Identification"**, *in IEEE signal processing magazine*, Vol. 11, no. 4, pp.33-41, October 1994

[3] Marc *A.* Zissman, "**Language Identification Using Phoneme Recognition And Phonotactic Language Modeling",** Lincoln Laboratory, Massachusetts Institute of Technology 244 Wood Street Lexington. MA03, 173-91 08 USA

[4] Y. K. Muthisamy. N. Jain and R. A. Cole**. "Perceptual Benchmarks for Automatic Language Identification"**. In *IC.44SP '94 Proceedings,* volume 1, pages 333-336, April 1994.

[5] Mary, L.; Yegnanarayana, B.; **"Prosodic Features for Language Identification"** Signal Processing, Communications and Networking, ICSCN '08. International Conference on 4-6 Jan.2008 Page(s):57–62Digital ObjectIdentifier 10.1109/ICSCN.2008.4447161, 2008.

[6] Schultz T., et al, *"***Multilingual speech proces*sing***"**, *ISBN* 13: 978-0-12-088501-5, Elsevier, 2006.

[7] Navrátil J., "**Automatic Language Identification**", in *multilingual speech processing*, T. Schultz & K. Kirchhoff eds., Elsevier, 2006.

[8] Zissman M., Berkling K., "**Automatic Language Identification**", *Speech Communication*, pp.115-124, Vol. 35, Issues 1-2, August, 2001.

[9] C. Lin and H. Wang, **"Language Identification Using Pitch Contour Information,"** *in Proc. Int. Conf. Acoust., Speech and Signal processing, Philadelphia*, USA, Apr.2005, vol. 1, pp. 601-604.

[10] Ana Lilia Reyes-Herrera, Luis Villaseñor-Pineda And Manuel Montes-y-Gómez **"Automatic Language Identification Using Wavelets"** Language Technologies Group Computer Science Department National Institute of Astrophysics, Optics and Electronics Luis Enrique Erro #1, Tonantzintla, Puebla, 72840, M

[11] Zissman M., Gleason T., Rekart D., Losiewicz B., "**Automatic Dialect Identification of Extemporaneous Conversational, Latin American Spanish Speech**", *Proc. of ICASSP-96*, Atlanta, Georgia, 1996.

[12] Kumpf K., King R.W., "**Automatic accent classification of foreign Accented Australian English Speech**", *Proc. of ICSLP-96*, Philadelphia, October, 1996.

[13] Torres-Carrasquillo P., Gleason T., Reynolds D., "**Dialect identification using Gaussian Mixture Models**", *Proc. of Odyssey-04*, Toledo, 2004. Spanish dialects

[14] Barkat M., et al, "**Prosody as a Distinctive Feature for the Discrimination of Arabic Dialects**", *Proc. of Eurospeech-99*, Budapest, Hungary, September 1999.

[15] Bartkova K, Jouvet D., "**On using units trained on foreign data for improved multiple accent speech recognition**", *Speech Communication*, Vol.49, 10-11, pp.836-846, 2007.

[16] Compernolle (van) D., "**Speech recognition by Goats, Wolves, Sheep and… Non natives**", *Speech Communication*, Volume 35, 2001, pp.71-79.

[17] Teixeira C., Trancoso I., Serralheiro A., "**Recognition of Non-Native Accents**", *Proc.of Eurospeech-97*, Rhodos, Greece, September, 1997.

[18] Wanneroy R., Bilinski E., Barras C., Adda-Decker M., Geoffrois E., "**Acoustic-Phonetic Modeling of Non-Native Speech for Language Identification**", *Proc. of MIST-99*, Leusden, The Netherlands, September, 1999.

[19] Witt, S., Young, S., "**Off-line acoustic modelling of non-native accents",** *Proc. Of Eurospeech-99*,, pp.1367-1370, Budapest, Hungary, September, 1999.

[20] Crystal D. "**The Cambridge Encyclopedia of Language**", *Cambridge University Press*, Cambridge UK, 1997.

[21] Comrie B., "**The World's Major Languages**", *Oxford University Press*, Oxford UK, 1990.

[22] Gordon R.G., (ed.),. *Ethnologue: "***Languages of the World**", *15th edition. Dallas, Tex. SIL International. Online version*: 2005 http://www.ethnologue.com/

[23] Kirchhoff K., "**Language Characteristics**", in *multilingual speech processing*, T.Schultz & K. Kirchhoff eds., Elsevier, 2006.

[24] J. Mehler, P. Jusczyk, G. Lambertz, N. Halsted, J. Bertoncini, and C. Amiel-Tison,"**A precursor to language acquisition in young infants**," *Cognition*, 29:143-178, 1988.

[25] T. J. Hazen and V.W. Zue. "**Recent improvements in an approach to segment-based automatic language identification**". *ICSLP*, Yokohama, Japan, 1994

[26] E. Singer, P.A. Torres-Carrasquillo, T.P. Gleason, W.M. Campbell, and D.A. Reynolds, "**Acoustic, Phonetic, and Discriminative approaches to Automatic Language Identification**," in Proc. *Eurospeech* 2003, pp. 1345-1348, Geneva, Switzer-land, Sept. 2003

[27] Rong Tong, Bin Ma, Donglai Zhu, Haizhou Li and Eng Siong Chng "**Integrating Acoustic, Prosodic and Phonotactic features for Spoken langauge identification**",*ICASSP* , May 14-19 Toulous, France, 2006

[28] Y.K. Muthusamy, R. Cole, B. Oshika, "**The OGI Multilanguage Telephone Speech Corpus**". *International Conference on Spoken Language Processing*, volume 2, Alberta Canada, 1992

[29] Mariani J., Paroubek P. "**Human Language Technologies Evaluation in the European Framework**", *Proc. of the DARPA Broadcast News Workshop*, Herndon, VA, pp.237-242, 1998.

[30] Mariani J., "**Developing Language Technologies with the Support of Language Resources and Evaluation Programs**", *Journal of Language Resources and Evaluation*, Springer Netherlands, 2005.

[31] ELDA: European Language Resources Distribution Agency. http://www.elda.org

[32] LDC, http://www.ldc.upenn.edu/Catalog/

[33] Leonard R., Doddington G., "**Automatic Language Identification**", *Technical report* RADC-TR-74-200, Air Force Rome Air Development Center, August, 1974.

[34] House A.S., Neuburg E.P., "**Toward Automatic Identification of the Language of an Utterance Preliminary Methodological Considerations**", *Journal of the Acoustical Society of America*, (JASA), Vol.62, no3, pp.708-713, September, 1977.

[35] Leonard, R.G. "**Language Recognition Test and Evaluation**". *Technical Report RADCTR-80-83*, Air Force Rome Air Development Center, March 1980.

[36] Li, K.P., Edwards, T.J. "**Statistical Models for Automatic Language Identification**". *Proc. ICASSP'80*, pp 884-887, April 1980.

[37] Cimarusti, D., Ives, R.B." **Development of an Automatic Identification System of Spoken Languages**": Phase 1. *Proc. ICASSP'82*, pp. 1661-1664, May 1982.

[38] Foil, J.T." **Language Identification Using Noisy Speech**", *Proc. ICASSP'86*, pp. 861-864, April 1986.

[39] Goodman, F.J., Martin, A.F., Wohlford, R.E. "**Improved Automatic Language Identification in Noisy Speech**". *Proc. ICASSP'89*, pp. 528-531, May 1989.

[40] Sugiyama, M. "**Automatic Language Recognition Using Acoustic Features**". *Proc. ICASSP'91*, pp. 813-816, May 1991.

[41] Nakagawa, S., Ueda, Y., Seino, T." **Speaker-independent, Text-independent Language Identification by HMM**". *Proc. ICSLP'92*, pp. 1011-1014, October 1992.

[42] Muthusamy, Y.K. *"***A Segmental Approach to Automatic Language Identification**". PhD thesis, *Oregon Graduate Institue of Science and Technology*, October 1993.

[43] OGI Multi Language Telephone Speech. *www.cslu.ogi.edu/corpora/mlts/*, Januar 2004.

[44] Yan, Y. "**Development of an Approach to Language Identification Based on Languagedependent Phone Recognition**". PhD thesis, *Oregon Graduate Institue of Science and Technology*, October 1995.

[45] Schultz, T., Rogina, I., Waibel, A. "**LVCSR-Based Language Identification**". *Proc. ICASSP'96*, pp. 781-784, May 1996.

[46] Berkling, K., Reynolds, D., Zissman, M.A." **Evaluation of Confidence Measures for Language Identification**". *Proc. Eurospeech'99*, vol. 1, pp. 363-366, September 1999.

[47] Hombert, J.M., Maddieson, I. "**the Use of 'Rare' Segments for Language Identification**". *Proc. Eurospeech'99*, vol. 1, pp. 379-382, September 1999.

[48] Navrátil, J. "**Spoken Language Recognition–a Step Toward Multilinguality in Speech Processing**", *IEEE Trans. Speech Audio Processing*, vol. 9, pp. 678-685, September 2001.

[49] Jayram, A.K.V.Sai, Ramasubramanian, V., Sreenivas, T.V. "**Automatic Language Recognition Using Acoustic Sub-word Units**". *Proc. ICSLP'02*, pp. 81-84, 2002.

[50] Adami, A.G., Hermansky, H." **Segmentation of Speech for Speaker and Language Recognition**". *Proc. Eurospeech'03*, pp. 841-844, September 2003.

[51] Singer, E., Torres-Carrasquillo, P.A., Gleason, T.P., Campbell, W.M., Reynolds, D.A. "**Acoustic, Phonetic and Discriminative Approaches to Automatic Language Identification**". *Proc. Eurospeech'03*, pp. 1345-1348, September 2003.

[52] U. M. Fayyad and K. B. Irani, **"Multi-Interval Discretization of Continuous-Valued Attributes"**, Proc.*13th Int'l Joint Conf.artificial Intelligence,* PP, 1022-1027, 1993.

[53] Bo Yin; Ambikairajah, E.Fang Chen. **"Combining Cepstral and Prosodic Features in Language Identification"** *Pattern Recognition, 2006. ICPR 2006.18th International Conference* on Volume 4, 0-0 0 Page(s):254 – 257 Digital Object Identifier 10.1109/ICPR.2006.381

[54] J. Rouas, J. Farinas, F. Pellegrino, and R. André- Obrecht,"**Modeling Prosody For Language Identification on Read and Spontaneous Speech**," in *Proc. of ICASSP'2003*, Hong Kong, China, April 2003, Vol. I, pp. 40-43.

[55] J. Mariethoz and S. Bengio, "**an Alternative to Silence Removal for Text-Independent Speaker Verification,**" *IDIAP*, *Research Report IDIAP*-RR 03-51, December 19, 2003 2003.

[56] B. Milner, "**A Comparison of Front-End Configurations for Robust Speech Recognition,**" *presented at Acoustics, Speech, and Signal Processing, IEEE International Conference on (ICASSP),* 2002

[57] J. Gutierrez, J. L. Rouas, and R. Andre-Obrecht, "**Fusing Language Identification Systems Using Performance Confidence Indexes**," presented at *Acoustics, Speech, and Signal Processing, IEEE International Conference on (ICASSP)*, 2004.

[58] P. Boersma, "**Accurate Short-Term Analysis of The Fundamental Frequency and the Harmonics-to-Noise Ratio of a Sampled Sound,**" in *IFA Processing 17*, University of Amsterdam, pp. 97-110, 1993.

[59] F. Cummins, F. Gers, and J. Schmidhuber,"**Language Identification from Prosody Without Explicit Features**," in *EUROSPEECH'99*,Budapest, Hungary, September 1999, pp.371-374.

[60] M. **A.** Zissman and E. Singer. **"Automatic Language Identification of Telephone Speech Messages Using Phoneme Recognition and N-Gram Modeling.**" In *ICASSP '94 Proceedings,* volume 1, pages 305-308, April 1994.

[61] T. J. Hazen and V. W. Zue."**Recent Improvements in an Approach to Segment-Based Automatic Language Identification**." In *ICSLP '94 Proceedings,* 1994.

[62] R. C. F. Tucker, M. J. Carey, and E. *S.* Parris. "**Automatic Language Identification Using Sub-Word Models**". In *ICASSP'94 Proceedings,* volume 1, pages: 301-304**,** April 1994.

[63] L. F. Lamel and J. L. Gauvain. "**Language Identification Using Phone-Based Acoustic Likelihoods".** In *ICASSP '94 proceedings,* volume 1, pager 293-296, April 1994.

[64] B. J. Shanon, Phd thesis, "**Speech Recognition and Enhancement Using Autocorrelation Domain Processing**"; *School of Engineering*, Griffith University,Brisbane,Australia, 2006.

[65] J. W, Picone,"**Signal Modeling Techniques in Speech Recognition**", Proceedings of the IEEE, Vol.81, No.9, pp.1215-1247., 1993.

[66] J. Makhol "**Linear Prediction: aTutorial Review**", *Procceding of the IEEE*, Vol.63, No.4, pp.561-580.**,** 1975.

[67] L. Rabiner and R. W. Schafer, **"Digital Processing of Speech Signals"**, Prentice Hall, USA. 1978

[68] L. R. Rabiner and R.W. Schafer, **"Fundamentals of Speech Recognition"**, Prentice Hall, USA, 1993..

[69] X. Huang, A. Acero and H. W. Hon, **"Spoken Language Processing: a Guide to Theory, Algorithm, and System Development"**, Prentice Hall, USA, 2001.

[70] S. B. Davis and P. Mermelstein**, "Fundamentals of Speech Recognition"**, Prentice Hall, USA, 1980.

[71] A. Koc, MS thesis, "**Acoustic Feature Analysis for Robust Speech Recognition"**, Bogazici University, Turkey, 2002.

[72] M. Rosell, "**an Introduction to Front-End Processing and Acoustic Features for Automatic Speech Recognition**", Term Paper in Swedish National Graduate School of Language Technology,2006.

[73] B. J. Sannon, PhD thesis, "**Speech Recognition and Enhancement using Autocorrelation Domain Processing**", *School of Engineering Griffith University*, Brisbane, Australia,2006.

[74] H. Hermansky, "**Perceptual Linear Predictive (PLP) Analysis of Speech**", J.Acoust.Soc.Am, Vol.87, No.4, pp.1738-1752**,** 1989.

[75] J. Psutka, L. Muller and J.V. Psutka, "**Comparison of MFCC and PLP Parameterizations in the Speaker Independent Continuous Speech Recognition Task** ", Eurospeech, Scandinavia, 2001.

[76] M. Gupta and A. Gilbert, "**Robust Speech Recognition Using Wavelet Coefficient Features**", in *IEEE Automatic Speech Recognition And Understanding Workshop*, USA, pp. 445-448, 2001.

[77] R. Modic, B. Lindberg, B. Petek. "**Comparative Wavelet And MFCC Speech Recognition Experiments On the Slovenian and English SpeechDat2**", in *ISCA Tutorial and Research Workshop on non-linear speech processing (NOLISP 03),* Le Croisic, France, 2003.

[78] Ching-Tang Hsieh, Eugene Lai, You-Chuang Wang: "**Robust Speaker Identification System Based on Wavelet Transform and Gaussian Mixture Model**". J. Inf. Sci. Eng. 19(2): 267-282, 2003**.**

[79] Johnson Ihyeh Agbinya **"Discrete Wavelet Transform Techniques in Speech Processing"** In *IEEE TENCON - Digital Signal Processing Applications1996 CSIRO Division of Radio physics*, Marsfield, P 0 Box 76, Epping, New South Wales, Australia**,**1996.

[80] S. Mallat, "**A Wavelet Tour of Signal Processing**", *Academic Press*. USA 1998.

[81] Hall, M. and Holmes, G. (2003)." **Benchmarking Attribute Selection Techniques for Discrete Class Data Mining**". *IEEE Transactions on Knowledge and Data Engineering*. 15(3), November/December 2003.

[۸۲] ح. اخلاق، (۱۳۸۸)، پایان نامه ارشد، "استخراج ویژگی مبتنی بر پردازش در حوزه اتوکرولیشن جهت بازشناخت گفتار با استفاده از HTK"، دانشکده مهندسی برق و رباتیک، دانشگاه صنعتی شاهرود.

پیوست

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% wavelet coefficient1
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

clc;clear all;
L=[];
w=1;%  define the first value for w
a=struct('wav',[],'wav2',[],'s',[],'num',[],'num2',[]);%creates a structure
array with the specified fields and values.
P=64;%(the number of sampel files)
for p=1:P

filename=['C:\users\asr\Desktop\OGI2\45\En-Ger\a\','k',int2str(p),'.wav'];

%filename=['C:\users\asr\Desktop\OGI2\45\En-Jap\a\','k',int2str(p),'.wav'];

%filename=['C:\users\asr\Desktop\OGI2\45\En-Man\a\','k',int2str(p),'.wav'];

%filename=['C:\users\asr\Desktop\OGI2\45\En-Spa\a\','k',int2str(p),'.wav'];

%filename=['C:\users\asr\Desktop\OGI2\45\Ge-Jap\a\','k',int2str(p),'.wav'];

%filename=['C:\users\asr\Desktop\OGI2\45\Ge-Man\a\','k',int2str(p),'.wav'];

%filename=['C:\users\asr\Desktop\OGI2\45\Ge-Spa\a\','k',int2str(p),'.wav'];

%filename=['C:\users\asr\Desktop\OGI2\45\Jap-Man\a\','k',int2str(p),'.wav'];

%filename=['C:\users\asr\Desktop\OGI2\45\Jap-Spa\a\','k',int2str(p),'.wav'];

%filename=['C:\users\asr\Desktop\OGI2\45\Man-Spa\a\','k',int2str(p),'.wav'];

    b =wavread(filename);% reads a WAVE file

    b2=b.';% change row to column without

    siz2=size(b2,2); % calculate the number of columns of b2.
    a(w).wav=b2;% save the vector b2.
    a(w).s=siz2;%save the size of the vector b2.
    w=w+1;

    L(1,p)=siz2; %the size of the vector b2.
end
```

```
Le=min(L); % find minimum size between all vectors b2.


for w=1:P %(the number of sampel files)
    a(w).wav2(1,:)=a(w).wav(1,1:Le);% resize all vectors to minimum size of
b2.
    c=a(w).wav2(1,:);% uses any rows of data.

    [cA,cD]=dwt(c,'db20'); %uses the Daubechies db2 wavelet transform.

    % maintaining just the 1% of the originals.
    lca1=numel(cA);

    Z=zeros(1,lca1);
    [sortca,IXmax]=sort(abs(cA),'descend');

    for i=1:((1/100)*lca1)
        Z(IXmax(i))=cA(IXmax(i));
    end

    if w==1
        M2=1:lca1;
    end

    M2(w+1,:)=Z; % make a matrix from vectors M

    for i=1:lca1
        ZZ(1,i,w)= cA(i);
    end
    M(w,:)=M2(w+1,:);
end

for w=1:P
    ZZZ(w,:)=ZZ(1,:,w); % make a matrix from vectors M
end

% select the relevant coefficient for both language(those with non-zero
values) .
R1=[];
count=zeros(1,size(M,2));
j=1;
for r=1:(size(M,2))
    co=0;

    for p=1:P
        if M(p,r)==0
            co=co+1;
        end
    end
    count(r)=co;
    if count(r)~=P
        R1(1,j)=IXmax(r);

        j=j+1;
```

```
    end
end

RR1=size(M,2);


for r=RR1:-1:1
    if M(:,r)==0
        M2(:,r)=[];
        M(:,r)=[];

    end
end

Mclass=[1*ones(32,1); 2*ones(32,1)];
HCA=zeros(1,size(M,2));
HC=zeros(1,size(M,2));
IG=zeros(1,size(M,2));
m=1;
for p=1:size(M,2)
    Nf=0; Ne=0;
    N1=0; N2=0;
    n1=0; n2=0; n3=0; n4=0;
    for k=1:size(M,1)
        if Mclass(k)==1
            N1=N1+1;
            if M(k,p)==0
                n1=n1+1;
            else
                n2=n2+1;
            end
        elseif  Mclass(k)==2
            N2=N2+1;
            if M(k,p)==0
                n3=n3+1;
            else
                n4=n4+1;
            end
        end
    end

    Pc1a1=n1/(n1+n3);
    Pc1a2=n2/(n2+n4);
    Pc2a1=n3/(n1+n3);
    Pc2a2=n4/(n2+n4);
    Pa1=(n1+n3)/(N1+N2);
    Pa2=(n2+n4)/(N1+N2);
    Pc1=N1/(N1+N2);
    Pc2=N2/(N1+N2);
    if Pc1a1~=0
        HCA(1,p)=HCA(1,p)+(Pa1*(Pc1a1)*(log2(Pc1a1)));
    end
    if Pc2a1~=0
        HCA(1,p)=HCA(1,p)+(Pa1*(Pc2a1)*(log2(Pc2a1)));
    end
```

```
    if   Pc1a2~=0
        HCA(1,p)=HCA(1,p)+(Pa2*(Pc1a2)*(log2(Pc1a2)));
    end
    if Pc2a2~=0
        HCA(1,p)=HCA(1,p)+(Pa2*(Pc2a2)*(log2(Pc2a2)));
    end

    HCA(1,p)=-1*(HCA(1,p));

    if Pc1~=0
        HC(1,p)=HC(1,p)+(Pc1*log2(Pc1));
    end
    if Pc2~=0
        HC(1,p)=HC(1,p)+(Pc2*log2( Pc2));
    end
    HC(1,p)=-1*(HC(1,p));
    IG(1,p)=(HC(p)-HCA(p));

end
[A,indA]=sort(IG,'descend');
for m=1:4000
    sss(m)=M2(1,indA(m));
end


for l=1:numel(sss)
    MM(:,l)=ZZZ(:,(sss(l)));
end
M=MM;
Mclass=[1*ones(32,1); 2*ones(32,1)];
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% wavelet coefficient2
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

L=[];
w=1;%  define the first value for w
a=struct('wav',[],'wav2',[],'s',[]);%creates a structure array with the
specified fields and values.
P=64;%(the number of sampel files)
for p=1:P

filename=['C:\users\asr\Desktop\OGI2\45\En-Ger\a\','k',int2str(p),'.wav'];

%filename=['C:\users\asr\Desktop\OGI2\45\En-Jap\a\','k',int2str(p),'.wav'];

%filename=['C:\users\asr\Desktop\OGI2\45\En-Man\a\','k',int2str(p),'.wav'];

%filename=['C:\users\asr\Desktop\OGI2\45\En-Spa\a\','k',int2str(p),'.wav'];

%filename=['C:\users\asr\Desktop\OGI2\45\Ge-Jap\a\','k',int2str(p),'.wav'];

%filename=['C:\users\asr\Desktop\OGI2\45\Ge-Man\a\','k',int2str(p),'.wav'];

%filename=['C:\users\asr\Desktop\OGI2\45\Ge-Spa\a\','k',int2str(p),'.wav'];

% filename=['C:\users\asr\Desktop\OGI2\45\Jap-Man\a\','k',int2str(p),'.wav'];

%filename=['C:\users\asr\Desktop\OGI2\45\Jap-Spa\a\','k',int2str(p),'.wav'];

%filename=['C:\users\asr\Desktop\OGI2\45\Man-Spa\a\','k',int2str(p),'.wav'];


    b =wavread(filename);% reads a WAVE file
    b2=b.';% change row to column without

    siz2=size(b2,2); % calculate the number of columns of b2.
    a(w).wav=b2;% save the vector b2.
    a(w).s=siz2;%save the size of the vector b2.
    w=w+1;

    L(1,p)=siz2; %the size of the vector b2.
end

Le=min(L); % find minimum size between all vectors b2.

for w=1:P %(the number of sampel files)
    a(w).wav2(1,:)=a(w).wav(1,1:Le);% resize all vectors to minimum size of
b2.
    c=a(w).wav2(1,:);% uses any rows of data.


    [cA,cD]=dwt(c,'db20'); %uses the Daubechies db2 wavelet transform.
```

```
    for i=1:lca1
        M3(1,i,w)=0;
    end
    % maintaining just the 1% of the originals.
    lca=numel(cA);

    for i=1:lca
        M3(1,i,w)=cA(i);
    end
    msize=size(M3,2);
end

for w=1:P
    Mnew(w,:)=M3(1,:,w); % make a matrix from vectors M
end

Mclassnew=[1*ones(32,1); 2*ones(32,1)];
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% wavelet coefficient3
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

L=[];
w=1;%  define the first value for w
a=struct('wav',[],'wav2',[],'s',[]);%creates a structure array with the
specified fields and values.
P=36;%(the number of sampel files)
for p=1:P

filename=['C:\users\asr\Desktop\OGI2\45\En-Ger\c\','k',int2str(p),'.wav'];

%filename=['C:\users\asr\Desktop\OGI2\45\En-Jap\c\','k',int2str(p),'.wav'];

%filename=['C:\users\asr\Desktop\OGI2\45\En-Man\c\','k',int2str(p),'.wav'];

%filename=['C:\users\asr\Desktop\OGI2\45\En-Spa\c\','k',int2str(p),'.wav'];

%filename=['C:\users\asr\Desktop\OGI2\45\Ge-Jap\c\','k',int2str(p),'.wav'];

%filename=['C:\users\asr\Desktop\OGI2\45\Ge-Man\c\','k',int2str(p),'.wav'];

%filename=['C:\users\asr\Desktop\OGI2\45\Ge-Spa\c\','k',int2str(p),'.wav'];

%filename=['C:\users\asr\Desktop\OGI2\45\Jap-Man\c\','k',int2str(p),'.wav'];

%filename=['C:\users\asr\Desktop\OGI2\45\Jap-Spa\c\','k',int2str(p),'.wav'];

%filename=['C:\users\asr\Desktop\OGI2\45\Man-Spa\c\','k',int2str(p),'.wav'];

    b =wavread(filename);% reads a WAVE file
    b2=b.';% change row to column without

    siz2=size(b2,2); % calculate the number of columns of b2.
    a(w).wav=b2;% save the vector b2.
    a(w).s=siz2;%save the size of the vector b2.
    w=w+1;

    L(1,p)=siz2; %the size of the vector b2.
end

Le=min(L); % find minimum size between all vectors b2.

for w=1:P %(the number of sampel files)
    a(w).wav2(1,:)=a(w).wav(1,1:Le);% resize all vectors to minimum size of
b2.
    c=a(w).wav2(1,:);% uses any rows of data.

    [cA,cD]=dwt(c,'db20'); %uses the Daubechies db2 wavelet transform.

    for i=1:lca1
        M4(1,i,w)=0;
    end
```

```
    % maintaining just the 1% of the originals.
    lca=numel(cA);
    for i=1:lca
        M4(1,i,w)=cA(i);
    end
    msize=size(M4,2);
end

for w=1:P
    Mtest(w,:)=M4(1,:,w); % make a matrix from vectors M
end

Mclasstest=[1*ones(18,1); 2*ones(18,1)];
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% PLP coefficients
% row vector of PLP coefficients from all frames
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% finding min of length
ntrain=64 ;
for p=1:ntrain

filename=['C:\Users\HQ\Desktop\flash\HQ\Ge-Man\a\','k',int2str(i),'.wav'];

%filename=['C:\users\asr\Desktop\OGI2\45\En-Ger\a\','k',int2str(p),'.wav'];

%filename=['C:\users\asr\Desktop\OGI2\45\En-Jap\a\','k',int2str(p),'.wav'];

%filename=['C:\users\asr\Desktop\OGI2\45\En-Man\a\','k',int2str(p),'.wav'];

%filename=['C:\users\asr\Desktop\OGI2\45\En-Spa\a\','k',int2str(p),'.wav'];

%filename=['C:\users\asr\Desktop\OGI2\45\Ge-Jap\a\','k',int2str(p),'.wav'];

%filename=['C:\users\asr\Desktop\OGI2\45\Ge-Man\a\','k',int2str(p),'.wav'];

%filename=['C:\users\asr\Desktop\OGI2\45\Ge-Spa\a\','k',int2str(p),'.wav'];

%filename=['C:\users\asr\Desktop\OGI2\45\Jap-Man\a\','k',int2str(p),'.wav'];

%filename=['C:\users\asr\Desktop\OGI2\45\Jap-Spa\a\','k',int2str(p),'.wav'];

%filename=['C:\users\asr\Desktop\OGI2\45\Man-Spa\a\','k',int2str(p),'.wav'];


    [s,f]=wavread(filename);
    L(p)=length(s);
end
ntest=36;
for p=1:ntest

 filename=['C:\Users\HQ\Desktop\flash\HQ\Ge-Man\b\','k',int2str(i),'.wav'];%

%filename=['C:\users\asr\Desktop\OGI2\45\En-Ger\c\','k',int2str(p),'.wav'];

%filename=['C:\users\asr\Desktop\OGI2\45\En-Jap\c\','k',int2str(p),'.wav'];

%filename=['C:\users\asr\Desktop\OGI2\45\En-Man\c\','k',int2str(p),'.wav'];

%filename=['C:\users\asr\Desktop\OGI2\45\En-Spa\c\','k',int2str(p),'.wav'];

%filename=['C:\users\asr\Desktop\OGI2\45\Ge-Jap\c\','k',int2str(p),'.wav'];

%filename=['C:\users\asr\Desktop\OGI2\45\Ge-Man\c\','k',int2str(p),'.wav'];
 %filename=['C:\users\asr\Desktop\OGI2\45\Ge-Spa\c\','k',int2str(p),'.wav'];
```

```
 %filename=['C:\users\asr\Desktop\OGI2\45\Jap-Man\c\','k',int2str(p),'.wav'];

 %filename=['C:\users\asr\Desktop\OGI2\45\Jap-Spa\c\','k',int2str(p),'.wav'];

 %filename=['C:\users\asr\Desktop\OGI2\45\Man-Spa\c\','k',int2str(p),'.wav'];

    [s,f]=wavread(filename);
    L(p+60)=length(s);
end
cut=min(L);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%---------parameter of PLP function------------
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
modelorder = 12;
wintime = 8.25;
steptime =wintime/2;
%---------------
dorasta = 0;
dither = 1;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%-------------------- train---------------------
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Mtrain=[];
Mclass=[];
ntrain=64 ;
for p=1:ntrain

filename=['C:\Users\HQ\Desktop\flash\HQ\Ge-Man\a\','k',int2str(i),'.wav'];

%filename=['C:\users\asr\Desktop\OGI2\45\En-Ger\a\','k',int2str(p),'.wav'];

%filename=['C:\users\asr\Desktop\OGI2\45\En-Jap\a\','k',int2str(p),'.wav'];

%filename=['C:\users\asr\Desktop\OGI2\45\En-Man\a\','k',int2str(p),'.wav'];

%filename=['C:\users\asr\Desktop\OGI2\45\En-Spa\a\','k',int2str(p),'.wav'];

%filename=['C:\users\asr\Desktop\OGI2\45\Ge-Jap\a\','k',int2str(p),'.wav'];

%filename=['C:\users\asr\Desktop\OGI2\45\Ge-Man\a\','k',int2str(p),'.wav'];

%filename=['C:\users\asr\Desktop\OGI2\45\Ge-Spa\a\','k',int2str(p),'.wav'];

%filename=['C:\users\asr\Desktop\OGI2\45\Jap-Man\a\','k',int2str(p),'.wav'];

%filename=['C:\users\asr\Desktop\OGI2\45\Jap-Spa\a\','k',int2str(p),'.wav'];

%filename=['C:\users\asr\Desktop\OGI2\45\Man-Spa\a\','k',int2str(p),'.wav'];



    [samples,sr]=wavread(filename);
```

```
    samples(cut:end)=[];
    %%%%---------------- PLP function-----------------
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %     pspectrum = powspec(samples, sr);
    winpts = round(wintime*sr);
    steppts = round(steptime*sr);
    NFFT = 2^(ceil(log(winpts)/log(2)));
    WINDOW = hamming(winpts);
    NOVERLAP = winpts - steppts;
    SAMPRATE = sr;

    % Values coming out of rasta treat samples as integers,
    % not range -1..1, hence scale up here to match (approx)
    y = abs(specgram(samples*32768,NFFT,SAMPRATE,WINDOW,NOVERLAP)).^2;

    % imagine we had random dither that had a variance of 1 sample
    % step and a white spectrum.  That's like (in expectation, anyway)
    % adding a constant value to every bin (to avoid digital zero)
    if (dither)
        y = y + winpts;
    end
    pspectrum=y;
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    aspectrum = audspec(pspectrum, sr);
    nbands = size(aspectrum,1);
    postspectrum = postaud(aspectrum, sr);
    lpcas = dolpc(postspectrum, modelorder);
    % convert lpc to cepstra
    cepstra = lpc2cep(lpcas, modelorder+1);
    % .. or to spectra
    %     [spectra,F,M] = lpc2spec(lpcas, nbands);
    cepstra = lifter(cepstra, 0.6);
    c=cepstra;
    c=reshape(c,1,size(c,1)*size(c,2));
    Mtrain(p,:)=c;
end
M=Mtrain;
Mclass=[1*ones(ntrain/2,1); 2*ones(ntrain/2,1)];
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%--------------------test-------------------------
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Mtest=[];
Mclasstest=[];
ntest=36 ;
for p=1:ntest

filename=['C:\Users\HQ\Desktop\flash\HQ\Ge-Man\b\','k',int2str(i),'.wav'];

%filename=['C:\users\asr\Desktop\OGI2\45\En-Ger\c\','k',int2str(p),'.wav'];

%filename=['C:\users\asr\Desktop\OGI2\45\En-Jap\c\','k',int2str(p),'.wav'];

%filename=['C:\users\asr\Desktop\OGI2\45\En-Man\c\','k',int2str(p),'.wav'];

 %filename=['C:\users\asr\Desktop\OGI2\45\En-Spa\c\','k',int2str(p),'.wav'];
```

```
%filename=['C:\users\asr\Desktop\OGI2\45\Ge-Jap\c\','k',int2str(p),'.wav'];

%filename=['C:\users\asr\Desktop\OGI2\45\Ge-Man\c\','k',int2str(p),'.wav'];

%filename=['C:\users\asr\Desktop\OGI2\45\Ge-Spa\c\','k',int2str(p),'.wav'];

%filename=['C:\users\asr\Desktop\OGI2\45\Jap-Man\c\','k',int2str(p),'.wav'];

%filename=['C:\users\asr\Desktop\OGI2\45\Jap-Spa\c\','k',int2str(p),'.wav'];

%filename=['C:\users\asr\Desktop\OGI2\45\Man-Spa\c\','k',int2str(p),'.wav'];

    [samples,sr]=wavread(filename);
    samples(cut:end)=[];
    %%%%---------------- PLP function------------------
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %     pspectrum = powspec(samples, sr);
    winpts = round(wintime*sr);
    steppts = round(steptime*sr);
    NFFT = 2^(ceil(log(winpts)/log(2)));
    WINDOW = hamming(winpts);
    NOVERLAP = winpts - steppts;
    SAMPRATE = sr;

    % Values coming out of rasta treat samples as integers,
    % not range -1..1, hence scale up here to match (approx)
    y = abs(specgram(samples*32768,NFFT,SAMPRATE,WINDOW,NOVERLAP)).^2;

    % imagine we had random dither that had a variance of 1 sample
    % step and a white spectrum.  That's like (in expectation, anyway)
    % adding a constant value to every bin (to avoid digital zero)
    if (dither)
        y = y + winpts;
    end
    pspectrum=y;
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    aspectrum = audspec(pspectrum, sr);
    nbands = size(aspectrum,1);
    postspectrum = postaud(aspectrum, sr);
    lpcas = dolpc(postspectrum, modelorder);
    % convert lpc to cepstra
    cepstra = lpc2cep(lpcas, modelorder+1);
    % .. or to spectra
    %     [spectra,F,M] = lpc2spec(lpcas, nbands);
    cepstra = lifter(cepstra, 0.6);
    c=cepstra;
    c=reshape(c,1,size(c,1)*size(c,2));
    Mtest(p,:)=c;
end
Mclasstest=[1*ones(ntest/2,1); 2*ones(ntest/2,1)];


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
% MFCC coefficients
% row vector of mfcc coefficients from all frames
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% finding min of length
ntrain=64 ;
for p=1:ntrain

filename=['C:\Users\HQ\Desktop\flash\HQ\Ge-Man\a\','k',int2str(i),'.wav'];

%filename=['C:\users\asr\Desktop\OGI2\45\En-Ger\a\','k',int2str(p),'.wav'];

%filename=['C:\users\asr\Desktop\OGI2\45\En-Jap\a\','k',int2str(p),'.wav'];

%filename=['C:\users\asr\Desktop\OGI2\45\En-Man\a\','k',int2str(p),'.wav'];

%filename=['C:\users\asr\Desktop\OGI2\45\En-Spa\a\','k',int2str(p),'.wav'];

%filename=['C:\users\asr\Desktop\OGI2\45\Ge-Jap\a\','k',int2str(p),'.wav'];

%filename=['C:\users\asr\Desktop\OGI2\45\Ge-Man\a\','k',int2str(p),'.wav'];

%filename=['C:\users\asr\Desktop\OGI2\45\Ge-Spa\a\','k',int2str(p),'.wav'];

%filename=['C:\users\asr\Desktop\OGI2\45\Jap-Man\a\','k',int2str(p),'.wav'];

%filename=['C:\users\asr\Desktop\OGI2\45\Jap-Spa\a\','k',int2str(p),'.wav'];

%filename=['C:\users\asr\Desktop\OGI2\45\Man-Spa\a\','k',int2str(p),'.wav'];

    [s,f]=wavread(filename);
    L(p)=length(s);
end
ntest=36;
for p=1:ntest

filename=['C:\Users\HQ\Desktop\flash\HQ\Ge-Man\b\','k',int2str(i),'.wav'];

%filename=['C:\users\asr\Desktop\OGI2\45\En-Ger\c\','k',int2str(p),'.wav'];

%filename=['C:\users\asr\Desktop\OGI2\45\En-Jap\c\','k',int2str(p),'.wav'];

%filename=['C:\users\asr\Desktop\OGI2\45\En-Man\c\','k',int2str(p),'.wav'];

%filename=['C:\users\asr\Desktop\OGI2\45\En-Spa\c\','k',int2str(p),'.wav'];

%filename=['C:\users\asr\Desktop\OGI2\45\Ge-Jap\c\','k',int2str(p),'.wav'];

%filename=['C:\users\asr\Desktop\OGI2\45\Ge-Man\c\','k',int2str(p),'.wav'];

%filename=['C:\users\asr\Desktop\OGI2\45\Ge-Spa\c\','k',int2str(p),'.wav'];
```

```
%filename=['C:\users\asr\Desktop\OGI2\45\Jap-Man\c\','k',int2str(p),'.wav'];

%filename=['C:\users\asr\Desktop\OGI2\45\Jap-Spa\c\','k',int2str(p),'.wav'];

%filename=['C:\users\asr\Desktop\OGI2\45\Man-Spa\c\','k',int2str(p),'.wav'];

    [s,f]=wavread(filename);
    L(p+60)=length(s);
end
cut=min(L);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%---------parameter of mfcc function------------
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
numcep=13;
wintime=6;
hoptime=wintime/2;
%---------------
lifterexp=0.6;sumpower=1;preemph=0.97;
dither=0;minfreq=0;maxfreq=4000;
nbands=40;bwidth=1.0;dcttype=2;
fbtype='mel';usecmp=0;modelorder=0;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%-------------------- train---------------------
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
M=[];
Mclass=[];
ntrain=64 ;
for p=1:ntrain

filename=['C:\Users\HQ\Desktop\flash\HQ\Ge-Man\a\','k',int2str(i),'.wav'];

%filename=['C:\users\asr\Desktop\OGI2\45\En-Ger\a\','k',int2str(p),'.wav'];

%filename=['C:\users\asr\Desktop\OGI2\45\En-Jap\a\','k',int2str(p),'.wav'];

%filename=['C:\users\asr\Desktop\OGI2\45\En-Man\a\','k',int2str(p),'.wav'];

%filename=['C:\users\asr\Desktop\OGI2\45\En-Spa\a\','k',int2str(p),'.wav'];

%filename=['C:\users\asr\Desktop\OGI2\45\Ge-Jap\a\','k',int2str(p),'.wav'];

%filename=['C:\users\asr\Desktop\OGI2\45\Ge-Man\a\','k',int2str(p),'.wav'];

%filename=['C:\users\asr\Desktop\OGI2\45\Ge-Spa\a\','k',int2str(p),'.wav'];

%filename=['C:\users\asr\Desktop\OGI2\45\Jap-Man\a\','k',int2str(p),'.wav'];

% filename=['C:\users\asr\Desktop\OGI2\45\Jap-Spa\a\','k',int2str(p),'.wav'];

%filename=['C:\users\asr\Desktop\OGI2\45\Man-Spa\a\','k',int2str(p),'.wav'];


    [samples,sr]=wavread(filename);
```

```
    samples(cut:end)=[];
 %%%%---------------- mfcc function-----------------
    if preemph ~= 0
       samples = filter([1 -preemph], 1, samples);
    end
    % Compute FFT power spectrum
      pspectrum = powspec(samples, sr, wintime, hoptime, dither);
      aspectrum = audspec(pspectrum, sr, nbands, fbtype, minfreq, maxfreq,
sumpower, bwidth);
    if (usecmp)
       % PLP-like weighting/compression
       aspectrum = postaud(aspectrum, maxfreq, fbtype);
    end
    if modelorder > 0
      if (dcttype ~= 1)
          disp(['warning: plp cepstra are implicitly dcttype 1 (not ',
num2str(dcttype), ')']);
      end
      % LPC analysis
       lpcas = dolpc(aspectrum, modelorder);
      % convert lpc to cepstra
      cepstra = lpc2cep(lpcas, numcep);
    else
       % Convert to cepstra via DCT
       cepstra = spec2cep(aspectrum, numcep, dcttype);
    end
    cepstra = lifter(cepstra, lifterexp);
    c=cepstra;
    c=reshape(cepstra,1,size(c,1)*size(c,2));
    M(p,:)=c;
end
Mclass=[1*ones(ntrain/2,1); 2*ones(ntrain/2,1)];
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%-------------------test-------------------------
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Mtest=[];
Mclasstest=[];
ntest=36 ;
for p=1:ntest

filename=['C:\Users\HQ\Desktop\flash\HQ\Ge-Man\b\','k',int2str(i),'.wav'];

%filename=['C:\users\asr\Desktop\OGI2\45\En-Ger\c\','k',int2str(p),'.wav'];

%filename=['C:\users\asr\Desktop\OGI2\45\En-Jap\c\','k',int2str(p),'.wav'];

%filename=['C:\users\asr\Desktop\OGI2\45\En-Man\c\','k',int2str(p),'.wav'];

%filename=['C:\users\asr\Desktop\OGI2\45\En-Spa\c\','k',int2str(p),'.wav'];

%filename=['C:\users\asr\Desktop\OGI2\45\Ge-Jap\c\','k',int2str(p),'.wav'];

%filename=['C:\users\asr\Desktop\OGI2\45\Ge-Man\c\','k',int2str(p),'.wav'];

%filename=['C:\users\asr\Desktop\OGI2\45\Ge-Spa\c\','k',int2str(p),'.wav'];
```

```
%filename=['C:\users\asr\Desktop\OGI2\45\Jap-Man\c\','k',int2str(p),'.wav'];

%filename=['C:\users\asr\Desktop\OGI2\45\Jap-Spa\c\','k',int2str(p),'.wav'];

%filename=['C:\users\asr\Desktop\OGI2\45\Man-Spa\c\','k',int2str(p),'.wav'];

    [samples,sr]=wavread(filename);
    samples(cut:end)=[];
 %%%%---------------- mfcc function-----------------
    if preemph ~= 0
       samples = filter([1 -preemph], 1, samples);
    end
    % Compute FFT power spectrum
      pspectrum = powspec(samples, sr, wintime, hoptime, dither);
      aspectrum = audspec(pspectrum, sr, nbands, fbtype, minfreq, maxfreq,
sumpower, bwidth);
    if (usecmp)
       % PLP-like weighting/compression
       aspectrum = postaud(aspectrum, maxfreq, fbtype);
    end
    if modelorder > 0
      if (dcttype ~= 1)
          disp(['warning: plp cepstra are implicitly dcttype 1 (not ',
num2str(dcttype), ')']);
      end
      % LPC analysis
       lpcas = dolpc(aspectrum, modelorder);
      % convert lpc to cepstra
      cepstra = lpc2cep(lpcas, numcep);
    else
       % Convert to cepstra via DCT
       cepstra = spec2cep(aspectrum, numcep, dcttype);
    end
    cepstra = lifter(cepstra, lifterexp);
    c=cepstra;
    c=reshape(cepstra,1,size(c,1)*size(c,2));
    Mtest(p,:)=c;
end
Mclasstest=[1*ones(ntest/2,1); 2*ones(ntest/2,1)];
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% LPC coefficient
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%------------ finding min of length------------

ntrain=64 ;
for p=1:ntrain

filename=['C:\Users\HQ\Desktop\flash\HQ\Ge-Man\a\','k',int2str(i),'.wav'];

%filename=['C:\users\asr\Desktop\OGI2\45\En-Ger\a\','k',int2str(p),'.wav'];

%filename=['C:\users\asr\Desktop\OGI2\45\En-Jap\a\','k',int2str(p),'.wav'];

%filename=['C:\users\asr\Desktop\OGI2\45\En-Man\a\','k',int2str(p),'.wav'];

%filename=['C:\users\asr\Desktop\OGI2\45\En-Spa\a\','k',int2str(p),'.wav'];

%filename=['C:\users\asr\Desktop\OGI2\45\Ge-Jap\a\','k',int2str(p),'.wav'];

%filename=['C:\users\asr\Desktop\OGI2\45\Ge-Man\a\','k',int2str(p),'.wav'];

%filename=['C:\users\asr\Desktop\OGI2\45\Ge-Spa\a\','k',int2str(p),'.wav'];

%filename=['C:\users\asr\Desktop\OGI2\45\Jap-Man\a\','k',int2str(p),'.wav'];

%filename=['C:\users\asr\Desktop\OGI2\45\Jap-Spa\a\','k',int2str(p),'.wav'];

%filename=['C:\users\asr\Desktop\OGI2\45\Man-Spa\a\','k',int2str(p),'.wav'];

    [s,f]=wavread(filename);
    L(p)=length(s);
end
ntest=36;
for p=1:ntest

 filename=['C:\Users\HQ\Desktop\flash\HQ\Ge-Man\b\','k',int2str(i),'.wav'];

%filename=['C:\users\asr\Desktop\OGI2\45\En-Ger\c\','k',int2str(p),'.wav'];

%filename=['C:\users\asr\Desktop\OGI2\45\En-Jap\c\','k',int2str(p),'.wav'];

%filename=['C:\users\asr\Desktop\OGI2\45\En-Man\c\','k',int2str(p),'.wav'];

%filename=['C:\users\asr\Desktop\OGI2\45\En-Spa\c\','k',int2str(p),'.wav'];

%filename=['C:\users\asr\Desktop\OGI2\45\Ge-Jap\c\','k',int2str(p),'.wav'];

%filename=['C:\users\asr\Desktop\OGI2\45\Ge-Man\c\','k',int2str(p),'.wav'];

%filename=['C:\users\asr\Desktop\OGI2\45\Ge-Spa\c\','k',int2str(p),'.wav'];
```

```
%filename=['C:\users\asr\Desktop\OGI2\45\Jap-Man\c\','k',int2str(p),'.wav'];

 %filename=['C:\users\asr\Desktop\OGI2\45\Jap-Spa\c\','k',int2str(p),'.wav'];

 %filename=['C:\users\asr\Desktop\OGI2\45\Man-Spa\c\','k',int2str(p),'.wav'];

    [s,f]=wavread(filename);
    L(p+60)=length(s);
end
cut=min(L);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%-------------- parameter of LPC----------------
Plpc=9;
wintime=5.25;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%-------------------- train---------------------
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
M=[];
Mclass=[];
ntrain=64 ;
for p=1:ntrain

filename=['C:\Users\HQ\Desktop\flash\HQ\Ge-Man\a\','k',int2str(i),'.wav'];

%filename=['C:\users\asr\Desktop\OGI2\45\En-Ger\a\','k',int2str(p),'.wav'];

%filename=['C:\users\asr\Desktop\OGI2\45\En-Jap\a\','k',int2str(p),'.wav'];

%filename=['C:\users\asr\Desktop\OGI2\45\En-Man\a\','k',int2str(p),'.wav'];

%filename=['C:\users\asr\Desktop\OGI2\45\En-Spa\a\','k',int2str(p),'.wav'];

%filename=['C:\users\asr\Desktop\OGI2\45\Ge-Jap\a\','k',int2str(p),'.wav'];

%filename=['C:\users\asr\Desktop\OGI2\45\Ge-Man\a\','k',int2str(p),'.wav'];

%filename=['C:\users\asr\Desktop\OGI2\45\Ge-Spa\a\','k',int2str(p),'.wav'];

%filename=['C:\users\asr\Desktop\OGI2\45\Jap-Man\a\','k',int2str(p),'.wav'];

%filename=['C:\users\asr\Desktop\OGI2\45\Jap-Spa\a\','k',int2str(p),'.wav'];

%filename=['C:\users\asr\Desktop\OGI2\45\Man-Spa\a\','k',int2str(p),'.wav'];

    [s,f]=wavread(filename);
    s(cut:end)=[];
    samples=wintime*f;
    m=[];
    a=floor(length(s)/samples)+floor((length(s)-samples/2)/samples);
    for j=1:a
        x=s(1+(j-1)*(samples/2):1+(j-1)*(samples/2)+samples-1);
        c=lpc(x,Plpc);
        m(:,j)=c';
    end
```

```
    m=reshape(m,1,size(m,1)*size(m,2));
    M(p,:)=m;
end
Mclass=[1*ones(ntrain/2,1); 2*ones(ntrain/2,1)];
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%-------------------test-------------------------
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Mtest=[];
Mclasstest=[];
ntest=36 ;
for p=1:ntest

filename=['C:\Users\HQ\Desktop\flash\HQ\Ge-Man\b\','k',int2str(i),'.wav'];

%filename=['C:\users\asr\Desktop\OGI2\45\En-Ger\c\','k',int2str(p),'.wav'];

%filename=['C:\users\asr\Desktop\OGI2\45\En-Jap\c\','k',int2str(p),'.wav'];

%filename=['C:\users\asr\Desktop\OGI2\45\En-Man\c\','k',int2str(p),'.wav'];

%filename=['C:\users\asr\Desktop\OGI2\45\En-Spa\c\','k',int2str(p),'.wav'];

%filename=['C:\users\asr\Desktop\OGI2\45\Ge-Jap\c\','k',int2str(p),'.wav'];

%filename=['C:\users\asr\Desktop\OGI2\45\Ge-Man\c\','k',int2str(p),'.wav'];

%filename=['C:\users\asr\Desktop\OGI2\45\Ge-Spa\c\','k',int2str(p),'.wav'];

%filename=['C:\users\asr\Desktop\OGI2\45\Jap-Man\c\','k',int2str(p),'.wav'];

%filename=['C:\users\asr\Desktop\OGI2\45\Jap-Spa\c\','k',int2str(p),'.wav'];

%filename=['C:\users\asr\Desktop\OGI2\45\Man-Spa\c\','k',int2str(p),'.wav'];

    [s,f]=wavread(filename);
    s(cut:end)=[];
    samples=wintime*f;
    m=[];
    a=floor(length(s)/samples)+floor((length(s)-samples/2)/samples);
    for j=1:a
        x=s(1+(j-1)*(samples/2):1+(j-1)*(samples/2)+samples-1);
        c=lpc(x,Plpc);
        m(:,j)=c';
    end
    m=reshape(m,1,size(m,1)*size(m,2));
    Mtest(p,:)=m;
end
Mclasstest=[1*ones(ntest/2,1); 2*ones(ntest/2,1)];
Mnew=M;
```

```
%*************************************************************************
%Original program for language identification
%*************************************************************************
%*************************************************************************
accuracy=zeros(1,size(M,2));
numbercoefficients=[];
accuracycoeficients=[];
numbercoefftest=[];
accuracycoefftest=[];
tlb=struct('tfs',[],'cn',[],'probc',[],'ixtT',[]);
YY=0;
% sizTf=YY;
ii=2;
A=[];
for p=1:size(M,2)
    N=size(M,1); %the number of examples present in a subset S.
    c=2; % the number of classes to process each pair of languages.
    %%%c=9;%  the number of classes.

    lableM=[];
    lable1=[];
    lable2=[];
    lable=1:N;
    a=M(:,p);


    Msort=zeros(N,1);
    Mclasssort=zeros(N,1);
    [Msort,in]=sort(a);%  sort of M, "in" is  the index of Msort.
    lableM=lable;

    for i=1:(size(in,1))
        Mclasssort(i,1)=Mclass(in(i,1),1);% sort of classes (label of Msort).
    end
    lableclass=Mclasssort; %lable class is first Mclasssort.
    Zv=[];Tfinal=[];
    w=1; q=1;v=1;z=1;

    b=struct('s2',[],'ix2',[],'lb2',[]);

    while (z>0)
        N=size(Msort,1);
        T=[];
        indexT=[];
        e=[];
        Ents=[];
        Ents1=[];
        Ents2=[];

%*************************************************************************

        j=1;
        for i=1:(numel(Mclasssort)-1)
            if Mclasssort(i,1)~=Mclasssort(i+1,1) & Msort(i,1)~=Msort(i+1,1)
```

```
                %Mclasssort show the classes present in a subset(the indices
for languages)
                T(1,j)=(Msort(i,1)+Msort(i+1,1))/2;
                indexT(1,j)=i; %indexT is the index of T.

                s1=Msort(1:i,1);          %show a subset s1.
                s2=Msort(i+1:N,1);        %show a subset s2.
                ix1=Mclasssort(1:i,1);    %show the classes present in S1 .
                ix2=Mclasssort(i+1:end,1);  %show the classes present in S2 .
                nums1=numel(s1);          %the number of examples present in
a subset S1.
                nums2=numel(s2);          %the number of examples present in
a subset S2.
                %%%%%%%%%%%%%%%%%%%%%
                pc=zeros(1,c);
                Pcc=zeros(1,c);
                for k=1:c
                    for x=1:length(Mclasssort)
                        if Mclasssort(x,1)==k
                            pc(1,k)=pc(1,k)+1;

                        end
                    end

                    pcc(1,k)=(pc(1,k)/(nums1+nums2));
                end
                ents=0;
                for k=1:c
                    if(pcc(1,k)~=0)
                        ents=ents+(pcc(1,k)*log2(pcc(1,k)));
                    end
                end
                Ents(1,j)=-ents; %Ent(s) is the class entropy of a subset S.
                %%%%%%%%%%%%%%%%%%
                pc=zeros(1,c);
                Pcc=zeros(1,c);
                for k=1:c
                    for x=1:length(ix1)
                        if ix1(x,1)==k
                            pc(1,k)=pc(1,k)+1;

                        end
                    end

                    pcc(1,k)=(pc(1,k)/nums1);
                end
                ents1=0;
                for k=1:c
                    if(pcc(1,k)~=0)
                        ents1= ents1+(pcc(1,k)*log2(pcc(1,k)));
                    end
                end
                Ents1(1,j)=-ents1; %Ent(s1) is the class entropy of a subset
S1.
                %%%%%%%%%%%%%%%%
                pc=zeros(1,c);
```

```
                Pcc=zeros(1,c);
                for k=1:c
                    for x=1:length(ix2)
                        if ix2(x,1)==k
                            pc(1,k)=pc(1,k)+1;

                        end
                    end

                    pcc(1,k)=(pc(1,k)/nums2);
                end
                ents2=0;
                for k=1:c
                    if(pcc(1,k)~=0)
                        ents2= ents2+(pcc(1,k)*log2(pcc(1,k)));
                    end
                end
                Ents2(1,j)=-ents2; %Ent(s2) is the class entropy of a subset
S2.


e(1,j)=(nums1/(nums1+nums2))*Ents1(1,j)+(nums2/(nums1+nums2))*Ents2(1,j);
                j=j+1;
            end
        end

%*************************************************************************

%*************************************************************************
        if numel(e)==0
            z=w-1;
            if numel(Zv)==0
                Zv(1)=0;
                Zv(2)=0;

            end

            cw=1;
            while (cw==1)
                cw=0;
                for v=1:length(Zv)

                    if Zv(v)==z
                        z=z-1;
                        cw=1;
                    end

                end
            end

            if z>0
                v=length(Zv)+1;
                Zv(v)=z;
            else break
            end
```

```
            Msort= b(z).s2;
            Mclasssort=b(z).ix2;
            lableM=b(z).lb2;
            Msave= b(z).s2;



        else
%*************************************************************************
%*************************************************************************
            [emin,indexemin]=min(e);          %find minimum e and its index.
            ixcut=indexT(indexemin);%find the index of T corresponding to
indexemin.

            s1min=Msort(1:ixcut,1);          %find a new subset s1
corresponding to emin.
            s2min=Msort(ixcut+1:end,1);       %find a new subset s2
corresponding to emin.

            num1=numel(s1min); % show the number of s1(s1min).
            num2=numel(s2min);% show the number of s2(s2min).

            ix1min=Mclasssort(1:ixcut,1);     %show the classes present in
s1min .
            ix2min=Mclasssort(ixcut+1:end,1); % show the classes present in
s2min).

            Entsmin=Ents(1,indexemin);  %find the ent(s)  for emin.
            Ents1min=Ents1(1,indexemin); %find the ent(s1) for emin.
            Ents2min=Ents2(1,indexemin); %find the ent(s2) for emin.

            %*********************************************
            count=c;   %the number of distinct classes present in S.

            h1=zeros(c,1);
            co1=0;
            for h=1:(numel(ix1min))
                for u=1:c
                    if ix1min(h)~=u
                        h1(u)=h1(u)+1;
                    end
                end
            end
            for u=1:c
                if h1(u)==numel(ix1min)
                    co1=co1+1;
                end
            end
            count1=c-co1;  %the number of distinct classes present in S1min
            %*****************************
            h2=zeros(c,1);
            co2=0;
```

```
            for h=1:numel(ix2min)
                for u=1:c
                    if ix2min(h)~=u
                        h2(u)=h2(u)+1;
                    end
                end
            end
            for u=1:c
                if h2(u)==numel(ix2min)
                    co2=co2+1;
                end
            end
            count2=c-co2;  %the number of distinct classes present in S2min

            %*****************************************************
            gain= Entsmin-((num1/N)*(Ents1min))-((num2/N)*Ents2min);
            delta=log2((3^count)-2)-((count* Entsmin)-(count1* Ents1min)-
(count2* Ents2min));

            cw=0;

            if gain>(((log2(N-1))/N)+(delta/N))


                Tfinal(q)=T(indexemin);
                lable1=lableM(1:ixcut); %
                lable2=lableM(ixcut+1:end); %

                b(w).s2=s2min;
                b(w).ix2=ix2min;
                b(w).lb2=lable2;
                lableM=lable1;

                Msort=s1min;
                Mclasssort=ix1min;

                w=w+1;
                q=q+1;


%*****************************************************************

%*****************************************************************
            else
                z=w-1;
                if numel(Zv)==0
                    Zv(1)=0;
                    Zv(2)=0;

                end

                cw=1;
                while (cw==1)
                    cw=0;
```

```
                    for v=1:length(Zv)

                        if Zv(v)==z
                            z=z-1;
                            cw=1;
                        end

                    end
                end

                if z>0
                    v=length(Zv)+1;
                    Zv(v)=z;
                else break
                end

                Msort= b(z).s2;
                Mclasssort=b(z).ix2;
                lableM=b(z).lb2;


            end
        end
    end


    %**********
    %**********
    %the classification phase1
    classname=[]; % tarif avaliye classname ke braye neshan dadane class har
baze be kar miravad.

    % sotuni ke daste bandi nemishavad ra moshakhas konim.
    A(:,1)=0;
    if numel(Tfinal)==0
        A(:,ii)=p;
        ii=ii+1;
    else

        [Tfinalsort,ixt]=sort(Tfinal); % sort kardane t haye peyda shode dar
bakhshe ghabl.

        if size(Tfinal,2)>YY
            sizTf=size(Tfinal,2); % tedad sotun ra baraye  C ,C1 midahad.
        end
        tlb(p).tfs=Tfinalsort; %zakhire sazi Tfinalsort baraye har
atribute(sotun)
        tlb(p).ixtT=ixt;%zakhire sazi  andise Tfinalsort baraye har
atribute(sotun)
        YY=sizTf;
    end

end
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% classification
Msort1=zeros(size(M,1),size(M,2));
[Msort1,in1]=sort(M);
for p=1:size(M,2)
    if p~=A(1,:)
        for i=1:(size(in1,1))
            Mclasssort1(i,p)=Mclass(in1(i,p),1);% sort of classes (label of
Msort1).
        end
    end
end
C=zeros(size(M,1),sizTf+1,size(Msort1,2));
C1=zeros(size(M,1),sizTf+1,size(Msort1,2));
for p=1:size(M,2)
    if p~=A(1,:)

        X=[-1 tlb(p).tfs 1];
        for j=1:size(X,2)-1
            K=find(X(j)<=Msort1(:,p)   &    Msort1(:,p)<=X(j+1));
            cc=Msort1(K,p);
            cc1=Mclasssort1(K,p);
            for i=1:numel(cc)
                C(i,j,p)=cc(i,1,1);    %C(i,j,p)=Msort1(K,p)
                C1(i,j,p)=cc1(i,1,1); %C1(i,j,p)=Mclasssort1(K,p)
            end
        end
    end
end

for p=1:size(C,3)
    if p~=A(1,:)
        for j=1:size(C,2)
            pc=zeros(1,c);
            pcc=0;
            for k=1:c
                for i=1:size(C,1)
                    if  C1(i,j,p)==k
                        pc(1,k)=pc(1,k)+1;
                        pcc=pcc+1;
                    end
                end
            end

            for i=1:c
                prpc(i,j,p)=pc(1,i)/pcc;
                C2(i,j,p)=prpc(i,j,p);
            end
            [pcmax,ixp]=max(pc);
            if pcmax==0
                ixp=0;
            end

            classname(j)=ixp;
            prc(j)=(pcmax/pcc);
```

```
        end
        tlb(p).cn=classname; %tlb(p).cn(j) classname baze j om az sotune p om
ra midahad.
        tlb(p).probc=prc;
    end
end

%************************************************************
%testing prog

Nnew=size(Mnew,1);
accuracy1=zeros(1,min(size(Mnew,2),size(M,2)));

for p=1:min(size(Mnew,2),size(M,2))
    if p~=A(1,:)
        true=0;
        false=0;
        pp=sss(p);
        for i=1:Nnew
            ph=Mnew(i,pp);
            phc=Mclassnew(i);
            pnum=0;

            if numel(tlb(p).tfs)~=0
                if ph<tlb(p).tfs(1)
                    phclass =tlb(p).cn(1);

                elseif ph>=tlb(p).tfs(end)
                    phclass=tlb(p).cn(end);
                else

                    for j=2:length(tlb(p).tfs)
                        if tlb(p).tfs(j-1)<=ph & ph<tlb(p).tfs(j)
                            phclass=tlb(p).cn(j);
                        end
                    end

                end

                if phclass==phc
                    true=true+1;
                else false=false+1;
                end
            end
        end

        accuracy1(p)=true*100/(true+false); %accuracy baraye har sotun be
dast miayad.
    end
end

[acc,pmax]=max(accuracy1)%firstattT=tlb.tfs(pmax);%First attribute
ppmax=sss(pmax);
%************************************************************************
classprobmax1= zeros(length(Mclassnew),1);
```

```
classprobmax2= zeros(length(Mclassnew),1);

probm1= zeros(c,1);
probm2= zeros(c,1);
probm3= zeros(c,1);
probm= zeros(c,1);
opt=struct('numco',[],'accco',[],'Tco',[],'Lco',[],'Lco2',[],'numcotest',[],'
acccotest',[]);
numattL=[];
probattL=[];
accuracy=[];
x=1;
if pmax~=A(1,:)

    opt(x).Tco=tlb(pmax).tfs;
    opt(x).Lco=tlb(pmax).cn;
    opt(x).accco=accuracy1(pmax);
    opt(x).numco=pmax;

    true=zeros(1,size(M,2));
    l=1;
    for k=1:size(Mnew,1)
        a1=Mnew(k,ppmax);
        T1=tlb(pmax).tfs;
        class1=tlb(pmax).cn;
        prob1=tlb(pmax).probc;

        if a1< T1(1)
            a1class=class1(1);
            a1prob1=prob1(1);

            for i=1:c
                probm1(i,l)=C2(i,1,pmax);
            end

        elseif a1>= T1(end)
            a1class=class1(numel(T1)+1);
            a1prob1=prob1(numel(T1)+1);

            for i=1:c
                probm1(i,l)=C2(i,numel(T1)+1,pmax);
            end

        else
            for j=2:length(T1)
                if T1(j-1)<=a1  & a1<T1(j)
                    a1class=class1(j);
                    a1prob1=prob1(j);

                    for i=1:c
                        probm1(i,l)=C2(i,j,pmax);
                    end
                end
            end
        end
```

```
        l=l+1;
    end
end


accuracy=zeros(1,min(size(Mnew,2),size(M,2)));
for p=1:min(size(Mnew,2),size(M,2))
    if p~=A(1,:)
        if p~=pmax
            pp=sss(p);
            if numel( tlb(p).tfs)~=0
                true=0;
                false=0;
                l=1;
                for k=1:size(Mnew,1)
                    a2=Mnew(k,pp);
                    T2=tlb(p).tfs;
                    class2=tlb(p).cn;
                    prob2=tlb(p).probc;

                    if a2< T2(1)
                        a2class=class2(1);
                        a2prob2=prob2(1);
                        for i=1:c
                            probm2(i,l)=C2(i,1,p);
                        end

                    elseif a2>= T2(end)
                        a2class=class2(numel(T2)+1);
                        a2prob2=prob2(numel(T2)+1);
                        for i=1:c
                            probm2(i,l)=C2(i,numel(T2)+1,p);
                        end

                    else
                        for j=2:length(T2)
                            if T2(j-1)<=a2 & a2<T2(j)
                                a2class=class2(j);
                                a2prob2=prob2(j);
                                for i=1:c
                                    probm2(i,l)=C2(i,j,p);
                                end
                            end


                        end
                    end
                    l=l+1;

                end


                for l=1:length(Mclassnew)
                    for i=1:c
                        probm(i,l)=(probm1(i,l)).* (probm2(i,l));
```

```
                    end
                end

                [prob,clsssname]=max(probm);
                opt(p).Lco2=probm;

                for k=1:size(Mnew,1)
                    if clsssname(1,k)== Mclassnew(k,1)
                        true=true+1;
                    else
                        false=false+1;
                    end
                end

            end
        else p=p+1;
        end
        accuracy(p)=true*100/(true+false);
    end
end

[a,b]=max(accuracy); %max accuracy and its index
i=1;
for p=1:min(size(Mnew,2),size(M,2))
    if p~=A(1,:)
        if p~=pmax
            if p~=b
                if accuracy(p)==a
                    b3(1:i)=p;
                    i=i+1;
                end
            end
        end
    end
end

C3=zeros(c,size(Mnew,1));
C3(:,:)=opt(b).Lco2;

while( a>opt(x).accco)
    x=x+1;
    opt(x).Tco=tlb(b).tfs; %the T of next attribute
    opt(x).accco=a; %the accuracy of next attribute
    opt(x).numco=b; %the number of next attribute

    accuracy=zeros(1,min(size(Mnew,2),size(M,2)));

    for p=1:min(size(Mnew,2),size(M,2))
        if p~=A(1,:)
            l=0;
            for u=1:x
                if p==opt(u).numco
                    l=l+1;
                end
            end
```

```
if l==0
    true=0;
    false=0;
    l=1;
    pp=sss(p);
    for k=1:length(Mclassnew)
        a1=Mnew(k,pp);
        T=tlb(p).tfs;
        class=tlb(p).cn;
        prob=tlb(p).probc;

        if a1< T(1)
            aclass=class(1);
            aprob=prob(1);

            for i=1:c
                probm3(i,l)=C2(i,1,p);
            end

        elseif a1>= T(end)
            aclass=class(end);
            aprob=prob(end);

            for i=1:c
                probm3(i,l)=C2(i,numel(T)+1,p);
            end

        else
            for j=2:length(T)
                if T(j-1)<=a1 & a1<T(j)
                    aclass=class(j-1);
                    aprob=prob(j-1);

                    for i=1:c
                        probm3(i,l)=C2(i,j,p);
                    end

                end
            end
        end
        l=l+1;
    end

    for l=1:length(Mclassnew)
        for i=1:c
            probmax(i,l)=C3(i,x-1).* (probm3(i,l));
        end
    end
    [prob2,classname2]=max(probmax);
    opt(p).Lco2=probmax;


    for k=1:length(Mclassnew)
```

```
                    if classname2(k)== Mclassnew(k)
                        true=true+1;
                    else
                        false=false+1;
                    end
                end
            end
            accuracy(p)=true*100/(true+false);
        end
    end

    [a,b]=max(accuracy)
    C3(:,:)=opt(b).Lco2;

end

for i=1:x
    l =opt(i).numco
    numbercoefficients(i)=sss(l)
    accuracycoeficients(i)=opt(i).accco
end


%%%*************************************************************************
% TEST

C3=ones(c,size(Mtest,1))
for n=1:x

    p=opt(n).numco;
    acc=opt(n).accco;
    T=opt(n).Tco;
    pp=sss(p)
    if p<=size(Mtest,2)
        true=0;
        false=0;


        for k=1:size(Mtest,1)

            ph=Mtest(k,pp);
            phc=Mclasstest(k);

            if ph<T(1)
                phclass =tlb(p).cn(1);
                for i=1:c
                    probtest(i,k)=C2(i,1,p);
                end

            elseif ph>=T(end)
                phclass=tlb(p).cn(end);
                for i=1:c
                    probtest(i,k)=C2(i,numel(T)+1,p);
                end
```

```
            else

                for j=2:length(T)
                    if T(j-1)<=ph & ph<T(j)
                        phclass=tlb(p).cn(j);

                        for i=1:c
                            probtest(i,k)=C2(i,j,p);
                        end
                    end
                end

            end

        end
        for l=1:size(Mtest,1)
            for i=1:c
            probtestmax(i,l)=C3(i,l).* probtest(i,l);
            C3(i,l)=probtestmax(i,l);
            end
        end

        [probmtest,classnametest]=max(probtestmax);
        for k=1:size(Mtest,1)
            if classnametest(k)==Mclasstest(k)
                true=true+1;
            else
                false=false+1;
            end

        end

        accuracytest(n)=true*100/(true+false); %accuracy baraye har sotun be
dast miayad.
        opt(n).numcotest=p;
        opt(n).acccotest=accuracytest(n);
    else
        opt(n).numcotest=p;
        opt(n).acccotest=0;
    end

end
for i=1:x
    l =opt(i).numcotest
    numbercoefftest(i)=sss(l)
    accuracycoefftest(i)=opt(i).acccotest

end
```

accuracy rate than samples of 10 seconds. The previous methods only could identify pair-wise languages, but our proposed method is able to identify language type, among 9 languages in OGI_TS.

Keywords: Wavelet transform, information gain, feature selection, spoken language identification, multi- interval discretisation, benchmarking attribute algorithm, cepstral coefficients, MFCC ,PLP, LPC

Abstract

Automatic language identification consists in recognizing a language based on a sample of speech from an unknown speaker. Automatic language identification can help relation between people of various areas. It has multiple usages in development of tourism, free trade, amplification of national security by means of pre-processing and filtering of doubtful conversations, emergency service and simultaneous translations in international congresses and conversations.

In this thesis, the system of automatic language identification is designed and simulated by assisting of classifying of various features. Therefore, we find suitable features for every language and train the classifying algorithm based on features which selected by benchmarking attribute algorithm and multi-interval discretisation for various languages. Then, we determine decision rules for each language and use this classification rules to identify the language of testing samples.

We use the samples of 10 seconds and 45 seconds in the OGI_TS database to test the proposed method. There are acoustic samples in 11 languages, consist of: English, Farsi, French, German, Hindi, Japanese, Korean, Mandarin, Spanish, Tamil, and Vietnamese. The utterances ranged in duration from one second to 50 seconds, with an average duration of 13.4 seconds. Language identification systems, usually, use 9 initial languages. Thus, we implement our experiments on these 9 languages, and compare our results with other reported methods. The experiments are accomplished based on various features such as wavelet transforms, MFCC, PLP and LPC.

So far, different approaches are proffered for automatic language identification which are dependent on phonotactics information and their utilization is difficult. In this research, we present an approach independent of phonotactics that it is very easy and can identify languages with a good accuracy rate. In this method, we utilize the wavelet transform and cepstral coefficients that are available on different languages which don't have any requirements to linguistic information. Cepstral coefficients achieve the accuracy rate better than wavelet transform. Also the samples of 45 seconds wavelet transform and cepstral coefficient have better

Shahrood University of Technology

Faculty of Electrical and Robotic Engineering

Design and Implementation of Automatic Spoken Language Identification System

Pariya Moharlooei

Supervisions:

Dr.Omid Reza Maroozi

Dr.Hossein Marvi

Associate Supervisor:

Dr.Hadi Grayloo

February, 2011