
بسم الله الرحمن الرحيم

دانشکده مهندسی برق و رباتیک

گروه الکترونیک

پایان نامه کارشناسی ارشد

پیاده سازی سخت افزاری جستجوی برون خط کلید واژه‌ی صوتی در گفتار ثبت شده در بستر پردازشگر TMS320C55xx

محمد جوادی

استاد راهنما:

دکتر هادی گرایلو

شهریور ماه ۱۳۹۴

تقدیم به...

انجام این اثر جز با حمایت های پدر و مادر مهربانم و همراهی همسر و فرزند عزیزم انجام نمی‌پذیرفت. آنها که درس صبر و وفاداری را به من آموختند، آنها که در تشخیص راه‌های درست مرا یاری کردند، آنها که در لحظات سخت زندگی پشت گرمی‌ام و در لحظات شیرین آن در کنارم بودند. آنها که در مسیر پر پیچ و تاب و طولانی تحصیل و تا انجام این پایان نامه همواره مشوق و همراه من بودند و لحظه ای محبتشان را دریغ نکردند.

اینک این تلاش با افتخار به آنان تقدیم می‌شود.

سپاس گزاری...

پس از حمد ثنای بی‌حد بر آستان بی‌همتای احدیت که در کمال رأفت و در نهایت عطوفت رخصت اتمام این پایان نامه را به بنده عطا فرمود، بر خود واجب می‌دانم که از استاد راهنمای گرانقدر، جناب دکتر هادی گرایلو که با راهنمایی‌های ارزشمند خود، مرا در انجام این پایان‌نامه یاری رساندند نهایت تشکر و قدردانی را داشته باشم، باشد که نشأت علم گران‌بارشان آبادگر سرزمین عزیزمان ایران باشد.

از دوست و همراه همیشگی‌ام آقای احمد معینی که در انجام این مسیر بسیار یاری بخش و دلگرمی بنده بودند نیز کمال تشکر را دارم.

تعهــد نامه

اینجانب **محمد جوادی** دانشجوی دوره کارشناسی ارشد رشته **الکترونیک-دیجیتال** دانشکده برق و رباتیک دانشگاه شاهرود نویسنده پایان نامه **پیاده سازی سخت افزاری جست و جوی برون خط کلید واژه صوتی در گفتار ثبت شده در بستر پردازشگر TMS320C55xx** تحت راهنمائی دکتر **هادی گرایلو** متعهد می شوم.

- تحقیقات در این پایان نامه توسط اینجانب انجام شده است و از صحت و اصالت برخوردار است.
- در استفاده از نتایج پژوهشهای محققان دیگر به مرجع مورد استفاده استناد شده است.
- مطالب مندرج در پایان نامه تاکنون توسط خود یا فرد دیگری برای دریافت هیچ نوع مدرک یا امتیازی در هیچ جا ارائه نشده است.
- کلیه حقوق معنوی این اثر متعلق به دانشگاه شاهرود می باشد و مقالات مستخرج با نام « دانشگاه شاهرود » و یا « Shahrood University» به چاپ خواهد رسید.
- حقوق معنوی تمام افرادی که در به دست آمدن نتایج اصلی پایان نامه تأثیرگذار بوده اند در مقالات مستخرج از پایان نامه رعایت می گردد.
- در کلیه مراحل انجام این پایان نامه ، در مواردی که از موجود زنده (یا بافتهای آنها) استفاده شده است ضوابط و اصول اخلاقی رعایت شده است.
- در کلیه مراحل انجام این پایان نامه، در مواردی که به حوزه اطلاعات شخصی افراد دسترسی یافته یا استفاده شده است اصل رازداری ، ضوابط و اصول اخلاق انسانی رعایت شده است .

تاریخ : ۹۴/۰۶/۲۹

امضای دانشجو :

مالکیت نتایج و حق نشر

- کلیه حقوق معنوی این اثر و محصولات آن (مقالات مستخرج، کتاب، برنامه های رایانه ای، نرم افزار ها و تجهیزات ساخته شده است) متعلق به دانشگاه شاهرود می باشد. این مطلب باید به نحو مقتضی در تولیدات علمی مربوطه ذکر شود.
- استفاده از اطلاعات و نتایج موجود در پایان نامه بدون ذکر مرجع مجاز نمی باشد.

چکیده

جستجوی کلید واژه در گفتار، یکی از شاخه‌های پردازش گفتار می‌باشد که در کاربردهای امنیتی و سیستم‌هایی که نیاز به ارتباط برقرار کردن با انسان از طریق گفتار دارند کاربرد زیادی دارد. اگر چه در مقالات مختلف بررسی‌هایی در مورد سیستم‌های جستجوی کلید واژه انجام شده است اما معمولا این بررسی‌ها در زمینه بهبود درصد شناسایی کلمات در گفتار و بهبود کارایی سیستم بوده است و معمولا کمتر از منظر پیاده سازی سخت‌افزاری به این موضوع پرداخته شده است. در این پژوهش تمرکز بر روی پیاده سازی سخت افزاری این روش و بررسی روش‌هایی می‌باشد که می‌تواند پیاده سازی سخت افزاری را تسهیل کند.

پس از بررسی‌های انجام شده، روش‌های طیفی زمان کوتاه[1] انتخاب شده و به صورت خاص از میان این دسته از روش‌های استخراج ویژگی، روش‌های MFCC و LPCC برای پیاده‌سازی سخت افزاری انتخاب شده‌اند. برای مدل کردن کلمات از مدل مخفی مارکوف استفاده شده است.

نتایج آزمایشات در محیط نرم افزار Matlab نشان دادند که روش استخراج ویژگی MFCC نسبت به روش استخراج ویژگی LPCC نتایج بهتری به دست می‌دهد. سیستم جستجوی کلید واژه به دو صورت سیستم‌های بر مبنای جستجو در گفتار گسسته و سیستم‌های بر مبنای جستجو در گفتار پیوسته شبیه سازی شد. هنگامی که از پایگاه داده ساخته شده جهت جستجوی اعداد به صورت گسسته استفاده شد، حداکثر میانگین درصد شناسایی کلمات ۹۳/۳۳ به دست آمد. نتیجه به دست آمده برای سیستم جستجوی کلمات در گفتار گسسته برای زمانی که نمونه‌های آموزش از گفتار پیوسته استخراج شده، برابر با ۸۱/۵۸ به دست آمد که نسبت به حالتی که پایگاه داده برای سیستم-های جستجو در گفتار گسسته طراحی شده است کمتر می‌باشد. حداکثر میانگین درصد شناسایی کلمات در گفتار پیوسته، زمانی که برای آموزش کلمات از نمونه‌های استخراج شده در جملات پیوسته استفاده شد، برابر با ۷۰/۶۶ به دست آمد. در نهایت درصد شناسایی برای پیاده سازی سخت افزاری سیستم جستجوی کلمات در گفتار گسسته ۷۰/۳۶ به دست آمد.

کلمات کلیدی: جستجوی کلید واژه، MFCC، LPCC، مدل مخفی مارکوف، پردازنده سیگنال TMS320C5509A

[1] Short Time Spectral Features

مقالات استخراج شده از پایان نامه

۱. مروری بر روش‌های یافتن کلیدواژه در زنجیره گفتار بر مبنای مدل مخفی مارکوف، کنفرانس ملی فناوری، انرژی و داده با رویکرد مهندسی برق و کامپیوتر – کرمانشاه، خرداد ماه ۱۳۹۴.

فهرست مطالب

فهرست شکل‌ها

فهرست جدول‌ها

فصل اول:

مقدمه

۱-۱ مقدمه

امروزه سیستم‌های زیادی وجود دارند که کارکرد آنها به نوعی ارتباط بین انسان و ماشین از طریق گفتار نیاز دارد. گرچه معمولا کمتر نیاز داریم که تمام عناصر جمله گفته شده توسط انسان را شناسایی کنیم،اما شناسایی اجزایی از جمله که برای ما نقش کلیدی دارند به وفور مورد نیاز می‌باشد. این حوزه از پردازش سیگنال‌های گفتار مربوط به شناسایی کلیدواژه(KWS[1]) می‌باشد. از سیستم‌های KWS در سیستم‌هایی که نیاز است انسان بتواند با ماشین به وسیله گفتار ارتباط برقرار کند، استفاده می‌شود و یا از دیگر کاربردهای این سیستم می‌توان در زمینه کاربردهای امنیتی اشاره کرد[1].

۱-۲ سیستم‌های جستجوی کلیدواژه

سیستم‌های جستجوی کلید واژه را می‌توان به دو دسته سیستم‌های بر مبنای جستجو در گفتار گسسته و سیستم‌های بر مبنای جستجو در گفتار پیوسته تقسیم کرد. در سیستم‌های بر مبنای گفتار گسسته هدف این است که کلمه ورودی به سیستم را با کلماتی که در پایگاه داده سیستم ذخیره شده‌اند مقایسه کرده و تعیین کرد که کلمه ورودی، با کدام یک از کلمات موجود در پایگاه داده معادل می‌باشد. در سیستم‌های بر مبنای جستجو در گفتار پیوسته هدف این است که یک یا چند کلمه کلیدی که مد نظر می‌باشد را در جمله‌ای که توسط گوینده ادا می‌شود پیدا کرد. در سیستم بر مبنای گفتار پیوسته هیچ قید و محدودیتی در مورد جمله‌ای که ادا می‌شود وجود ندارد ولی در سیستم بر مبنای گفتار گسسته کلماتی که برای ورود به سیستم ادا می‌شوند باید به صورت جداگانه و با مکث وارد سیستم بشوند[2].

[1]- Keyword Spotting

۱-۳هدف پایان نامه

هدف این پایان نامه پیاده سازی سخت افزاری یک سیستم جست جوی کلیدواژه توسط مدار طراحی شده بر مبنای پردازنده سیگنال TMS320C5509A می‌باشد. به همین دلیل ابتدا به بررسی روش‌های مختلف جست و جوی کلیدواژه‌ای پرداخته شده است که توانایی پیاده‌سازی سخت‌افزاری به صورت بهینه را داشته باشند. در ادامه، شبیه‌سازی‌هایی انجام شده و بهترین روش با توجه به امکاناتی که پردازنده سیگنال موجود در اختیار کاربر قرار می‌دهد و کیفت جست و جوی کلیدواژه روش مورد بررسی، انتخاب شده است. در نهایت مداری بر مبنای پردازنده TMS320C5509A طراحی شده و الگوریتم جست و جوی کلیدواژه انتخابی بر روی آن پیاده سازی و بررسی شده است.

۱-۴ساختار پایان نامه

در فصل دوم به بررسی مفاهیم سیستم‌های جست و جوی کلیدواژه پرداخته شده و روش‌های پرکاربرد استخراج ویژگی، مدل کردن و کلاسه‌بندی مورد بحث و بررسی قرار گرفته است. در ادامه در فصل سوم سیستم سخت‌افزاری پیشنهادی معرفی شده است. بعد از معرفی سخت افزار، در فصل چهارم، نتایج آزمایشات در محیط Matlab برای انتخاب مناسب‌ترین روش جست و جوی کلیدواژه و همینطور نتیجه آزمایش بر روی سخت افزار طراحی شده ارائه شده است و در نهایت در فصل پنجم به جمع بندی و ارائه پیشنهادات برای کارهای آینده پرداخته شده.

فصل دوم

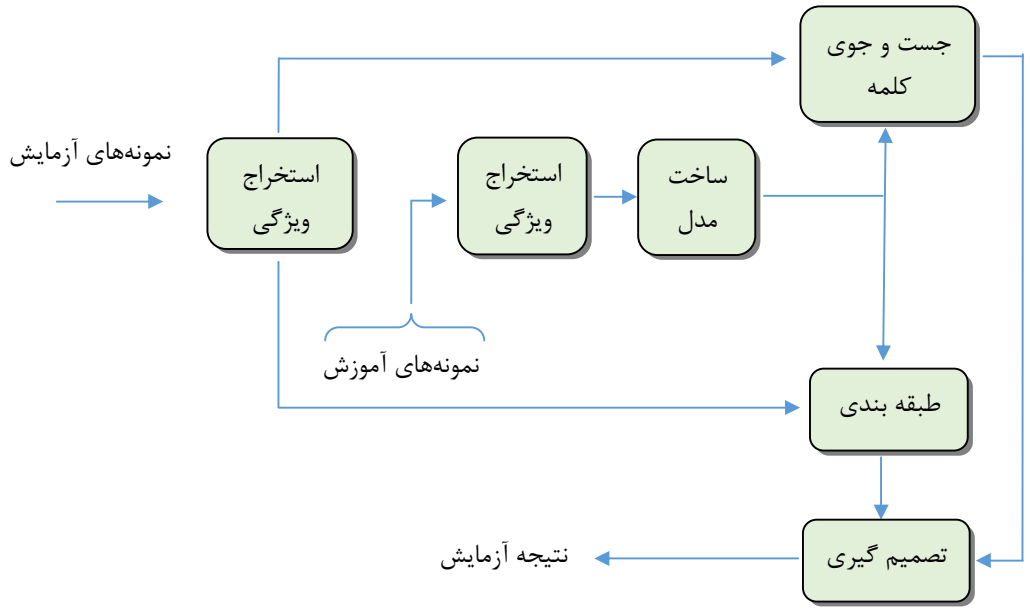
مروری بر مفاهیم و روش‌های جست و جوی کلیدواژه

مقدمه

در این فصل ابتدا به طور کلی به معرفی سیستم جست و جوی کلید واژه پرداخته می‌شود و در ادامه قسمت‌های مختلف این سیستم معرفی می‌شود. روش‌های مختلف استخراج ویژگی، مدل کردن کلمه که در مراجع گوناگون ارائه شده مورد بررسی قرار گرفته است.

۲-۱معرفی بخش‌های سیستم جست و جوی کلیدواژه

در شکل (۲-۱) نمودار بلوکی یک سیستم جست و جوی کلیدواژه مشاهده می‌شود. در قسمت آموزش ابتدا با استفاده از روش‌های مختلف استخراج ویژگی، ویژگی‌های نمونه‌های آموزش استخراج می‌شود و سپس بسته به روش مورد استفاده، برای کلمه مورد نظر یک مدل ساخته می‌شود. در قسمت آزمایش ابتدا از نمونه‌های آزمایش ویژگی‌های لازم استخراج شده و سپس بسته به این که سیستم مورد نظر یک سیستم تشخیص کلمات کلیدی گسسته باشد و یا این که یک سیستم تشخیص کلمات در گفتار پیوسته باشد ، به ترتیب از روش‌های مختلف طبقه بندی و جست و جوی کلمات استفاده می‌شود. در نهایت قسمت پایانی مربوط به تصمیم گیری است که مشخص می‌کند سیگنال ورودی به سیستم، کلمه مورد نظر سیستم می‌باشد و یا اینکه کلمه‌ای غیر از کلمات کلیدی موجود در سیستم است[3].



شکل(۲-۱): دیاگرام بلوکی سیستم جست و جوی کلیدواژه[3]

۲-۲ استخراج ویژگی

اولین مرحله در یک سیستم جست و جوی کلیدواژه، استخراج ویژگی‌های مناسب از سیگنال گفتار می‌باشد. استخراج ویژگی یک مرحله حیاتی برای سیستم بوده و تاثیر بسزایی در کارایی سیستم دارد[4]. طبیعتا نمی‌توان از هر نوع روش استخراج ویژگی استفاده کرد و باید ویژگی‌هایی انتخاب شوند که به بهترین وجه بتوانند سیگنال گفتار را مدل کنند و البته برای هدف ما، یعنی جست و جوی کلیدواژه، نیز مناسب باشند. در ضمن در کاربردهای مد نظر این پایان نامه، روش استخراج ویژگی که انتخاب می‌شود باید برای کاربردهای سخت افزاری مناسب بوده و قابلیت پیاده‌سازی را نیز داشته باشد. اما ویژگی‌هایی که برای استفاده در سیستم جست و جوی کلیدواژه استفاده می‌شوند باید چه خصوصیاتی داشته باشند؟ در ادامه به چند نمونه از این خصوصیات اشاره شده است. به ویژگی‌ای ویژگی مناسب می‌گویند که[5, 6, 7]:

- در مقابل نویز و اعوجاج مقاوم باشد

- به دفعات زیاد و به صورت طبیعی در سیگنال گفتار اتفاق بیفتد
- اندازه‌گیری آن از سیگنال گفتار آسان باشد
- برای مدل کردن کلمه نیاز به تعداد بسیار زیادی بردار ویژگی نباشد

ابعاد بردار ویژگی و همینطور تعداد این بردارها به صورت نمایی بر میزان حجم محاسبات در هنگام آموزش و آزمایش سیستم تاثیر می‌گذارند و با افزایش تعداد و خصوصا ابعاد بردار ویژگی، بار محاسباتی بیشتر می‌شود[8].

روش‌های مختلفی برای طبقه‌بندی کردن ویژگی‌ها وجود دارد. برای نمونه به منظور دسته‌بندی ویژگی‌ها، می‌توان دسته‌های زیر را معرفی کرد[5]:

- ویژگی‌های طیفی زمان کوتاه[1].
- ویژگی‌های ریتمیک[2].
- ویژگی‌های سطح بالا

سیگنال گفتار وقتی که در بازه‌های زمانی کوتاه(در حدود چند میلی ثانیه) بررسی شود، دارای خصوصیات شبه ایستا[3] می‌باشد. روش طیفی زمان کوتاه از این خاصیت بهره می‌برد و با بررسی سیگنال گفتار در بازه‌های زمانی کوتاه، سیگنال را در آن بازه مدل می‌کند و مشخصه‌های سیگنال را استخراج می‌کند. در این روش می‌توان ویژگی‌های سیگنال گفتار را در حوزه زمان و یا فرکانس استخراج کرد[9].

گاهی برای مدل کردن یک سیگنال گفتار به خصوصیات پیشرفته‌تری در سطح محاوره مراجعه می‌کنند، برای مثال خصوصیاتی مانند معنای کلمات و یا نوع کلماتی که استفاده میکنیم. به ویژگی-

[1]Short Term Spectral Features
[2]Prosodic Features
[3]Quasi Stationary

هایی که با استفاده از این خصوصیات استخراج می‌شوند، ویژگی‌های سطح بالا می‌گویند[5].

برای محققی که در زمینه پردازش گفتار کار می‌کند این سوال پیش می‌آید که کدام یک از این روش‌های استخراج ویژگی روش بهتری است؟ باید گفت که قبل از هر چیز انتخاب روش استخراج ویژگی به نوع کاربرد بستگی دارد. برای مثال وقتی که قرار است یک سیستم برخط طراحی شود، استفاده از ویژگی‌های طیفی زمان کوتاه انتخاب عاقلانه‌تری نسبت به ویژگی‌های سطح بالا می‌باشد. زیرا زمانی که صرف استخراج ویژگی‌های زمان کوتاه می‌شود زمان کمتری است. علاوه بر این انتخاب مدلی که قرار است با ویژگی‌های استخراج شده آموزش داده شود نیز در انتخاب روش تاثیر گذار است. روش‌های استخراج ویژگی که معرفی شد، به صورت عمومی یک سری خصوصیات دارند که در هنگام انتخاب آنها باید به این خصوصیات دقت کرد. در مورد ویژگی‌های طیفی زمان کوتاه یکی از مهمترین خصوصیاتی که می‌توان به آن اشاره کرد، سرعت بالای استخراج ویژگی است. در ضمن برای آموزش مدل به کمک ویژگی‌های طیفی زمان کوتاه، نیاز به مقادیر زیادی داده برای آموزش نیست و به راحتی نیز قابل استخراج هستند[10, 11].

بر خلاف روش‌های طیفی زمان کوتاه، روش‌های استخراج ویژگی سطح بالا به زمان بیشتری برای استخراج ویژگی نیاز دارند. در ضمن این ویژگی‌ها به میزان زیادی داده برای استخراج ویژگی و مدل کردن گوینده نیاز دارند. شاید به بزرگترین حسنی که برای ویژگی‌های سطح بالا در مقابل ویژگی‌های طیفی زمان کوتاه می‌توان اشاره کرد، مقاومت آنها در مقابل نویز و اثرات کانال انتقال باشد. ویژگی‌های ریتمیک از نظر قدرت، در بین دو روش استخراج ویژگی طیفی زمان کوتاه و سطح بالا قرار می‌گیرند. برای مثال سرعت آنها از روش‌های سطح بالا بیشتر ولی از روشهای طیفی زمان کوتاه کمتر می‌باشد و یا در مقایسه با ویژگی‌های سطح بالا به میزان کمتری داده برای استخراج ویژگی نیاز دارند[4, 12].

با توجه به مطالبی که ارائه شد، برای کاربردهای شناسایی گفتار و نیز در سیستم‌های

جستجوی کلید واژه، معمولا از ویژگی‌های طیفی زمان کوتاه استفاده می‌شود[16-13]. دلیلی که در این پایان نامه، این دسته از روش‌ها برای استخراج ویژگی انتخاب شد این است که برای پیاده‌سازی سخت افزاری مناسب‌اند. در ادامه روش‌های استخراج ویژگی طیفی زمان کوتاه مورد بررسی قرار گرفته است.

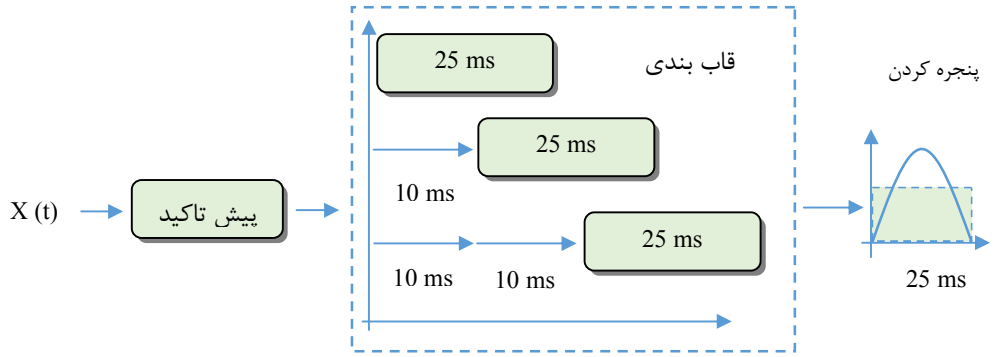
۲-۳ روش استخراج ویژگی طیفی زمان کوتاه

۲-۳-۱ پیش پردازش

همانگونه که در بخش قبل گفته شد، در روش استخراج ویژگی طیفی زمان کوتاه سیگنال گفتار به قسمت‌های کوچک تقسیم می‌شود تا بتوان ویژگی‌های مورد نظر را از آنان استخراج کرد. طرح کلی این روش در شکل (۲-۲) نمایش داده شده است. مرحله اول عبارت است از یک پیش پردازش که عبارت است از اعمال یک فیلتر بالاگذر مرتبه اول به سیگنال. این فیلتر به صورت معمول به صورت معادله (۲-۱) می‌باشد [17].

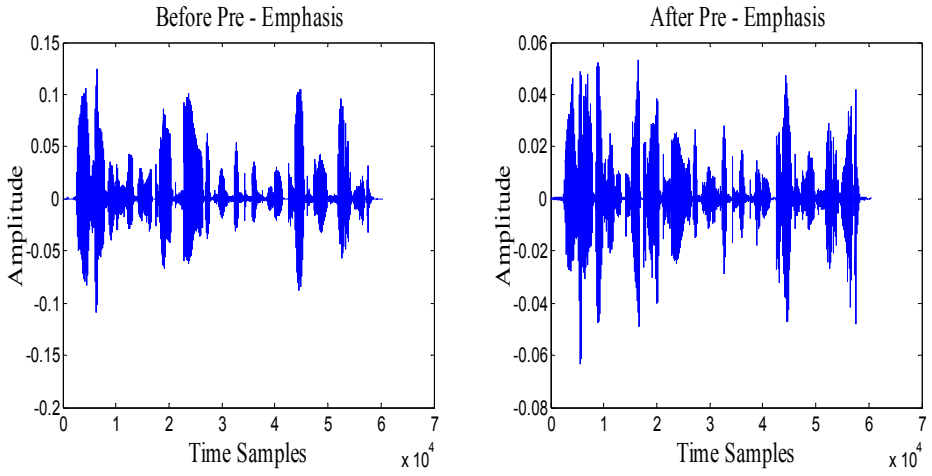
$$y(t) = x(t) - 0.97x(t-1) \qquad (۱-۲)$$

که در آن x(t) عبارت است از سیگنال فیلتر نشده و y(t) عبارت است از سیگنالی که فیلتر به آن اعمال شده است.



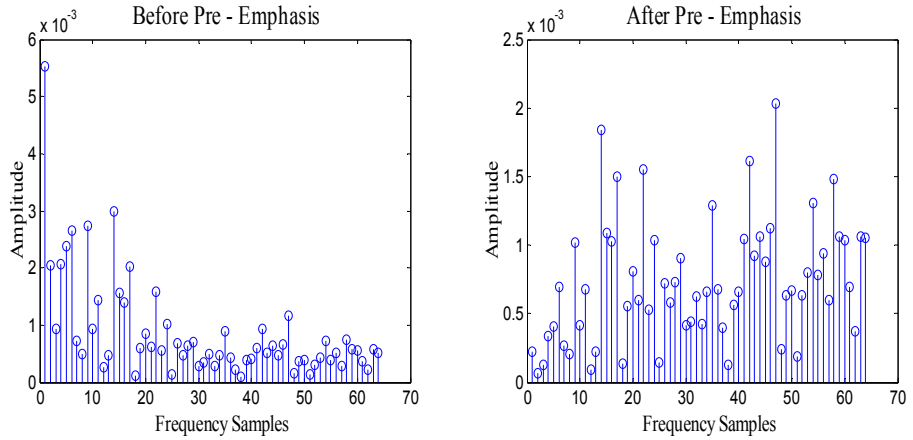
شکل(۲-۲): دیاگرام بلوکی پیش‌پردازش سیگنال گفتار[17]

سیستم تولید گفتار در انسان به طور طبیعی به صورتی می‌باشد که فرکانس‌های بالای سیگنال گفتار را تضعیف می‌کند. اعمال فیلتر معادله (۲-۱) به محتویات فرکانس بالای سیگنال گفتار اجازه می‌دهد که در مقابل محتویات مربوط به فرکانس‌های پایین‌تر خود را نشان دهند و اثر فرکانس‌های بالا در ویژگی‌های استخراج شده کم نشود. برای مثال در شکل (۲-۳) یک سیگنال گفتار قبل و بعد از پس پردازش نمایش داده شده است. شکل سمت چپ مربوط به قبل از اعمال فیلتر و شکل سمت راست مربوط به بعد از فیلتر کردن می‌باشد.



شکل(۲-۳): نمودار سیگنال گفتار قبل و بعد از اعمال فیلتر بالاگذر(شکل سمت چپ مربوط به قبل از اعمال فیلتر و شکل سمت راست مربوط به بعد از اعمال فیلتر می‌باشد)

در ادامه تبدیل فوریه هر دو را محاسبه و نتیجه در شکل (۲-۴) نمایش داده شده است. همانطور که مشاهده می‌شود، در شکل سمت چپ که مربوط به قبل از اعمال فیلتر بالا گذر می‌باشد، توزیع محتویات فرکانسی به صورت نامتقارن می‌باشد در صورتی که در شکل سمت راست که مربوط به بعد از اعمال فیلتر می‌باشد، محتویات فرکانسی توزیع متقارن‌تری دارند.



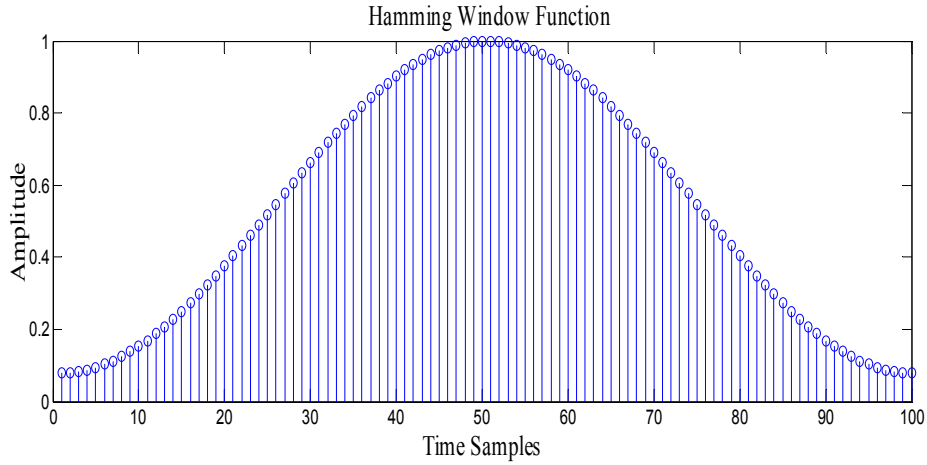
شکل(۲-۴): نمایش تبدیل فوریه گسسته سیگنال گفتار (شکل سمت چپ مربوط به قبل از اعمال فیلتر و شکل سمت راست مربوط به بعد از اعمال فیلتر می‌باشد)

وقتی که فیلتر بالا گذر اعمال شد، در مرحله بعد باید سیگنال قاب[1] بندی شود. طول هر قاب را را معمولا بین ۲۰ تا ۳۰ میلی ثانیه(به طور متوسط ۲۵ میلی ثانیه) در نظر می‌گیرند. معمولا برای اینکه انتقال از یک قاب به قاب بعدی به صورت آرام[2] انجام شود، قاب‌ها را به صورتی از سیگنال گفتار انتخاب می‌کنند که با هم همپوشانی داشته باشند. مثلا در شکل (۲-۲) هر دو قاب متوالی با یکدیگر ۱۵ میلی ثانیه همپوشانی دارند. پس با فرض اینکه قاب‌ها هر کدام ۲۵ میلی ثانیه طول دارند، هر ۱۰ میلی‌ثانیه یک بار قاب برداری انجام می‌شود[18] .

در مرحله پایانی پیش پردازش یک تابع پنجره به هر کدام از قاب‌های استخراج شده از سیگنال گفتار اعمال می‌کنند. این کار به این دلیل است که وقتی یک قاب را از سیگنال گفتار انتخاب می‌کنند عملا از یک تابع پنجره مستطیلی استفاده می‌شود، از طرفی بسیاری از روش‌های استخراج ویژگی طیفی زمان کوتاه بر اساس تبدیل فوریه عمل می‌کنند و استفاده از پنجره مستطیلی در حوزه زمان باعث اعوجاج در حوزه فرکانس می‌شود. برای غلبه بر این مشکل از توابع پنجره‌ای مانند همینگ[3] و یا هنینگ[4] استفاده می‌شود. این پنجره‌ها به آرامی دامنه دو سمت قاب را صفر می‌کنند و

[1]Frame
[2]Smooth
[3]Hamming
[4]Hanning

اثر یک‌باره صفر شدن تابع پنجره مستطیلی را تا حدودی رفع می‌کنند و باعث می‌شوند کـه درحـوزه فرکانس اعوجاج کمتری ایجاد شود[19] . برای نمونه در شکل (۲-۵) پنجـره همینـگ ۱۰۰ نقطـه‌ای نمایش داده شده است.



شکل(۲-۵): پنجره همینگ ۱۰۰ نقطه‌ای

با اعمال تابع پنجره به هر یک از قاب‌ها، مرحله پیش‌پردازش به اتمام می‌رسد و بسته بـه نـوع روش استخراج ویژگی که انتخاب می‌شود، مراحل بعدی متفاوت اسـت. در ادامـه روش‌هـای مختلـف استخراج ویژگی طیفی زمان کوتاه معرفی شده است.

۲-۳-۲ دسته‌بندی روش‌های استخراج ویژگی طیفی زمان کوتاه

بعد از مرحله پیش پردازش نوبت استخراج ویژگی‌ها می‌رسد. روش‌های استخراج ویژگی طیفی زمان کوتاه را می‌توان به دو دسته کلی تقسیم کرد. دسته اول روش‌هایی هستند که بر اساس سیستم شنوایی انسان ویژگی‌های سیگنال گفتار را استخراج می‌کنند و دسته دوم روش‌هایی هسـتند کـه بـر اساس مشخصات سیستم تولید صدای انسان، سیگنال گفتار را مدل می‌کنند[6]. در ادامه هر دو دسته و روش‌های معروف و پر استفاده‌ای که بر اساس آن‌ها پیشنهاد شده، ارائه شده‌اند.

۲-۳-۳ روش‌های استخراج ویژگی طیفی زمان کوتاه بر مبنای شنوایی انسان

روش‌های استخراج ویژگی بر مبنای سیستم شنوایی انسان در چند قسمت کلی مشترک هستند. بعد از اینکه سیگنال از مرحله پیش‌پردازش عبور کرد، با استفاده از تبدیل فوریه گسسته[1] (DFT) و یا تبدیل موجک گسسته[2] (DWT) سیگنال گفتار را به فضای دیگری انتقال داده و سپس از یک فیلتر بانک استفاده می‌شود[11]. فیلتر بانک‌ها شامل فیلترهای میان‌گذری هستند که بر اساس نوع روشی که استفاده می‌شود متفاوت می‌باشند و مقیاس‌بندی آن‌ها نیز بسته به اینکه از چه روشی استفاده می‌شود می‌تواند خطی باشد و یا اینکه از نوع مقیاسی که انتخاب شده است پیروی کند. وقتی که تبدیل فوریه و یا تبدیل موجک یک قاب محاسبه می‌شود، معمولا اطلاعات زیادی به دست می‌آید که همه این اطلاعات مورد نیاز نمی‌باشد و علاوه بر این حجم زیاد اطلاعات نیز باعث پیچیدگی محاسباتی زیادی می‌شود. برای مثال فرض کنید تبدیل فوریه یک قاب را محاسبه کرده و نتیجه این عمل ۲۵۶ عدد است. وقتی که از یک فیلتر بانک شامل ۳۰ فیلتر استفاده می‌شود، تمام اطلاعات این ۲۵۶ عدد در ۳۰ عدد ذخیره می‌شود. به این ترتیب اطلاعات زیادی از دست داده نمی‌شود و در عین حال پیچیدگی محاسباتی به میزان زیادی کاهش می‌یابد. روش‌ها را بر حسب اینکه از DFT و یا DWT استفاده کنند و اینکه چه نوع فیلتر بانکی را انتخاب کنند دسته‌بندی و نام‌گذاری می‌کنند. در ادامه چند روش پیشنهاد شده در مقالات مختلف بررسی شده است.

۲-۳-۳-۱ روش استخراج ویژگی ضرایب کپسترال در مقیاس مِل (MFCC)

شناخته شده‌ترین روش این دسته که در بسیاری از مقالات به عنوان روش استاندارد استخراج ویژگی شناخته می‌شود روش ضرایب کپسترال در مقیاس مِل[3](MFCC) می‌باشد[7, 11, 15, 20, 21].

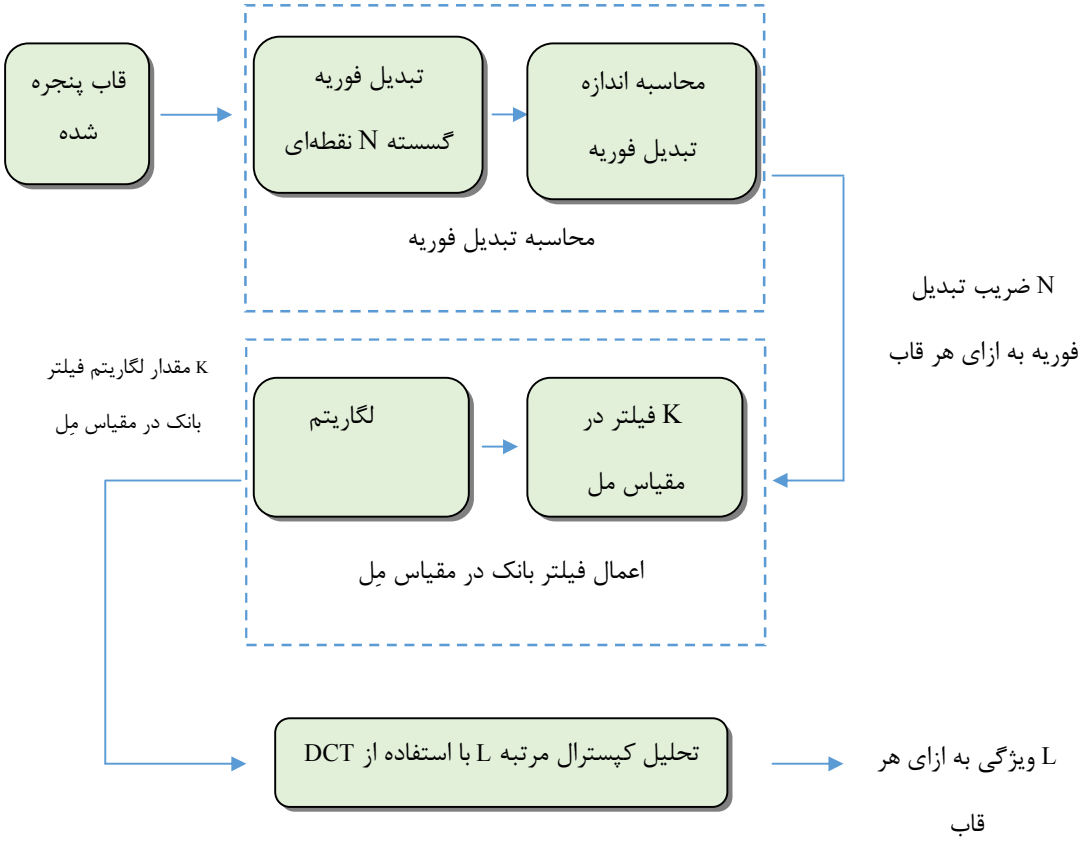
[1]Discrete Fourier Transform
[2]DiscreteWavelet Transform
[3]Mel Frequency Cepstral Coefficient

در روش MFCC از تبدیل فوریه گسسته استفاده می‌شود و فیلتر بانک مورد استفاده، فیلتر بانک مِل می‌باشد. اما روش به دست آوردن ویژگی‌های MFCC دقیقا چگونه است؟ در شکل (۲-۶) نمودار بلوکی این روش نمایش داده شده است[17].

هنگامی که در مرحله پیش پردازش سیگنال گفتار به قاب‌های پنجره شده تقسیم شدند، آنگاه تبدیل فوریه گسسته را برای هر قاب محاسبه می‌کنند. این کار با استفاده از الگوریتم‌های محاسبه سریع تبدیل فوریه گسسته[1] (FFT) انجام می‌شود. برای مثال فرض شده که برای هر قاب، تبدیل فوریه گسسته N = 512 نقطه‌ای محاسبه شده است. در استخراج ویژگی برای جست و جوی کلید واژه به اندازه تبدیل فوریه نیاز است و اطلاعات فاز در نظر گرفته نمی‌شود. پس اندازه هر ۵۱۲ عدد را محاسبه می‌کنیم. در ادامه به دلیل تقارن FFT نیمی از داده‌ها را نگه داشته و از نیمه دیگر صرفنظر می‌شود. پس اکنون به ازای یک قاب، ۲۵۶ عدد حقیقی به دست آمده است ولی هنوز هم این تعداد عدد زیاد است و نیازی به همه آن‌ها نیست. پس با استفاده از فیلتر بانک این تعداد داده را خلاصه‌تر کرده و در K = 30 عدد نمایش می‌دهند. برای این کار از فیلتر بانک در مقیاس مِل استفاده می‌شود. برای تبدیل مقیاس‌بندی خطی به مقیاس مِل از معادله (۲-۲) استفاده می‌شود[6].

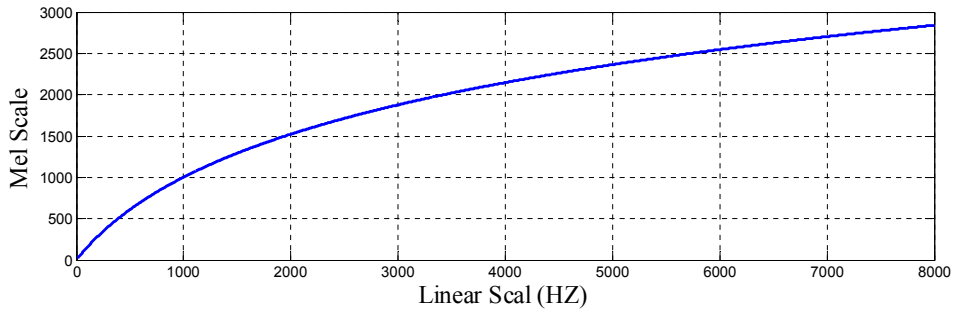
$$F_{Mel} = 2595\, log_{10}\left(1 + \frac{F_{Linear}}{700}\right) \qquad (۲-۲)$$

[1] Fast Fourier Transform



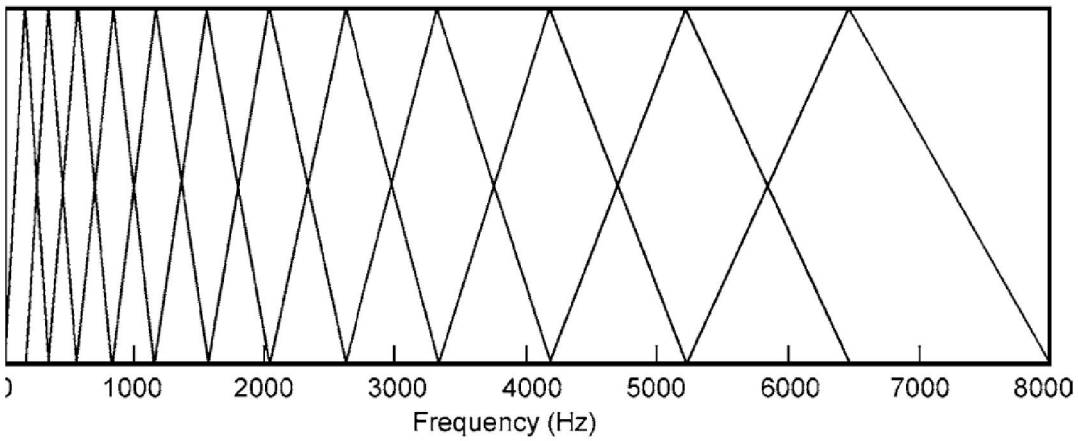
شکل(۲-۶): نمودار بلوکی استخراج ویژگی به روش MFCC [1]

همانطور که در شکل (۲-۷) مشاهده می‌شود، مقیاس مل بین ۰ تا ۱۰۰۰ هرتز، تقریبا به صورت خطی می‌باشد اما بعد از آن به صورت غیر خطی رفتار می‌کند.



شکل(۲-۷): نمودار مقیاس مِل بر مبنای مقیاس خطی

هر کدام از فیلترهای میان‌گذری که در فیلتر بانک استفاده می‌شود، یک فیلتر مثلثی می‌باشد و تمام فیلترها با همدیگر همپوشانی دارند. ابتدا باید مرکز هر فیلتر محاسبه شود. اگر فرض شود حداکثر فرکانس ۸۰۰۰ هرتز باشد(فرکانس نمونه‌برداری ۱۶۰۰۰ هرتز است)، با استفاده از معادله (۲-۲) اندازه حداکثر و حداقل فرکانس در حوزه مِل را بر اساس فرکانس خطی محاسبه می‌کنند. مثلا معادل فرکانس ۸۰۰۰ هرتز در حوزه مِل برابر با 2840 است . در ادامه محدوده بین کمترین و بیشترین فرکانسی در حوزه مل را به تعداد فیلتری که مورد نظر است تقسیم می‌کنند. مثلا در اینجا تعداد فیلترها۳۰ فرض شده است. با این کار مراکز فیلترها در حوزه مِل به دست می‌آید. سپس با استفاده از معکوس معادله (۲-۲) این مراکز را به حوزه فرکانس خطی هرتز انتقال می‌دهند. ابتدای هر فیلتر مثلثی از مرکز فیلتر قبل شروع شده و انتهای آن نیز در مرکز فیلتر بعد از خودش قرار می‌گیرد. یک نمونه از این فیلتر بانک در شکل (۲-۸) نمایش داده شده است.



شکل(۲-۸): یک نمونه فیلتر بانک در مقیاس مِل

در ادامه مقادیر اندازه FFT را در اندازه فیلتر مثلثی مربوط به هر فرکانس ضرب کرده و نتایج حاصلضرب هر مثلث را با هم جمع می‌کنند و به این ترتیب ۲۵۶ عدد به ۳۰ عدد تبدیل می‌شوند. معمولا به این دلیل که سیستم شنوایی انسان از یک منحنی لگاریتمی پیروی می‌کند، برای نشان دادن این اثر از خروجی هر کدام از فیلترها لگاریتم می‌گیرند.

با استفاده از تحلیل کپسترال می‌توان باز هم داده‌ها را خلاصه‌تر کرد و به ساده‌تر شدن محاسبات و سریع‌تر شدن روش استخراج ویژگی کمک کرد. در اینجا برای به دست آوردن ضرایب کپسترال از تبدیل کسینوسی گسسته[1] (DCT) استفاده می‌شود. معادله‌ای که برای این تبدیل استفاده می‌شود عبارت است از[22] :

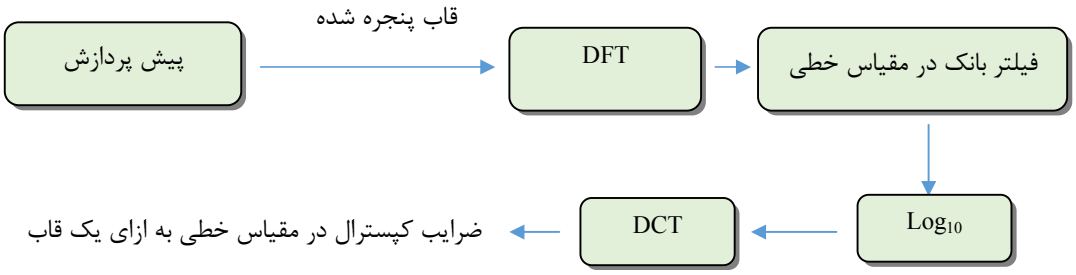
$$C_n = \sum_{k=1}^{K} (S_k) \cos\left(n\,(k-0.5)\left(\frac{\pi}{K}\right)\right) \quad , \quad n = 1,2,3 \ldots, L \qquad (۳-۲)$$

که در این فرمول S_k عبارت است از لگاریتم خروجی فیلتر k ام و C_n ضریب کپسترال n ام می‌باشد. به طور معمول L = 12 فرض می‌شود و ضریب C_0 را بسته به کاربرد می‌توان استفاده کرد و یا از آن صرفنظر کرد.

۲-۳-۳-۲ روش استخراج ویژگی ضرایب کپسترال در مقیاس خطی[2] (LFCC)

این روش همانند روش MFCC می‌باشد با این تفاوت که فیلتر بانکی که استفاده می‌شود دیگر بر مبنای مقیاس مِل نمی‌باشد بلکه در مقیاس خطی تقسیم‌بندی شده و طول تمام فیلترهای مثلثی با یکدیگر یکسان می‌باشد. دیاگرام بلوکی این روش در شکل (۲-۹) ارائه شده است[21].

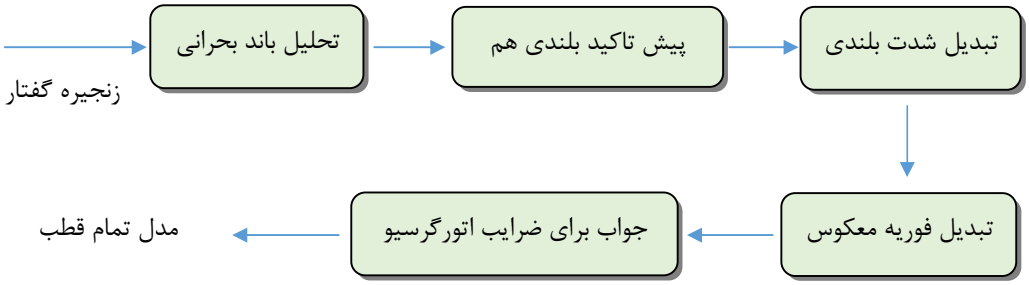
[1] Discrete Cosine Transform
[2] Linear Frequency Cepstral Coefficients



شکل(۲-۹): دیاگرام بلوکی استخراج ویژگی به روش LFCC

۲-۳-۳-۳ روش استخراج ویژگی ادراکی پیش‌گویی خطی[1](PLP)

روش PLP طیف شنیداری را به صورت یک مدل تمام قطب با مرتبه پایین مدل می‌کند. دیاگرام بلوکی این روش در شکل (۲-۱۰) نمایش داده شده است[13].



شکل(۲-۱۰): دیاگرام بلوکی روش استخراج ویژگی PLP

در ادامه قسمت‌های مختلف این دیاگرام بلوکی شرح داده شده است.

✓ تحلیل باند بحرانی

در این قسمت ابتدا مرحله پیش‌پردازش، همانگونه که در قسمت (۲-۴-۱) شرح داده شده است، انجام می‌شود. سپس FFT قاب پنجره شده را محاسبه کرده و اندازه طیف را به دست می‌آورند.

[1]Perceptual Linear Prediction

در مرحله بعد باید مقیاس خطی را به مقیاس بارک[1] انتقال داد. این کار را می‌توان با استفاده از معادله (۲-۴) انجام داد[6, 13].

$$\Omega(\omega) = 6\, ln\left\{\frac{\omega}{1200\pi} + \left[\left(\frac{\omega}{1200\pi}\right)^2 + 1\right]^{0.5}\right\} \quad (۴-۲)$$

که در آن ω فرکانس زاویه‌ای با یکای رادیان بر ثانیه می‌باشد. در ادامه طیف توان انتقال یافته به مقیاس بارک را با تابع $\Psi(\Omega)$ کانولوشن می‌کنند. تابعی که برای کانولوشن استفاده می‌شود معمولا به صورت معادله (۲-۵) می‌باشد.

$$\Psi(\Omega) = \begin{cases} 0 & , \quad \Omega < -1.3 \\ 10^{2.5(\Omega+0.5)} & , \quad -1.3 \le \Omega \le -0.5 \\ 1 & , \quad -0.5 < \Omega < 0.5 \\ 10^{-(\Omega-0.5)} & , \quad 0.5 < \Omega < 2.5 \\ 0 & , \quad \Omega > 2.5 \end{cases} \quad (۵-۲)$$

✓ پیش تاکید بلندی هم تراز

در این مرحله نتیجه کانولوشن مرحله قبل را در تابع پیش تاکید$E(\omega)$ ضرب می‌کنند. این کار به منظور مدل کردن سیستم شنوایی انسان می‌باشد. تابع پیش تاکید با معادله (۲-۶) نمایش داده می‌شود.

$$E(\omega) = \frac{[(\omega^2 + 5.68 \times 10^6)\omega^4]}{[(\omega^2 + 6.3 \times 10^6)^2 \times (\omega^2 + 0.38 \times 10^9) \times (\omega^6 + 9.58 \times 10^{26})]} \quad (۶-۲)$$

✓ تبدیل شدت بلندی و محاسبه ویژگی‌ها

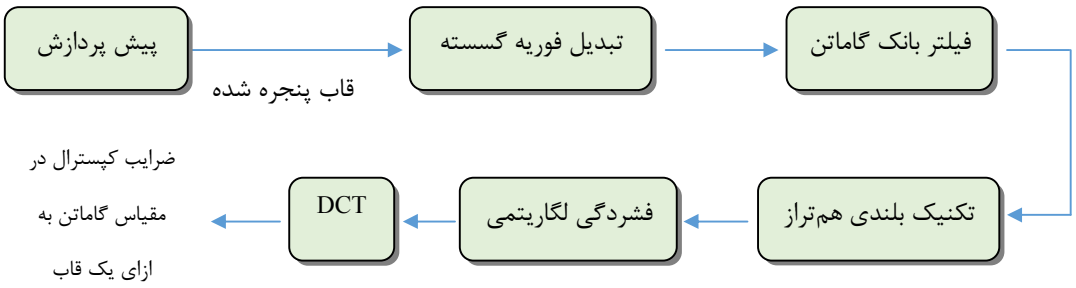
بعد از مرحله پیش تاکید، برای اینکه تبدیل شدت بلندی انجام شود، نتیجه ضرب مرحله قبل را به توان ۰/۳۳ می‌رسانند. این کار به شبیه سازی رابطه غیر خطی بین شدت صدا و بلندی صدای

[1] Bark Scale

درک شده کمک می‌کند. در ادامه تبدیل فوریه معکوس گسسته برای مقادیر به دست آمده را محاسبه می‌کنند و از M+1 ضریب اولیه خود همبستگی استفاده کرده و معادله Yule – Walker را برای ضرایب اتورگرسیو[1] یک مدل تمام قطب مرتبه M حل می‌کنند. ضرایب به دست آمده، ویژگی‌های به دست آمده به ازای یک قاب هستند[15].

۲-۳-۳-۴ روش استخراج ویژگی ضرایب کپسترال در مقیاس گاماتن[2] (GFCC)

این روش ترکیبی از روش‌های MFCC و PLP می‌باشد. به این ترتیب که برای قسمتی که به جای فیلتر بانک در مقیاس مِل، از فیلتر بانک در مقیاس گاماتن استفاده می‌کنند. بعد از اعمال فیلتر بانک، مانند روش PLP از تکنیک بلندی هم‌تراز استفاده می‌کنند[27 ,26]. دیاگرام بلوکی این روش استخراج ویژگی در شکل (۲-۱۱) نمایش داده شده است.



شکل(۲-۱۱): دیاگرام بلوکی استخراج ویژگی به روش GFCC [26]

فیلتر بانک گاماتن عبارت است از یک سری فیلتر میان گذر که سیستم شنیداری انسان را مدل می‌کنند. تابع ضربه هر کدام از این فیلترها به صورت معادله (۲-۷) می‌باشد. در این معادله a یک ثابت است که معمولا برابر با ۱ در نظر گرفته می‌شود. φ عبارت است از جابه‌جایی فاز و n مرتبه فیلتر می‌باشد. در این معادله f_c و b به ترتیب فرکانس مرکزی و پهنای باند فیلتر در حوزه هرتز

[1] Autoregressive
[2] Gammatone Frequency Cepstarl Coefficients

هستند[23].

$$g(t) = at^{n-1}e^{-2\pi t}\cos(2\pi f_c t + \varphi) \quad (۷-۲)$$

۲-۳-۳-۵ روش‌های بر مبنای تبدیل موجک

در روش‌هایی که بر مبنای تبدیل موجک گسسته هستند، در طی فرایند استخراج ویژگی به جای بخشی که از تبدیل فوریه گسسته استفاده شده، از تبدیل موجک گسسته استفاده می‌شود. روش‌های این دسته نیز بر طبق اینکه از چه نوع فیلتری برای ساخت فیلتر بانک استفاده می‌شود، با همدیگر متفاوت هستند. برای این دسته می‌توان به روش‌هایی مانند روش Farooq & Datta و یا روش Sarikaya & Hansen اشاره کرد که هر دو به نام پیشنهاد دهندگانشان نامگذاری شده‌اند[21]. در مقالات، از روش‌های بر پایه تبدیل فوریه گسسته نتایج بهتری نسبت به روش‌های بر پایه تبدیل موجک گزارش شده است.

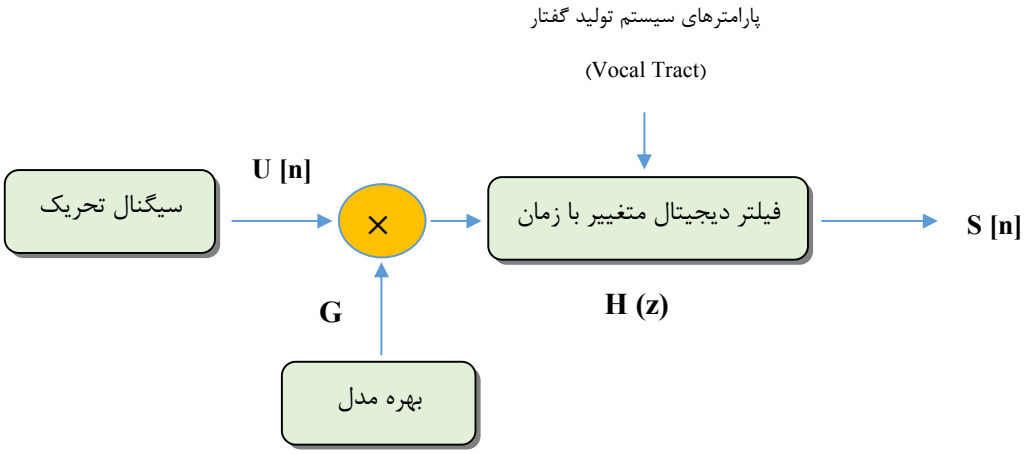
۲-۳-۴ روش‌های استخراج ویژگی بر مبنای سیستم تولید گفتار انسان

در روش‌های بر مبنای سیستم شنوایی انسان سعی می‌شود که سیستم شنوایی انسان را مدل کرده و پارامترهای آن را به دست آورد. در این بخش به بررسی دو روش کد کردن پیش‌بینی خطی[1](LPC) وفرکانس طیف خطی[2](LSF) پرداخته می‌شود.

۲-۳-۴-۱ روش استخراج ویژگی کد کردن پیش‌بینی خطی (LPC)

روش LPC بر این مبنا بنا شده است که می‌توان نمونه‌های سیگنال گفتار را با استفاده از یک ترکیب خطی از نمونه‌های قبل از آن‌ها تقریب زد. در این روش سیستم گفتار را به صورت یک مدل تمام قطب مدل می‌کنند و با استفاده از روش‌های مختلف، این پارامترها را تخمین می‌زنند[6]. فرض کنید مدلی که برای سیستم تولید گفتار انسان ارائه شده است به صورت شکل (۲-۱۲) باشد[22].

[1]Linear Predictive Coding
[2]Line Spectral Frequency



شکل(۲-۱۲): دیاگرام بلوکی ساده شده سیستم تولید گفتار انسان

برای این سیستم تابع تبدیل را به صورت معادله (۲-۸) تعریف می‌کنند.

$$H(z) = \frac{S(z)}{U(z)} = \frac{G}{1 - \sum_{k=1}^{P} a_k z^{-k}} \qquad (۲-۸)$$

نقطه قوت این مدل این است که می‌توان پارامترهای مدل، یعنی a_kها را با روش‌های سر راست و از نظر محاسباتی بهینه، از طریق تحلیل پیش‌بینی خطی به دست آورد. رابطه بین یک سیستم پیش‌بینی خطی و یک سیستم تولید گفتار در انسان در ادامه بررسی شده است.

برای سیستم شکل (۲-۱۲) می‌توان رابطه بین نمونه‌های تولید شده و سیگنال تحریک در حوزه زمان را به صورت معادله (۲-۹) نوشت[22].

$$s[n] = \sum_{k=1}^{P} a_k s[n-k] + Gu[n] \qquad (۲-۹)$$

یک سیستم پیش‌بینی خطی با ضرایب پیش‌بینی خطی α_k ، به صورت معادله (۲-۱۰) تعریف می‌شود.

$$\tilde{s}[n] = \sum_{k=1}^{P} \alpha_k s[n-k] \qquad (۲-۱۰)$$

در ادامه برای یک سیستم پیش‌بینی خطی با مرتبه P ، خطای پیش‌بینی، یعنی e[n] به صورت معادله (۲-۱۱) تعریف می‌شود.

$$e[n] = s[n] - \tilde{s}[n] = s[n] - \sum_{k=1}^{P} \alpha_k s[n-k] \qquad (۱۱-۲)$$

معادله (۲-۱۱) نشان می‌دهد که خطای پیش‌بینی را می‌توان به صورت خروجی یک سیستم با تابع تبدیل معادله (۲-۱۲) فرض کرد.

$$A(z) = 1 - \sum_{k=1}^{P} \alpha_k z^{-k} \qquad (۱۲-۲)$$

از مقایسه معادلات (۲-۸) و (۲-۱۲) می‌توان استنباط کرد که اگر سیگنال گفتار از مدل معادله (۲-۹) پیروی کند، و اگر فرض شود که $a_k = \alpha_k$، آنگاه می‌توان گفت $e[n] = Gu[n]$ است. بنابراین می‌توان گفت که تابع فیلتر خطای پیش‌بینی برابر است با عکس تابع معکوس سیستم تولید گفتار انسان. از همین خاصیت استفاده می‌شود و برای سیگنال گفتار با استفاده از روش پیش‌بینی خطی، پارامترهای مدل سیستم تولید گفتار را تخمین می‌زنند.

در سیستم پیش‌بینی خطی، هدف این است که یک سری α_k را یافت که با استفاده از آن‌ها بتوان خطای پیش‌بینی را به کمترین مقدر ممکن رساند[6]. برای انجام این کار می‌توان به سه روش اشاره کرد که این روش‌ها عبارتند از:

- روش حصیری[۱]
- روش کوواریانس[۲]
- روش خودهمبستگی[۳]

از میان این روش‌ها، در کاربردهای پردازش گفتار به دلیل بهینه بودن محاسبات روش

[1] Lattice Method
[2] Covariance Method
[3] Autocorrelation Method

خودهمبستگی، معمولا از این روش استفاده می‌شود[6]. در این روش تابع خودهمبستگی را به صورت معادله (۲-۱۳) تعریف می‌کنند.

$$R_n(K) = \sum_{m=0}^{N-1-k} s_w(m)s_w(m+k) \qquad (۱۳-۲)$$

که در این معادله R تابع خودهمبستگی و s_w قاب پنجره شده است. برای به دست آوردن ضرایب α_k باید معادله (۲-۱۴) حل شود.

$$R_n(i) = \sum_{k=1}^{P} \alpha_k R_n(|i-k|) \qquad (۱۴-۲)$$

می‌توان با استفاده از روش Levinson – Durbin معادله (۲-۱۴) را حل کرده و پارامترهای سیستم را محاسبه کرد. پس از محاسبه پارمترهای مدل تولید گفتار، می‌توان با استفاده از معادلات (۲-۱۵) تا (۲-۱۷) ضرایب کپسترال را محاسبه کرد[25].

$$C_0 = \ln(G) \qquad (۱۵-۲)$$

$$C_m = \alpha_m + \sum_{k=1}^{m-1} \left(k/m\right) C_k a_{m-k} \quad , \quad 1 \le m \le P \qquad (۱۶-۲)$$

$$C_m = \sum_{k=1}^{m-1} \left(k/m\right) C_k a_{m-k} \quad , \quad m > P \qquad (۱۷-۲)$$

۲-۴-۳-۲ استخراج ویژگی به روش فرکانس طیف خطی (LSF)

روش استخراج ویژگی LSF یک روش فرعی است که از روش LPC مشتق شده است. فرض

کنید که برای یک سیستم پیش‌بینی خطی، فیلتر خطای پیش‌بینی به صورت معادله (۲-۱۸) تعریف شده باشد.

$$A(z) = 1 - \sum_{k=1}^{P} a(k)A(z^{-k}) \qquad (۱۸-۲)$$

که در این معادله $a(\mathrm{k})$ ها ضرایب پیش‌بینی خطی هستند. می‌توان دو معادله متقارن (۲-۱۹) و نامتقارن (۲-۲۰) را به صورت زیر تعریف کرد.

$$F_1(\mathrm{z}) = \mathrm{A(z)} + \mathrm{z}^{-(\mathrm{P}+1)}\mathrm{A}(\mathrm{z}^{-1}) \qquad (۱۹-۲)$$

$$F_1(\mathrm{z}) = \mathrm{A(z)} - \mathrm{z}^{-(\mathrm{P}+1)}\mathrm{A}(\mathrm{z}^{-1}) \qquad (۲۰-۲)$$

ریشه‌های چندجمله‌ای‌های معادلات (۲-۱۹) و (۲-۲۰) ویژگی‌های LSF هستند.

۲-۳-۵ بهبود ویژگی‌ها[1]

بعد از اینکه ویژگی‌های مورد نیاز از سیگنال گفتار استخراج شد، معمولا یک سری عملیات بر روی آن‌ها به منظور بهبود کیفیت و یا کم کردن بار محاسباتی با فشرده کردن بردارهای ویژگی انجام می‌دهند. در ادامه به چند روش از روش‌های پرکاربرد اشاره شده است.

۲-۳-۵-۱ نرمالیزه کردن میانگین کپسترال[2] (CMN)

وقتی که بردارهای ویژگی استخراج شدند و ضرایب کپسترال نیز برای آنها محاسبه شد، برای اینکه بتوان اثرات نویز موجود در پایگاه داده و یا میکروفن و البته اثرات اعوجاج مربوط به کانال انتقال را کمتر کرد، می‌توان از روش CMN استفاده کرد[17]. در این روش ابتدا ضرایب کپسترال را برای تمام قاب‌های سیگنال گفتار محاسبه و سپس میانگین تمام بردارهای ویژگی را بر روی هر یک از ابعاد

[1]

[2] Cepstral Mean Normalization

بردار ویژگی محاسبه می‌کنند. این کار با استفاده از معادله (۲-۲۱) انجام می‌شود.

$$\vec{\mu}_T = \frac{1}{\mathrm{T}}\sum_{\mathrm{t}=1}^{\mathrm{T}} \overrightarrow{\mathrm{C_t}} \qquad (۲۱-۲)$$

در این معادله فرض شده است که سیگنال گفتار دارای T بـردار ویژگـی مـی‌باشـد. بردارهـای ویژگی به صورت زیر نمایش داده می‌شوند:

$$\overrightarrow{C_t} = [\, C_1, C_2, \ldots., C_L]^t \qquad (۲۲-۲)$$

در معادله (۲-۲۲) اندازه بردارهای ویژگی برابر با L می‌باشد.

حال که میانگین‌ها محاسبه شدند، بردارهای جبران‌سازی شده جدید را به صـورت معادلـه(۲-۲۳) محاسبه می‌کنند.

$$\overrightarrow{C_t^{\,c}} = \overrightarrow{C_t} - \vec{\mu}_T \qquad (۲۳-۲)$$

این کار را برای تمام بردارهای ویژگی انجام می‌دهند و به این ترتیب ویژگی‌های جبران سـازی شده CMN به دست می‌آیند. البته می‌توان به جای روش CMN از روش‌هایی مانند نرمالیزه کردن با استفاده از واریانس و میانه با هم نیز استفاده کرد. روش CMN محاسـبات کمتـری دارد و معمـول‌تـر می‌باشد[26].

۲-۳-۵-۲ مشتقات زمانی[1]

در پردازش گفتار یک نکته مهم این است که بتوان ارتباط بین بردارهای ویژگـی و البتـه پویـا بودن سیگنال گفتار را در ویژگی‌های به دست آمده اعمال کرد. برای این کار معمولا از مشتقات زمانی مرتبه اول، دوم و یا سوم استفاده می‌شود. مشتقات زمانی کمک می‌کنند کـه ارتبـاط بـین بردارهـای ویژگی و پویایی سیگنال گفتار در ویژگی‌های استخراج شده لحاظ شوند. گرچه مـی‌تـوان از مشـتقات مرتبه سوم نیز استفاده کرد اما به دلیل ملاحظات محاسباتی و جلوگیری از بزرگ شدن بیش از اندازه

[1]Temporal Derivatives

بردار ویژگی معمولا از مشتقات مرتبه اول و دوم استفاده می‌شود. برای به دست آوردن مشتقات مرتبه اول از ضرایب جبران سازی شده CMN استفاده می‌شود تا بتوان علاوه بر نمایش پویایی سیگنال گفتار، مقاومت در مقابل نویز را نیز به ویژگی‌ها افزود. مشتقات مرتبه اول که با نام" ضرایب دلتا" نیز شناخته می‌شوند، از معادله (۲-۲۴) به دست می‌آیند[17, 20].

$$\overrightarrow{d_t} = \frac{\sum_{p=1}^{P} p\left(\overrightarrow{C^c}_{t+p} - \overrightarrow{C^c}_{t-p}\right)}{2\sum_{p=1}^{P} p^2} \qquad (۲۴-۲)$$

در این معادله معمولا p=2 فرض می‌شود. $\vec{C}^c{}_t$ عبارت است از ضرایب جبران سازی شده به روش CMN و $\vec{d}_t$ نیز ضرایب دلتا می‌باشند. برای محاسبه مشتق مرتبه دوم که با نام" ضرایب دلتا – دلتا" و یا "ضرایب شتاب[1]" شناخته می‌شوند، از معادله (۲-۲۵) استفاده می‌شود.

$$\overrightarrow{a_t} = \frac{\sum_{p=1}^{P} p\left(\overrightarrow{d^c}_{t+p} - \overrightarrow{d^c}_{t-p}\right)}{2\sum_{p=1}^{P} p^2} \qquad (۲۵-۲)$$

همانگونه که ملاحظه می‌شود، رابطه (۲-۲۵) همانند رابطه (۲-۲۴) می‌باشد که برای به دست آوردن ضرایب شتاب، یعنی $\overrightarrow{a_t}$ ، از ضرایب دلتا استفاده شده است. پارامتر P نیز همانند معادله (۲-۲۴) معمولا برابر ۲ در نظر گرفته می‌شود[17].

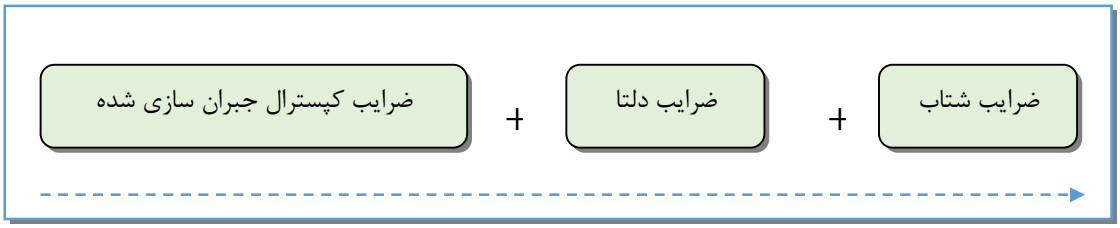
وقتی که ضرایب CMN و با استفاده از آن‌ها ضرایب دلتا و شتاب نیز محاسبه شدند، می‌توان با کنار هم قرار دادن این ویژگی‌ها در یک بردار، یک بردار ویژگی نهایی شامل هر سه دسته ضریب ایجاد کرد. شکل (۲-۱۳) این مطلب را نمایش می‌دهد.

۲-۳-۵-۳ روش فیلتر کردن نسبی طیف[2] (RASTA)

روش RASTA اولین بار توسط Hermansky برای بهبود ویژگی‌های استخراج شده از روش PLP پیشنهاد شد[4]. گرچه در ادامه برای سایر روش‌های استخراج ویژگی مانند MFCC و یا روش-های بر مبنای تحلیل کپسترال نیز به کار رفت. روش RASTA کیفیت شناسایی گفتار را در حضور

[1]Acceleration
[2]Relative Spectra Filtering

نویز حاصل از انتقال افزایش می‌دهد. این روش با اعمال فیلتر میان‌گذر بر سیگنال، طیف‌هایی از سیگنال که نرخ تغییرات آنها به نسبت نرخ تغییرات سیگنال گفتار کندتر و یا تندتر می‌باشد را حذف می‌کند[27].



شکل(۲-۱۳): ترکیب ضرایب CMN، دلتا و شتاب با یکدیگر

۲-۳-۶ جمع بندی

در این قسمت ابتدا روش‌های استخراج ویژگی دسته‌بندی شده و سپس با انتخاب روش استخراج ویژگی طیفی زمان کوتاه به عنوان یکی از پرکاربردترین روش‌های استخراج ویژگی، به تشریح این دسته از روش‌های استخراج ویژگی پرداخته شد. انواع روش‌های استخراج ویژگی طیفی زمان کوتاه شرح داده شدند و در ادامه روش‌های بهبود این ویژگی‌ها نیز بیان شد.

حال سوال این است که برای استخراج ویژگی کدام یک از روش‌ها مناسب‌تر است؟ هدف این پایان نامه این است که یک سیستم جست و جوی کلیدواژه بر روی پردازنده DSP پیاده سازی شود. پس طبیعتا روش استخراج ویژگی که انتخاب می‌شود باید به فضای حافظه کمی نیاز داشته و قابلت پیاده سازی بر روی پردازنده DSP را داشته باشد و همچنین درصد شناسایی بالایی داشته باشد. با توجه به این معیارها و البته مطالب ذکر شده در مقالات مختلف و روش‌هایی که انتخاب شده است [4,6, 14]، تصمیم گرفته شد که از دو روش پرکاربرد MFCC و LPCC استفاده شود. هر دو روش دارای ضرایب کپسترال بوده و می‌توان از آنها برای محاسب ضرایب CMN نیز استفاده کرد. در ضمن برای قسمت بهبود کیفیت ویژگی از روش‌های CMN، پارامترهای دلتا و همینطور پارامترهای شتاب استفاده خواهد شد.

۲-۴ مدل کردن کلمه کلیدی[1]

یک سیستم KWS را می‌توان به عنوان یک مسأله طبقه بندی در نظر گرفت. این سیستم یک طبقه بند دودویی می‌باشد که یک کلمه داده شده را از سایر کلمات جدا می‌کند. می‌توان این طبقه بندی را با استفاده از روش‌های بر مبنای HMM[2] انجام داد و یا اینکه از روش‌های بر مبنای متمایز سازی[3] استفاده نمود. در روش بر مبنای HMM با استفاده از نمونه‌های آموزش، یک مدل برای هر کدام از کلمات کلیدی ساخته می‌شود و هنگامی که کلمه‌ای وارد سیستم می‌شود، با مدل‌های از قبل ساخته شده مقایسه می‌شود و در نهایت تعیین می‌شود که کدام یک از کلمات کلیدی مورد نظر می-باشد[1]. در روش‌های متمایز کننده، بر خلاف روش‌های بر مبنای HMM، از اطلاعات آماری استفاده نمی‌شود. در این روش‌ها یک ابر صفحه متمایز کننده طراحی می‌شود که فضا را به دو قسمت تقسیم می‌کند و کلمات را از هم متمایز می‌کند. مثال در مرجع [28] یک روش متمایز کننده بر مبنای روش‌های تکاملی ارائه شده است. از دیگر روش‌های KWS می‌توان به الگوریتم DTW[4] اشاره کرد. در روش‌های بر مبنای DTW از مقایسه نمونه‌های زمانی استفاده می‌شود[۴۲].

۲-۴-۱ روش DTW

روش DTW یک روش موثر برای یافتن مسیر هم‌تراز غیر خطی بین دو دنباله‌ی زمانی است که از نظر طولی با یکدیگر برابر نیستند. این الگوریتم در محاسبه شباهت بین دنباله‌های زمانی کارآمد بوده و اثر جابجایی و اعوجاج سیگنال را به حداقل می‌رساند. روش DTW جزء اولین روش‌هایی است که به منظور جست و جوی کلمات کلیدی ارائه شده است. در این روش یک کلمه کلیدی مرجع در

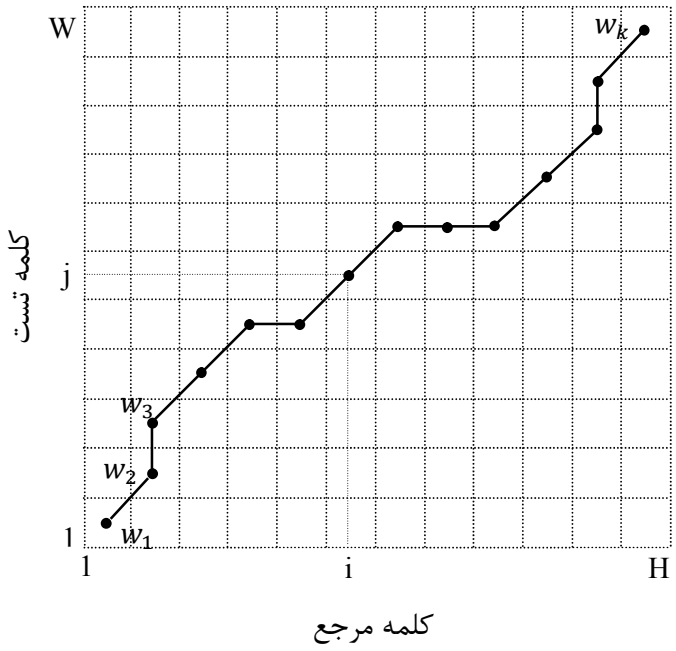
[1] Keyword Spotting
[2] Hidden Markove Model
[3] Discriminative
[4] Dynamic Time Warping

سیستم ذخیره می‌شود و با کلمات ورودی به سیستم مقایسه می‌شود. در شکل (۲-۱۴) یک نمونه از کارکرد سیستم DTW را مشاهده می‌کنید. در این شکل بردارهای ویژگی کلمه کلیدی مرجع و کلمه ورودی در امتداد دو محور افقی و عمودی شبکه قرار داده شده است. هرکدام از بلوک‌های شبکه محتوی فاصله بین بردارهای ویژگی متناظر با آن بلوک می‌باشند. بهترین تطابق بین دو دنباله را می-توان از مسیری در امتداد شبکه که فاصله تجمعی کل را کمینه می‌کند، بدست آورد.

هنگامی که طول کلمه افزایش می‌یابد، تعداد مسیرهای ممکن در طول شبکه نیز افزایش خواهد یافت. با اعمال برخی محدودیت‌ها می‌توان تعداد مسیرهای ممکن را کاهش داده و به این ترتیب محاسبات را ساده‌تر کرد. از جمله‌ی این محدودیت‌ها می‌توان به موارد زیر اشاره نمود[۴۲]:

- **شرط یکنوایی:** مسیر باید به صورت یک تابع یکنوا افزایش پیدا کند و امکان برگشت وجود ندارد. اگر i و j به ترتیب مشخص کننده‌ی اندیس‌های کلمه مرجع و کلمه آزمایش باشند، مقدار آنها یا باید ثابت بماند و یا افزایش یابد.
- **شرط پنجره‌ی تنظیم:** مسیر بهینه نباید از مسیر قطری خیلی فاصله بگیرد. این شرط از این حقیقت ناشی می‌شود که مسیر ایده‌آل در امتداد خط قطری قرار دارد.
- **شرایط مرزی:** مسیر از گوشه پایین سمت چپ آغاز شده و در گوشه سمت راست بالا پایان می‌یابد. به لحاظ منطقی، این شرط برای اطمینان از مقایسه کامل دو کلمه گذاشته شده است.
- **شرط محدودیت شیب:** مسیر نباید دارای شیب خیلی زیاد و یا خیلی کم باشد. شیب خیلی زیاد و یا خیلی کم، باعث مقایسه غیر واقعی یک الگوی کوچک با یک الگوی بزرگ و یا برعکس آن می‌شود.

البته با توجه به نوع کاربرد ممکن است این شرط‌ها تغییراتی داشته باشند.



شکل(۲-۱۴): شبکه DTW همراه با مسیر همترازی[۴۲]

۲-۴-۲ مدل مخفی مارکوف

مدل مخفی مارکوف یک مدل احتمالاتی است که از آن می‌توان برای توصیف پدیده‌های طبیعی کخ ماهیت تصادفی دارند(مانند سیگنال گفتار)، استفاده کرد. هر مدل HMM گسسته با استفاده از پارامترهای زیر تعریف می‌شود[29,30]:

۱. تعداد حالت‌هایی که سیستم می‌تواند در آن قرار داشته باشد و با N نمایش داده می‌شود. به طور کلی حالت‌ها با هم پیوند دارند و بسته به ساختار مدل، ارتباط حالت‌ها با یکدیگر متفاوت می‌باشد. حالت‌های مجزا با $S = \{s_1, s_2, \dots, s_N\}$ نمایش داده می‌شود و حالت در زمان t را با g_t نمایش می‌دهند.

۲. تعداد نمادهای مشاهده شده مجزا در هر حالت که با M نشان داده می‌شود. نمادهای مشاهده شده متناظر با خروجی فیزیکی سیستمی هستند که مدل می‌شود. نمادهای

مجزا را با $V = \{v_1, v_2, \dots, v_m\}$ نمایش می‌دهند.

۳. توزیع احتمال انتقال حالت که با $A = \{a_{ij}\}$ نمایش داده می‌شود، جایی که داریم:

$$a_{ij} = P(q_{t+1} = s_j | q_t = s_i) \quad , \ 1 \le i, j \le N \tag{۲۶-۲}$$

۴. توزیع احتمال نماد مشاهده در حالت jام که عبارت است از:

$$b_j(k) = P(v_k \ at \ t | q_t = s_j) \quad , \ 1 \begin{cases} 1 \le j \le N \\ 1 \le k \le M \end{cases} \tag{۲۷-۲}$$

۵. توزیع احتمال حالت اولیه سیستم به صورت $\pi = \{\pi_i\}$ که عبارت است از:

$$\pi_i = P(q_1 = s_i) \quad , \ 1 \le i \le N \tag{۲۸-۲}$$

با دانستن مقادیر N, M, A, B, π یک مدل HMM به طور کامل توصیف می‌شود. می‌توان مدل HMM را به صورت خلاصه به صورت $\lambda = (A, B, \pi)$ نمایش داد.

برای استفاده از یک مدل HMM در کاربردهای عملی باید به سه سوال اساسی پاسخ داد. این سوال‌ها عبارتند از[30]:

۱. فرض کنید یک دنباله مشاهده به صورت $O = o_1 o_2 \dots o_T$ داده شده است و پارامترهای مدل یعنی $\lambda = (A, B, \pi)$ نیز مشخص شده باشند. چگونه می‌توان به صورت بهینه و بدون نیاز به محاسبات طولانی، $P(O|\lambda)$ یعنی احتمال تولید دنباله مشاهده توسط مدل λ را محاسبه کرد؟

۲. مانند قسمت قبل، دنباله مشاهدات O و پارامترهای مدل λ مشخص شده‌اند. حال چگونه می‌توان دنباله حالات $Q = q_1, q_2, \dots, q_T$ متناظر با دنباله مشاهدات O را که به شکل معنی داری بهینه شده باشد، معین کرد؟

۳. فرض کنید دنباله مشاهدات O مشخص شده است، چگونه می‌توان پارامترهای مدل، یعنی $\lambda = (A, B, \pi)$ را محاسبه کرد به صورتی که $P(O|\lambda)$ ماکزیمم شود؟

در ادامه به بررسی جواب‌های سوالات طرح شده، پرداخته شده است.

۲-۴-۲-۱ جواب سوال اساسی اول

برای محاسبه $P(O|\lambda)$ می‌توان از الگوریتم جلوروندهٔ[1] استفاده کرد. متغیر مربوط به پارامتر جلو رونده به صورت زیر محاسبه می‌شود[30,31]:

$$\alpha_i(i) = P(o_1 o_2 \dots o_t, \quad q_t = s_i|\lambda) \tag{۲۹-۲}$$

که $\alpha_t(i)$ عبارت است از احتمال جزئی دنباله مشاهده شده $o_1, o_2, \dots, o_t$ (یعنی تا زمان t) و حالت s_i در زمان t توسط مدل λ. برای محاسبه α_t به ازای $\mathrm{t} = 1,2, \dots, \mathrm{T}$ داریم:

۱. مقدار دهی اولیه:

$$\alpha_1(i) = \pi_i b_i(o_1) \qquad ,1 \le i \le N \tag{۳۰-۲}$$

۲. برای سایر مقادیر t و i می‌توان به صورت زیر عمل کرد:

$$\alpha_{t+1}(j) = \left[\sum_{i=1}^{N} \alpha_t(i) a_{ij}\right] b_j(o_{t+1}) \qquad ,\begin{cases} 1 \le t \le T-1 \\ 1 \le j \le N \end{cases} \tag{۳۱-۲}$$

۳. در نهایت برای محاسبه $P(O|\lambda)$ می‌توان نوشت:

$$P(o|\lambda) = \sum_{i=1}^{N} \alpha_T(i) \tag{۳۲-۲}$$

تا این مرحله، سوال اول پاسخ داده شده است. برای پاسخ دادن بـه سـوالات شـماره ۲ و ۳ بـه متغیر دیگری به نام متغیر پس‌رونده[2] نیاز است که در ادامه شرح داده خواهد شد.

متغیر پس‌رونده به صورت زیر تعریف می‌شود:

$$\beta_t(i) = P(o_{t+1} o_{t+2} \dots o_T)|q_t = s_i, \lambda \tag{۳۳-۲}$$

[1] Forward
[2] Backward

که عبارت است از احتمال دنباله مشاهدات جزئی از $t+1$ تا T به ازای اینکه سیستم در زمان t در حالت s_i باشد و پارامترهای مدل نیز λ باشند.

برای محاسبه پارامترهای β می‌توان به صورت زیر عمل کرد:

۱. مقدار دهی اولیه:

$$\beta_T(i) = 1 \qquad ,1 \le i \le N \tag{۳۴-۲}$$

۲. محاسبه β به ازای سایر t و iها:

$$\beta_t(i) = \sum_{i=1}^{N} a_{ij}\, b_j(o_{t+1})\beta_{t+1}(j) \qquad ,\begin{cases} t = T-1, T-2, \dots, 1 \\ 1 \le i \le N \end{cases} \tag{۳۵-۲}$$

۲-۲-۴-۲ نرمالیزه کردن α و β

هنگامی که α و β را محاسبه می‌کنند و مقادیر s به دست می‌آیند، این مقادیر از نظر انـدازه بسیار کوچک می‌شوند و برای پیاده سازی سخت افزاری و یا حتی نرم افزاری مشکل ایجاد می‌کننـد. به همین دلیل معمولا مقادیر α و β را نرمالیزه می‌کنند. روش نرمالیزه کردن به صورت زیر می‌باشد:

بر طبق تعریف، متغیر $\alpha_t(i)$ بارت است از:

$$\alpha_t(i) = \left[\sum \alpha_{t-1}(j)a_{ij}\right] b_i(o_t) \tag{۳۶-۲}$$

به ازای $t = 1$ فرض می‌شود:

$$\tilde{\alpha}_{\,1}(i) = \alpha_1(i) \qquad ,i = 1,2,\dots,N \tag{۳۷-۲}$$

سپس ضریب نرمالیزه اول به صورت زیر تعریف می‌شود:

$$c_1 = \frac{1}{\sum_{j=1}^{N} \tilde{\alpha}_1(j)} \quad (۳۸-۲)$$

و در نهایت می‌توان نوشت

$$\hat{\alpha}_1(i) = c_1 \tilde{\alpha}_1(i) \quad (۳۹-۲)$$

در معادله (۲-۳۹)، $\tilde{\alpha}_1(\mathrm{i})$ متغیر $\alpha_1(i)$ نرمالیزه شده می‌باشد. در ادامه برای محاسبه α_t به ازای $t = 2,3,\dots,T$ می‌توان به صورت زیر عمل کرد:

۱. ابتدا به ازای $i = 1,2,\dots,N$ باید معادله (۲-۴۰) محاسبه شود، یعنی:

$$\tilde{\alpha}_t(i) = \left[\sum_{j=1}^{N} \tilde{\alpha}_{t-1}(j) a_{ij}\right] b_i(o_t) \quad (۴۰-۲)$$

۲. در مرحله دوم باید ضریب نرمالیزه کردن در زمان t محاسبه شود، یعنی:

$$c_t = \frac{1}{\sum_{j=1}^{N} \tilde{\alpha}_1(j)} \quad (۴۱-۲)$$

در نهایت در مرحله پایانی، α_t نرمالیزه شده محاسبه می‌شود. یعنی:

$$\tilde{\alpha}_t(i) = c_t \tilde{\alpha}_t(i) \quad (۴۲-۲)$$

وقتی که به این روش، پارامترهای α نرمالیزه می‌شوند، $P(o|\lambda)$ به صورت زیر محاسبه می‌شود:

$$P(o|\lambda) = \frac{1}{\prod_{j=1}^{T} c_j} \quad (۴۳-۲)$$

هنگامی که ضراب c_t به ازای $t = 1,\dots,T$ محاسبه شدند، می‌توان از آن‌ها برای نرمالیزه کردن پارامترهای β نیز استفاده کرد. به این ترتیب که وقتی $\beta_T(i)$ در مرحله اول الگوریتم پس‌رونده محاسبه شد، سپس برای نرمالیزه کردن داریم:

$$\tilde{\beta}_T(i) = c_T \beta_T(i) \quad (۴۴-۲)$$

در ادامه باز هم مانند معادله (۲-۴۴)، هر کدام از پارامترهای β که محاسبه شدند، با استفاده از

معادله (۲-۴۵) نرمالیزه می‌شوند.

$$\tilde{\beta}_t(i) = c_t \beta_t(i) \quad (۴۵-۲)$$

۲-۴-۲-۳ جواب سوال سوم

وقتی که یک دنباله مشاهده $O = o_1 o_2 \dots o_T$ موجود باشد، برای محاسبه‌ی پارامترهای $\lambda = (A, B, \pi)$ معمولا از الگوریتم Baun-Welch استفاده می‌شود. برای این کار باید چند متغیر جدید تعریف شود. اولین متغیر عبارت است از:

$$y_t(i) = P(q_t = s_i | o, \lambda) \quad (۴۶-۲)$$

که در این معادله $y_t(i)$ عبارت است از احتمال اینکه سیستم در زمان t در حالت iام باشد. با فرض اینکه $\alpha_t(i)$ و $\beta_t(i)$ متغیرهای نرمالیزه شده باشد، می‌توان $y_t(i)$ را به صورت معادله (۲-۴۷) محاسبه کرد:

$$y_t(i) = \frac{\alpha_t(i)\beta_t(i)}{\sum_{j=1}^{N} \alpha_t(j)\beta_t(j)} \quad (۴۷-۲)$$

متغیر دیگری که باید تعریف شود عبارت است از:

$$y_t(i,j) = P(q_t = s_i, q_{t+1} = s_j | O, \lambda) \quad (۴۸-۲)$$

متغیر $y_t(i,j)$ عبارت است از احتمال اینکه سیستم در زمان t در حالت s_i باشد و در زمان $t+1$ به حالت s_j انتقال پیدا کند. برای محاسبه این متغیر می‌توان از معادله (۲-۴۹) استفاده کرد:

$$y_t(i,j) = \frac{\alpha_t(i) a_{ij} b_j(o_{t+1}) \beta_{t+1}(j)}{\sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_t(i) a_{ij} b_j(o_{t+1}) \beta_{t+1}(j)} \quad (۴۹-۲)$$

برای محاسبه پارامترهای مدل HMM با استفاده از متغیرهایی که تعریف شده است، ابتدا باید تعریف کلی پارامترهای مدل را در نظر گرفت. برای پارامترهای مدل HMM می‌توان نوشت:

$\bar{\pi}_i$ = تعداد دفعاتی که سیستم در حالت s_i در زمان $t = 1$ بوده است.

$\bar{a}_{ij}$ = (تعداد دفعات انتقال از حالت s_i به حالت s_j)/ (تعداد دفعات انتقال از حالت s_i)

$\bar{b}_j(k)$ = (تعداد دفعاتی که در حالت jام، نماد v_k مشاهده شده است)/ (تعداد دفعاتی که سیستم در حالت jام بوده است).

با توجه به تعریف‌های ارائه شده شده و متغییرهایی تعریف شده، می‌توان پارامترهای مدل HMM را به صورت زیر تعریف کرد:

$$\bar{\pi}_i = y_1(i) \qquad (۵۰-۲)$$

$$\bar{a}_{ij} = \frac{\sum_{i=1}^{T-1} y_t(i,j)}{\sum_{i=1}^{T-1} y_t(1)} \qquad (۵۱-۲)$$

$$\bar{b}_j(k) = \frac{\sum_{\substack{t=1 \\ s.t.\ o_t=v_k}}^{T} y_t(j)}{\sum_{t=1}^{T} y_t(j)} \qquad (۵۲-۲)$$

در عمل برای محاسبه پارامترهای مدل HMM به ازای یک دنباله مشاهدات، ابتدا برای پارامترهای مدل، به صورت تصادفی یک مقدار اولیه در نظر گرفته می‌شود و سپس با استفاده از مشاهدات و پارامترهای اولیه و معادلات (۲-۵۰) تا (۲-۵۲) پارامترهای مدل محاسبه می‌شوند. در ادامه با استفاده از پارامترهای محاسبه شده جدید و دنباله مشاهدات، دوباره با استفاده از معادلات (۲-۵۰) تا (۲-۵۲) پارامترهای مدل محاسبه می‌شوند و این کار تا زمانی که پارامترها با دقت مناسب تخمین زده شوند، ادامه پیدا می‌کند.

۲-۴-۲-۴ مدل کردن یک کلمه با استفاده از HMM

برای مدل کردن سیگنال گفتار، با استفاده از HMM معمولا از حالت پیوسته HMM استفاده می‌شود. مواردی که تا این قسمت ذکر شد، مربوط به حالت گسسته HMM بود. منظور از گسسته بودن HMM این است که احتمال مشاهده نماد v_k در حالت jام، یعنی $b_j(k)$، فقط تعداد محدودی مقدار می‌تواند داشته باشد در صورتی که در حالت پیوسته، $b_j(k)$ می‌تواند هر مقداری بین صفر و

یک داشته باشد. در این حالت، تابع احتمال مشاهده نماد را به صورت یک مدل آمیخته گاوسی تعریف می‌کنند[30, 32].

$$b_j(o_t) = \sum_{l=1}^{M} c_{jl} N\left(o_t \middle| \mu_{jl}, \sum_{jl}\right) = \sum_{l=1}^{M} c_{jl} b_{jl}(o_t) \quad (۵۳-۲)$$

در این معادله M تعداد توزیع‌های نرمال چند متغیره تشکیل دهنده *HMM*، c_{jl}ها ضرایب ترکیب و N توزیع توزیع نرمال چند متغیره می‌باشد. μ_{jl} عبارت است از برادار میانگین به ازای توزیع نرمال شماره l و $\sum_{jl}$ها ماتریس کواریانس ترکیب lام می‌باشد.

برای محاسبه پارامترهای مدل در این حالت، باید یک متغیر جدید تعریف شود که عبارت است از:

$$y_{il}(t) = y_i(t) \frac{c_{il} b_{il}(o_t)}{b_i(o_t)} \quad (۵۴-۲)$$

در عمل برای آموزش دادن یک کلمه معمولا تعداد زیادی از نمونه‌های مختلف آن کلمه که توسط گویندگان مختلف ادا شده شده است،استفاده می‌شود[33,34]. اگر تعداد E نمونه آموزش در اختیار داشته باشد، آن‌گاه پارامترهای مدل به صورت زیر محاسبه می‌شوند. در این معادلات، T_e عبارت است از طول دنباله مشاهده‌ی eام.

$$\bar{\pi}_i = \frac{\sum_{e=1}^{E} y_i^e(1)}{E} \quad (۵۵-۲)$$

$$\bar{c}_{il} = \frac{\sum_{e=1}^{E}\sum_{t=1}^{Te} y_{il}^e(t)}{\sum_{e=1}^{E}\sum_{t=1}^{Te} y_{il}^e(t)} \quad (۵۶-۲)$$

$$\bar{\mu}_{il} = \frac{\sum_{e=1}^{E}\sum_{t=1}^{Te} y_{il}^e(t) o_t^e}{\sum_{e=1}^{E}\sum_{t=1}^{Te} y_{il}^e(t)} \quad (۵۷-۲)$$

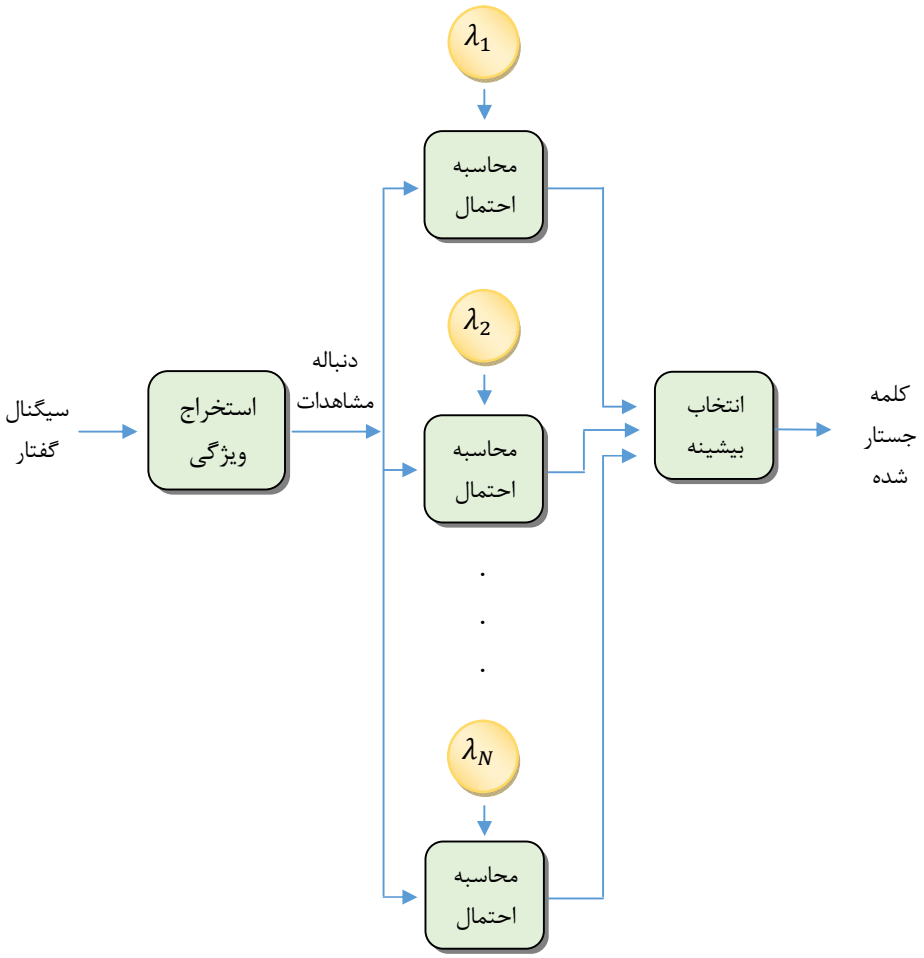
$$\overline{\sum_{il}} = \frac{\sum_{e=1}^{E}\sum_{t=1}^{Te} y_{il}^e(t)(o_t^e - \mu_{il})(o_t^e - \mu_{il})'}{\sum_{e=1}^{E}\sum_{t=1}^{Te} y_t^e(t)} \quad (۵۸-۲)$$

$$\bar{a}_{ij} = \frac{\sum_{e=1}^{E}\sum_{t=1}^{Te} y_t^e(i,j)}{\sum_{e=1}^{E}\sum_{t=1}^{Te} y_t^e(i)} \quad (۵۹-۲)$$

۲-۵ سیستم جست و جوی کلمات کلیدی

۲-۵-۱ سیستم جست و جوی کلمات کلیدی در گفتار گسسته

دیاگرام بلوکی سیستم جستجوی کلمات کلیدی در گفتار در شکل (۲-۱۵) نمایش داده شـده است[33,34].

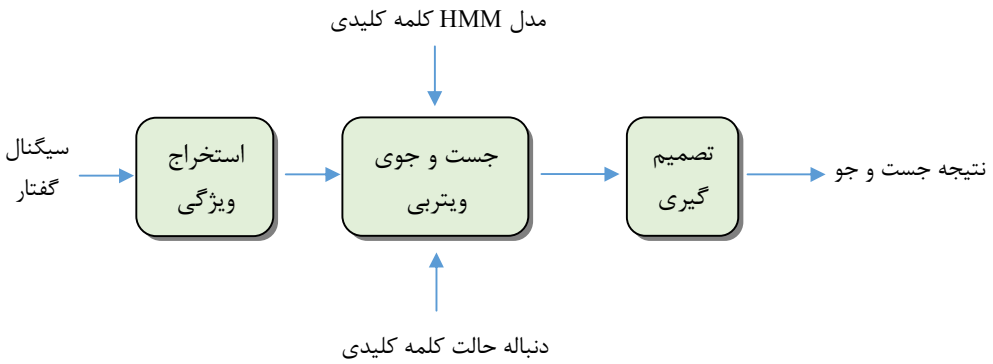


شکل(۲-۱۵): دیاگرام بلوکی سیستم جست و جوی کلمات کلیدی در گفتار

همانگونه که در این شکل دیده می‌شود، وقتی سیگنال آزمایش وارد سیستم می‌شود، ابتدا ویژگی‌های سیگنال ورودی استخراج می‌شود و سپس با استفاده از معادله (۲-۴۳) به ازای مدل هر کدام از کلماتی که از قبل آموزش داده شده‌اند، $P(o|\lambda)$ یعنی میزان شباهت به ازای دنباله مشاهدات ورودی به سیستم و مدل کلمه کلیدی λ محاسبه می‌شود. در نهایت هر کدام از این شباهت‌های محاسبه شده که بیشترین مقدار را داشته باشد، دنباله مشاهدات ورودی معادل به کلمه‌ی مدل برنده می‌باشد[35].

۲-۵-۲ سیستم جست و جوی کلمات کلیدی در گفتار پیوسته

در شکل (۲-۱۶) دیاگرام بلوکی یک سیستم جست و جوی کلمات کلیدی در گفتار پیوسته نمایش داده شده است[36].



شکل(۲-۱۶): دیاگرام بلوکی سیستم جست و جوی کلمه کلیدی در گفتار پیوسته

شیوه عملکرد این سیستم به این ترتیب است که وقتی از سیگنال ورودی به سیستم ویژگی-های لازم استخراج شد، با استفاده از الگوریتم ویتربی، محتمل‌ترین دنباله‌ی حالت‌های مطابق با دنباله مشاهدات ورودی به سیستم و بر اساس مدل کلمه کلیدی مورد نظر استخراج می‌شود. این دنباله حالت‌ها با دنباله‌ی حالت مربوط به کلمه کلیدی مرجع مقایسه شده و در مرحله‌ی تصمیم گیری مکان کلمه کلیدی مورد نظر را در گفتار ورودی به سیستم، مشخص می‌شود.

اگر که دنباله مشاهدات $O = \{o_1, o_2, \dots, o_T\}$ فرض شود و λ مدل HMM مربوط به کلمه کلیدی باشد و $Q = \{q_1, q_2, \dots, q_T\}$ محتمل‌ترین دنباله‌ی حالت برای دنباله مشاهدات باشد، آنگاه برای به دست آوردن محتمل‌ترین دنباله‌ی حالت برای دنباله مشاهدات، ابتدا باید متغیر $\delta_t(i)$ تعریف شود[37, 30]:

$$\delta_t(i) = \max_{q_1, q_2, \dots, q_{T-1}} P[q_1 q_2 \dots q_t = i, o_1 o_2 \dots o_t | \lambda] \qquad (۶۰-۲)$$

$\delta_t(i)$ عبارت است از بیشترین احتمال در یک مسیر در زمان t که برای t مشاهده اول محاسبه شده و در حالت s_i پایان می‌یابد. برای نگهداری محتمل‌ترین حالت در هر زمان نیز متغیر $\psi_t(j)$ تعریف می‌شود. مراحل بدست آوردن محتمل‌ترین دنباله حالت عبارت است از:

الف- مقدار دهی اولیه:

$$\begin{aligned} &\delta_1(i) = \pi_i b_i(o_1) \quad , \quad 1 \le i \le N \\ &\psi_1(i) = 0 \end{aligned} \qquad (۶۱-۲)$$

ب- محاسبه سایر مقادیر δ و ψ:

$$\delta_t(j) = \max_{1 \le i \le N} [\delta_{t-1}(i) a_{ij}] b_j(o_t) \quad , \quad \begin{cases} 2 \le t \le T \\ 1 \le j \le N \end{cases} \qquad (۶۲-۲)$$

$$\psi_t(j) = arg \max_{1 \le i \le N} [\delta_{t-1}(i) a_{ij}] \quad , \quad \begin{cases} 2 \le t \le T \\ 1 \le j \le N \end{cases} \qquad (۶۳-۲)$$

ج- مرحله پایانی:

$$P^* = \max_{1 \le i \le N} [\delta_T(i)] \qquad (۶۴-۲)$$

$$q_T^* = arg \max_{1 \le i \le N} [\delta_T(i)] \qquad (۶۵-۲)$$

د- دنباله کلی حالت‌ها:

$$\psi_t^* = \psi_{t+1}(q_{t+1}^*) \quad , \quad t = T-1, T-2, \dots, 1 \qquad (۶۶-۲)$$

فصل سوم

معرفی سیستم سخت‌افزاری

برای ساخت سخت افزار جست و جوی کلیدواژه، از پردازنده سیگنال TMS320C5509A استفاده شده که ساخت شرکت Texas Instrument می‌باشد. قبل از اینکه مدار سخت افزاری را معرفی کنیم، ابتدا خود پردازنده مورد استفاده و اطلاعات مربوط به آن را ارائه می‌دهیم.

۳-۱پردازشگرهای سیگنال

در دهه ۷۰ میلادی همزمان با ساخت اولین پردازنده‌ها توسط شرکت‌های مختلف، شرکت TI نیز پردازنده‌های مخصوص پردازش سیگنال را وارد بازار کرد. این پردازنده‌ها که بیشتر با نام DSP[1] معروف هستند، همگی با نام TMS320 شروع می‌شوند. پردازنده‌های DSP در مدت زمان حدود ۴۰ سال که از حضورشان در بازار قطعات الکترونیک می‌گذرد بسیار تکامل یافته‌اند. اولین سری این پردازنده‌ها با نام TMS320C10 وارد بازار شد. پس از چند سال، حضور سری TMS320C25 باعث معروف شدن DSPها گردید. این پردازنده می‌توانست تبدیل فوریه را با سرعتی محاسبه کند که اولین سری‌های پردازنده پنتیوم ساخت شرکت اینتل ۲۰ سال بعد توانستند به آن سرعت برسند[۴۰].

۳-۱-۱ پردازنده‌های مهم شرکت TI

شرکت TI دارای چندین دسته مختلف از پردازنده‌ها می‌باشد که برای کاربردهای گوناگون استفاده می‌شوند و هر کدام از دسته‌های این پردازنده‌ها ویژگی‌های منحصر به فرد و مخصوص به خود را دارند. در ادامه به معرفی سری‌های مختلف پردازنده‌های شرکت TI می‌پردازیم.

الف– سری 5000 (یا 5XXXX): این سری دارای دو خانواده اصلی 55XX و 54XX می‌باشد. این پردازنده‌ها کم مصرف‌ترین پردازنده‌های شرکت TI می‌باشند که در بسیاری از تجهیزاتی که نیاز به پردازش با سرعت بالا و مصرف توان کم دارند استفاده می‌شوند. در سری 5000 سرعت پردازنده‌ها بین ۱۰۰ تا ۳۰۰ مگا هرتز می‌باشد و به صورت خاص در سری 55XX قدرت محاسبات

[1] Digital Signal Processor

ریاضی می‌تواند دو برابر فرکانس کاری پردازنده باشد. یعنی سری 55XX حداکثر توان انجام ۶۰۰ میلیون محاسبه ریاضی را در یک ثانیه دارد[۴۰].

کاربرد اصلی پردازشگرهای سری 5000 در پردازش صوت و الگوریتم‌هایی است که نیاز به سرعت بالا دارند. از بعضی سری‌ها که حجم حافظه بیشتر از ۱۲۸ کیلو بایت دارند می‌توان برای کاربردهای پردازش تصویر نیز استفاده کرد.

ب– سری 2000 (2XXX): این سری شامل دو خانواده اصلی 24XX و 28XX می‌باشد. سری 28XX یک خانواده با عملکرد نزدیک به میکروکنترلرها می‌باشد. ویژگی مهمی که این دسته را از سایر دسته‌ها مجزا می‌سازی، وجود حافظه فلش در این سری می‌باشد. وجود حافظه فلش داخلی سبب شده است که این سری از پردازنده‌ها را بتوان راحت‌تر از سری‌های دیگر DSP ها برنامه‌ریزی کرد، . در این خانواده حجم حافظه داخلی از نوع SRAM کمتر از ۳۲ کیلوبایت بوده و کاربرد این سری از پردازنده‌ها بیشتر به عنوان یک میکروکنترلر پرسرعت می‌باشد. این میکروکنترلرها به خوبی جای خود را در صنعت باز کرده‌اند.[۴۰].

ج – سری 6000 (6XXX): این سری شامل سه خانواده اصلی 62XX، 64XX و 67XX است. در این خانواده‌ها فرکانس کاری پردازنده بین ۱۵۰ مگاهرتز تا ۱/۲ گیگاهرتز می‌باشد اما سرعت کاری واقعی آنها ۸ برابر فرکانس کاری آنها است. در این پردازنده‌ها در هر کلاک تا حداکثر ۸ دستور به شکل همزمان قابل اجرا می‌باشد و به همین دلیل می‌توانند تا حدود ۱۰ گیگا دستورالعمل در ثانیه[1](GIPS) را اجرا نمایند. این خانواده برای تمامی انواع پردازش‌های پرسرعت مناسب می‌باشند، اما سری 64xx با قابلیت‌های خاص آن مناسب‌ترین سری برای پردازش‌های تصویر می‌باشد. [۴۰].

[1]Giga Instruction Per Second

۳-۱-۲ پردازنده TMS320C5509A

از بین خانواده‌های معرفی شده، پردازنده 5509A برای طراحی سخت افزاری انتخاب شده است که از سری خانواده 5000 بوده و حالت بهینه شده پردازنده 5509 می‌باشد. می‌توان از خصوصیات این پردازنده به موارد زیر اشاره کرد[44] :

- قابلیت سه بار خواندن و دوبار نوشتن در هر ثانیه
- سیستم محاسباتی ممیز ثابت
- دو واحد MAC با قابلیت ضرب دو عدد ۱۷ بیتی
- فرکانس کاری قابل تنظیم تا حدود ۲۰۰ مگاهرتز
- واحد EMIF جهت دسترسی به حافظه
- معماری پیشرفته چندگذرگاهی شامل یک گذرگاه برنامه، سه گذرگاه داده و چهار گذرگاه آدرس
- قابلیت اجرای موازی چند دستور در یک سیکل
- توان مصرفی پایین
- اجرای دستورالعمل‌های پیچیده پردازش سیگنال مانند کانولوشن، تبدیل فوریه و ....
- قابلیت محاسبه ۴۰۰ میلیون محاسبه ریاضی در ثانیه در فرکانس ۲۰۰ مگاهرتز
- پشتیبانی از پروتکل‌های ارتباطی McBSP و I2C برای ارتباط با انواع مبدل‌ها

از این پردازنده می‌توان در کاربردهای مختلف پردازش سیگنال مانند کاربردهای زیر استفاده کرد.

- انواع کاربردهای پردازش گفتار شامل جست و جوی کلیدواژه و ....
- انواع الگوریتم‌های رفع نویز
- مدولاسیون و دمدولاسیون

- فشرده سازی صوت

برای برنامه نویسی و شبیه سازی و انتقال برنامه به DSP از نرم افزار Code Composer Studio استفاده می‌شود و برای اتصال کامپیوتر به پردازنده و همچنین به منظور انتقال برنامه از JTAG[1] استفاده می‌شود. مدل استفاده شده در این پایان نامه Spectrum Digital XDS510 USB می‌باشد که در شکل (۳-۱) نمایش داده شده است.

شکل(۳-۱): JTAG جهت ارتباط کامپیوتر با پردازنده DSP

۳-۲ مدار پردازشگر سیگنال

مدار طراحی شده دارای قسمت‌های مختلفی می‌باشد که قسمت‌های اصلی آن عبارتند از:

- منبع تغذیه
- مبدل داده‌ها (کُدِک)
- پردازنده سیگنال

در ادامه هر کدام از این قسمت‌ها را بررسی کرده و در مورد ویژگی‌ها و کارکرد آن‌ها صحبت

[1] Joint Test Action Group

خواهیم کرد[۴۱].

۳-۲-۱ منبع تغذیه

برای تامین انرژی مورد نیاز بخش‌های مختلف مدار از یک آی سی توان با نـام TPS767D301 استفاده شده است. یکی از مهمترین ویژگی‌هایی که این آی سی دارد این اسـت کـه توسـط شـرکت سازنده پردازنده سیگنال، یعنی Texas Instrument طراحی و ساخته شده است و دقیقا با مشخصـات پردازنده سیگنال همخوانی داشته و نیازهای آن را برآورده می‌کند[45]. ویژگی‌هایی که می‌توان بـرای این آی سی نام برد عبارتند از:

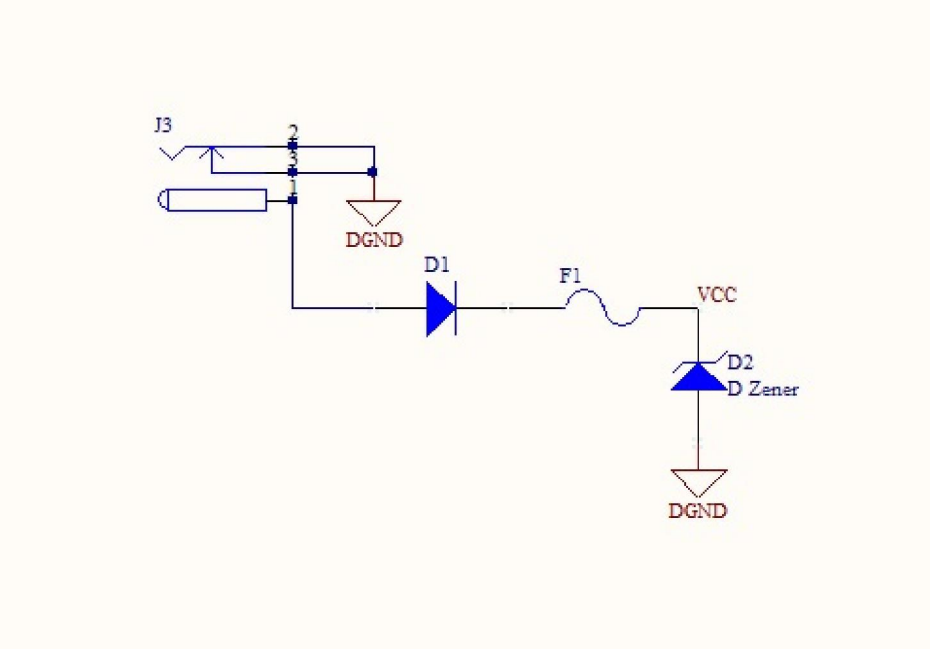
- دارای دو ولتاژ خروجی است که برای استفاده‌های متفاوت در مدار مزیت مهمی مـی-باشد.

پردازنده DSP برای کار به دو سطح ولتاژ نیاز دارد که از ولتاژ ۳/۳ ولت برای استفاده در ادوات جانبی و پورت‌های ورودی و خروجی استفاده می‌شود و از ولتاژ قابل تنظیم که روی ۱/۶تنظـیم شـده است، برای کار پردازنده استفاده می‌شود.

- هر کدام از خروجی‌ها می‌توانند حداکثر ۱ آمپر جریان خروجـی را تحمـل کننـد و بـه ازای هر یک آمپر خروجی حداکثر ۳۵۰ میلی ولت افت جریان دارند.
- زمان پایدار شدن رگولاتور ۲۰۰ میلی ثانیه می‌باشد.
- دارای پاسخ گذاری سریعی است.
- دارای محافظ حرارتی می‌باشد که اگر دما بیشتر از اندازه مجاز شود، به صورت خودکار ولتاژ خروجی را قطع می‌کند.

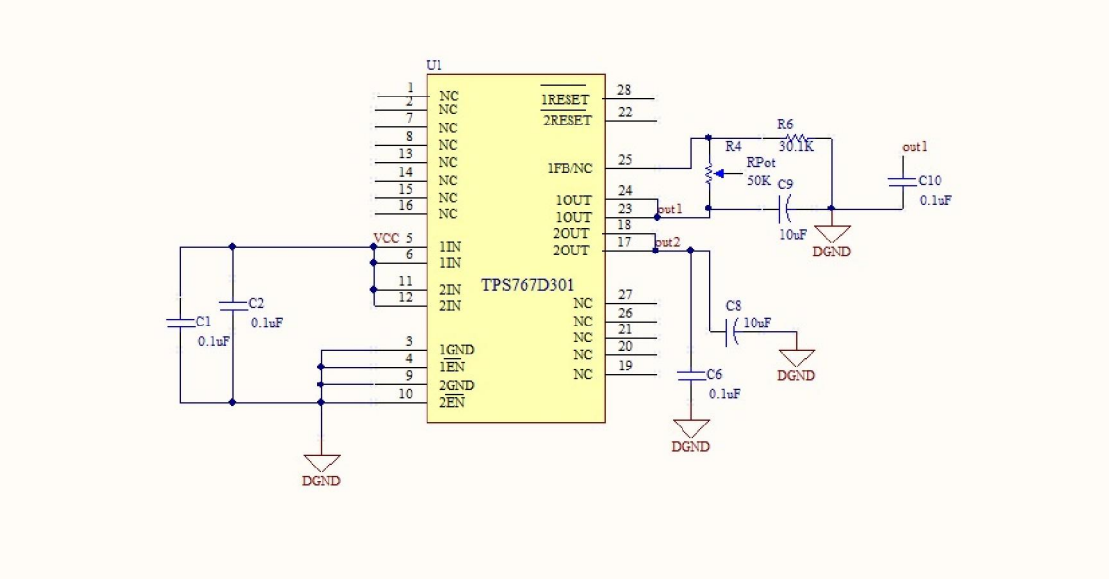
همانگونه که در شکل (۳-۲) نمایش داده شده است، در قسمت ورودی منبع تغذیه از یک مدار محافظ استفاده شده است که از یک فیوز و یک دیود زنر تشکیل شده است. این دو در کنار همـدیگر

وظیفه محافظت از مدار را به عهده دارند. مدار محافظ به گونه‌ای عمل می‌کند که فیوز از عبور جریان بیشتر از ۵۰۰ میلی آمپر جلوگیری می‌کند و دیود زنر نیز از افزایش ولتاژ بیشتر از ۵/۶ ولت جلوگیری می‌کند.



شکل(۳-۲): مدار محافظ منبع تغذیه

خروجی مدار شکل (۳-۲) وارد آی سی تغذیه می‌شود. در ادامه آی سی تغذیه دو ولتاژ خروجی ۱/۶ ولت و ۳/۳ ولت را ایجاد می‌کند که از طریق کلیدها این دو ولتاژ به سرتاسر مدار انتقال پیدا می-کنند. آی سی تغذیه و ادوات جانبی آن را می‌توان در شکل (۳-۳) مشاهده کرد.



شکل(۳-۳): آی سی تغذیه و ادوات جانبی آی سی

۳-۲-۲ مبدل داده‌ها

برای تبدیل داده‌های آنالوگ به دیجیتال از آی سی کُدِک[1] TLV320AIC23B استفاده شده است. برای استفاده از آی سی باید قبل از شروع به کار مدار، با استفاده از نرم افزار Code Composer Studio ثبات‌های درون کدک را به طور مناسبی برنامه‌ریزی کرد. کلاک[2] مورد نیاز برای آی سی نیز توسط یک کریستال نوسان‌ساز خارجی ۱۲ مگاهرتز تامین می‌شود. از ویژگی‌های این آی سی می‌توان به موارد زیر اشاره کرد[46]:

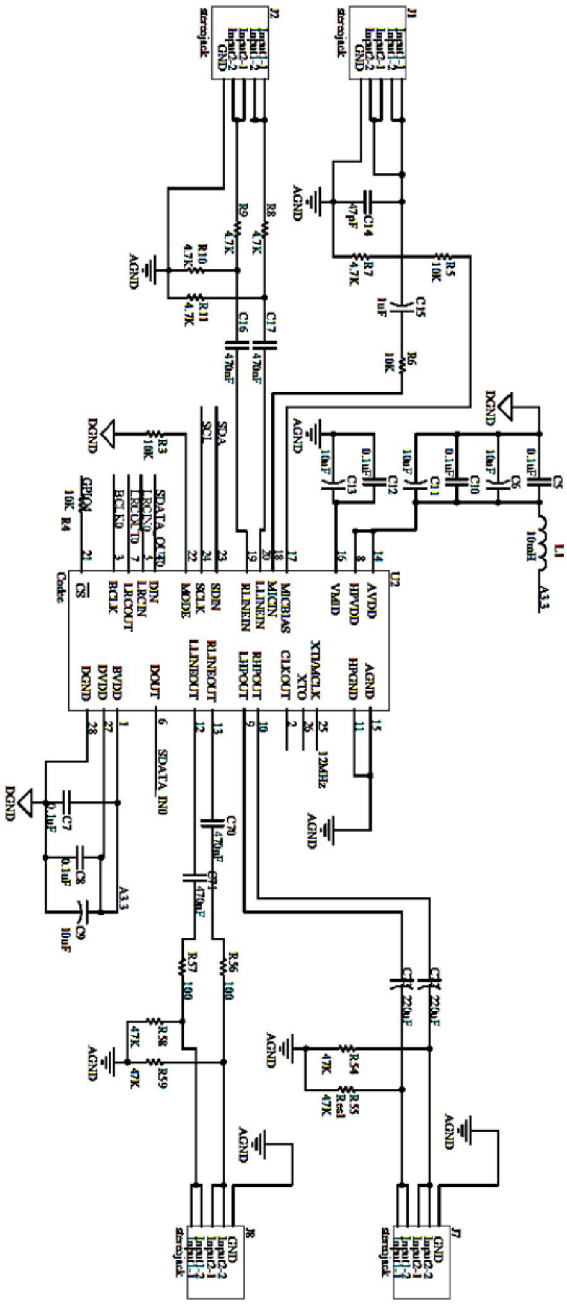
- توسط شرکت Texas Instrument ساخته شده است و با پردازنده‌های DSP ساخت این شرکت هماهنگی کامل دارد.
- کدک توانایی راه اندازی یک میکروفن به صورت مستقیم را داشته و ولتاژ بایاس میکروفن را نیز تامین می‌کند.

[1] Codec
[2] Clock

- شامل یک کانال خط ورودی[1] دوتایی، یک ورودی میکروفن به صورت جداگانه، یک کانال خروجی[2] دوتایی و یک خروجی هدفون می‌باشد.
- فرکانس نمونه‌برداری این آی سی قابل تنظیم بوده و بازه بین ۸ الی ۹۶ کیلوهرتز را پوشش می‌دهد.
- امکان ارسال و دریافت داده‌ها را به صورت همزمان و در اندازه‌های ۸، ۱۶، ۲۴ و ۳۲ بیت توسط پورت McBSP دارد.
- ولتاژ کاری کدک مشابه DSP و برابر با ۱/۶ است.
- تنظیم رجیسترهای کدک توسط روشهای مختلفی مانند SPI، I2C و McBSP قابل انجام است.
- شامل تقویت کننده داخلی جداگانه برای ورودی میکروفن و ورودی‌های دوتایی جهت تقویت سیگنال‌های ورودی می‌باشد.
- به منظور آزمایش عملکرد کدک، در داخل آن یک مسیر کنارگذر تعبیه شده است و می‌تواند هر سیگنالی که وارد آن می‌شود را به صورت مستقیم و بدون تغییر به خروجی بدهد.

مدار کدک و ادوات جانبی آن را در شکل (۳-۴) مشاهده می‌کنید.

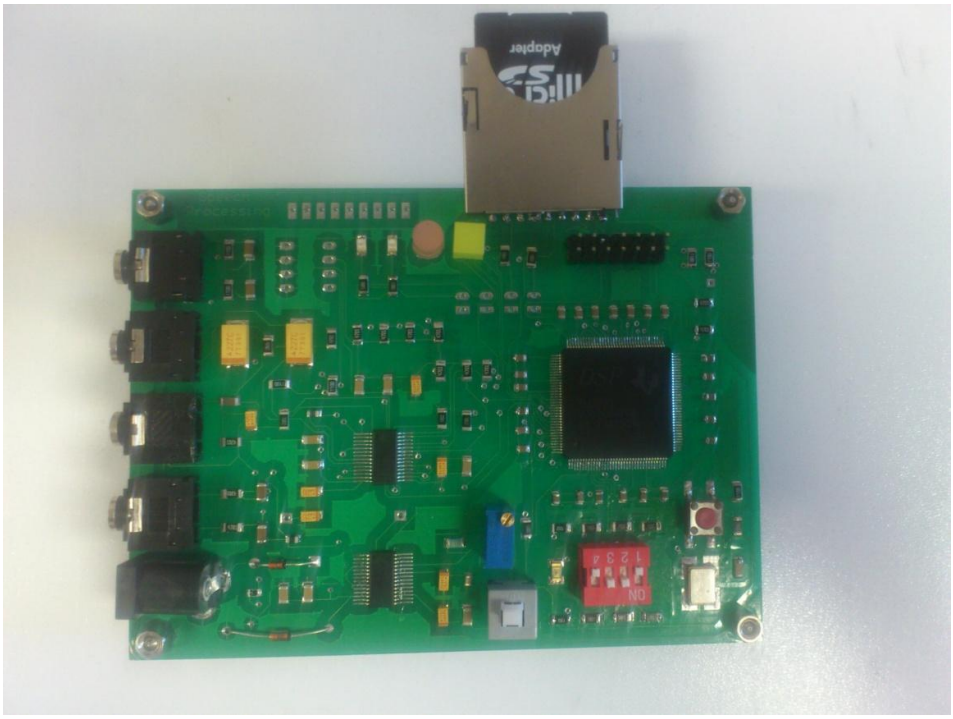
[1] Line In
[2] Line Out



شکل(۳-۴): مدار کدک و ادوات جانبی همراه با آی سی

۳-۲-۳ نمای کلی مدار

در نهایت اصلی‌ترین قسمت مدار، پردازنده سیگنال می‌باشد که در بخش‌های قبل در مـورد آن صحبت کرده‌ایم. نمای کلی مدار طراحی شده به صورت شکل (۳-۵) می‌باشد.



شکل(۳-۵): مدار نهایی جهت جست و جوی کلیدواژه

از دیگر ادوات نصب شده بر روی مدار می‌توان به موارد زیر اشاره کرد:

- تعداد ۳ عدد LED که بر روی پایه‌های GPIO 4, 6, 7 نصب شده‌اند و به منظور نمایش خروجی‌های مدار از آن‌ها استفاده می‌شود.
- یک عدد dip switch چهار کاناله که بر روی پایه‌های GPIO 1, 2, 3, 4 نصب شده و از آن برای تنظیم نوع Boot load استفاده می‌شود. در مدار طراحی شده، برای Boot load کردن پردازنده می‌توان از دو روش استفاده کرد. روش اول از طریق کامپیوتر(با استفاده از JTAG) می‌باشد و روش دوم از طریق حافظه EEPROM است.

فصل چهارم:

بررسی نتایج آزمایش‌ها

۷-۱مقدمه

در این بخش ابتدا در مورد پایگاه داده‌های استفاده شده صحبت شده است و در ادامه ابتدا نتایج آزمایشات در محیط نرم‌افزار Matlab ارائه و بررسی شده و در نهایت نتایج پیاده‌سازی سخت-افزاری ارائه و بررسی شده است. در بخش مربوط به آزمایشات در محیط نرم‌افزار Matlab، برای هر دو سیستم جستجوی کلمات کلیدی در گفتار گسسته و پیوسته آزمایشاتی انجام شده است.

۷-۲ پایگاه داده‌های استفاده شده در آزمایشات

۷-۲-۱ پایگاه داده TIMIT

پایگاه داده TIMIT یک پایگاه داده استاندارد می‌باشد که برای کاربردهای پردازش گفتار مانند شناسایی گوینده، جستجوی کلمه کلیدی و .... گردآوری شده است. این مجموعه شامل ۶۳۰ گوینده می‌باشد که ۴۳۸ نفر مرد و ۱۹۲ نفر از این مجموعه زن هستند. هر کدام از گویندگان ۱۰ جمله را ادا کرده‌اند که هر جمله به صورت جداگانه در یک فایل صوتی ذخیره شده است. در ضمن فایل‌های این پایگاه داده بدون نویز می‌باشند[47].

برای استفاده از این پایگاه داده به این ترتیب عمل شده است که از ۶۰۰ گوینده به عنوان مجموعه آموزش و از ۳۰ گوینده به عنوان مجموعه آزمایش استفاده شده است. برای آموزش مدل‌های HMM تعداد ۱۰ کلمه از هر کدام از گویندگان استخراج شده است. بنابراین به ازای هر کلمه، ۶۰۰ نمونه آموزش و ۳۰ نمونه آزمایش در اختیار است.

۷-۲-۲ پایگاه داده Spoken Arabic Digit Data Set

این پایگاه داده شامل اعداد ۰ تا ۹ عربی می‌باشد که دارای ۴۴ گوینده مرد و ۴۴ گوینده زن بین ۱۸ تا ۴۰ سال می‌باشد. در قسمت آموزش به ازای هر کدام از اعداد، ۶۶۰ نمونه آموزش تهیه

شده است و در قسمت آزمایش به ازای هر یک از اعـداد، ۲۲۰ نمونـه آزمـایش تهیـه شـده اسـت. در قمست آموزش، ۳۳۰ نمونه اول به ازای هر عدد مربوط به گویندگان مرد و ۳۳۰ نمونه دوم بـرای هـر عدد به گویندگان زن اختصاص دارد. در قسـمت آزمـایش نیـز بـه ازای هـر عـدد ۱۱۰ نمونـه بـرای گویندگان مرد و ۱۱۰ نمونه برای گویندگان زن موجود می‌باشد. در این پایگاه داده، ویژگی‌های داده-های آموزش و آزمایش از قبل و به صورت ویژگی‌های MFCC استخراج شده است. بردارهای ویژگـی دارای ۱۳ درایه می‌باشد که ضریب انرژی نیز در آن‌ها لحـاظ شـده اسـت. در ضـمن بـرای اسـتخراج ویژگی از تابع پنجره همینگ استفاده شده است. همچنین گویندگان داده‌های آمـوزش بـا گوینـدگان داده‌های آزمایش متفاوت هستند و البته داده‌های این پایگاه داده بدون نویز هستند[48].

۷-۳ بررسی نتایج آزمایشات در محیط Matlab

در بررسی‌هایی که در فصل دوم داشتیم، دو روش را برای استخراج ویژگی انتخاب کردیم. این دو روش عبارتند از روش MFCC و روش LPCC که هر دو می‌توانند بدون استفاده از بهبود ویژگی‌ها استفاده بشوند و یا اینکه ویژگی‌ها را جهت افزایش مقاومت آن‌ها در مقابل نویز و اثرات کانال انتقال بهبود داد. برای به دست آوردن بهترین نتیجه، در آزمایشات از هر دو روش MFCC و LPCC استفاده کرده‌ایم و برای بهبود کیفیت ویژگی‌ها از ضرایب دلتا و ضرایب شتاب استفاده کرده‌ایم. بسته به آزمایشات متفاوت از بردارهای ویژگی بدون ضرایب دلتا و شتاب یا همراه با ضرایب دلتا و شتاب استفاده شده است. برای مدل کردن کلمات نیز از مدل مخفی مارکوف استفاده شده است.

۷-۳-۱ نتایج آزمایش مربوط به اعداد گسسته

در این بخش از پایگاه داده Spoken Arabic Digits برای انجام آزمایشات استفاده شده است. در این پایگاه داده بردارهای ویژگی از قبل توسط سازنده پایگاه داده استخراج شده و به صورت ضرایب MFCC می‌باشند. هر کدام از بردارهای ویژگی ۱۳ درایه دارند که درایه شماره یک برابر با ضریب انرژی می‌باشد. در این پایگاه داده ۱۰ عدد موجود می‌باشد که عبارتند از اعداد صفر تا ۹ و تعداد داده-های آموزش به ازای هر کدام از اعداد برابر با ۶۶۰ نمونه و تعداد داده‌های آزمایش به ازای هر کدام از اعداد برابر با ۲۲۰ نمونه می‌باشد.

برای آموزش دادن مدل‌های HMM در محیط متلب از HMM all toolbox استفاده شده است. روش آموزش به این ترتیب است که باید برای هر کدام از کلمات به صورت جداگانه با استفاده از الگوریتم Baum-welch یک مدل HMM آموزش داده شود. برای محاسبه پارامترهای مدل مخفی مارکوف با استفاده از روش Baum-welch ابتدا باید ماتریس‌های حالت اولیه، انتقال حالت و مشاهده نماد را به صورت تصادفی مقداردهی اولیه کرد. شرطی که باید برای مقدار دهی اولیه رعایت شود این

است که باید ماتریس‌های حالت اولیه و انتقال حالت به صورت ردیفی احتمالاتی باشند، یعنی حاصلجمع هر ردیف از ماتریس برابر با یک باشد. در ضمن توزیع احتمال مشاهده نماد دارای توزیع احتمالی آمیخته گاوسی می‌باشد. برای این توزیع احتمال آمیخته گاوسی، ماتریس کواریانس به منظور کاهش بار محاسباتی، قطری در نظر گرفته شده است. بعد از اینکه مقداردهی اولیه انجام شد، در مرحله بعد با استفاده از این مقادیر اولیه، پارامترهای آلفا و بتا محاسبه می‌شوند و با استفاده از آن‌ها پارامترهای مدل احتمالاتی مخفی مارکوف تخمین زده می‌شود. در ادامه این مقادیر که با استفاده از الگوریتم Baum-welch محاسبه شده‌اند، به عنوان مقادیر اولیه فرض شده و در الگوریتم قرار می‌گیرند و با استفاده از این مقادیر اولیه جدید، مراحل قبل تکرار شده و پارامترهای مدل مخفی مارکوف تخمین زده می‌شوند. این کار تا زمانی که پارامترها با دقتی که تعیین شده است تخمین زده شوند ادامه پیدا می‌کند.

در این آزمایشات تعداد حالت‌های هر مدل و تعداد ترکیب‌های گاوسی تشکیل دهنده هر کدام از حالت‌ها متغییر فرض شده‌اند و با تعداد حالت و ترکیب‌های مختلف آزمایش انجام شده است. در ضمن مدل‌ها به صورت چپ به راست در نظر گرفته شده‌اند. نتایج در جدول (۴-۱) نمایش داده شده است.

جدول(۷-۱): نتایج آزمایش برای شناسایی اعداد در گفتار گسسته بر حسب درصد برای روش MFCC

شماره آزمایش	تعداد حالت‌های مدل	تعداد ترکیب‌های هر حالت		
		۲	۳	۵
آزمایش اول	۲	۸۱/۱۶	۸۵	۸۳/۳۳
آزمایش دوم	۳	۸۷/۶۶	۹۳/۳۳	۹۰/۵۰
آزمایش سوم	۴	۸۶/۶۶	۹۰/۱۲	۸۸/۳۳
آزمایش چهارم	۵	۸۵/۴۴	۸۷/۱۶	۸۶/۶۶

نتایجی که در جدول ثبت شده است، میانگین درصد شناسایی تمام اعداد صفر تا ۹ می‌باشد. نتایج حاصل از استفاده از الگوریتم Baum-welch کاملا به مقداردهی اولیه بستگی دارد و بر اساس مقداردهی اولیه می‌تواند به نتایج خوبی برسد و یا اینکه به نتایج ضعیفی برسد. در HMM all toolbox تمهیداتی برای رفع این مشکل اندیشیده شده است. برای مثال، مقدار دهی اولیه ماتریس-های کواریانس بر اساس واریانس داده‌های آموزشی انجام می‌شود و یا اینکه مقداردهی اولیه میانگین-های هر کدام از ترکیب‌های مدل آمیخته گاوسی را بر اساس میانگین داده‌های آموزش تعیین می‌کند. با این حال باز هم به دلیل دقیق نبودن مقداردهی اولیه، به ازای هر بار که الگوریتم اجرا می‌شود، درصد شناسایی کلمات با نتیجه قبلی مقداری متفاوت می‌باشد. به همین دلیل به ازای هر کدام از حالت‌ها و ترکیب‌ها چند بار الگوریتم اجرا شده و میانگین نتیجه‌های حاصل شده در جدول وارد شده‌اند.

روش کار به این صورت است که برای هر کدام از اعداد مدل‌های HMM جداگانه آموزش داده شده است. این مدل‌ها را در پایگاه داده سیستم قرار داده و سپس برای هر کدام از داده‌های آزمایش با استفاده از فرمول (۲-۴۳) میزان شباهت به ازای هر کدام از مدل‌های موجود در پایگاه داده سیستم

محاسبه می‌شود. در نهایت تمام این امتیازات را با همدیگر مقایسه می‌کنند و نمونه آزمایش معادل با مدلی است که امتیاز بیشتری را در این مقایسه کسب کرده باشد.

در جدول (۴-۱) دو پارامتر وجود دارد که بر کیفیت کارایی سیستم تاثیر می‌گذارد. این دو پارامتر عبارتند از تعداد حالت‌ها و تعداد ترکیب‌ها به ازای هر حالت. بیشترین درصد به ازای تعداد حالت ۳ و تعداد ترکیب ۳ به ازای هر حالت می‌باشد. در این شرایط درصد شناسایی برابر با ۸۹ درصد می‌باشد. مشاهده می‌شود که با افزایش تعداد حالت‌ها، درصد شناسایی کاهش پیدا کرده است. و البته با کمتر شدن تعداد حالت‌ها از عدد ۳ نیز، درصد شناسایی کاهش پیدا می‌کند. علاوه بر این تعداد ترکیب‌های تشکیل دهنده هر حالت نیز بر درصد شناسایی تاثیر می‌گذارد به این ترتیب که تعداد ترکیب ۳، بهترین نتیجه را به دست داده است. برای بررسی اثر تعداد ترکیب، نتایج حاصل از ترکیب‌های ۲ و ۵ نیز در جدول نمایش داده شده است. مشاهده می‌شود که تعداد ترکیب هنگامی که از عدد ۳ کمتر باشد، درصد شناسایی از ۸۹ درصد کمتر می‌باشد ولی هنگامی که به ۳ می‌رسد، درصد شناسایی هم افزایش پیدا می‌کند. البته با افزایش تعداد ترکیب‌ها، درصد شناسایی کاهش پیدا می‌کند. علت اینکه با افزایش تعداد حالت‌ها، درصد شناسایی کاهش پیدا می‌کند، این است که با تعداد حالت برابر با ۳ به بهترین شکل؛ اعداد مدل می‌شوند ولی با افزایش تعداد حالت‌ها، مدل باید طوری تخمین زده شود که از اطلاعاتی که برای یک کلمه دارد، پارامترهای تعداد حالت‌های بیشتر از حد نیاز را تخمین بزند که این موضوع باعث می‌شود که مدل بهینه‌ای به دست نیاید و بنابراین درصد شناسایی کلمات کاهش پیدا می‌کند. در مورد تعداد ترکیب‌ها به ازای هر حالت نیز همین موضوع صادق می‌باشد. به این ترتیب که با فزایش تعداد ترکیب‌ها، اطلاعات لازم برای تخمین تعداد پارامترهای ترکیب‌های بیشتر در دسترس نمی‌باشد و بنابراین پارامترهایی که تخمین زده می‌شوند، دقت خوبی نخواهند داشت و این موضوع باعث کاهش درصد شناسایی کلمات می‌شود.

۷-۳-۲ نتایج آزمایش مربوط به جستجوی کلمات در گفتار گسسته

در این بخش به بررسی نتایج آزمایش در مورد شناسایی کلمات در گفتار گسسته پرداخته شده است. روش کار به این صورت است که از پایگاه داده TIMIT به ازای هر گوینده ۱۰ کلمه استخراج شده و سپس با استفاده از داده‌های تمام گویندگان، برای هر کلمه یک مدل HMM ساخته شده است. تعداد نمونه‌های آموزش به ازای هر کلمه ۶۰۰ نمونه می‌باشد و برای آزمایش نیز به ازای هر کلمه ۳۰ نمونه در نظر گرفته شده است. کلماتی که برای آزمایش در نظر گرفته شده‌اند در جدول (۴-۲) نمایش داده شده است.

جدول(۷-۲): کلمات استفاده شده به عنوان کلمات کلیدی در آزمایشات

شماره کلمه	۱	۲	۳	۴	۵	۶	۷	۸	۹	۱۰
نام کلمه	she	Your	suit	greasy	Water	year	ask	carry	like	that

آزمایش‌های انجام شده را می‌توان به چند بخش تقسیم کرد. ابتدا برای بررسی اثر استخراج ویژگی بر درصد شناسایی کلمات، یک مقایسه بین دو روش استخراج ویژگی MFCC و LPCC انجام شده است. در این مقایسه از روش‌های بهبود ویژگی با استفاده از ضرایب دلتا و شتاب استفاده شده است. در ادامه پس از انجام این آزمایشات، اثر بهبود ویژگی‌ها با استفاده از ضرایب دلتا و شتاب بر روی درصد شناسایی کلمات در گفتار بررسی شده است.

۷-۳-۲-۱ مقایسه روش‌های MFCC و LPCC

برای استخراج ویژگی طول قاب‌ها ۲۵ میلی ثانیه در نظر گرفته شده و میزان هم‌پوشانی بین قاب‌ها نیز ۱۵ میلی ثانیه انتخاب شده است. پس هر ۱۰ میلی ثانیه یک بار قاب برداری انجام می‌شود. در روش MFCC به طور معمول اندازه بردار ویژگی ۱۳ فرض می‌شود. این عدد با احتساب ضریب

انرژی، (C_0)، می‌باشد. در آزمایشاتی که انجام شده است از ضریب انرژی صرفنظر شده است. دلیل این موضوع این است که ضریب انرژی به نسبت دیگر ضرایب بردار ویژگی اندازه بسیار بزرگتری دارد و این تفاوت در اندازه باعث کاهش دقت در پیاده سازی سخت افزاری می‌شود. زیرا هنگامی که اعداد نرمالیزه شده و به محیط سخت افزار انتقال داده می‌شوند، مقداری از اطلاعات از بین می‌روند و این باعث کاهش کارایی سیستم می‌شود. اگر ضریب انرژی که اندازه بسیار بزرگتری نسبت به سایر اعداد دارد حذف شود، هنگام نرمالیزه کردن اعداد اطلاعات کمتری از بین می‌رود و به این ترتیب کارایی سیستم بهبود پیدا می‌کند. بنابراین در روش MFCC، بردارهای ویژگی ۱۲ درایه‌ای فرض شده‌اند. در روش LPCC به این منظور که با روش MFCC هماهنگ شود، از ضریب (C_0) صرفنظر شده و بردارهای ویژگی ۱۲ درایه‌ای در نظر گرفته شده‌اند.

بعد از به دست آوردن ضرایب کپسترال در هر دو روش استخراج ویژگی، ابتدا از ضرایب کپسترال، ضرایب دلتا استخراج می‌شود که تعداد این ضرایب ۱۲ عدد می‌باشد و در مرحله بعد با استفاده از ضرایب دلتا، ضرایب شتاب استخراج می‌شود که در نهایت با کنار هم قرار دادن این سه دسته ضریب، یک بردار ویژگی ۳۶ درایه‌ای به دست می‌آید.

هنگامی که بردارهای ویژگی مربوط به هر کلمه استخراج شد، برای آموزش مدل‌های HMM همانند بخش ۲-۳-۱ از HMM all toolbox استفاده شده است و روند آموزش نیز همانند قبل است. در این آزمایش نیز پارامترهایی که متغیر هستند عبارتند از تعداد حالت‌های مدل و تعداد ترکیب‌هایی که یک حالت را تشکیل می‌دهند. نتایج آزمایش به ازای هر کدام از کلمات و به ازای روش استخراج ویژگی MFCC در جداول (۴-۳) تا (۴-۱۲) نمایش داده شده‌اند.

جدول(۷-۳): میزان نتایج شناسایی برای کلمه **she** بر حسب درصد برای روش MFCC

شماره آزمایش	تعداد حالت‌های مدل	تعداد ترکیب‌های هر حالت		
		۲	۳	۵
آزمایش اول	۲	۷۰	۷۶/۶۶	۷۳/۳۳
آزمایش دوم	۳	۷۶/۶۶	۸۴/۱۶	۷۸/۳۳
آزمایش سوم	۴	۷۳/۳۳	۸۰	۷۶/۶۶
آزمایش چهارم	۵	۷۰	۷۶/۶۶	۷۳/۳۳

جدول(۷-۴): میزان نتایج شناسایی برای کلمه **your** بر حسب درصد برای روش MFCC

شماره آزمایش	تعداد حالت‌های مدل	تعداد ترکیب‌های هر حالت		
		۲	۳	۵
آزمایش اول	۲	۷۳/۳۳	۸۰	۷۶/۶۶
آزمایش دوم	۳	۸۱/۶۶	۸۶/۶۶	۸۳/۳۳
آزمایش سوم	۴	۷۶/۶۶	۸۴/۱۶	۸۰
آزمایش چهارم	۵	۷۳/۳۳	۸۰	۷۶/۶۶

جدول(۷-۵): میزان شناسایی برای کلمه **suit** بر حسب درصد برای روش MFCC

شماره آزمایش	تعداد حالت‌های مدل	تعداد ترکیب‌های هر حالت		
		۲	۳	۵
آزمایش اول	۲	۶۶/۶۶	۷۳/۳۳	۷۰
آزمایش دوم	۳	۷۶/۶۶	۸۰	۷۸/۳۳
آزمایش سوم	۴	۷۳/۳۳	۷۸/۳۳	۷۳/۳۳
آزمایش چهارم	۵	۷۰	۷۵	۷۳/۳۳

جدول(۷-۶): میزان شناسایی برای کلمه **greasy** بر حسب درصد برای روش MFCC

شماره آزمایش	تعداد حالت‌های مدل	تعداد ترکیب‌های هر حالت		
		۲	۳	۵
آزمایش اول	۲	۶۶/۶۶	۷۳/۳۳	۷۰
آزمایش دوم	۳	۷۶/۶۶	۸۳/۳۳	۸۰
آزمایش سوم	۴	۸۰	۸۵	۷۸/۳۳
آزمایش چهارم	۵	۷۶/۶۶	۸۰	۷۳/۳۳

جدول(۷-۷): میزان شناسایی برای کلمه **water** بر حسب درصد برای روش MFCC

شماره آزمایش	تعداد حالت‌های مدل	تعداد ترکیب‌های هر حالت		
		۲	۳	۵
آزمایش اول	۲	۶۶/۶۶	۷۰	۶۸/۳۳
آزمایش دوم	۳	۷۳/۳۳	۷۸/۳۳	۷۶/۶۶
آزمایش سوم	۴	۷۶/۶۶	۸۰	۷۸/۳۳
آزمایش چهارم	۵	۷۵	۷۶/۶۶	۷۳/۳۳

جدول(۷-۸): میزان شناسایی برای کلمه **year** بر حسب درصد برای روش MFCC

شماره آزمایش	تعداد حالت‌های مدل	تعداد ترکیب‌های هر حالت		
		۲	۳	۵
آزمایش اول	۲	۷۰	۷۶/۶۶	۷۳/۳۳
آزمایش دوم	۳	۷۸/۳۳	۸۳/۳۳	۸۰
آزمایش سوم	۴	۷۶/۶۶	۸۰	۷۸/۳۳
آزمایش چهارم	۵	۷۵	۷۸/۳۳	۷۳/۳۳

جدول(۷-۹): میزان شناسایی برای کلمه **ask** بر حسب درصد برای روش MFCC

شماره آزمایش	تعداد حالت‌های مدل	تعداد ترکیب‌های هر حالت		
		۲	۳	۵
آزمایش اول	۲	۶۳/۳۳	۷۰	۶۶/۶۶
آزمایش دوم	۳	۶۸/۳۳	۷۸/۳۳	۷۶/۶۶
آزمایش سوم	۴	۶۶/۶۶	۷۵	۶۸/۳۳
آزمایش چهارم	۵	۷۰	۷۳/۳۳	۷۰

جدول(۷-۱۰): میزان شناسایی برای کلمه **carry** بر حسب درصد برای روش MFCC

شماره آزمایش	تعداد حالت‌های مدل	تعداد ترکیب‌های هر حالت		
		۲	۳	۵
آزمایش اول	۲	۶۳/۳۳	۷۰	۶۶/۶۶
آزمایش دوم	۳	۷۱/۶۶	۷۸/۳۳	۷۰
آزمایش سوم	۴	۷۸/۳۳	۸۱/۶۶	۸۰
آزمایش چهارم	۵	۷۵	۸۰	۷۶/۶۶

جدول(۷-۱۱): شناسایی برای کلمه **like** بر حسب درصد برای روش MFCC

شماره آزمایش	تعداد حالت‌های مدل	تعداد ترکیب‌های هر حالت		
		۲	۳	۵
آزمایش اول	۲	۶۸/۳۳	۷۳/۳۳	۷۰
آزمایش دوم	۳	۷۵	۸۰	۷۸/۳۳
آزمایش سوم	۴	۷۶/۶۶	۷۸/۳۳	۷۵
آزمایش چهارم	۵	۷۳/۳۳	۷۵	۷۴/۱۶

جدول(۷-۱۲): شناسایی برای کلمه **that** بر حسب درصد برای روش MFCC

شماره آزمایش	تعداد حالت‌های مدل	تعداد ترکیب‌های هر حالت		
		۲	۳	۵
آزمایش اول	۲	۶۸/۳۳	۷۵	۷۰
آزمایش دوم	۳	۷۸/۳۳	۸۳/۳۳	۸۰
آزمایش سوم	۴	۷۶/۶۶	۸۰	۷۸/۳۳
آزمایش چهارم	۵	۷۳/۳۳	۷۶/۶۶	۷۵

همانند نتایج آزمایش مربوط به اعداد گسسته، در آزمایشات مربوط به کلمات در گسسته گفتار مشاهده می‌شود که با تغییر تعداد حالت‌ها درصد شناسایی کلمات نیز تغییر می‌کند. هنگامی که تعداد حالت‌ها کم باشد، مثلا در این آزمایشات عدد ۲، درصد شناسایی کلمات نیز کم می‌باشد و با افزایش تعداد حالت‌ها، درصد شناسایی کلمات افزایش پیدا می‌کند. ولی این افزایش درصد شناسایی کلمات، با افزایش تعداد حالت‌ها از یک تعداد حالتی به بعد متوقف شده و روند نزولی به خود می‌گیرد. تعداد ترکیب‌های گاوسی تشکیل دهنده هر حالت نیز بر نتیجه آزمایش تاثیر می‌گذارد و همانند حالت‌ها، ابتدا با افزایش تعداد ترکیب‌ها، درصد شناسایی کلمات نیز افزایش می‌یابد که در ادامه با افزایش تعداد ترکیب‌ها، روند افزایش درصد شناسایی کلمات متوقف شده و درصد شناسایی کلمات رو به کاهش می‌گذارد.

در آزمایشات انجام شده برای شناسایی کلمات در گفتار گسسته، در بعضی از کلمات انتخاب شده، تعداد حالت ۳ بیشترین درصد شناسایی کلمات را به دست داده است و در بعضی دیگر تعداد حالت ۴ بیشترین درصد را به خود اختصاص داده است. در ادامه در جدول (۴-۱۳)، میانگین درصد شناسایی برای تمام کلمات و به ازای حالات و ترکیبات مختلف نمایش داده شده است.

جدول(۷-۱۳): میانگین درصد شناسایی کلمات برای کلمات در گفتار گسسته برای روش MFCC

شماره آزمایش	تعداد حالت‌های مدل	تعداد ترکیب‌های هر حالت		
		۲	۳	۵
آزمایش اول	۲	۶۷/۶۶	۷۳/۸۳	۷۰/۵۰
آزمایش دوم	۳	۷۵/۶۶	۸۱/۵۸	۷۶/۱۶
آزمایش سوم	۴	۷۵/۵۰	۸۰/۲۵	۷۶/۶۶
آزمایش چهارم	۵	۷۳/۱۶	۷۷/۱۶	۷۳/۹۱

همانگونه که مشاهده می‌شود، میانگین درصد شناسایی به ازای هر ۱۰ کلمه نشان می‌دهد که ، بیشترین درصد شناسایی هنگامی به دست می‌آید که تعداد حالات ۳ باشد و هر حالت نیز با ۳ ترکیب گاوسی مدل شده باشد و روند تغییرات درصد شناسایی کلمات در جدول به خوبی نمایش داده شده است. در ادامه همین آزمایشات با استفاده از روش استخراج ویژگی LPCC انجام شده است و میانگین نتایج شناسایی کلمات به ازای هر ۱۰ کلمه در جدول (۴-۱۴) ارائه شده است.

جدول(۷-۱۴): متوسط درصد شناسایی کلمات برای کلمات در گفتار گسسته برای روش LPCC

شماره آزمایش	تعداد حالت‌های مدل	تعداد ترکیب‌های هر حالت		
		۲	۳	۵
آزمایش اول	۲	۶۰/۵۰	۶۸/۷۳	۶۶/۶۶
آزمایش دوم	۳	۷۳/۲۱	۷۷/۳۳	۷۳/۱۷
آزمایش سوم	۴	۷۱/۶۶	۷۵/۱۶	۶۹/۱۲
آزمایش چهارم	۵	۶۸/۳۳	۷۲/۶۶	۷۰/۱۶

با توجه به جدول (۴-۱۴)، در روش بر مبنای LPCC تغییرات درصد شناسایی کلمات در گفتار گسسته همانند روش بر مبنای MFCC بوده و تغییرات درصد شناسایی بر اساس تغییرات تعداد حالت‌ها و ترکیب‌های تشکیل دهنده حالت‌ها همانند روش بر مبنای MFCC می‌باشد.

با مقایسه نتایج حاصل از آزمایشات در جدول‌های (۲-۱۳) و (۲-۱۴) با استفاده از دو روش MFCC و LPCC مشاهده می‌شود که درصد شناسایی کلمات در گفتار گسسته با استفاده از روش MFCC نتیجه بهتری به دست می‌شود. همانطور که مشاهده می‌شود بیشترین درصد شناسایی در روش بر مبنای MFCC حدود ۴.۲۵ درصد نسبت به روش بر مبنای LPCC بیشتر است.

۷-۳-۲-۲ بررسی اثر استفاده از مشتقات زمانی

همانگونه که در فصل دوم شرح داده شد، هنگام استخراج ویژگی می‌توان به ضرایب کپسترال اکتفا کرد و یا اینکه علاوه بر ضرایب کپسترال از روش‌هایی مانند مشتقات زمانی و .. برای بهبود کیفیت ویژگی‌های استخراج شده استفاده کرد. از طرفی هنگامی که ابعاد بردارهای ویژگی بزرگ می-شود، متناسب با آن‌ها باید پارامترهای بیشتری برای مدل تخمین زد. مثلا هنگامی که بردار ویژگی دارای ۱۲ درایه می‌باشد، ماتریس‌های کواریانس ترکیب‌های گاوسی تشکیل دهنده حالت‌ها، دارای ابعاد ۱۲ در ۱۲ هستند، اما هنگامی که بردارهای ویژگی دارای ۳۶ درایه باشند، ماتریس‌های کواریانس ابعادی برابر با ۳۶ در ۳۶ دارند. افزایش تعداد پارامترهای مدل باعث افزایش حجم پایگاه داده می‌شود و از طرفی باعث افزایش بار محاسباتی نیز می‌شود. از آن جهت که هدف پایان نامه پیاده‌سازی سخت افزاری سیستم جستجوی کلید واژه می‌باشد و میزان حافظه در دسترس DSP نیز محدود می‌باشد، باید بررسی شود که آیا می‌توان با کاهش ابعاد بردار ویژگی، بار محاسباتی را کاهش داد و در عین حال دقت کارکرد سیستم نیز تغییر چندانی نکند؟ برای بررسی این موضوع یک بار آزمایشات مراحل قبل بر روی کلمات گسسته با استفاده از بردار ویژگی ۱۲ درایه‌ای و بدون استفاده از مشتقات زمانی، انجام شده است. میانگین نتایج حاصل از این آزمایش در جدول (۴-۱۵) ارائه شده است.

جدول(۷-۱۵): متوسط درصد شناسایی کلمات در گفتار گسسته برای روش MFCC با ۱۲ درایه

شماره آزمایش	تعداد حالت‌های مدل	تعداد ترکیب‌های هر حالت		
		۲	۳	۵
آزمایش اول	۲	۶۳/۳۳	۷۰	۶۸/۸۳
آزمایش دوم	۳	۷۲/۸۲	۷۸/۱۶	۷۵
آزمایش سوم	۴	۷۱/۶۶	۷۶/۶۶	۷۲/۸۱
آزمایش چهارم	۵	۷۰/۱۶	۷۴/۳۳	۷۰/۴۴

اگر به جدول‌های (۴-۱۳) و (۴-۱۴) دقت شود، مشاهده می‌شود که استفاده از روش مشتقات زمانی برای بهبود بردارهای ویژگی، تاثیر مثبتی در افزایش درصد شناسایی کلمات دارد. زمانی که از پارامترهای مشتقات زمانی دلتا و شتاب استفاده شده است، متوسط درصد شناسایی نسبت به زمانی که از روش مشتقات زمانی استفاده نشده است، حدود ۳ درصد بیشتر می‌باشد. نتیجه جالب توجه دیگری که می‌توان از مقایسه جدول‌های (۴-۱۴) و (۴-۱۵) مشاهده کرد این است که روش MFCC بدون استفاده از روش مشتقات زمانی، به نسبت روش LPCC همراه با مشتقات زمانی، درصد شناسایی بهتری به دست می‌دهد.

۳-۳-۷ نتایج آزمایش مربوط به جستجوی کلمات در گفتار پیوسته

در این بخش آزمایشاتی برای بررسی نتایج استفاده از مدلسازی به روش مدل مخفی مارکوف برای جستجوی کلید واژه در گفتار پیوسته انجام شده است. بدین منظور همانند بخش‌های قبل برای هر کدام از کلماتی کلیدی، با استفاده از الگوریتم Baum – Welch یک مدل HMM آموزش داده شده است و یک جمله که حاوی کلمه کلیدی مورد نظر می‌باشد از پایگاه داده انتخاب شده و ویژگی-های آن به روش MFCC استخراج شده است. در ضمن از ضرایب مشتقات زمانی دلتا و شتاب نیز استفاده شده و بردار ویژگی دارای ۳۶ درایه می‌باشد. برای جستجوی کلید واژه در جمله از الگوریتم ویتربی استفاده شده است. برای هر کدام از کلمات کلیدی، علاوه بر اینکه یک مدل HMM آموزش داده شده است، یک کلمه مرجع نیز با طولی معادل میانگین طول کلمات آموزشی انتخاب شده، ویژگی‌های MFCC آن استخراج شده و با استفاده از الگوریتم ویتربی، محتملترین دنباله حالتی که این کلمه مرجع از آن پیروی می‌کند به دست آورده شده است. در ادامه با این شرط که در جمله حتما کلید واژه وجود دارد و تعداد کلیدواژهای موجود در جمله نیز تنها یک عدد می‌باشد، آزمایشاتی انجام شده است. بنابراین وظیفه الگوریتم ویتربی این است که مکان آن یک کلمه کلیدی که در جمله موجود است را با استفاده از تطبیق دادن دنباله حالت کلمه مرجع با دنباله حالت مشاهدات جمله مورد آزمایش بیابد. همانند مراحل قبلی آزمایش، تعداد نمونه‌های آزمایش ۳۰ عدد می‌باشد. در ضمن با توجه به جدول‌های (۴-۳) تا (۴-۱۲)، به ازای هر کلمه کلیدی، مدل HMM بر اساس تعداد حالت و ترکیب گاوسی‌ای ساخته شده است که توانسته در آزمایشات مربوط به حالت گسسته بیشترین امتیاز را کسب کند. در جدول (۴-۱۶)، نتایج حاصل از این آزمایش ارائه شده است.

جدول(۷-۱۶): متوسط درصد شناسایی کلمات در گفتار پیوسته برای روش MFCC

شماره آزمایش	نام کلمه	تعداد تشخیص‌های صحیح	تعداد تشخیص‌های غلط	متوسط درصد شناسایی
۱	she	۲۰	۱۰	۶۶/۶۶
۲	your	۲۱	۹	۷۰
۳	suit	۱۸	۱۲	۶۰
۴	greasy	۲۰	۱۰	۶۶/۶۶
۵	water	۱۹	۱۱	۶۳/۳۳
۶	year	۱۹	۱۱	۶۳/۳۳
۷	ask	۱۷	۱۳	۵۶/۶۶
۸	carry	۱۹	۱۱	۶۶/۶۶
۹	like	۱۸	۱۲	۶۰
۱۰	that	۱۹	۱۱	۶۶/۶۶
		متوسط درصد شناسایی		۶۳/۹۹

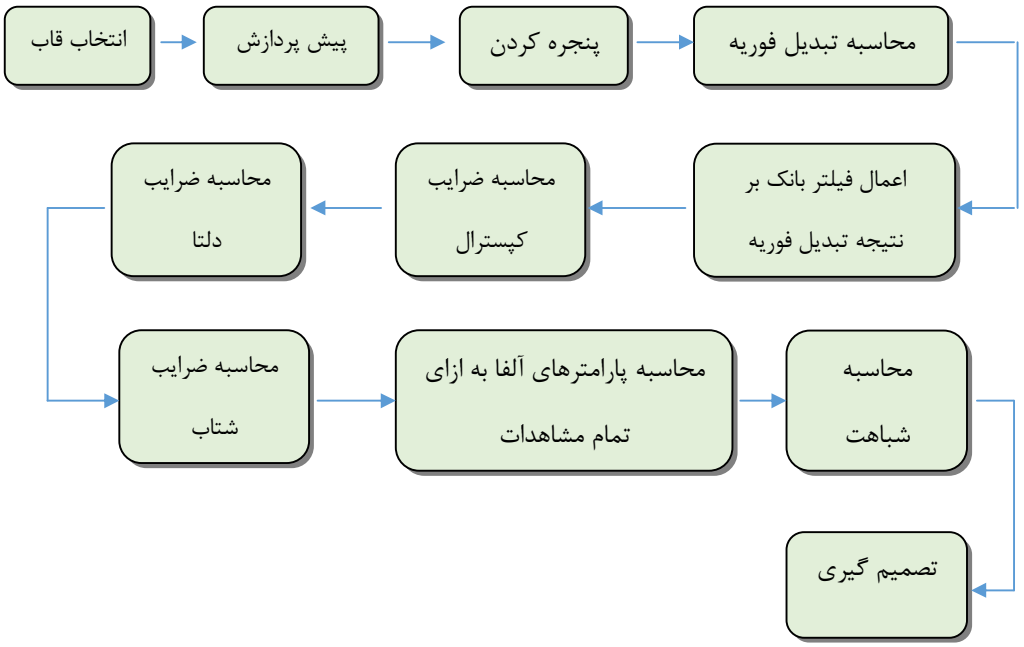
همانگونه که مشاهده می‌شود، نتایج نسبت به حالت گسسته افت بسیار زیادی داشته است. آزمایش-هایی که بر روی سیستم جستجوی کلمه در گفتار پیوسته انجام شده است با این پیش شرط بوده‌اند که در جمله‌ای که در در آن جستجو انجام می‌شود تنها یک بار کلمه کلیدی مورد نظر تکرار شده است. بنابراین معیاری که برای بررسی کیفیت کارایی این سیستم وجود دارد این است که به ازای نمونه‌های آزمایش، یعنی تعداد جمله‌هایی که در آن‌ها جستجو انجام می‌شود، چند بار مکان کلمه کلیدی به صورت صحیح تشخیص داده شده است. با توجه به نتایج جدول (۴-۱۶) مشاهده می‌شود

که متوسط کارایی سیستم به ازای تمام کلمات و نمونه‌های آموزش برابر با ۷۰/۶۶ می‌باشد که نسبت به حالتی که جستجو در گفتار گسسته انجام می‌شود، بسیار کمتر می‌باشد.

۴-۷ بررسی نتایج آزمایشات بر روی سخت افزار

در این بخش به بررسی پیاده‌سازی الگوریتم جستجوی کلمه کلیدی در گفتار گسسته پرداخته شده است. برای استخراج ویژگی دو انتخاب وجود دارد. اولین انتخاب این است که از بردارهای ویژگی ۱۲ درایه‌ای MFCC استفاده شود و از ضرایب مشتقات زمانی دلتا و شتاب استفاده نشود. در این حالت طبق جدول (۴-۱۵) به صورت متوسط بیشترین درصد شناسایی کلمات کلیدی به ازای ۱۰ کلمه انتخاب شده از پایگاه داده TIMIT برابر با ۷۸/۱۶ می‌باشد. در حالت دوم می‌توان از روش MFCC همراه با ضرایب مشتقات زمانی استفاده کرد که در این صورت بردارهای ویژگی دارای ۳۶ درایه می‌باشند و بر اساس داده‌های جدول(۴-۱۳) حد اکثر درصد شناسایی کلمات کلیدی در گفتار گسسته به ازای ۱۰ کلمه استخراج شده از پایگاه داده TIMIT برابر با ۸۱/۵۸ می‌باشد. به دلیل بیشتر بودن متوسط درصد شناسایی در حالتی که از ضرایب مشتقات زمانی استفاده می‌شود، از این روش استخراج ویژگی بر روی پردازنده DSP استفاده شده است. همانگونه که در ابتدای فصل ذکر شد، در استخراج ویژگی و مدل کردن و آزمایشات انجام شده از ضریب انرژی صرفنظر شده است. به این دلیل که اندازه ضریب انرژی به نسبت سایر درایه‌های بردار ویژگی بسیار بزرگتر می‌باشد، هنگام نرمالیزه کردن و وارد کردن داده‌ها به حافظه پردازنده، میزان زیادی از اطلاعات از دست می‌رود که این موضوع باعث کاهش کارایی سیستم می‌شود. به همین دلیل از استفاده از ضریب انرژی صرفنظر شده است و در آزمایشات مربوط به نرم افزار Matlab نیز از ضریب انرژی استفاده نشده است.

برای پیاده‌سازی الگوریتم بر روی سخت افزار باید از استخراج ویژگی شروع کرد و به ترتیب پارامترهای لازم برای محاسبات را به دست آورد. این فرایند را می‌توان در شکل (۴-۱) مشاهده کرد.



شکل(۴-۱): دیاگرام بلوکی پیاده‌سازی سیستم جستجوی کلمه در گفتار گسسته

برای پیاده‌سازی سخت افزاری مانند قسمت آزمایشات در محیط Matlab ابتدا برای کلماتی که از پایگاه داده TIMIT استخراج شده‌اند باید یک مدل HMM ساخت که بتوان با آن بیشترین درصد شناسایی را به دست آورد. این کار قبلا در هنگام آزمایشاتی در محیط نرم افزار TIMIT انجام شده است، و در جدول‌های (۴-۳) تا (۴-۱۲) اطلاعات مربوط به بهینه‌ترین پارامترهای مدل به ازای هر کلمه وارد شده است. هنگامی که مدل‌های کلمات کلیدی ساخته شد، باید این مدل‌ها را به حافظه پردازنده انتقال داد. در مرحله بعد باید نمونه‌های آزمایش را وارد پردازنده کرد. در این مرحله باید ابتدا از نمونه آزمایش استخراج ویژگی به عمل آید و دنباله مشاهدات را به ازای نمونه آزمایش به دست آورد. مراحل کار مطابق با شکل (۴-۱) می‌باشد. به این دلیل که حجم حافظه پردازنده DSP محدود می‌باشد، برای استخراج ویژگی، ابتدا یک قاب انتخاب می‌شود، مراحل استخراج ویژگی بر روی آن

انجام می‌شود و سپس نتایج در حافظه ذخیره شده و در ادامه قاب دوم انتخاب شده و ویژگی‌های آن استخراج شده در حافظه ذخیره می‌شود و این کار به همین ترتیب ادامه پیدا می‌کند تا اینکه ویژگی-های تمام قاب‌های سیگنال آزمایش استخراج شود.

مراحل استخراج ویژگی برای یک قاب به این ترتیب می‌باشد که ابتدا عمل پیش پردازش بر روی قاب انجام می‌شود که عبارت است از اعمال یک فیلتر بالاگذر مرتبه اول بر سیگنال گفتار. در مرحله بعد برای کم کردن اثر اعوجاج قاب بندی در حوزه سیگنال، قاب پیش پردازش شده در یک تابع پنجره همینگ ضرب شده است. در ادامه تبدیل فوریه ۵۱۲ نقطه‌ای برای قاب پنجره شده محاسبه می‌شود. در مرحله بعد فیلتر بانک مل بر روی ضرایب تبدیل فوریه به دست آمده اعمال می-شود و ۲۵۶ ضریب تبدیل فوریه به ۳۰ عدد خروجی فیلتر بانک تبدیل می‌شود. هنگامی که این کار انجام شد، با استفاده از تبدیل کسینوسی، ضرایب کپسترال برای قاب مورد نظر استخراج می‌شود. تا این مرحله، ضرایب کپسترال محاسبه شده‌اند و اکنون می‌توان با استفاده از این ضرایب، ضرایب مشتقات زمانی را محاسبه کرد. یعنی اینکه ابتدا ضرایب دلتا از ضرایب کپسترال استخراج شده و سپس ضرایب شتاب از ضرایب دلتا استخراج می‌شوند. اکنون فرایند استخراج ویژگی به ازای یک قاب به اتمام رسیده است. در ادامه این بردار ویژگی ۳۶ درایه‌ای در حافظه ذخیره شده و فرایند استخراج ویژگی برای قاب بعدی شروع می‌شود. این فرایند به همین ترتیب ادامه پیدا می‌کند تا زمانی که ویژگی‌های سیگنال آزمایش به طور کامل استخراج شوند.

حال که ویژگی‌های سیگنال آزمایش استخراج شده‌اند باید در مرحله بعد، پارامترهای آلفا محاسبه شوند. به این ترتیب که برای هر کدام از کلمات کلیدی که در پایگاه داده ذخیره شده‌اند باید یک بار پارمترهای آلفا محاسبه شوند. برای اینکه دقت سیستم جستجوی کلمات بیشتر شود، باید در هنگام محاسبه آلفاها، آن‌ها را نرمالیزه کرد. این کار بر طبق بخش ۲-۴-۲-۲ انجام شده است. در نهایت با استفاده از آلفاهای به دست آمده به ازای هر مدل HMM و سیگنال آزمایش ورودی می‌توان

میزان شباهت را با استفاده از معادله (۲-۴۳) به دست آورد. وقتی که اندازه شباهت برای تمام مدل‌ها محاسبه شد، امتیازات را با هم مقایسه می‌کنند و هر کدام که امتیاز بیشتری داشته باشد، سیگنال ورودی معادل با آن کلمه می‌باشد.

در آزمایشی که بر روی سخت افزار انجام شده است، برای ۵ کلمه مدل HMM ساخته شده و به حافظه پردازنده انتقال داده شده است. تعداد نمونه‌های آموزش نیز ۳۰ عدد می‌باشد. نتایج این آزمایش و مقایسه این نتایج با نتایج به دست آمده از آزمایش در نرم افزار Matlab در جدول (۴-۱۷) قابل مشاهده می‌باشد.

جدول(۷-۱۷): مقایسه نتایج حاصل از شبیه سازی Matlab با نتایج آزمایش بر روی سخت افزار

شماره آزمایش	نام کلمه	نتایج آزمایش در محیط Matlab	نتایج آزمایش بر روی سخت افزار
۱	she	۸۴/۱۶	۷۳/۳۳
۲	greasy	۸۵	۷۱/۱۶
۳	year	۸۳/۳۳	۶۸/۹۱
۴	carry	۸۱/۶۶	۷۰
۵	like	۸۰	۶۸/۴۴
	میانگین درصد شناسایی سخت‌افزار		۷۰/۳۶

همانگونه که مشاهده می‌شود نتایج آزمایش سخت افزاری نسبت به نتایج حاصل از آزمایشات سخت افزاری پایین‌تر می‌باشد. سیستم محاسباتی و حافظه پردازنده بر مبنای سیستم ممیز ثابت می‌باشد و به دلیل حجم بالای محاسباتی که در پردازنده برای آزمایشات انجام می‌شود، میزان اتلاف اطلاعات بالا می‌باشد. برای بهبود درصد کارایی سیستم می‌توان از حافظه‌های جانبی در کنار پردازنده کمک گرفت که به این ترتیب می‌توان اعداد را با دقت بالاتری نرمالیزه کرد و در حین محاسبات نیز

داده‌های کمتری از دست می‌روند.

در ادامه به صورت نمونه برای کلمه greasy ، زمـان و دوره سـاعت هـر کـدام از قسـمت‌هـای آزمایش در جدول (۴-۱۸) نمایش داده شده است.

جدول(۷-۱۸): زمان و دوره‌های ساعت برای آزمایش به ازای کلمه greasy

	مرحله الگوریتم	تعداد دوره ساعت	زمان به میلی ثانیه
استخراج ویژگی	پیش پردازش	۴۳۴۱	۰/۰۳
	پنجره کردن	۲۲۸۴	۰/۰۱۶
	محاسبه تبدیل فوریه	۱۳۵۴۸۸	۰/۰۹۴
	اعمال فیلتر بانک	۶۷۷۶	۰/۰۴۷
	محاسبه ضرایب کپسترال	۴۹۳۲	۰/۰۳۴
	محاسبه ضرایب دلتا	۳۵۲۱	۰/۰۲۵۱
	محاسبه ضرایب شتاب	۳۵۲۱	۰/۰۲۵
	زمان کلی استخراج ویژگی	۱۶۰۸۶۳	۱/۱۲
	زمان استخراج ویژگی به ازای کلمه	۴۵۰۴۱۶۴	۳۱/۲۸
محاسبه شباهت	محاسبه پارامترهای آلفا	۷۵۳۲۹۱	۵/۲۳
	محاسبه شباهت	۵۳۴۶	۰/۰۳۷
	زمان کلی بخش محاسبه شباهت	۷۵۸۶۳۷	۵/۲۷
زمان کل			۳۶/۵۵

در جدول (۴-۱۸)، منظور از زمان کلی استخراج ویژگی، زمان بـه ازای یـک قـاب مـی‌باشـد و زمـان استخراج ویژگی به ازای کلمه عبارت است از زمان لازم برای استخراج ویژگی از تمام قاب‌های کلمه‌ی مورد نظر. برای کلمه greasy، تعداد قاب‌ها به صورت متوسط برابر با ۲۸ قاب می‌باشد. برای به دسـت آوردن زمان ، باید تعداد دوره‌های ساعت را بر فرکانس کاری مدار، یعنی ۱۴۴کیلو هرتز، تقسیم کرد.

فصل پنجم:

جمع‌بندی و ارائه پیشنهادات

۸-۱جمع‌بندی و نتیجه‌گیری

در این پژوهش ابتدا مروری بر روش‌های استخراج ویژگی از سیگنال گفتار انجام شد و روش-های مختلف استخراج ویژگی بررسی شد. از میان این روش‌ها، روش‌های استخراج ویژگی زمان کوتاه انتخاب شده و از میان این دسته از روش‌ها به صورت خاص روش‌های MFCC و LPCC انتخاب، پیاده‌سازی و بررسی شدند. در ادامه برای مدل کردن کلمات از مدل مخفی مارکوف که به عنوان یک مدل قدرتمند در زمینه مدل کردن پدیده‌های احتمالاتی مانند سیگنال گفتار است، استفاده شد. سیستم‌های جستجوی کلمات کلیدی در گفتار پیوسته و گسسته به صورت جداگانه بررسی شدند. نتایج آزمایشات نشان دادند که:

- روش استخراج ویژگی MFCC نسبت به روش استخراج ویژگی LPCC برای کاربردهای جستجوی کلید واژه نتیجه مناسب‌تری به دست می‌دهد.
- برای جستجوی کلمه در گفتار گسسته، پایگاه‌های داده‌ای که کلمات را به صورت جداگانه تهیه کرده‌اند نسبت به پایگاه داده‌هایی که کلمات را از گفتار گسسته استخراج کرده‌اند، درصد شناسایی کلمه بیشتری به دست می‌دهند.
- حداکثر میانگین درصد شناسایی اعداد عربی در گفتار گسسته که از پایگاه داده ساخته شده برای شناسایی اعداد گسسته شده طراحی شده است، برابر با ۹۳/۳۳ می‌باشد.
- حداکثر میانگین درصد شناسایی به دست آمده برای کلماتی که از پایگاه داده TIMIT استخراج شده و آموزش داده شده‌اند برابر با ۸۱/۵۸ است.
- حداکثر میانگین درصد شناسایی برای جستجوی کلمات در گفتار پیوسته ۷۰ درصد می‌باشد که نسبت به حالتی که در گفتار گسسته جستجو انجام می‌شود کمتر می-باشد.

پس از اینکه شبیه سازی‌های در محیط Matlab به اتمام رسید، ۵ کلمه انتخاب شد و برای هر کدام با توجه به نتیجه‌هایی که در شبیه‌سازی‌ها به دست آمد، یک مدل مخفی مارکوف ساخته شد. میانگین درصد شناسایی بر روی سخت افزار برای حالتی که جستجو در کلمات گسسته انجام می‌شود برابر ۷۰/۳۶ می‌باشد.

۸-۲ پیشنهاد برای کارهای آینده

- تهیه پایگاه داده مناسب برای آموزش کلمات در هر دو حالت جستجوی گسسته و جستجوی پیوسته که به میزان کافی داده‌های آموزشی در اختیار کاربر قرار دهد.
- استفاده از حافظه جانبی در کنار پردازنده برای بهبود دقت سیستم سخت افزاری.
- طراحی یک روش جستجو بهینه‌تر نسبت به روش ویتربی برای بهبود درصد شناسایی جستجوی کلمات در گفتار پیوسته.
- بررسی و بهینه‌سازی روش‌های آموزش مدل‌های HMM برای افزایش دقت مدل‌ها و رفع نقص اثر داده‌های تصادفی اولیه بر کارکرد سیستم آموزش مدل‌ها.
- استفاده از نرم‌افزارهای خاص پردازش گفتار مانند HMM Toolkit (HTK) جهت بهبود دقت مدل‌های HMM.

مراجع

[1] M. Gales and S. Young, "The application of hidden Markov models in speech recognition," *Foundations and trends in signal processing,* vol. 1, pp. 195-304, 2008.

[2] Y. Ling, "Keyword spotting in continuous speech utterances," McGill University, Montreal, 1999.

[3] I. Bhardwaj and N. D. Londhe, "Hidden Markov Model based isolated Hindi word recognition," in *Power, Control and Embedded Systems (ICPCES), 2012 2nd International Conference on*, 2012, pp. 1-6.

[4] I. Trabelsi and D. Ben Ayed, "On the use of different feature extraction methods for linear and non linear kernels," in *Sciences of Electronics, Technologies of Information and Telecommunications (SETIT), 2012 6th International Conference on*, 2012, pp. 797-802.

[5] T. Kinnunen and H. Li, "An overview of text-independent speaker recognition: From features to supervectors," *Speech communication,* vol. 52, pp. 12-40, 2010.

[6] M. P. Kesarkar, "Feature extraction for speech recognition," in *M. Tech. Credit Seminar Report, Electronic Systems Group, EE. Dept, IIT Bombay*, 2003, pp. 1-11.

[7] V. S. Baidwan and S. Gujral, "Comparative Analysis of Prosodic Features and Linear Predictive Coefficients for Speaker Recognition Using Machine Learning Technique," in *Devices, Circuits and Communications (ICDCCom), 2014 International Conference on*, 2014, pp. 1-8.

[8] M. Nilsson and M. Ejnarsson, "Speech recognition using hidden markov model," 2002.

[9] U. Shrawankar and V. M. Thakare, "Techniques for feature extraction in speech recognition system: A comparative study," *arXiv preprint arXiv:1305.1145,* 2013.

[10] B. Singh, R. Kaur, N. Devgun, and R. Kaur, "The Process Of Feature Extraction In Automatic Speech Recognition System For Computer Machine Interaction With Humans: A Review," *International Journal Of Advanced Research In Computer Science And Software Engineering, ISSN,* vol. 2277, 2012.

[11] S. Shanthi Therese and C. Lingam, "Review of Feature Extraction Techniques in Automatic Speech Recognition," *International Journal of*

Scientific Engineering and Technology, vol. 2, pp. 479-484, 2013.

[12] H. Kameoka, K. Yoshizato, T. Ishihara, K. Kadowaki, Y. Ohishi, and K. Kashino, "Generative Modeling of Voice Fundamental Frequency Contours," *Audio, Speech, and Language Processing, IEEE/ACM Transactions on,* vol. 23, pp. 1042-1053, 2015.

[13] H. Hermansky, "Perceptual linear predictive (PLP) analysis of speech," *the Journal of the Acoustical Society of America,* vol. 87, pp. 1738-1752, 1990.

[14] T. Kinnunen, I. Sidoroff, M. Tuononen, and P. Fränti, "Comparison of clustering methods: A case study of text-independent speaker modeling," *Pattern Recognition Letters,* vol. 32, pp. 1604-1617, 2011.

[15] N. Dave, "Feature extraction methods LPC, PLP and MFCC in speech recognition," *International Journal for Advance Research in Engineering and Technology,* vol. 1, pp. 1-4, 2013.

[16] S. O. Sadjadi and J. H. Hansen, "Mean Hilbert envelope coefficients (MHEC) for robust speaker and language identification," *Speech Communication,* 2015.

[17] R. Togneri and D. Pullella, "An overview of speaker identification: Accuracy and robustness issues," *Circuits and Systems Magazine, IEEE,* vol. 11, pp. 23-61, 2011.

[18] N. Dave, "Feature extraction methods LPC, PLP and MFCC in speech recognition," *International Journal for Advance Research in Engineering and Technology,* vol. 1, pp. 1-4, 2013.

[19] A. V. Oppenheim, R. W. Schafer, and J. R. Buck, "Discrete-Time Signal Processing," 1999.

[20] Q. Zhu and A. Alwan, "Non-linear feature extraction for robust speech recognition in stationary and non-stationary noise," *Computer speech & language,* vol. 17, pp. 381-402, 2003.

[21] I. Mporas, T. Ganchev, M. Siafarikas, and N. Fakotakis, "Comparison of speech features on the speech recognition task," *Journal of Computer Science,* vol. 3, pp. 608-616, 2007.

[22] L. R. Rabiner and R. W. Schafer, *Digital processing of speech signals*: Prentice Hall, 2010.

[23] X. Zhao and D. Wang, "Analyzing noise robustness of MFCC and GFCC features in speaker identification," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, 2013, pp. 7204-7208.

[25] X. Huang, A. Acero, H.-W. Hon, and R. Foreword By-Reddy, *Spoken language processing: A guide to theory, algorithm, and system development*: Prentice Hall PTR, 2001.

[26] J. Hai and E. M. Joo, "Improved linear predictive coding method for speech recognition," in *Information, Communications and Signal Processing, 2003 and Fourth Pacific Rim Conference on Multimedia. Proceedings of the 2003 Joint Conference of the Fourth International Conference on*, 2003, pp. 1614-1618.

[27] S. S. Nidhyananthan and R. S. S. Kumari, "Text independent voice based students attendance system under noisy environment using RASTA-MFCC feature," in *Communication and Network Technologies (ICCNT), 2014 International Conference on*, 2014, pp. 182-187.

[28] S. Tabibian, A. Shokri, A. Akbari, and B. Nasersharif, "Performance evaluation for an HMM-based keyword spotter and a Large-margin based one in noisy environments," *Procedia Computer Science,* vol. 3, pp. 1018-1022, 2011.

[29] J. A. Bilmes, "A gentle tutorial of the EM algorithm and its application to parameter estimation for Gaussian mixture and hidden Markov models," 1998.

[30] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proceedings of the IEEE,* vol. 77, pp. 257-286, 1989.

[31] J. A. Bilmes, "A gentle tutorial of the EM algorithm and its application to parameter estimation for Gaussian mixture and hidden Markov models," 1998.

[32] S. Binte Zinnat, R. M. A. Siddique, M. I. Hossain, D. Md Abdullah, and M. N. Huda, "Automatic word recognition for bangla spoken language," in *Signal Propagation and Computer Technology (ICSPCT), 2014 International Conference on*, 2014, pp. 470-475.

[33] J. G. Wilpon, L. R. Rabiner, C.-H. Lee, and E. Goldman, "Automatic recognition of keywords in unconstrained speech using hidden Markov models," *Acoustics, Speech and Signal Processing, IEEE Transactions on,* vol. 38, pp. 1870-1878, 1990.

[34] I. Bhardwaj and N. D. Londhe, "Hidden Markov Model based isolated Hindi word recognition," in *Power, Control and Embedded Systems (ICPCES), 2012 2nd International Conference on*, 2012, pp. 1-6.

[35] S. Sunil, S. Palit, and T. Sreenivas, "HMM based fast keyworld spotting algorithm with no garbage models," in *Information, Communications*

and Signal Processing, 1997. ICICS., Proceedings of 1997 International Conference on, 1997, pp. 1020-1023.

[36] R. Rose, "Keyword detection in conversational speech utterances using hidden Markov model based continuous speech recognition," *Computer Speech & Language,* vol. 9, pp. 309-333, 1995.

[37] J. Junkawitsch, L. Neubauer, H. Höge, and G. Ruske, "A new keyword spotting algorithm with pre-calculated optimal thresholds," in *Spoken Language, 1996. ICSLP 96. Proceedings., Fourth International Conference on*, 1996, pp. 2067-2070.

[38] Mark Stamp, "A Revealing Introduction to Hidden Markov Models", pp.1-20, September 28, 2012.

[39] Dawei Shen, "Some Mathematics for HMM", pp.1-9, October 13th, 2008.

[۴۰] ک. شفاعی, مرجع کامل پردازنده های سری ۲۰۰۰ و ۵۰۰۰ و ۶۰۰۰. تهران: انتشارات آستان قدس, ۱۳۸۹.

[۴۱] علی رجائیان، طراحی، شبیه سازی و پیاده سازی سخت افزاری روشی جهت تعیین سطح هوشیاری رانندگان خودرو با استفاده از سیگنال‌های مغزی EEG و مبتنی بر پردازشگرهای سیگنال TMS320C55X، دانشگاه صنعتی شاهرود، ۱۳۹۲.

[۴۲] هادی نادری، جستجوی کلیدواژه در زنجیره گفتار، دانشگاه صنعتی شاهرود، ۱۳۹۱.

[42] Michael Collins, " Notes on the EM Algorithm", pp.1-10, September 24th 2005.

[43] www.ti.com.

[44] T. Instruments, "TMS320C5509A Data Manual," ed, 2008.

[45] T. Instruments, "TPS767D301-EP Data Sheet," ed, 2010.

[46] T. Instruments, "TLV320AIC23B Data Manual," ed, 2004.

[47] "The DARPA TIMIT Acoustic-Phonetic Continuous Speech Corpus (TIMIT)," ed, updated 10/12/90.

[48] www.archive.ics.uci.edu/ml/datasets.

ABSTRACT

Keyword spotting is an important branch in speech signal processing which has many application in security and human-intraction-based systems. Literature review shows that the main effort has been done on improving the identification accuracy of spoken keywords not the hardware implementation of an approprate approach from aspects such as real-time processing, low power consumption, etc. The aim of this thesis is to hardware implementation of an approprate method based on the TMS320C55xx digital signal processors.

After reviewing the litrature , we identified the short-time spectral methods as approprate family and among which, we considered the MFCC and LPCC approaches to be implemented on hardware.

Simulation resulte show the MFCC approach result in better performance compared against the LPCC one. We simulated the automatic keyword spotting system in two modes of discrete and continuous speech. The average identification accuracy, when we used individually spoken numeral keywords, was computed as 93.33% and 81.58% in the discrete and continuous modes, respectively. In addition, the average identification accuracy, when we used keywords from spoken speech, was computed as 70.36% and 70.66% in the discrete and continuous modes, recpectively.

Keywords: keyword spotting, MFCC, LPCC, hidden markov model (HMM), TMS320C55xx digital signal processor.

University of Shahrood

Faculty of Electrical and Robotic Engineering

Hardware Implementation of a Keyword Spotting Algorithm on TMS320C55xx Platform

Mohammad Javadi

Supervisor:

Dr. Hadi Grailu

September 2015