نام و نام خانوادگی دانشجو: پانیذ اسلامی تمیجانی

شماره دانشجویی: ۴۰۱۱۴۱۴۰۱۱۱۰۲۶

کد پروژه : ۸۹

لینک پروژه: https://github.com/ShuhuaGao/vchsm

دانشکده فنی مهندسی واحد تهران جنوب

پاییز ۱۴۰۲-۱۴۰۱

توسعه یک سیستم تبدیل صوتی کارآمد محاسباتی در تلفن های همراه

هدف تبدیل صدا تغییر صدای گوینده منبع به گونه ای است که با حفظ اطلاعات زبانی، صدای گوینده هدف را شبیه صدای گوینده هدف کند. علیرغم پیشرفت سریع الگوریتم های تبدیل صدا در دهه گذشته، بسیاری از آنها هنوز برای عموم پیچیده هستند. با محبوبیت دستگاه های تلفن همراه به ویژه تلفن های هوشمند، برنامه های تبدیل صدای موبایل بسیار مطلوب هستند به طوری که همه می توانند از لذت تقلید صوتی با کیفیت بالا لذت ببرند و افراد مبتلا به اختلالات گفتاری نیز می توانند به طور بالقوه از آن بهره مند شوند. با توجه به محدودیت منابع محاسباتی در تلفن های همراه، نگرانی اصلی این است که بازده زمانی چنین اپلیکیشن موبایلی برای تضمین تجربه کاربر مثبت باشد. در این مقاله، توسعه یک سیستم تبدیل صدای تلفن همراه بر اساس مدل مخلوط گاوسی (GMM)و روش‌های تاب برداشتن فرکانس وزنی را شرح می‌دهیم. ما سعی می‌کنیم با استفاده از بهترین ویژگی‌های سخت‌افزاری تلفن‌های همراه امروزی، مانند محاسبات موازی بر روی چندین هسته و پشتیبانی پیشرفته بردارسازی، کارایی محاسباتی را افزایش دهیم. نتایج ارزیابی تجربی نشان می‌دهد که سیستم ما می‌تواند به عملکرد تبدیل صوتی قابل قبولی دست یابد، در حالی که زمان تبدیل برای یک جمله پنج ثانیه‌ای فقط کمی بیشتر از یک ثانیه در iPhone 7 طول می‌کشد.

مقدمه

تبدیل صدا به تغییر صدای یک گوینده که گوینده منبع نامیده می شود، اشاره دارد تا صدا را طوری ایجاد کند که گویی توسط گوینده دیگری به نام گوینده هدف بیان شده است، در حالی که محتوای زبانی بدون تغییر باقی می ماند. بنابراین، ویژگی‌های صوتی گوینده منبع باید توسط یک سیستم تبدیل صدا برای تقلید از گوینده هدف، بدون تغییر پیام ارسال شده در گفتار، شناسایی و تبدیل شود.

تبدیل صدا پتانسیل زیادی در توسعه کاربردهای صنعتی مختلف دارد، از جمله نه تنها ادغام در سیستم‌های سنتز متن به گفتار (TTS) برای سفارشی کردن صدا همانطور که می‌خواهیم [۱]، بلکه نقشی در پزشکی توانبخشی نیز ایفا می‌کند. مناطق سرگرمی و آموزشی. به عنوان مثال، با تبدیل مصوت های یک گوینده مبتلا به دیس آرتری، یک اختلال حرکتی گفتاری، به فضای مصوت یک گوینده غیر دیسارتری، درک گفتار به طور قابل توجهی بهبود یافته است [۲]. علاوه بر این، فناوری‌های تبدیل صدا نیز برای تولید صداهای چند خواننده متنوع با استفاده از یک مدل تبدیل به یک خواننده تک استفاده شده است.

پایگاه داده [۳]، برای ترکیب گفتار احساسی از گفتار خواندن استاندارد (خنثی) [۴] یا تبدیل احساسات منتقل شده در یک گفتار به دیگری [۵]، و تولید نسخه های تصحیح شده عروضی از گفته های زبان آموزان زبان خارجی در آموزش تلفظ به کمک کامپیوتر [۶، ۷]. برعکس، تبدیل صدا ممکن است تهدیدی برای تأیید صوت بلندگو

باشد و در نتیجه باعث افزایش ظرفیت ضد جعل سیستم‌های تأیید بلندگوی معمولی شود [۸]. در مجموع، سزاوار تلاش ما برای مطالعه تبدیل صدا برای اهداف تحقیقات علمی و کاربردی صنعتی است.

به طور کلی، گفتار انسان را می توان به سه جزء تقسیم کرد: محتوای زبان، الگوی طیفی و عروض [۸]. دو عامل آخر تمرکز مطالعات تبدیل صدای فعلی را تشکیل می‌دهند که در دو سطح واقع شده‌اند: سطح فوق‌بخشی، که شامل ویژگی‌های عروضی مانند خطوط فرکانس اساسی، مدت زمان کلمات، واج‌ها، زمان‌بندی، ریتم و سطوح شدت و غیره است. ; و سطح سگمنتال شامل شدت گام متوسط، پاسخ فرکانسی دستگاه صوتی، و ویژگی های منبع گلوتال. از این نظر، تبدیل صدا با هدف نگاشت ویژگی های طیفی و عروضی گوینده مبدا به گوینده هدف به منظور اصلاح فردیت صدا است. چنین وظیفه نگاشت ویژگی معمولاً به عنوان یک مشکل یادگیری نظارت شده در ادبیات فرموله می شود که به مقدار معینی از داده های گفتاری در دسترس هم از گوینده منبع و هم از گوینده هدف برای آموزش نیاز دارد. تفاوت کلیدی بین انواع مختلف مطالعات در مدل‌های نقشه‌برداری خاص اتخاذ شده، عمدتاً شامل روش‌های مبتنی بر کتاب کد، آماری و مبتنی بر شبکه‌های عصبی مصنوعی است که در یک بررسی اخیر به طور کامل بررسی شده‌اند [۹]. در ادامه، ما فقط برخی از رویکردهای نماینده را به طور مختصر مورد بحث قرار می دهیم تا عقلانیت روش شناسی خود را توضیح دهیم.

یک سیستم تبدیل صدای معمولی به دو نوع مدل نیاز دارد، یعنی مدل گفتار و مدل تبدیل. پس از بررسی‌های گسترده، مشخص شد که سه ویژگی گفتار مرتبط با فردیت گوینده عبارتند از: طیف متوسط، شکل‌دهنده‌ها و سطح متوسط گام [۶]. در نتیجه، بیشتر سیستم‌های تبدیل صدا تلاش می‌کنند تا پوشش‌های طیفی کوتاه‌مدت و ویژگی‌های عروضی از جمله مقدار زیر و بم، مدت و شدت را تغییر دهند [۹]. بنابراین، قبل از تبدیل صدا، یک مدل گفتار مناسب باید برای تجزیه و تحلیل سیگنال‌های گفتاری ورودی اتخاذ شود تا ویژگی‌های گفتاری مربوطه برای اصلاحات بعدی استخراج شود. علاوه بر این، یک مدل گفتار خوب باید بتواند سیگنال گفتار را از پارامترهای مدل با کیفیت بالا بازسازی کند، ظرفیتی که ما به آن نیاز داریم تا پس از تبدیل، گفتار را برای گوینده هدف ترکیب کنیم. معمولاً مدل گفتار فریم به فریم ساخته می شود در حالی که هر فریم یک بخش کوتاه مدت (معمولاً کمتر از ۲۰ میلی ثانیه) را نشان می دهد.

مدل‌های گفتاری رایج عبارتند از مدل[10] STRAIGHT ، مدل هم‌پوشانی-افزودن گام-همگام (PSOLA) [11]، و مدل نویز به علاوه هارمونیک[12] (HNM) ، و غیره STRAIGHT .متعلق به خانواده مدل فیلتر است که فرض می کند گفتار با فیلتر کردن سیگنال تحریک با فیلتر مجرای صوتی مستقل از تحریک تولید می شود. برای کاهش تداخل بین تناوب سیگنال و پوشش طیفی، STRAIGHTیک تحلیل طیفی تطبیقی را انجام می دهد، که می تواند جزئیات سطوح فرکانس زمانی را حفظ کند و در عین حال ساختارهای جزئی ناشی

از تناوب سیگنال را تقریباً کاملاً حذف کند. از سوی دیگر، روش‌های مبتنی بر سیگنال مانند PSOA و HNMگفتار را به‌عنوان ترکیبی از سیگنال منبع و فیلتر پوشش طیفی مدل‌سازی نمی‌کنند، در نتیجه از فرضیات محدودکننده مانند استقلال بین سیگنال منبع و فیلتر اجتناب می‌کنند. معمولا منجر به کیفیت بالاتر سنتز گفتار می شود. به طور خاص، روش‌های PSOLA می‌توانند سیگنال گفتار را در حوزه فرکانس (FD-PSOLA) یا مستقیماً در حوزه زمان (TD-PSOLA) تغییر دهند. به ویژه، TD-PSOLAشکل موج گفتار را به جریانی از سیگنال‌های تحلیل کوتاه‌مدت تنظیم می‌کند که با نرخ همگام گام تنظیم شده‌اند، که یکی از محبوب‌ترین و ساده‌ترین روش‌ها برای اصلاح عروضی با کیفیت بالا است. با این وجود، در جنبه تبدیل صدا، مدل‌های مبتنی بر تجزیه سینوسی گفتار مطلوب‌تر هستند، زیرا چنین مدل‌هایی با دستکاری پارامترهای مدل، تغییرات طیفی و عروضی انعطاف‌پذیر را امکان‌پذیر می‌کنند. به عنوان یک نماینده معمولی، HNMیکی از پرکاربردترین مدل‌ها برای سنتز و اصلاح گفتار است HNM .[12] سیگنال گفتار را به یک بخش قطعی به عنوان مجموع سینووئیدها با فرکانس های مربوط به زیر و بم و یک بخش تصادفی که با فیلتر کردن نویز گاوسی سفید به دست می آید، تجزیه می کند. یک مطالعه مقایسه ای در [۱۳] نشان می دهد که HNM عملکرد کلی بهتری نسبت به-TD PSOLAو در نتیجه یک مدل گفتاری مبتنی بر ارائه می دهد.

در چنین تجزیه تصادفی هارمونیک به علاوه در سیستم ما به تصویب رسید.

برگرفته از مرحله مدل‌سازی و تحلیل گفتار فوق، ویژگی‌های خروجی ممکن است مستقیماً مورد استفاده قرار گیرند یا برای نقشه‌برداری ویژگی بعدی پردازش شوند [۹]. همانطور که قبلا ذکر شد، در تبدیل صدای فعلی، اکثر روش ها پردازش فریم به فریم را فرض می کنند و بنابراین عمدتاً به ویژگی های گفتار محلی موجود در بخش های کوتاه مدت بستگی دارند. ویژگی های طیفی رایج عبارتند از ضرایب(Mel-cepstral (MCCs ، ضرایب پیش بینی خطی(Cepstral (LPCCs ، فرکانس طیف خط (LSF) و فرکانسformant ، و پهنای باند [۸، ۹]. با توجه به ویژگی های عروضی، ما معمولاً فقط الگوهای f0 و مدت زمان را در نظر می گیریم. سیستم‌های تبدیل صدای معمولی فقط تغییرات ساده‌ای از ویژگی‌های عروضی را انجام می‌دهند، به عنوان مثال، عادی‌سازی میانگین جهانی و واریانس مقادیر f0 با مقیاس ورود به سیستم [۸، ۱۴]، زیرا عروض یک ویژگی فوق‌بخشی است که بر اساس بخش‌بندی منتقل نمی‌شود، اما از طریق واحدهای بزرگتر، که چالش برانگیزتر است [۱۵]. در نتیجه، بیشتر تحقیقات بر روی نقشه برداری برای ویژگی های طیفی تمرکز دارند. از نظر ریاضی، تبدیل صدا عبارت است از یادگیری تابع نگاشت f (۰) از ویژگی گفتار مبدأ x به ویژگی گفتار هدف y با استفاده از پیکره آموزشی و سپس اعمال این تابع به یک نمونه گفتار منبع نادیده جدید برای تبدیل در زمان اجرا

شروع از کار اصلی مبتنی بر کمی سازی برداری (VQ) و کتاب های کد نگاشت توسعه یافته توسط آبه و همکاران. در سال ۱۹۸۸ [۱۶]، روش های بسیاری برای نگاشت ویژگی های طیفی در ادبیات ارائه شده است. اگرچه روش‌های مبتنی بر نگاشت کتاب کد ساده و از نظر محاسباتی کارآمد هستند، اما به دلیل تولید توالی ویژگی‌های ناپیوسته، عملکرد ضعیفی داشتند [۹، ۱۵]. در حال حاضر، روش‌های رایج‌تر عمدتاً در دسته‌های زیر طبقه‌بندی

می‌شوند: (۱) ترکیبی از نگاشت‌های خطی، مانند مدل‌های مخلوط گاوسی13] (GMM) ، ۱۷، ۱۸ [و مدل‌های پنهان مارکوف19] (HMM) ، ۲۰ (2) .[یک مدل نگاشت غیرخطی منفرد، شامل رگرسیون ماشین بردار پشتیبان [۲۱]، شبکه های عصبی مصنوعی22] (ANN) ، ۲۳[، و رویکردهای جدیدتر یادگیری عمیق [۲۴، ۲۵]. (۳) رویکردهای نقشه برداری مبتنی بر تاب فرکانس مانند تاب برداشتن فرکانس وزنی [۲۶]، تاب برداشتن فرکانس دینامیکی [۲۷] و تاب برداشتن فرکانس دوخطی [۲۸]. و (۴) روش های نگاشت ناپارامتریک مانند روش مبتنی بر مثال [۲۹، ۳۰]، رگرسیون فرآیند گاوسی [۳۱] و رویکرد هیستوگرام .[32] K در روش‌های مبتنی بر GMM، توزیع بردارهای ویژگی طیفی منبع (هدف) با گروهی از توزیع‌های نرمال چند متغیره مدل‌سازی می‌شوند، و تابع تبدیل معمولاً یک فرم خطی را اتخاذ می‌کند و بنابراین می‌تواند مستقیماً با حداقل مربعات حل شود [۱۳, ۱۵]. روش دیگر برای انجام نقشه برداری در GMM این است که بردارهای ویژگی منبع و هدف را با هم ترکیب کنیم تا یک GMM برای بردار تقویت شده z = [xT, yT]T که GMM مشترک نامیده می شود و بردار هدف نگاشت شده در طول تبدیل می تواند بسازد. با استفاده از بردارهای میانگین و ماتریس های کوواریانس که از آموزش به دست آورده ایم [۳۳] به دست می آید. در رویکردهای مبتنی برANN ، شبکه‌های کم عمق و عمیق برای نقشه‌برداری غیرخطی ویژگی‌ها از منبع به مقصد مورد بررسی قرار گرفته‌اند که ممکن است به دلیل ساختار بسیار منعطف و توانایی برازش غیرخطی برجسته به ویژه زمانی که از یک شبکه عصبی عمیق (DNN) استفاده می شود [۲۳، ۳۴]. با این وجود، طراحی و آموزش شبکه به تخصص زیادی نیاز دارد، زیرا فرآیند یادگیری به راحتی می تواند در حداقل های محلی گیر کند و DNN به طور کلی حجم زیادی از داده های آموزشی را درخواست می کند [۳۵]. مشخص است که روش های آماری مانند GMM تمایل به تولید گفتار بیش از حد صاف با کیفیت پایین دارند، اگرچه می توانند هویت گوینده را به خوبی تبدیل کنند [۲۷، ۳۰]. در مقابل، تاب فرکانس با هدف نگاشت محور فرکانس طیف بلندگوی منبع به بلندگوی هدف با تاب برداشتن پوشش طیفی منبع، در نتیجه حفظ جزئیات طیفی بیشتر و تولید کلام با کیفیت بالاتر به طور کلی [۳۰، ۳۶] است. با وجود امتیاز با کیفیت بالا، شباهت بین صداهای تبدیل شده و هدفمند، یعنی عملکرد تبدیل هویت، در تاب خوردگی فرکانس رضایت بخش نیست، زیرا شکل طیفی به طور کامل اصلاح نشده است [۲۶]. یک روش جایگزین برای تولید گفتار با کیفیت بالا، تبدیل صدای غیرپارامتری مبتنی بر نمونه است. یک نمونه به عنوان بخشی از طیف‌نگار گفتار که فریم‌های متعدد را در بر می‌گیرد، تعریف می‌شود، در حالی که مجموعه وزن‌های ترکیبی خطی یک بردار فعال‌سازی را تشکیل می‌دهند. برای جلوگیری از اثر هموارسازی بیش از حد، بردار فعال سازی باید پراکنده باشد، که می توان آن را با فاکتورسازی ماتریس غیرمنفی (NMF) با محدودیت پراکنده یا دکانولوشن ماتریس غیرمنفی تخمین زد [۲۹]. این چارچوب بیشتر گسترش یافته است تا شامل ضریب فشرده سازی طیفی و یک تکنیک جبران باقیمانده [۳۰] شود. نتایج آزمایش بر روی پایگاه داده VOICES برتری روش مبتنی بر نمونه را در شباهت و کیفیت گفتار تبدیل شده نشان می دهد. با این حال، این رویکرد دارای معایب هزینه محاسباتی

بالا است که آن را در حال حاضر برای برنامه های کاربردی تلفن همراه نامناسب می کند. مسلماً GMM هنوز یکی از موفق ترین و پرکاربردترین مدل ها در سیستم های تبدیل صدای عملی است [۸، ۹، ۱۵، ۳۷]. در سیستم تبدیل صدای خود که برای تلفن‌های هوشمند طراحی شده است، ما از روش تاب برداشتن فرکانس وزنی استفاده کردیم که GMM را با تاب برداشتن فرکانس ترکیب می‌کند تا تعادل بهتری بین کیفیت گفتار و شباهت حاصل شود.[26]

در سال های اخیر، دستگاه های تلفن همراه، به ویژه تلفن های هوشمند و تبلت ها، در زندگی روزمره ما غالب شده اند. یک گزارش نشان می‌دهد که زمان صرف شده روی دستگاه‌های تلفن همراه بین سال‌های ۲۰۱۴ تا ۲۰۱۵ ۱۱۷ رشد داشته است، در حالی که استفاده کلی از برنامه‌های تلفن همراه (برنامه) به طور متوسط ۵۸ سال به سال افزایش یافته است [۳۸]. با این حال، اگرچه الگوریتم‌های زیادی با مجموعه داده‌های خاص در ادبیات پیشنهاد و تأیید شده است، همانطور که در بالا ذکر شد، تا آنجا که دانش ما وجود دارد، هنوز هیچ سیستم تبدیل صوتی با استفاده از چنین روش‌های پیشرفته‌ای وجود ندارد که بتواند روی دستگاه‌های تلفن همراه، به‌ویژه در تلفن‌های همراه کار کند. علاوه بر این، از طریق جستجو در فروشگاه اپل و گوگل پلی برای برنامه‌های فعلی با برچسب تغییر/تبدیل صدا، متوجه شدیم که اکثر برنامه‌های تغییر صدای موجود صرفاً سعی می‌کنند برخی از ویژگی‌های طیفی یا عروضی را بدون هیچ هدف خاصی تغییر دهند و فقط به تغییر صدا کمک کنند. صداهای ربات مانند را برای سرگرمی تولید کنید، مانند VoiceLab و .Voice Changer تنها چند برنامه عمداً برای تقلید صدای شخص از پیش تعریف شده دیگر توسعه یافته اند، مانندTrump Voice Changer ، که می تواند متن ورودی را به صورت ترامپ مانند بیان کند.

صدا، و برنامهCelebrity Voice Changer Lite ، که لیست ثابتی از افراد مشهور را به عنوان اهداف تبدیل صوتی شما ارائه می دهد. با این حال، طبق بررسی‌های کاربران، این برنامه‌های کاربردی تلفن همراه (برنامه‌ها) محدودیت‌های آشکاری از خود نشان می‌دهند، به عنوان مثال، گفته‌های تولید شده معمولاً غیرطبیعی، غیرمشابه و فاقد قابلیت درک به نظر می‌رسد. علاوه بر این، برخی از برنامه ها برای دسترسی به سرورهای آنلاین نیاز به اتصال اینترنت دارند. مهمتر از آن، هیچ برنامه‌ای از سفارشی‌سازی بلندگوهای هدف پشتیبانی نمی‌کند، بلکه فقط اهداف ثابتی را ارائه می‌کند، و ویژگی‌های کلیدی یک سیستم تبدیل صوتی عمومی را از دست می‌دهد. بنابراین، این به ما انگیزه می‌دهد تا با اجرای کارآمد یک الگوریتم مبتنی برGMM ، که شامل بهترین استفاده از دستورالعمل‌های برداری سخت‌افزار، محاسبات موازی بر روی تلفن‌های همراه، یک سیستم تبدیل صوتی کامل، آفلاین و بلادرنگ در تلفن‌های همراه ایجاد کنیم. هسته های متعدد و طراحی خوب معماری نرم افزار. یک برنامه iOSبه نام Voichap برای نشان دادن امکان‌پذیری تبدیل صدا در زمان واقعی با امکانات کامل در دستگاه‌های آیفون توسعه یافته است.

این مقاله به شرح زیر سازماندهی شده است. در بخش دوم، به طور خلاصه چارچوب الگوریتم زیربنایی برای تبدیل صدا در سیستم خود را معرفی می‌کنیم. در بخش سوم، ما اصول و تکنیک‌های اصلی را برای اجرای کارآمد الگوریتم‌های اصلی در تلفن‌های همراه شرح می‌دهیم. بخش چهارم بعدی معماری نرم افزار و توسعه یک برنامه کاربردی کاربردی موبایل برای پلتفرم iOS را ارائه می دهد. بخش V برنامه iOS را نشان می دهد و سپس اثربخشی و کارایی این برنامه به صورت تجربی ارزیابی می شود. در بخش ششم، بحث مختصری در مورد رویکردهای مبتنی بر یادگیری عمیق ارائه می‌کنیم. در نهایت، مقاله را در بخش هفتم به پایان می‌رسانیم.

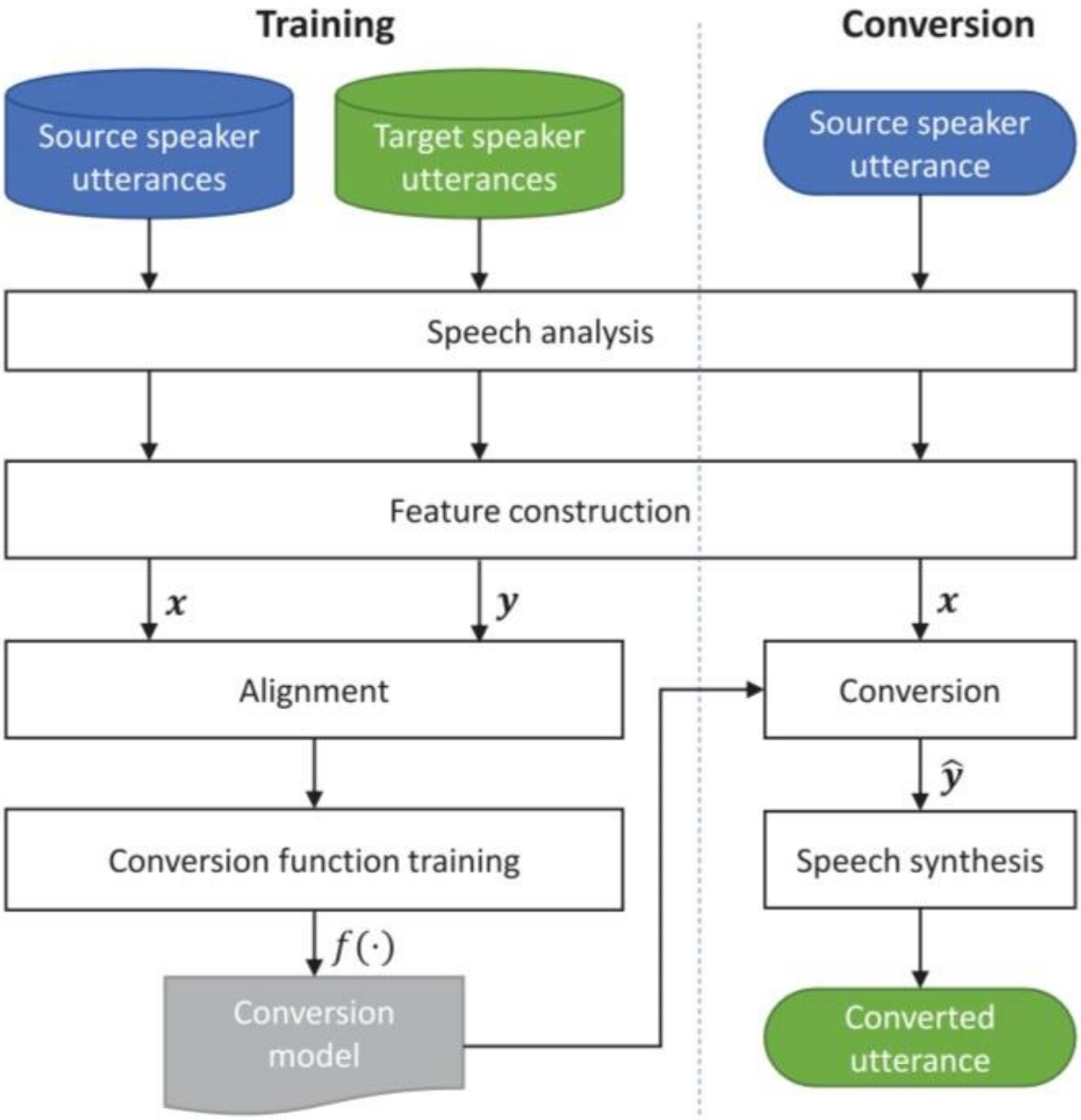
چارچوب تبدیل صدا

صدا، و برنامهCelebrity Voice Changer Lite ، که لیست ثابتی از افراد مشهور را به عنوان اهداف تبدیل صوتی شما ارائه می دهد. با این حال، طبق بررسی‌های کاربران، این برنامه‌های کاربردی تلفن همراه (برنامه‌ها) محدودیت‌های آشکاری از خود نشان می‌دهند، به عنوان مثال، گفته‌های تولید شده معمولاً غیرطبیعی، غیرمشابه و فاقد قابلیت درک به نظر می‌رسد. علاوه بر این، برخی از برنامه ها برای دسترسی به سرورهای آنلاین نیاز به اتصال اینترنت دارند. مهمتر از آن، هیچ برنامه‌ای از سفارشی‌سازی بلندگوهای هدف پشتیبانی نمی‌کند، بلکه فقط اهداف ثابتی را ارائه می‌کند، و ویژگی‌های کلیدی یک سیستم تبدیل صوتی عمومی را از دست می‌دهد. بنابراین، این به ما انگیزه می‌دهد تا با اجرای کارآمد یک الگوریتم مبتنی برGMM ، که شامل بهترین استفاده از دستورالعمل‌های برداری سخت‌افزار، محاسبات موازی بر روی تلفن‌های همراه، یک سیستم تبدیل صوتی کامل، آفلاین و بلادرنگ در تلفن‌های همراه ایجاد کنیم. هسته های متعدد و طراحی خوب معماری نرم افزار. یک برنامه iOSبه نام Voichap برای نشان دادن امکان‌پذیری تبدیل صدا در زمان واقعی با امکانات کامل در دستگاه‌های آیفون توسعه یافته است.

این مقاله به شرح زیر سازماندهی شده است. در بخش دوم، به طور خلاصه چارچوب الگوریتم زیربنایی برای تبدیل صدا در سیستم خود را معرفی می‌کنیم. در بخش سوم، ما اصول و تکنیک‌های اصلی را برای اجرای کارآمد الگوریتم‌های اصلی در تلفن‌های همراه شرح می‌دهیم. بخش چهارم بعدی معماری نرم افزار و توسعه یک برنامه کاربردی کاربردی موبایل برای پلتفرم iOS را ارائه می دهد. بخش V برنامه iOS را نشان می دهد و سپس اثربخشی و کارایی این برنامه به صورت تجربی ارزیابی می شود. در بخش ششم، بحث مختصری در مورد رویکردهای مبتنی بر یادگیری عمیق ارائه می‌کنیم. در نهایت، مقاله را در بخش هفتم به پایان می‌رسانیم.

۱.نمای کلی سیستم تبدیل صدا

یک نمای کلی از یک سیستم تبدیل صدای معمولی، شامل فاز آموزشی و فاز تبدیل در شکل ۱ نشان داده شده است. به طور کلی، در مرحله آموزش، تابع تبدیل f (·) برای ترسیم بردار ویژگی منبع x به بردار ویژگی هدف yسپس، در طول مرحله تبدیل، بردار ویژگی x یک گفته منبع جدید است.



تصویر۱: معماری کلی سیستم تبدیل صدا فاتیپیک

ساخته شده و متعاقباً توسط ŷ=f(x)تغییر شکل داده است، که در نهایت برای سنتز گفتار تبدیل شده در صدای

گوینده مورد نظر استفاده شد. در اصطلاحات یادگیری ماشین، چنین تبدیل صدا یک مشکل رگرسیونی است، اما به طور کلی یک فرآیند چالش برانگیز است زیرا کیفیت گفتار و تشابه تبدیل تبدیل به هدف معمولاً دو هدف متناقض هستند و باید تعادل مناسبی حاصل شود [۱۵]. . در بخش‌های فرعی زیر، مربوط به بلوک‌های شکل ۱، مدل‌سازی گفتار را برای تجزیه و تحلیل گفتار، ساخت ویژگی و یادگیری تابع تبدیل به طور جداگانه شرح می‌دهیم.

۲.مدل قطعی و تصادفی

در یک سیستم تبدیل صدا، یک مدل گفتار ابتدا باید برای اهداف سنتز مناسب باشد، به عنوان مثال، وقتی سیگنال گفتاری از پارامترهای مدل بازسازی می‌شود، باید وفاداری را حفظ کند تا از نظر ادراکی از مدل اصلی قابل تشخیص نباشد TD-PSOLA .فوق الذکر یک روش قدرتمند برای سنتز گفتار است [۱۱]. با این حال، برای تبدیل صدا مناسب نیست زیرا هیچ مدلی برای سیگنال گفتار در نظر نمی گیرد و در نتیجه هیچ دستکاری طیفی را نمی توان به راحتی اعمال کرد. بنابراین، سیستم ما از مدل قطعی به علاوه تصادفی بر اساس تجزیه سینوسی گفتار استفاده می کند [۳۹]. مشابه HNM کلاسیک [۱۲]، این مدل سیگنال گفتار s را به عنوان مجموع مجموعه‌ای از سینوسی‌ها با پارامترهای متغیر با زمان و یک جزء نویز مانند نشان می‌دهد که باقیمانده را نشان می‌دهد.

(1)

$$s[n] = \sum_{j=1}^{J} A_j[n] \cos\big(\theta_j[n]\big) + e[n]$$

جایی که قسمت قطعی فقط در قطعات صدادار ظاهر می شود و قسمت تصادفی شامل اجزای سیگنال غیر سینوسی باقی می متنده مانند اصطحکاک و صدای تنفس اگرچه پارامترهای مدل در سطح جهانی متغیر هستند، اما می‌توان آن‌ها را در فواصل زمانی کوتاه ثابت در نظر گرفت، و تحلیل محلی سیگنال توسط فریم‌ها را منطقی و آسان‌تر می‌کند، جایی که هر فریم با تعداد ثابتی از نمونه‌های گفتاری مطابقت دارد، مثلاًN ، در یک فاصله زمانی ۱۰ میلی ثانیه در مورد ما.

پس از تشخیص دامنه ها و فازها [۴۰] برای هر نقطه اندازه گیری k مربوط به لحظه زمانیkN ، $k \geq 1$، شکل موج قطعی در هر لحظه زمانی توسط

(2)

$$A_j[KN + m] = A_j^{(k)} + \frac{A_j^{(k+1)} - A_j^{(k)}}{N} m$$

از m= 0,1, ….., N-1 تا جایی که $A_j^{(k)}$ دامنه هارمونیکj را در نقطه k نشان می دهد.فاز ها و فرکانس ها توسط پلی سه مرتبه درون یابی میشود به شرح زیر:

(3)

$$\theta_j[KN+m] = am^3 + bm^2 + cm + d$$

که در آن پارامترهایa ، b، c، dبه صورت بهینه انتخاب می شون روش [۴۱]. اکنون با سینوسی های موجود، می توانیم بخش قطعی کامل d[n] را مطابق با آن بدست آورید معادله (۱) و مولفه تصادفی e[n] می تواند باشد متعاقباً به صورت زیر جدا شد

(4)

$$d[n] = \sum_{j}^{J^{(K)}} A_j[n]\cos\left(\theta_j[n]\right)$$

$$\boldsymbol{e[n] = s[n] - d[n]}$$

که در آن J (k) تعداد هارمونیک ها در قاب k ام است. کار باقی مانده آنالیز شکل طیفی قدر باقیمانده با روش کدگذاری اعتباری خطی (LPC) است. پس از به دست آوردن مدل قطعی به علاوه تصادفی (۱)، عروض شامل مدت و گام را می توان به طور مستقیم تغییر داد، که در [۴۱] به تفصیل آمده است. برای بازسازی سیگنال از پارامترهای اندازه گیری شده، می توان از تکنیک همپوشانی-افزودن (OLA) برای بازسازی بخش قطعی استفاده کرد، جایی که یک قاب از نمونه های ۲ Nدر آن ساخته شده است. هر نقطه اندازه گیری k که توسط

(5)

$$d[kN+m] = \sum_{j=1}^{J^{(k)}} \left(A_j^{(K)} \cos\left(w_j^{(k)} + \varphi_j^{(k)}\right)\frac{N-m}{N} + A_J^{(K+1)}\left(w_j^{(k+1)}(m-N) + \varphi_j^{(K+1)}\right)\frac{m}{N}\right)$$

در مرحله بعد، فریم های طول N نویز گاوسی سفید در حوزه فرکانس توسط فیلترهای LPC محاسبه شده قبلی برای تولید مولفه تصادفی شکل می گیرند. به این معنی، سیگنال گفتار را می توان با موفقیت از پارامترهای مدل سنتز کرد. نتایج تجربی نشان می دهد که خروجی سیستم از نظر ادراکی از گفتار اصلی قابل تشخیص نیست.

۳.فرکانس های طیفی خط دارای ویژگی ساختاری

اگرچه ما یک مدل هارمونیک به علاوه تصادفی ساخته‌ایم، باید توجه داشت که تبدیل صداها مستقیماً از این پارامترهای مدل بسیار دشوار است، زیرا دامنه‌ها و فازها پارامترسازی مناسبی از پوشش طیفی هارمونیک برای هدف صدا ارائه نمی‌کنند. تبدیل. این عمدتاً به دلیل این واقعیت است که تعداد هارمونیک ها متغیر و به طور کلی زیاد است که تبدیل را بسیار پیچیده می کند. بنابراین، همانطور که در شکل ۱ نشان داده شده است، یک مرحله ساخت ویژگی بیشتر مورد نیاز است تا امکان نمایش بهتر گفتار برای هدف تبدیل فراهم شود. در این مطالعه، ضرایب فرکانس‌های طیفی خطی (LSF) که دارای ویژگی‌های کوانتیزه‌سازی و درون‌یابی بهتری هستند [۴۲]،

به‌عنوان ویژگی‌هایی برای تخمین پارامترهایαi} ، μi، {i مدل مخلوط گاوسی استفاده می‌شوند. که در زیربخش بعدی معرفی می شود.

با توجه به نمایش تمام قطبی مرتبه pth از طیف، ۱/A(z)، ضرایب LSF ریشه های چند جمله ای های مرتبه (p + 1)هستند که توسط

(6)

$$P(z) = A(z) + z^{-(p+1)}A(z^{-1})$$
$$\mathrm{Q(z)} = \mathrm{A(z)} - \mathrm{z^{-(p+1)}A^{(z^{-1})}}$$

که در آن P یک چند جمله ای پالندرومیک و Q یک چند جمله ای ضد پالیندروم است. توجه داشته باشید که ریشه های P(z) و Q(z) به صورت جفت متقارن روی دایره واحد قرار دارند، به این معنی که می توان آنها را کاملاً با فرکانس های مربوط به مکان ریشه ها مشخص کرد و فقط فرکانس های p/2 باید مشخص شوند. برای هر چند جمله ای ذخیره می شود. بنابراین، ویژگی LSF در مجموع دارای صفات p است.

با توجه به اجرای عملی، تکنیک تکراری مدلسازی تمام قطبی ((DAPاستفاده می شود زیرا منجر به اعوجاج کمتری می شود و بنابراین کیفیت ادراکی بهتری را نسبت به پیشینیان خود ارائه می دهد [۴۳]. در تعیین ترتیب بهینه برای فیلترهای تمام قطبی هارمونیک، باید یک معاوضه انجام شود زیرا فیلترهای مرتبه بالا وضوح بالاتر و کیفیت بالاتری را ارائه می دهند، در حالی که فیلترهای مرتبه پایین را می توان با اطمینان بیشتری تبدیل کرد. از طریق آزمون و خطا، فیلترهای تمام قطبی مرتبه چهاردهم = pبرای ارائه بهترین نتایج یافت می شوند.

۴.تبدیل صد از طریق تاب برداشتن فرکانس وزنی

از مجموعه آموزشی داده شده، به عنوان مثال، یک پیکره موازی شامل سخنان گوینده منبع و سخنران هدف، بردار ویژگی نهایی LSF را می توان به ترتیب به صورت x و y بدست آورد. پس از تراز زمانی، استفاده از GMM برای نشان دادن توزیع ویژگی معمول است. ما می توانیم از یک بردار تقویت شده z = [xT, yT]T برای ساختن یک GMM مشترک با مولفه های m استفاده کنیم.

(7)

$$p(z) = \sum_{i=1}^{m} \alpha_I \, N(z; \mu_i, \varepsilon_i)$$

جایی که m; و E; به ترتیب بردار و میانگین هستند ماتریس کوواریانس برای مولفه ith گاوسی و(a;) مثبت هستند و تا ۱ جمع می شوند μi .و iرا می توان به شکل بلوکی که مربوط به x و y داده شده توسط

(8)

$$\mu_i = \begin{bmatrix} \mu_i^x \\ \mu_i^y \end{bmatrix}, \sum_i = \begin{bmatrix} \varepsilon_i^{xx} & \varepsilon_i^{xy} \\ \varepsilon_i^{yx} & \varepsilon_i^{yy} \end{bmatrix}$$

که می توان از روی داده ها با استفاده از تکنیک انتظار- حداکثرسازی (EM) تخمین زد .در طول تبدیل، برای بردار ورودی منبع xt از قابt ، توزیع شرطی yt داده شده xt دوباره با توزیع گاوسی مخلوط مدل‌سازی می‌شود و می‌توانیم بردار میانگین را به عنوان بردار خروجی هدف پیش‌بینی‌شده ŷt استفاده کنیم. بنابراین، تابع تبدیل کلاسیک [15] GMM به صورت زیر فرموله شده است:

(9)

$$F(x_t) = \sum_{i=1}^{m} \omega_i(x_t)\left[\mu_i^y + \varepsilon_i^{yx}(\varepsilon_i^{xx})^{-1}(x_t - \mu_i^x)\right]$$

که در آن ωi (xt) احتمال خلفی است که LSF داده شده است بردار متعلق به ITH گاووسی است که توسط

(10)

$$\omega_i(x_t) = \frac{\alpha_i N(z; \mu_i, \varepsilon_i)}{\alpha_j N\left(z; \mu_j, \varepsilon_j\right)}$$

اگرچه روش سنتی GMM که در بالا توضیح داده شد می‌تواند به شباهت خوبی برای تبدیل صدا دست یابد، کیفیت گفتار تبدیل شده هنوز رضایت‌بخش نیست، عمدتاً به دلیل اثر هموارسازی بیش از حد [۱۵]. در مقابل، روش‌های مبتنی بر تاب‌خوردگی فرکانس می‌توانند از افت قابل‌توجه کیفیت گفتار جلوگیری کنند، زیرا درجه اصلاح محدود است [۳۵]. بنابراین، ما روشی را که در ابتدا در [۳۶] پیشنهاد شده بود، اتخاذ کردیم، که هدف آن

به دست آوردن تشابه تبدیل شده به هدف بالا در عین حفظ کیفیت گفتار، با ترکیب GMM و تاب برداشتن فرکانس است. الهام اصلی این است که میانگین بردارهای LSF هر کلاس آکوستیک مربوط به هر جزء گاوسی درGMM ، μixوμiy ، ساختار فرمانت بسیار مشابهی دارند. به دنبال این مشاهدات، یک تابع تاب خوردگی فرکانس خطی تکه‌ای Wi (f) با استفاده از موقعیت این شکل‌دهنده‌ها برای هر کلاس آکوستیک ایجاد می‌شود و تابع تاب فرکانس برای قاب کامل شامل m کلاس‌های صوتی به دست می‌آید.
متعلق به من گاوسی است که توسط
(11)

$$W(f) = \sum_{i=1}^{m} w_i(x) W_i(f)$$

با W(f) تعیین شده، با فرض اینکه A(f) و θ(f) اندازه و فاز تخمین‌زننده طیف قاب فعلی هستند، تغییرات طیفی را می‌توان با تاب برداشتن پوشش‌های منبع به شرح زیر انجام داد.
(12)

$$A_w(f) = A\big(W^{-1}(f)\big), \theta_w(f) = \theta\big(W^{-1}(f)\big)$$

حتی پس از به دست آوردن طیف تاب خورده جریان فریم$S_w(f)$مانند بالا، توزیع انرژی همچنان با صدای هدف واقعی متفاوت است، زیرا شدت، پهنای باند و شیب طیفی تقریباً بدون تغییر باقی می‌مانند. تابع تبدیل GMM (9)در اینجا برای به دست آوردن استفاده می شود نسخه جدید طیف هدف Sg (f) از بردار LSF تبدیل شده .F (x)طیف تبدیل نهایی برای فریم فعلی به دست می آید
(13)

$$S'^{(f)} = G(f) S_w(f)$$

که در آن فیلتر تصحیح انرژی G (f) توسط
(14)

$$G(f) = \left| \frac{S_g(f)}{S_w(f)} \right| * B(f)$$

که از تابع پنجره هموارسازی B(f) برای انجام کانولوشن (عملگر ⁕) استفاده می کند. تابع B(f) می تواند تعادل بین شباهت و کیفیت گفتار تبدیل شده را با تنظیم شکل آن کنترل کند. تابع صاف کردن مثلثی معمولاً در عمل استفاده می شود.[26]
برای تغییر بزرگی شکل‌دهنده‌ها، می‌توان پوشش تمام قطبی ŷ = F (x) تبدیل‌شده را به‌دست آورد و انرژی در باندهای خاص مورد علاقه را اندازه‌گیری کرد. انرژی قاب گفتار تبدیل شده در هر باند به سادگی با عوامل ضرب ثابت تصحیح می شود.

یک ویژگی مهم عروضی، فرکانس بنیادیf0 ، با تبدیل جهانی ساده اما محبوب تغییر یافته است [۳۵]. از آنجایی که f0 از توزیع -log نرمال پیروی می کند، میانگین μf0 و واریانس σf0 log f0 را می توان در طول آموزش محاسبه کرد. سپس، f0سیگنال گفتار تبدیل شده را می توان به صورت خطی با مقیاس بندی کرد (15)

$$logf_0^{(c)} = \mu_{f_0}^{(t)} + \frac{\sigma_{f_0}^{(t)}}{\sigma_{f_0}^{(s)}}\left(logf_0^{(s)} - \mu_{f_0}^{(s)}\right)$$

که در آن حروف بالا (s) و (t) نشان دهنده منبع و هدف به ترتیب، و f (c) فرکانس اساسی است گفتار تبدیل شده در رابطه با مولفه تصادفی در رابطه (۱)، آن است تبدیل شناخته شده است که به اندازه تبدیل هارمونیک / قطعی مرتبط نیست [۱۳]. با این وجود، بهتر است مولفه تصادفی را برای گوینده هدف با استفاده از پارامترهای LSF دستگاه صوتی در فریم‌های صوتی، همانطور که در [۳۶] شرح داده شده، پیش‌بینی کنیم. در این مرحله، ما شرح تئوری اصلی تبدیل صدا را برای سیستم خود به پایان رساندیم. در بخش بعدی، پیاده سازی بسیار کارآمد چنین الگوریتم هایی را شرح خواهیم داد.

پیاده سازی کارآمد الگوریتم های اصلی

در مقایسه با لپ‌تاپ‌ها و رایانه‌های شخصی (رایانه‌های رومیزی شخصی)، دستگاه‌های تلفن همراه، به عنوان مثال، تلفن‌های هوشمند، معمولاً با حافظه، قدرت و منابع محاسباتی کاملاً محدود مشخص می‌شوند. با این وجود، هدف اصلی مطالعه ما توسعه یک سیستم تبدیل صدا کامل بر روی تلفن های همراه بدون پشتیبانی سمت سرور است. بنابراین، چالش کلیدی اجرای الگوریتم تبدیل به اندازه کافی کارآمد است به طوری که زمان مورد نیاز برای تبدیل صوتی، به ویژه فاز تبدیل، برای استفاده روزانه قابل قبول باشد. برای رفع این مشکل باید دو نکته زیر را عمدتاً در نظر گرفت :کارایی الگوریتم انتظار می‌رود روش‌هایی که برای مدل‌سازی، سنتز و تبدیل گفتار انتخاب می‌شوند، در زمان خود صرفه‌جویی کنند. در بخش دوم، ما الگوریتم‌های درگیر برای سیستم تبدیل صدای خود را معرفی کرده‌ایم که تا حدی به دلیل بار محاسباتی کم و در عین حال عملکرد قابل قبول آن‌ها پذیرفته شده‌اند. به عنوان مثال، مدل قطعی به علاوه تصادفی که قبلا برای مدل‌سازی گفتار توضیح داده شد، در واقع ناهمزمان است، که می‌تواند تحلیل را تا حد زیادی ساده‌تر کند، زیرا جداسازی دقیق دوره‌های سیگنال ضروری نیست. مهمتر از آن، این الگوریتم تبدیل می تواند حتی با تعداد کمی از نمونه های آموزشی، به عنوان مثال، ده ها جمله، به خوبی کار کند

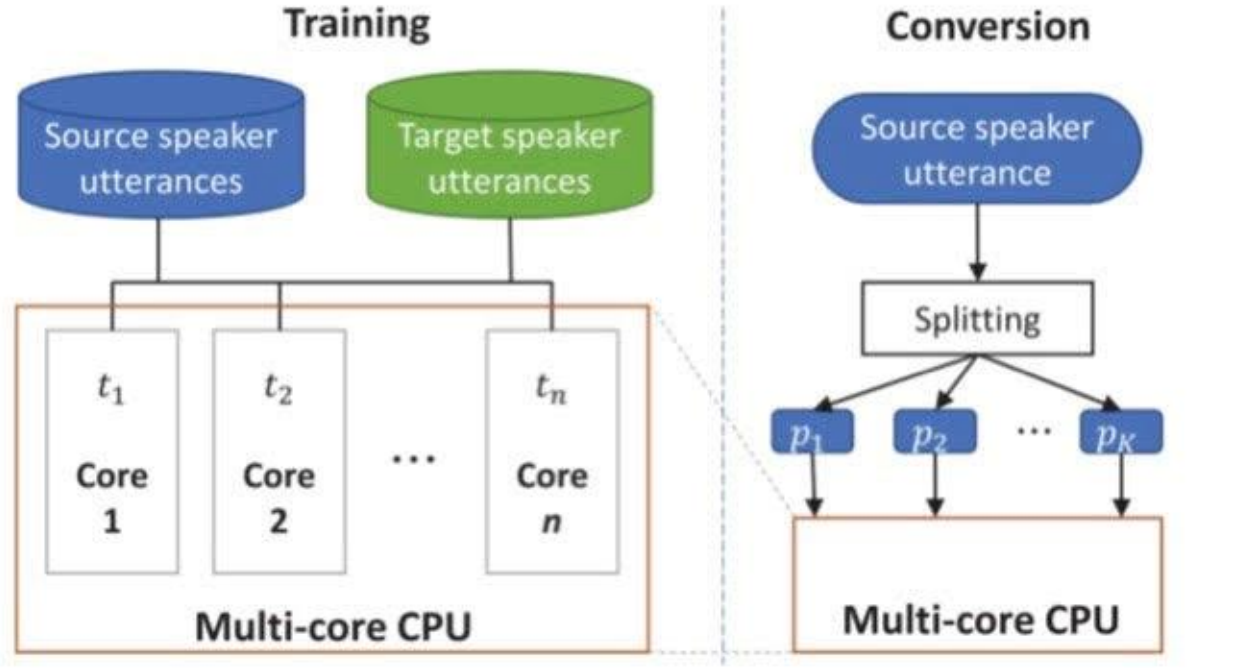
•کارایی پیاده سازی. این وظیفه ای است که ما در طول این مطالعه بیشتر بر آن تأکید می کنیم. به طور کلی، سخت افزار همیشه سریعتر از نرم افزار است. بنابراین، استفاده کامل از ویژگی های سخت افزاری پیشرفته موجود

در تلفن های همراه رایج، از جمله پشتیبانی از محاسبات برداری و محاسبات موازی چند هسته ای، حیاتی است. علاوه بر این، انتخاب یک زبان برنامه نویسی مناسب و دستکاری هوشمند ماتریس ها نیز نقش مهمی در شتاب بازی می کند.

در این بخش، جنبه‌های اجرایی کلیدی در رابطه با کارایی محاسباتی را به تفصیل شرح می‌دهیم. امروزه، ارائه یک CPU چند هسته ای، حتی برای تلفن های هوشمند ارزان قیمت، برای تلفن های هوشمند تقریباً استاندارد است. به عنوان مثال، آیفون ۷ که در سپتامبر ۲۰۱۶ عرضه شد، از Apple A10 Fusion 64بیتی سیستم روی تراشه (SoC) استفاده می کند که از دو هسته کم مصرف و دو هسته پرمصرف تشکیل شده است. به عنوان مثال دیگر، گوشی اندرویدی رده پایین Redmi 5 که در دسامبر ۲۰۱۷ توسط شیائومی عرضه شد، مجهز به تراشه اسنپدراگون ۴۵۰ با ۸ هسته است. بنابراین، برای استفاده کامل از ظرفیت محاسباتی تلفن‌های همراه، به‌ویژه پردازنده‌های چند هسته‌ای که به‌طور گسترده در تلفن‌های هوشمند در دسترس هستند، سیستم ما باید موازی‌سازی شود تا روی چندین هسته برای عملکرد تبدیل در زمان واقعی کار کند.

محاسبات موازی نوعی محاسبات است که در آن بسیاری از محاسبات یا اجرای فرآیندها به طور همزمان بر روی چندین هسته انجام می شود. به عبارت ساده، یک مسئله پیچیده بزرگ را می توان به مسائل کوچکتر و ساده تر تجزیه کرد، که می توان آنها را به هسته های مختلف اختصاص داد تا همزمان حل شوند و در نتیجه کل زمان محاسبات مورد نیاز را کاهش دهند [۴۴]. با توجه به محدودیت‌های فیزیکی که از مقیاس‌بندی فرکانس جلوگیری می‌کند، محاسبات موازی مورد توجه گسترده‌تری قرار گرفته و به پارادایم غالب در معماری کامپیوتر تبدیل شده است، همانطور که با محبوبیت پردازنده‌های چند هسته‌ای نشان داده شده است [۴۵]. برای مثال، پردازنده‌های چند هسته‌ای برای رمزگشایی کارآمد ویدیوها مورد سوء استفاده قرار گرفته‌اند [۴۶]. در سیستم تبدیل صدای ما، محاسبات عمدتاً در دو نقطه موازی می‌شوند، مرحله مدل‌سازی و تحلیل گفتار در طول آموزش و تبدیل، که در شکل ۲ نشان داده شده است.

موازی سازی در طول مرحله آموزش، همانطور که در سمت چپ شکل ۲ نشان داده شده است، واضح و ساده است.



تصویر۲: محاسبات موازی در مراحل آموزش و تبدیل

از آنجایی که تحلیل گفتار و ساخت ویژگی هر نمونه گفته مستقل است. بنابراین، می‌توانیم کل ۲ میلیون نمونه را از بلندگوی منبع و بلندگوی هدف بر روی هسته‌های C با ایجاد یک رشته ti بر روی هر هسته توزیع کنیم. به این ترتیب، در حالت ایده‌آل، نمونه‌های صوتی C می‌توانند همزمان بدون تداخل یکدیگر پردازش شوند. بنابراین، کل زمان اجرا بسیار کاهش می یابد زیرا زمان صرف شده توسط تجزیه و تحلیل گفتار و مهندسی ویژگی بخش بزرگی از کل هزینه زمان را اشغال می کند.

موازی کردن فاز تبدیل ظریف تر است. میانگین مدت جملات در مجموعه تقریباً ۴ یا ۵ ثانیه است، مشابه جملاتی که ما هر روز صحبت می کنیم. با توجه به ورودی گفته از بلندگوی منبع که باید تبدیل شود، ابتدا آن را به گروهی از پارتیشن‌های پیوستهpi ، i = 1،2،...، Kتقسیم می‌کنیم، که طول آن‌ها تقریباً برابر است، مثلاً، در حدود tpثانیه. سپس، مشابه مرحله آموزش، دوباره این پارتیشن های کوتاه را به هسته های مختلف برای پردازش همزمان اختصاص می دهیم، همانطور که در سمت راست در شکل ۲ نشان داده شده است. برای جبران مصنوعات احتمالی معرفی شده در مرز قطعه، همپوشانی اضافی وجود دارد. منطقه ای از مدت زمان تا ثانیه در هر طرف نقطه تقسیم بندی، نشان داده شده در شکل ۳ .(a)پس از هر پارتیشن پی به picما این بخش ها را با استفاده از یک تابع لجستیک در یک گفتار تبدیل شده کامل ادغام می کنیم تا به یک انتقال صاف در اطراف نقطه پارتیشن دست یابیم. تابع لجستیکی که برای انتقال هموار در اینجا استفاده می‌شود توسط داده می‌شود

(16)

$$\varphi(x) = \frac{1}{1 + e^{-kx}}$$

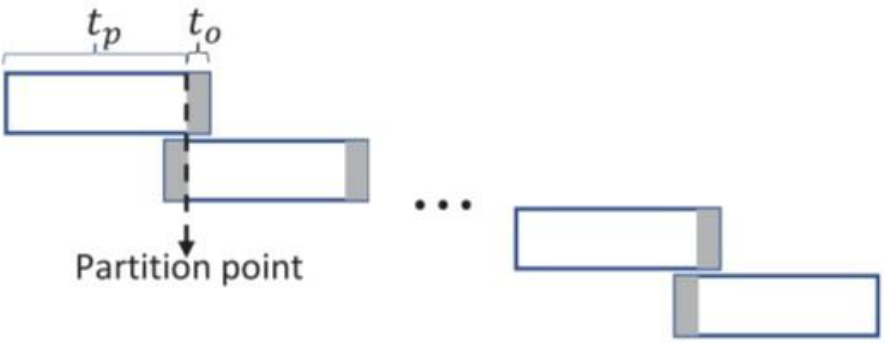
جایی که k شیب منحنی را تنظیم می کند.در اجرای فعلی ما، طول همپوشانی است 40میلی‌ثانیه، یعنی حدود ۶۴۰ نمونه. با فرض اینکه دو ناحیه همپوشانی اطراف نقطه تقسیم را از ۶۴۰- تا ۶۴۰ ایندکس کنیم و k = 0.015 را در تابع لجستیک (۱۶) که در شکل ۳ b) نشان داده شده است (انتخاب کنیم، نمونه در شاخص i دو نمونه متناظر را ادغام می کند .، s_i^l در پارتیشن سمت چپ و آقا در پارتیشن سمت راست، توسط

(17)

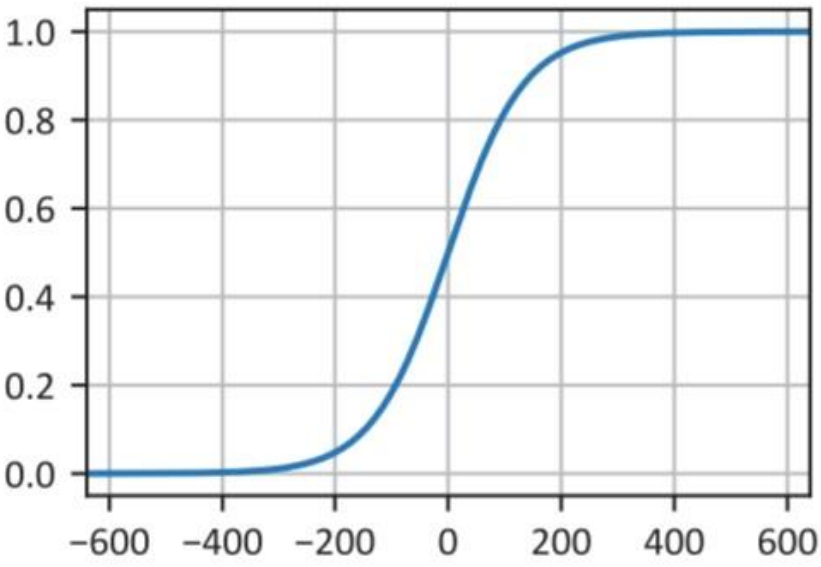
$$S_i = \big(1 - \varphi(i)\big)s_i^l + \varphi(i)s_i^r$$

اگرچه در اجرای فعلی ما عمدتاً مراحل تحلیل گفتار، ساخت ویژگی و تراز فریم را به دلیل مصرف زیاد آنها موازی

می‌کنیم، باید توجه داشت که موازی‌سازی بیشتر همچنان می‌تواند در مراحل بعدی مانند تخمین پارامترهای GMMاعمال شود. معادله (۸). به عنوان مثال، در کار[47] Wojciech Kwedlo ، یک موازی سازی حافظه مشترک از الگوریتم استاندارد EM بر اساس تجزیه داده ها برای یادگیری پارامترهای GMM در یک سیستم چند هسته ای برای عملکرد بالاتر پیشنهاد شده است. بنابراین، ما در حال برنامه ریزی برای پیاده سازی چنین ویژگی هایی برای تسریع بیشتر سیستم تبدیل صدای خود در نسخه بعدی آن هستیم.



Split a sentence for parallel conversion
(a)

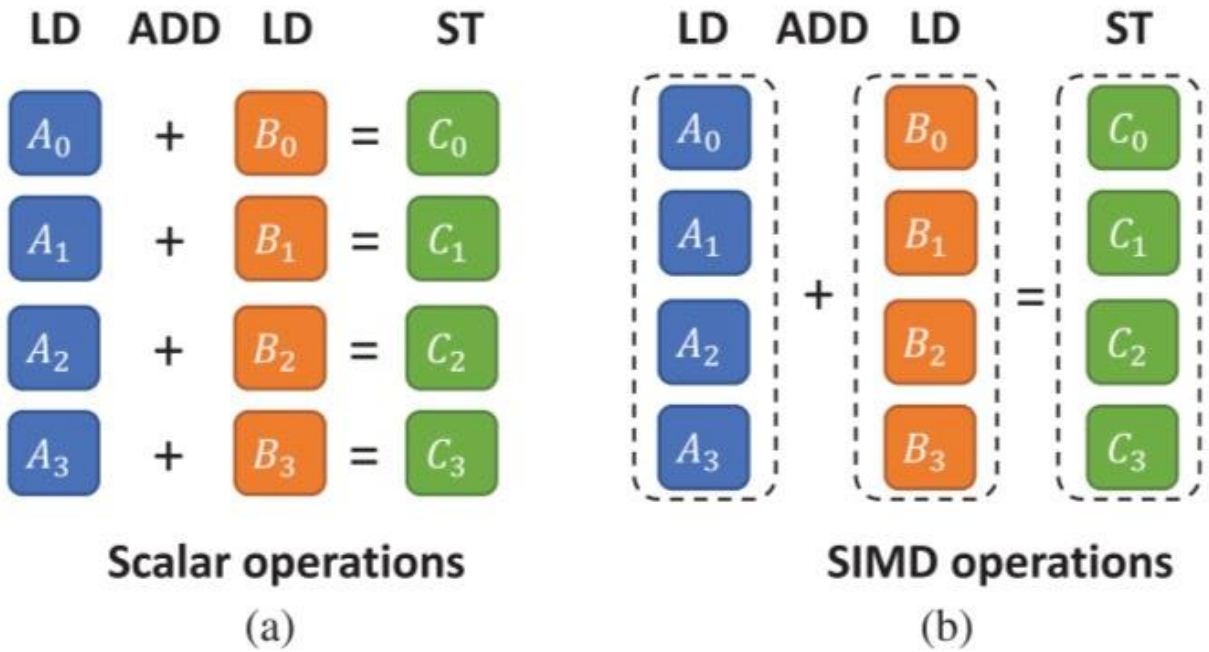


Logistic function for segment merging
(b)

تصویر ۳: موازی سازی چند هسته ای در مرحله تبدیل. اول، یک جمله ورودی به چند بخش تقسیم می شود و آن بخش ها به طور همزمان تبدیل می شوند. سپس، بخش های تبدیل شده با استفاده از یک تابع لجستیک برای مجموع وزنی به آرامی ادغام می شوند. (الف) یک جمله را برای تبدیل موازی تقسیم کنید (ب) تابع لجستیک برای ادغام بخش.

برداری از طریق SIMD

پس از اینکه کار محاسباتی روی چندین هسته گسترده شد، هدف بعدی ما این است که توان عملیاتی تک هسته‌ای را تا حد امکان بالا ببریم. مجدداً، به دلیل استفاده زیاد از ماتریس ها در الگوریتم های زیربنایی، ما باید به کاوش و استفاده کامل از ظرفیت سخت افزار، به ویژه موازی سازی درون هسته ای برای عملیات روی آرایه ها (بردارها) داده ها بپردازیم. در تلفن های همراه مدرن، موازی سازی درون هسته ای، یا بردارسازی، عمدتاً توسط SIMDپشتیبانی می شود که به طور گسترده در پردازنده های امروزی موجود است[44] ، ۴۸.[اکثر پردازنده های موبایل، از جمله پردازنده های آیفون و اندروید، با معماری ARM طراحی شده اند. به عنوان مثال، آیفون ۷ از سیستم روی تراشه ۶۴ بیتی Apple A10 Fusion استفاده می کند که مبتنی بر معماری ARMv8-A با پشتیبانی از دستورالعمل های ۶۴ بیتی است. به ویژه، پردازنده‌های مدرن موبایلARM ، مانند تمامی سری‌های Cortex-A8، معمولاً از افزونه‌های پیشرفتهSIMD ، معروف بهNEON ، پشتیبانی می‌کنند که یک مجموعه دستورات SIMD ترکیبی ۶۴ و ۱۲۸ بیتی است که شتاب استانداردی را برای پردازش رسانه و سیگنال ارائه می‌کند. برنامه های کاربردی NEON .[48] می تواند از اعداد صحیح ۸، ۱۶، ۳۲ و ۶۴ بیتی پشتیبانی کند.



تصویر ۴: عملیات اسکالر در مقابل SIMD برای افزودن های متعدد.

و داده های ممیز شناور تک دقیق (۳۲ بیتی) و عملیات SIMD تا ۱۶ عملیات را همزمان فراهم می کند. در یک مطالعه بر روی تثبیت‌کننده تصویر دیجیتال برای دستگاه‌های تلفن همراه، محاسبات بردار حرکت و محاسبات FFTبا استفاده از موتور نئون SIMD موجود در CPU ARM تسریع می‌شوند و بردار حرکت جهانی برای هر فریم را می‌توان در کمتر از ۲۰ میلی‌ثانیه محاسبه کرد [۴۹] . در معماری‌های x86/64 که معمولاً برای رایانه‌های شخصی استفاده می‌شود، پردازنده‌ها معمولاً مجموعه دستورالعمل‌های SIMD مبتنی بر SSE یا AVXرا ارائه می‌کنند.

به طور خلاصه، SIMDعملیات مشابهی را روی چندین عنصر داده از یک نوع که در یک بردار تراز شده اند به طور همزمان انجام می دهد. از نظر تئوری، اگر بتوانیم عناصر داده q را به طور کلی با یک SIMD پردازش کنیم، آنگاه سرعت افزایش در مقایسه با پردازش متوالی از طریق دستورالعمل های اسکالر به q نزدیک است، اگرچه سرعت واقعی به طور بالقوه توسط پهنای باند حافظه محدود می شود. شکل ۴ یک عملیات رایج در پردازش چند رسانه ای را نشان می دهد، که در آن مقدار یکسان به تعداد زیادی از عناصر داده اضافه می شود، که با دستورالعمل های اسکالر و دستورالعمل های SIMD به طور جداگانه برای مقاصد مقایسه اجرا می شوند. در زمینه پردازش صدا، داده های صوتی معمولاً در انواع اعداد صحیح ۱۶ بیتی ارائه می شوند. با فرض اینکه از رجیسترهای NEON 64بیتی برای برداری استفاده می کنیم، می توانیم چهار عدد صحیح ۱۶ بیتی را همزمان در این ثبات واحد بسته بندی کنیم، همانطور که در شکل ۴ (b) نشان داده شده است. فرآیندی که فقط با استفاده از دستورالعمل های اسکالر در شکل ۴ (a) نشان داده شده است. برای انجام یک جمع، ابتدا دو عملیات بارگذاری (LD)برای خواندن دو عدد A و B از حافظه اجرا می شود. سپس، یک افزودنی (ADD) برای به دست آوردن نتیجه C دستور داده می شود. در نهایت، مجموع C با یک دستور ذخیره (ST) به حافظه بازگردانده می شود. در مقابل، دستورالعمل‌های بارگذاری، افزودن و ذخیره‌سازی تقویت‌شده SIMD می‌توانند روی چهار عدد صحیح به طور همزمان اجرا شوند که در شکل ۴ (b) مشخص شده است. بنابراین، در مجموع، ۱۶ دستورالعمل اسکالر از طریق برداری SIMD به تنها چهار دستورالعمل کاهش می‌یابد که سرعت تئوری چهار را به دست می‌دهد.

برای استفاده از SIMD در برنامه نویسی عملی و توسعه نرم افزار، عمدتاً سه راه وجود دارد: (۱) کدگذاری در دستورالعمل های اسمبلی سطح پایین به طور مستقیم. (۲) استفاده از توابع ذاتی با رابط های C که کدهای اسمبلی را به صورت داخلی بسته بندی می کنند. و (۳) استفاده از بردارسازی خودکار در صورتی که توسط کامپایلرها پشتیبانی شود [۴۸]. به طور کلی، برنامه نویسی در زبان اسمبلی به طور مستقیم، علیرغم عملکرد بسیار زیاد، کار فشرده و بسیار مستعد خطا است. از سوی دیگر، بردارسازی خودکار می‌تواند به طور کامل توسط خود کامپایلر بدون نیاز به مداخله انسانی انجام شود. با این حال، در حال حاضر حتی کامپایلرهای پیشرفته هستند

به اندازه کافی هوشمند نیست، و در نتیجه، تنها بخش کوچکی از کدها می توانند به طور خودکار بردار شوند. بنابراین، مطالعه ما بر عملکردهای ذاتی به عنوان یک مبادله بین عملکرد سیستم و کار دستی متکی است.

شایان ذکر است که در یک پیاده سازی واقعی، وابستگی به توابع ذاتی لزوماً به این معنا نیست که ما باید آنها را مستقیماً توسط خودمان در برنامه نویسی فراخوانی کنیم. قابل توجه است که بخش عمده ای از الگوریتم های تجزیه و تحلیل گفتار و تبدیل در محاسبات ماتریسی فرموله می شوند. بنابراین، ما می‌توانیم به برخی از کتابخانه‌های جبر خطی بالغ متوسل شویم که رابط‌های سطح بالا را برای تسهیل توسعه برنامه‌ها در معرض دید قرار می‌دهند و در عین حال از توابع ذاتی برای بردار کردن محاسبات ماتریس/آرایه تا حد امکان استفاده می‌کنند. یعنی استفاده داخلی از دستورالعمل های SIMD برای کاربران تقریباً شفاف است. جزئیات مربوطه در زیر بخش بعدی توضیح داده شده است

در دو بخش فرعی بالا، موازی سازی سطح هسته و سطح دستورالعمل را برای تسریع الگوریتم های اصلی برای تبدیل صدا توضیح داده ایم. هدف از این مطالعه توسعه یک سیستم تبدیل صدا کارآمد در دستگاه های تلفن همراه است، نه محدود به یک نوع دستگاه خاص مانند گوشی هایiPhone ، iPadیا .Android اگرچه کیت‌های توسعه رابط کاربری گرافیکی در سیستم‌های عملیاتی مختلف تلفن همراه، مانند iOS برای آیفون و اندروید برای اکثر تلفن‌های هوشمند دیگر، تفاوت زیادی با یکدیگر دارند، اما ما می‌خواهیم اجرای الگوریتم اصلی این سیستم مستقل از پلتفرم باشد به طوری که این امر بسیار مهم باشد. بخش را می توان به راحتی در چندین پلت فرم با حداقل تغییرات منتقل کرد. از این رو، ما عملکرد اصلی و رابط کاربری گرافیکی را در معماری سیستم خود جدا می کنیم. برای برآوردن نیازهای بی طرفی پلت فرم و بازده بالا، زبان برنامه نویسی C++ بهترین گزینه برای کدگذاری الگوریتم های اصلی است. به عنوان یک زبان همه منظوره، C++عملکرد، کارایی و انعطاف‌پذیری را برجسته می‌کند، یعنی امکان دستکاری حافظه در سطح پایین را فراهم می‌کند و در عین حال انتزاعات سطح بالا را به شیوه‌ای شی گرا ارائه می‌کند. در زیر به طور خلاصه نحوه دستیابی به موازی سازی چند هسته ای و مبتنی بر SIMD در C++ را شرح می دهیم.

برای اعمال موازی‌سازی مبتنی بر رشته برای محاسبات چند هسته‌ای، یعنی برنامه‌نویسی چند رشته‌ای درC++ ، ساده‌ترین راه استفاده از OpenMP API استاندارد است که از برنامه‌نویسی موازی با حافظه مشترک چند پلتفرمی در C و C پشتیبانی می‌کند [50] C++ .تنها با استفاده از دستورالعمل های ساده. با این حال، OpenMPهنوز دریOS ، سیستم عامل تلفن همراه آیفون یا آی پد، پشتیبانی ضعیفی دارد. در عوض، همانطور که اپل توصیه می کند، باید از فناوری Grand Central Dispatch (GCD) برای موازی سازی وظایف استفاده کنیم، که به طور ویژه برای iOS طراحی شده و بسیار بهینه شده است. به طور خلاصه، آن چارچوب های موازی ساده هنوز تا حدودی به پلت فرم وابسته هستند. خوشبختانه، این مشکل قابل حمل را می توان با استاندارد جدید C++ 11 که در سال ۲۰۱۱ منتشر شد، حل کرد که با معرفی یک کتابخانه رشته جدید، پشتیبانی از برنامه نویسی چند رشته ای را اضافه می کند. به عبارت ساده، ما می‌توانیم با نمونه‌سازی کلاس std::threadبا ارسال تابعی که نشان‌دهنده کارهایی است که باید در این رشته انجام شود، یک رشته جدید ایجاد کنیم. باید توجه ویژه ای شود

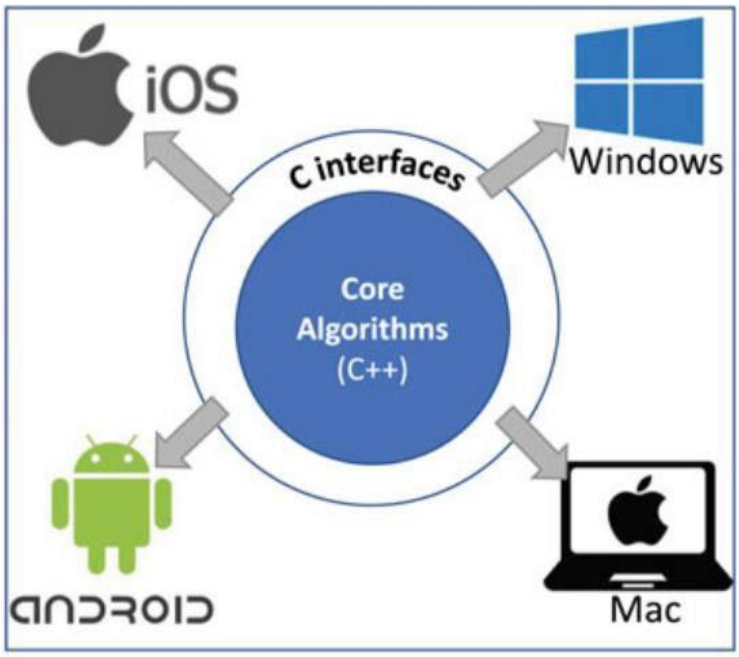
اجتناب از مسابقه داده در چنین برنامه های چند رشته ای با استفاده از اصول اولیه همگام سازی مانند و برای محافظت از داده های مشترک. خوانندگان علاقه مند می توانند برای جزئیات بیشتر در مورد برنامه نویسی چند رشته ای در C++ به این کتاب عالی [۵۱] مراجعه کنند.

در مورد برداری بر روی هر هسته با استفاده از دستورالعمل هایSIMD ، ما به کتابخانه معروفEigen ، یک کتابخانه الگوی C++ با کارایی بالا برای جبر خطی، که در بسیاری از پروژه های صنعتی استفاده شده است، تکیه می کنیم [۵۲]. ما Eigen را انتخاب می‌کنیم زیرا از تمام اندازه‌های ماتریس پشتیبانی می‌کند و الگوریتم‌های رایج برای تجزیه ماتریس، حل‌کننده‌های معادلات عددی و سایر الگوریتم‌های مرتبط مانند تبدیل فوریه سریع (FFT) را ارائه می‌دهد. مهمتر از آن، Eigenبا انجام بردارسازی صریح داخلی برای مجموعه‌های دستوراتSSE ، AVXو ARM NEON به اندازه کافی سریع است و این پیچیدگی را از کاربران پنهان می‌کند. به این معنا که ما می‌توانیم با اجرای ساده الگوریتم‌ها با استفاده از ماتریس‌ها و عملیات جبر خطی ارائه‌شده توسطEigen ، تا حد امکان به موازی‌سازی در سطح دستورالعمل دست یابیم. به عنوان یک مزیت جانبی، Eigenرابط هایی مشابه MATLAB ارائه می دهد که می تواند برنامه نویسی را ساده کرده و کدها را به ویژه برای مهندسان خواناتر کند.

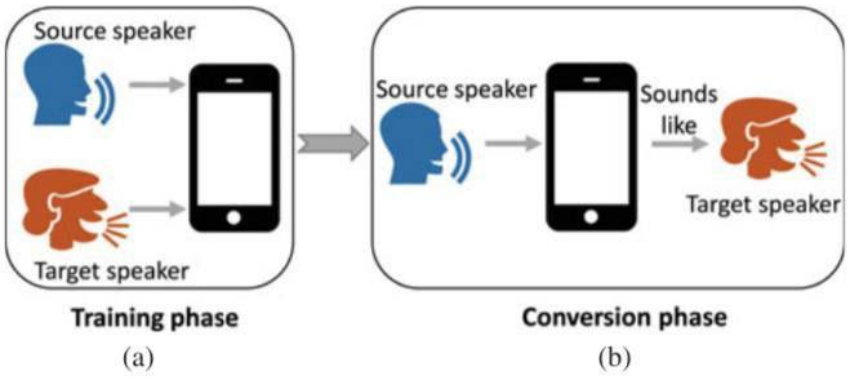
اکنون می‌توانیم معماری نرم‌افزاری سیستم تبدیل صدای خود را که در شکل ۵ نشان داده شده است، نمایانگر داشته باشیم. واضح است که سیستم ما الگوریتم‌های اصلی (موتور محاسباتی) و رابط کاربری گرافیکی (GUI) را به maxi تقسیم می‌کند. - قابلیت حمل پلت فرم mize در حالت ایده‌آل، هیچ تغییری در کد منبع الگوریتم‌های هسته‌ای هنگام انتقال به پلتفرم دیگر مورد نیاز نیست، و ما فقط باید کدهای C++ را با کامپایلرهای خاص پلتفرم برای استقرار مجدد کامپایل کنیم. برعکس، به طور کلی، بخش رابط کاربری گرافیکی با پلتفرم‌های خاصی مانندiOS ، اندروید و ویندوز به‌شدت مرتبط است و نمی‌توان آن را منتقل کرد. به عنوان مثال، دریOS ، ما رابط کاربری گرافیکی را با فریم ورک UIKit با استفاده از زبان Swift یا Objective-C می‌سازیم، در حالی که اندروید اجزای رابط کاربری خود را فراهم می‌کند که با زبان جاوا یا کاتلین قابل دسترسی هستند. یکی از مشکلات هسته محاسباتی مبتنی بر C++ این است که C++ فقط با برخی از زبان‌ها از جملهSwift ، زبان اصلی توسعهiOS ، قابلیت همکاری ضعیف یا حتی بدون همکاری را پشتیبانی می‌کند. بنابراین، ما تصمیم می‌گیریم رابط‌های ورودی/خروجی الگوریتم‌های اصلی را با استفاده از زبان برنامه‌نویسی C بپیچیم، که می‌تواند به راحتی با تمام زبان‌های رایج تعامل داشته باشد. در برنامه کامل تلفن همراه، رابط کاربری گرافیکی با انتقال اطلاعات عمدتاً از طریق قطار و تبدیل توابع ارائه شده به عنوان رابط های C با الگوریتم های اصلی ارتباط برقرار می کند. به این ترتیب، موتور محاسباتی، یعنی الگوریتم‌های هسته، می‌توانند به خوبی روی هر چهار سیستم عملیاتی نشان‌داده‌شده در شکل ۵ همانطور که ما آزمایش کرده‌ایم، دو مورد برای دستگاه‌های تلفن همراه و دو مورد برای رایانه‌های شخصی، کار کنند، بنابراین هدف ما از استقلال پلت فرم را محقق می‌کند. .

توسعه برنامه های IOS

پس از الگوریتم های اصلی سیستم تبدیل صدا با C++ پیاده سازی می شوند، کار باقی مانده است یک رابط کاربری گرافیکی دوستانه در تلفن های همراه ایجاد کنید، یعنی توسعه دهید.



تصویر۵: مروری بر معماری نرم افزار الگوریتم های اصلی (موتور محاسباتی) برای قابلیت حمل در C++ پیاده سازی شده اند. رابط گرافیکی کاربر (GUI) در بالای موتور قرار دارد و ممکن است با زبان ها/کتابخانه های مختلف در پلتفرم های مختلف ساخته شود. این موتور رابط های C را برای استفاده توسط رابط کاربری گرافیکی در چهار سیستم عامل رایجiOS :، Android، Windowsو Mac در معرض دید قرار می دهد.

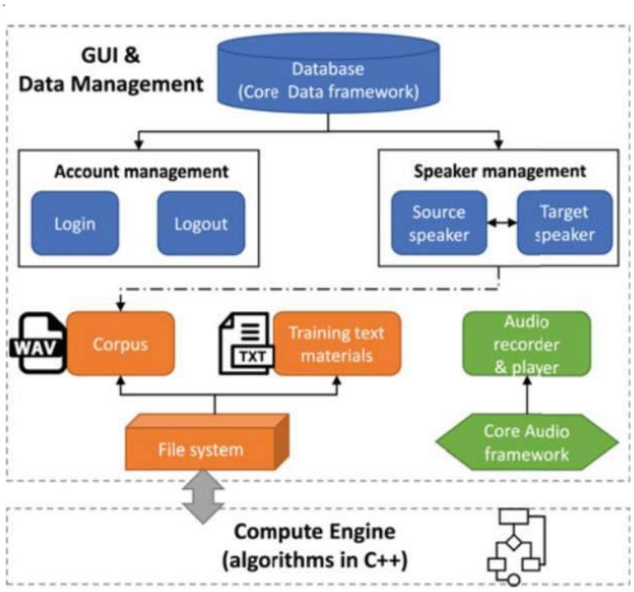


تصویر۶:گردش کار برنامه تبدیل صدا در تلفن های همراه

اپلیکیشن موبایل، برای تسهیل دسترسی به این سامانه. همانطور که قبلا در شکل ۵ نشان داده ایم، می توانیم رابط کاربری گرافیکی را با زبان ها و ابزارهای مختلف بر روی پلتفرم های مختلف توسعه دهیم که به خوبی از موتور محاسباتی در معماری ما جدا شده است. در این بخش، به عنوان یک مثال خاص، ما توسعه یک برنامه (برنامه) iOS را ارائه می‌کنیم که بر روی آیفون ۷، احتمالاً محبوب‌ترین تلفن هوشمند در سال ۲۰۱۷، مستقر شده است. به طور کلی، به عنوان یک نمونه اولیه برای تأیید امکان‌سنجی صوتی نسخه بر روی تلفن های همراه، برنامه ما بر قابلیت استفاده و مختصر تمرکز دارد. اساسی ترین استفاده از این برنامه فقط شامل دو مرحله است که در شکل ۶ نشان داده شده است. اولاً، هم گوینده منبع و هم بلندگوی هدف تعداد کمی از جملات را می خوانند در حالی که گفته های آنها توسط تلفن ضبط می شود تا یک پیکره موازی برای موارد بعدی تشکیل شود. آموزش مدل تبدیل ثانیاً، بلندگوی منبع می‌تواند دوباره چیزی را با برنامه صحبت کند و سعی می‌کند این پیام را به گونه‌ای تبدیل کند که به نظر برسد که توسط بلندگوی مورد نظر صحبت می‌شود. در ادامه جزئیات بیشتری در مورد ساختار ماژولار و برخی مسائل طراحی خاص این اپلیکیشن iOS ارائه شده است.

نمای کلی ماژول های کاربردی

در مهندسی نرم افزار، به خوبی شناخته شده است که ماژولار- ization یک اصل اساسی برای نرم افزارهای مقیاس بزرگ است



تصویر ۷: طرح کلی از ماژول ها در برنامه کاربردی

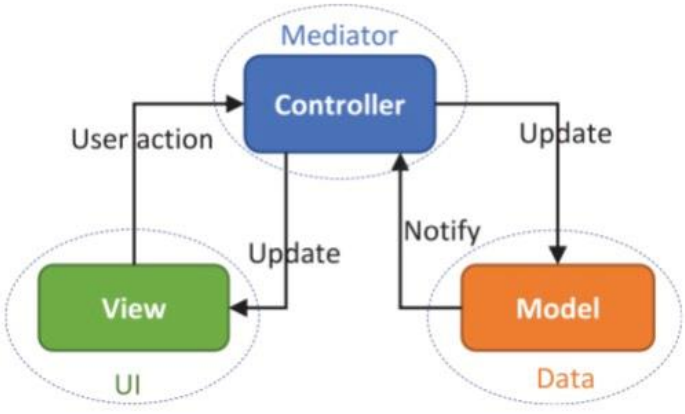
توسعه. ماژول های اصلی در سیستم ما در شکل ۷ نشان داده شده اند. برای مدیریت حساب های کاربری، از جمله ثبت نام، ورود به سیستم و خروج، اطلاعات کاربر را در یک پایگاه داده موبایل سبک مانند SQLite ذخیره می

کنیم که دسترسی به آن واسطه است. توسط چارچوب iOS Core Data برای ساده کردن کدنویسی. به طور مشابه، برای یک بلندگوی منبع (یعنی کاربر)، چندین بلندگوی هدف را می توان آموزش داد، و ارتباط بین بلندگوی منبع و بلندگوی هدف نیز در پایگاه داده ثبت می شود. بیشتر فضای ذخیره سازی هارد دیسک این برنامه توسط مجموعه ساخته شده برای آموزش تبدیل صوتی اشغال شده است. ما مطالب متنی را ارائه می دهیم که هر کدام حاوی حدود ۲۰ جمله عادی است تا توسط یک جفت گوینده منبع و گوینده هدف خوانده شود. در حالی که یک سخنران در حال خواندن متن است، صدای او می تواند به راحتی توسط برنامه ما ضبط شود (شکل ۶) و با پشتیبانی از چارچوب iOS Core Audio به عنوان فایل های WAV تک صدایی با فرکانس نمونه برداری ۱۶ کیلوهرتز ذخیره شود.

در نهایت، باید از شکل ۷ متوجه شد که در حال حاضر در سیستم ما تعامل بین لایه رابط کاربری گرافیکی و موتور محاسباتی، یعنی الگوریتم‌های اصلی پیاده‌سازی شده در++C ، کاملاً واضح و مختصر است. در طول آموزش، لایه رابط کاربری گرافیکی فقط باید به موتور بگوید فایل‌های WAV که مجموعه آموزشی را تشکیل می‌دهند چیست و موتور یک فایل مدل مطابق با تابع تبدیل تولید می‌کند. به طور مشابه، در نقطه تبدیل، لایه رابط کاربری گرافیکی موتور را در مورد فایل صوتی از بلندگوی مبدأ که قرار است تبدیل شود، مطلع می‌کند و موتور نهایتاً گفتار تبدیل شده را در یک فایل WAV جدید ذخیره می‌کند و به لایه رابط کاربری گرافیکی اطلاع می‌دهد. از مسیر فایل ما می توانیم صدای تبدیل شده را به سادگی با پخش این فایل صوتی بشنویم. علاوه بر این، همچنین می توان تنظیم کرد که پس از اتمام به طور خودکار پخش شود.

طراحی های خاص

ایجاد یک عملکرد خوب یک کار بسیار چالش برانگیز است برنامهiOS ، که شامل طراحیUI ، منطق تجاری است



تصویر ۸: الگوی طراحی MVC

مدل سازی، کدنویسی باSwift/Objective-C ، اشکال زدایی XCode و بسیاری کارهای خسته کننده دیگر. در اینجا، جزئیات برای حفظ فضا حذف شده است. در ادامه، ما فقط دو نگرانی طراحی خاص در مورد توسعه اپلیکیشن را به عنوان نمایندگان کل فرآیند فهرست کرده و به آن می پردازیم.

1.الگوی طراحی MVC برای رابط کاربری در توسعه برنامه های کاربردی مبتنی بر رابط کاربری گرافیکی، بهترین روش شناخته شده جداسازی داده ها از ارائه آن است، زیرا ارائه ممکن است با توجه به نیازهای خاص، به عنوان مثال، در نمودار، بسیار متفاوت باشد. یا در یک جدول از زمان اولین عرضهiOS ، اپل توصیه کرده است که الگوی طراحی Model-View-Controller (MVC) را برای تأکید بر این بهترین روش، که در شکل ۸ مشخص شده است، اتخاذ کند .نشان دهنده چیزی قابل مشاهده در رابط کاربری است که برای نمایش view داده ها اختصاص داده شده است. با این حال، مدل و نما نمی توانند به طور مستقیم به منظور کاهش جفت شدن ارتباط برقرار کنند. در عوض، آنها توسط شی کنترلر، معمولاً از طریق الگوی نمایندگی، واسطه می شوند.
در عمل، همانطور که در شکل ۷ نشان داده شده است، یک شی مدل را برای دسترسی به داده های ذخیره شده در پایگاه داده یا سیستم فایل تعریف می کنیم. کنترل کننده سعی می کند نما را با توجه به داده های تغییر یافته به روز کند. در جهت دیگر، هنگامی که کاربر با نمای تعامل برقرار می کند، اقدام او به کنترلر منتقل می شود، که بیشتر مدل را در مورد قصد کاربر آگاه می کند، به عنوان مثال، به روز رسانی یا حذف داده ها. علیرغم سادگی، لایه‌های یک برنامه رابط کاربری گرافیکی معمولی را می‌توان با الگوی MVC جدا کرد تا سازماندهی بهتر و ترویج استفاده مجدد از کد را تشویق کند. در برنامه ما، هر پنجره با الگوی MVC که در بالا توضیح داده شد ساخته شده است.
2.ارائه داده محور در نمای مجموعه برای نمایش بهتر چندین اطلاعات صوتی و آیتم های بلندگوی هدف، نمای جدول و مجموعه نماها به طور گسترده در رابط کاربری گرافیکی برنامه ما استفاده می شود. برای انعطاف پذیری بهتر و قابلیت استفاده مجدد برنامه، بهتر است داده ها را از تجسم آن از طریق سبک UI- Data-Operation جدا کنید. در چارچوب توسعهiOS ، داده ها در یک شی منبع داده ذخیره می شوند و عملیات توسط یک شی نماینده نشان داده می شود. توجه داشته باشید که نمای جدول و نمای مجموعه در iOS UIKit همگی با مکانیزم تخصصی MVC طراحی شده اند.
منبع داده تنها مسئول ارائه داده ها است و نمی داند داده ها چگونه نمایش داده می شوند. تفویض اختیار به اشیا فرصتی می دهد تا ظاهر و حالت خود را با تغییراتی که در جاهای دیگر برنامه اتفاق می افتد، که معمولاً توسط اقدامات کاربر ایجاد می شود، هماهنگ کنند. مهمتر از آن، تفویض این امکان را برای یک شی فراهم می کند تا رفتار شیء دیگر را بدون نیاز به ارث بردن از آن تغییر دهد. با این الگوی طراحی، می توانیم منبع داده و ارائه آن را بهتر جدا کنیم. هنگامی که داده ها در جایی به روز شوند، UIبه طور خودکار پاسخ می دهد تا تغییرات داده را منعکس کند. از این رو، چنین سبک ارائه مبتنی بر داده نیز به طور گسترده در برنامه ما به کار گرفته شده است تا توسعه رابط کاربری پیچیده را تسهیل کند.

آزمایش ها و نتایج

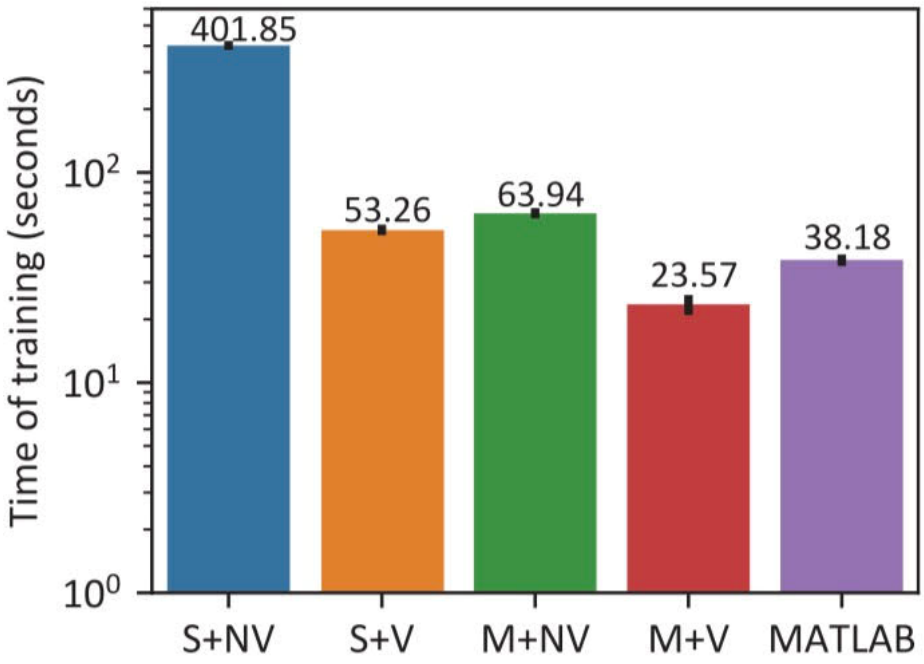
ما سیستم تبدیل صدا را به شیوه ای مناسب از پایین به بالا توسعه داده ایم زیرا دو لایه در شکل ۷ تقریباً کاملاً جدا شده اند. یعنی موتور محاسباتی مسئول محاسبات فشرده ابتدا با تمرکز بر دقت و کارایی با آزمایش و بهبود مکرر ساخته شد. گام بعدی طراحی رابط کاربری و مدیران داده مرتبط برای یک برنامه تلفن همراه با توجه به قابلیت استفاده و مختصر بودن بود. برنامه تلفن همراه تمام شده، به نامVoichap ، با موفقیت بر روی یک دستگاه آیفون ۷ مستقر شد. در ادامه، ابتدا سرعت الگوریتم هسته فعال شده توسط موازی سازی مبتنی بر چند هسته و بردار را ارزیابی می کنیم. سپس، رابط کاربری این اپلیکیشن را شرح می دهیم و استفاده از آن را از طریق مثال های خاص نشان می دهیم. در نهایت، کارایی و اثربخشی این سیستم تبدیل صدای تلفن همراه را با اندازه‌گیری زمان اجرای آن و انجام تست کیفیت تبدیل امتیازدهی شده توسط ۱۰ شنونده، ارزیابی می‌کنیم . الف) الگوریتم های هسته ای با سرعت بالا نتایج تست را افزایش می دهند

برای ارزیابی کارایی پیاده سازی خود با استفاده کامل از موازی سازی، عملکرد موتور را با تنظیمات محیطی مختلف آزمایش کردیم. برای سادگی، آزمایش‌ها را بر روی یک رایانه شخصی ویندوز ۱۰ با یک CPU Intel Core i7با ۴ هسته سخت‌افزاری انجام دادیم، زیرا الگوریتم‌ها برای اولین بار با Visual Studio IDE قدرتمند در ویندوز ۱۰ توسعه و آزمایش شدند. این همچنین نشان می‌دهد که الگوریتم اصلی ما پیاده‌سازی در واقع پلتفرم آگنوستیک است، زیرا می‌توانیم همان کد را در iOS (iPhone 7) بدون هیچ تغییری اجرا کنیم. در هر آزمون اندازه گیری زمان اجرا ذکر شده در زیر، آزمایش ها برای تخمین دقیق تر ۱۰ بار تکرار شد.

(1عملکرد کلی

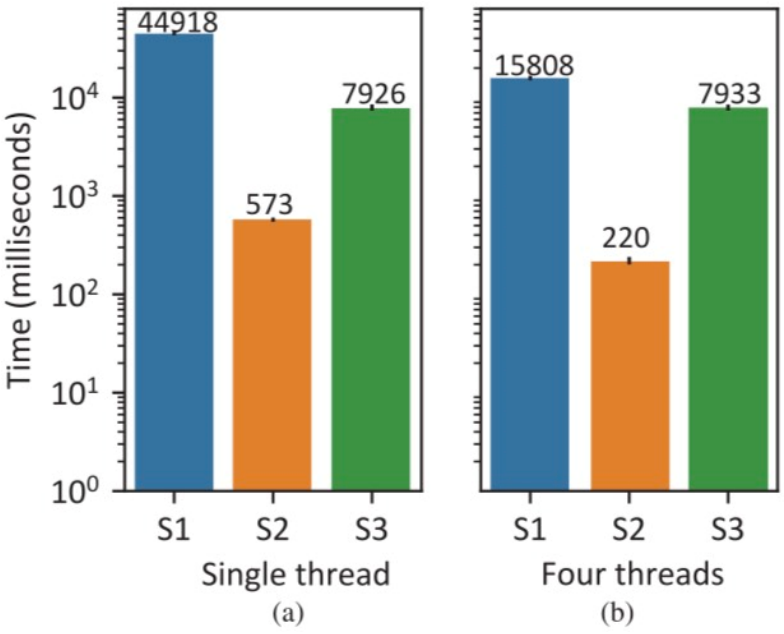
ما فرآیند آموزش را برای ۲۰ بیان زمان بندی کردیم که هر کدام به طور متوسط ۴,۵ ثانیه طول کشید، هم از بلندگوی منبع و هم از بلندگوی هدف، که پیکربندی پیش فرض برنامه تلفن همراه ما است. در این آزمون، تعداد مؤلفه‌های گاوسی مورد استفاده در مدل $m = 4$ (7) انتخاب شده است. به‌عنوان خط پایه، ما همچنین زمان اجرای کد اصلی متلب را در همان رایانه شخصی اندازه‌گیری کردیم. در اینجا لازم به یادآوری است که بسیاری از توابع داخلی متلب در واقع روال های C یا Fortran بسیار بهینه شده هستند که ممکن است به خوبی به صورت موازی در داخل پیاده سازی شوند. بنابراین، معمول است که یک برنامه عددی C++ ساده نتواند همتای خود را در MATLAB در بازده زمانی شکست دهد. آمار زمان اجرای اندازه گیری شده در شکل ۹ نشان داده شده است. علاوه بر این، باید توجه داشته باشیم که روش تاب برداشتن فرکانس فقط فریم های صدادار را تغییر می دهد. بنابراین، تعداد فریم های صوتی در گفته ها ممکن است مورد توجه بیشتری باشد. در مجموعه آموزشی ما شامل ۲۰ گفتار، میانگین تعداد فریم های صوتی در هر گفته ۳۰۲,۶ و تغییر فریم ۸ میلی ثانیه (۱۲۸ نمونه)

است. بنابراین، میانگین طول سیگنال صوتی در مجموعه آموزشی تنها ۲,۴۲ ثانیه است. به طور مشابه، می‌توانیم میانگین تعداد فریم‌های صدادار در مجموعه آزمایشی را برای تبدیل بشماریم، که در حدود ۲۹۷,۵ است که مربوط به میانگین مدت زمان ۲,۳۸ ثانیه است، اگر فقط بخش‌های صداگذاری شده را در نظر بگیری برای پیاده سازی واقعا کارامد در تلفن های همراه



تصویر ۹: زمان اجرای مرحله تمرین با تنظیمات مختلف) فاصله اطمینان ۹۵ .(از چپ به راست، S+NV: C++تک رشته‌ای بدون بردار، S+V: C++تک رشته‌ای با برداری، M+NV: C++چند رشته‌ای بدون برداری، M+V: C++چند رشته‌ای با بردارization - ، MATLAB: 64بیتی MATLAB 2016 با تنظیمات پیش فرض

در اجرای فعلی مرحله آموزش، دو مرحله اول شامل تجزیه و تحلیل گفتار، ساخت ویژگی، و تراز قاب (به شکل ۱ مراجعه کنید) روی چند هسته موازی می شوند. در حالت ایده آل، هر گفته یا جفت منبع-هدف



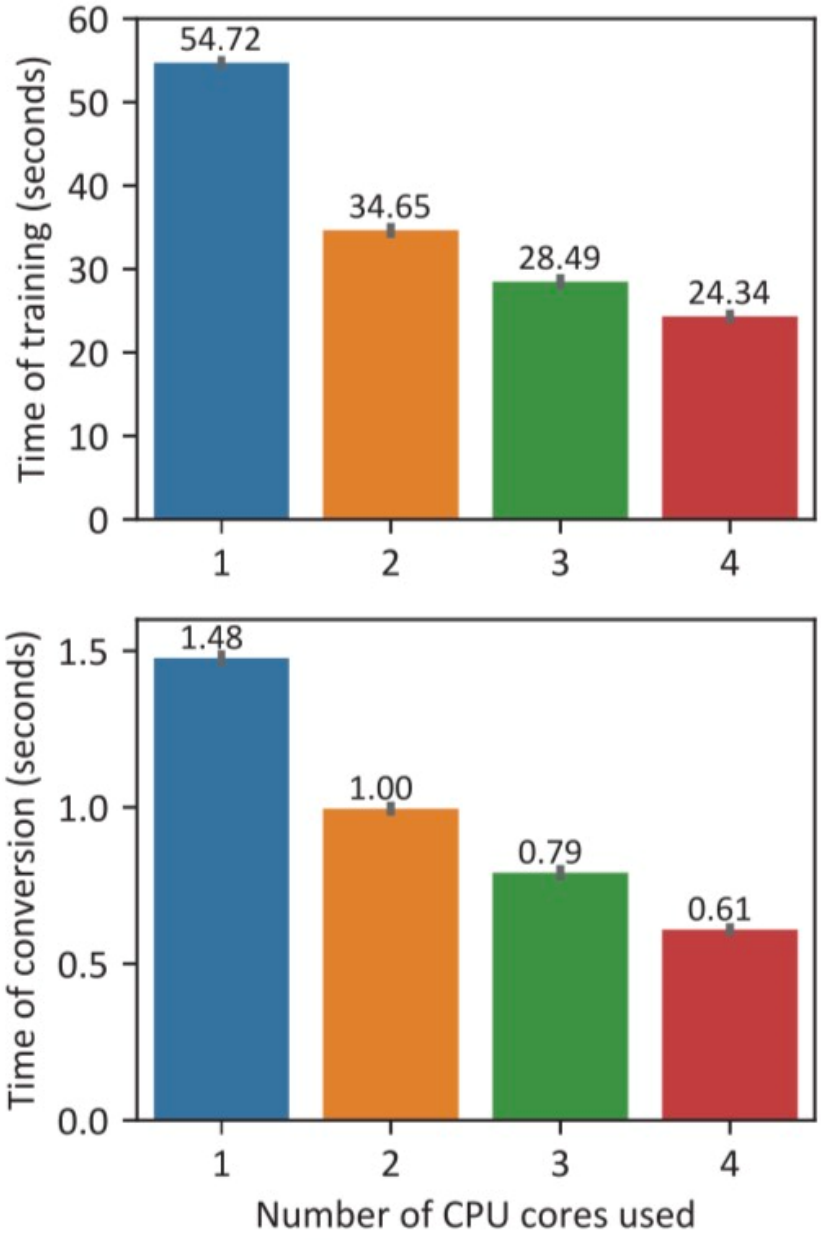
تصویر ۱۰: زمان اجرای هر مرحله در طول مرحله تمرین با تنظیمات مختلف :S1 .تجزیه و تحلیل گفتار و ساخت ویژگی :S2 .تراز قاب از بدنه موازی :S3 .آموزش تابع تبدیل. برداری برای همه شرایط فعال است.

گفته ها (برای تراز قاب) می توانند به طور مستقل و همزمان در هسته های جداگانه پردازش شوند. برای بررسی بیشتر توزیع بار کار، ما زمان اجرای هر مرحله در طول تمرین را با ۲۰ نمونه تمرین موازی، که هر کدام به طور متوسط حدود ۴,۵ ثانیه طول کشید، اندازه‌گیری کردیم. نتایج در شکل ۱۰ (a) ترسیم شده است.

به وضوح نشان داده شده است که مرحله اول، یعنی تجزیه و تحلیل گفتار و ساخت ویژگی، بیشتر زمان می برد)حدود ۸۴ .(در مقابل، مرحله دوم آموزش، یعنی هم ترازی فریم، تنها نسبت ناچیزی از کل زمان مجموعه آموزشی فعلی را می گیرد. در مورد مرحله آخر، آموزش تابع تبدیل، اگرچه زمان محاسباتی غیر قابل اغماض را می طلبد، موازی کردن این مرحله ساده نیست، زیرا به کل داده ها نیاز دارد. علاوه بر این، از آنجایی که آخرین مرحله در مقایسه با دو مرحله اول زمان بسیار کمتری می برد، ما در حال حاضر تنها دو مرحله اول را از طریق موازی سازی چند هسته ای اجرا می کنیم. با توزیع حجم کار دو مرحله اول به چهار هسته با استفاده از چهار رشته، زمان اجرای این دو مرحله به میزان زیادی به حدود ۳۵,۲ و کل زمان اجرا به حدود ۴۴,۸ از زمان مورد نیاز روی یک هسته کاهش می یابد. که در شکل ۱۰ (ب) نشان داده شده است. در نهایت، می‌خواهیم تأکید کنیم که اگرچه زمان صرف شده توسط مرحله هم‌ترازی فریم S2) در شکل ۱۰ (جزئی به نظر می‌رسد، اما همچنان لازم است این مرحله را موازی کنیم، زیرا پیچیدگی زمانی نظری آن به دلیل استفاده از تاب خوردگی زمانی پویا [۵۳]، در حالی که دو مرحله دیگر فقط پیچیدگی زمانی خطی را به صورت تجربی نشان می‌دهند.

(3اثر تعداد هسته ها برای تجزیه و تحلیل بیشتر اثر افزایش سرعت ناشی از موازی سازی چند هسته ای، زمان آموزش مجموعه آموزشی متشکل از ۲۰ گفته موازی با مدت متوسط حدود ۴,۵ ثانیه و همچنین زمان تبدیل

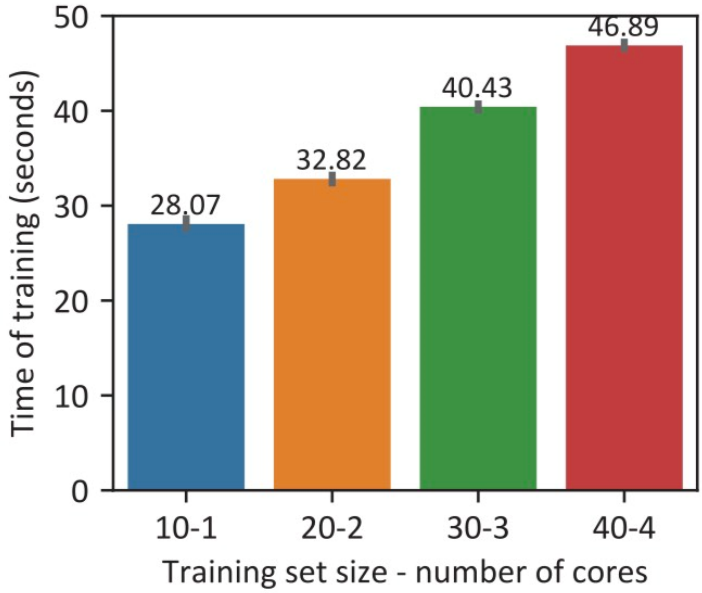
یک گفتار ورودی منبع ۵ ثانیه را اندازه گیری کردیم. از آنجایی که در کل ۴ هسته در CPU رایانه شخصی ما وجود دارد، برنامه به گونه ای پیکربندی شده است که از تعداد هسته های متفاوتی از ۱ تا ۴ برای موازی سازی استفاده کند. زمان اجرا با توجه به تعداد هسته های مجاز گزارش شده است



تصویر ۱۱: زمان اجرای مرحله آموزش (بالا) و مرحله تبدیل (پایین) زمانی که تعداد متفاوتی از هسته های CPUبرای موازی سازی چند هسته ای استفاده می شود) فاصله اطمینان ۹۵ .(در کل چهار هسته در CPU در دست بررسی وجود دارد. برداری فعال است.

در شکل ۱۱. از آنجایی که فقط دو مرحله اول مرحله آموزش از موازی سازی چند هسته ای بهره می برند، یعنی حدود ۸۴ میکرومتر از برنامه را می توان موازی کرد) شکل ۱۰((a)، اثر شتاب نشان داده شده است. شکل ۱۱ در واقع با قانون Amdahl سازگار است، که اغلب در محاسبات موازی برای پیش بینی سرعت تئوری در هنگام استفاده از پردازنده های متعدد استفاده می شود [۵۴]. به طور خاص، با چهار هسته موجود، سرعت مرحله آموزش حدود ۲,۳ برابر است، در حالی که قانون Amdahl سرعت تئوری را حداکثر ۲,۷ برابر پیش بینی می کند. به طور کلی، با توجه به رابطه بین زمان اجرا و تعداد هسته های نشان داده شده در شکل ۱۱، این سیستم تبدیل صدا با افزایش سرعت زیر خطی زمانی که موازی سازی چند هسته ای اعمال می شود مشخص می شود.

ما بیشتر مقیاس‌پذیری موازی‌سازی چند هسته‌ای را در سیستم خود با افزایش اندازه مجموعه آموزشی متناسب با تعداد هسته‌های استفاده شده بررسی کردیم. نتایج در شکل ۱۲ نشان داده شده است. بدیهی است که میانگین زمان مورد نیاز برای ۱۰ نمونه آموزشی در هر هسته با بزرگ شدن اندازه کل مشکل افزایش می یابد. این به دلیل این واقعیت است که همه بخش های چارچوب الگوریتم موازی نیستند (شکل ۱۰). با این وجود، استفاده از موازی سازی چند هسته ای در سیستم ما هنوز مقیاس پذیر است، زیرا زمان اجرا می تواند برای اندازه مشکل ثابت تا حد زیادی کاهش یابد که هسته های بیشتری در دسترس هستند (شکل ۱۱). از منظر عملی، باید توجه داشت که اگرچه اصولاً هنگام اجرای موازی سازی مبتنی بر multithreading می‌توانیم از تعداد رشته‌هایی که دوست داریم استفاده کنیم، زمانی که تعدادThread ها نسبت به هسته‌ها در یک CPU منفرد بیشتر باشد، سرعت افزایش نمی‌تواند بیشتر شود. به عنوان مثال، در مورد ما، مرحله آموزش سیستم ما به CPU فشرده است، و بنابراین حداکثر چهار رشته را می توان به طور همزمان روی یک CPU 4 هسته ای اجرا کرد.



تصویر ۱۲: زمان آموزش اندازه های مختلف مجموعه آموزشی و تعداد هسته ها) فاصله اطمینان ۹۵ .(□به طور متوسط، هر هسته مربوط به یک مجموعه آموزشی از ۱۰ بیان است. برچسب n-k به معنای n گفته در مجموعه آموزشی و k هسته است. برداری فعال است.

نمایش برنامه

برنامه موبایلی که ما برای iPhone 7 توسعه داده ایم، Voichap، در مجموع دارای شش پنجره است که عکس های صفحه نمایش آنها در شکل ۱۳ نشان داده شده است که با ماژول های مختلف در شکل ۷ مطابقت دارد. مهمترین اقدامات در شکل ها انجام شده است. ۱۳ (d)و ۱۳(f)، جایی که آموزش و تبدیل انجام می شود. همانطور که در شکل ۱۳ (d) نشان داده شده است، ما تعدادی رونوشت را برای خواندن توسط گوینده منبع و گوینده هدف برای ساختن یک پیکره موازی برای اهداف آموزشی ارائه کرده ایم، جایی که می توانیم اطلاعات گوینده هدف، از جمله او را نیز ویرایش کنیم. نام او و توضیحات کوتاه به طور خاص، در زمینهVoichap ، سخنران منبع در واقع به کاربر اشاره دارد. از این رو، کاربر فقط باید این جملات را یک بار بخواند و صداهای ذخیره شده را می توان دوباره برای مطابقت با همه بلندگوهای هدف برای آموزش در آینده استفاده کرد. پس از آن، کاربر می تواند از گوینده هدف، به عنوان مثال، دوست خود، بخواهد که این جملات را دوباره روی تلفن بخواند و گفته ها توسط این برنامه ضبط می شود تا مجموعه آموزشی ایجاد شود.

همانطور که قبلا در شکل ۷ نشان داده شد، اطلاعات بلندگو در یک پایگاه داده سبک ذخیره می شود و فایل های صوتی ضبط شده در یک فهرست خاص برای هر بلندگو در سیستم فایل محلی iOS سازماندهی می شوند. پس از اتمام ضبط، دکمه Train در شکل ۱۳ (d)را می توان برای شروع فرآیند آموزشی لمس کرد و در پایان، یک فایل مدل تبدیل مربوط به جفت منبع-هدف داده شده ایجاد می شود. توجه داشته باشید که فایل مدل حاوی تمام اطلاعات لازم برای بازیابی سریع تابع تبدیل f (۰) در صورت نیاز است (شکل ۱). اکنون، زمان بازی با تبدیل صدای بلادرنگ است که توسط صفحه Speak here در شکل ۱۳ (f)مشخص شده است. فقط کافی است دکمه میکروفون را فشار داده و نگه دارید و هر چیزی که دوست دارید بگویید. پس از رها شدن دکمه میکروفون، مرحله تبدیل به صورت خودکار و بلافاصله شروع می شود.

پس از مدت کوتاهی، می‌توانید آنچه را که اخیراً گفته‌اید تکرار می‌شود، اما با صدای گوینده هدف بشنوید.

به عنوان یک یادداشت جانبی، برنامه Voichap ما همچنین می تواند با فایل های صوتی به طور مستقیم به غیر از ضبط جملات بداهه توسط خودمان تغذیه شود. این پشتیبانی می تواند برنامه را بسیار جالب تر کند. همانطور که در بخش مقدمه ذکر شد، در حال حاضر، برخی از برنامه های پولی در گوگل پلی یا اپل استور در رابطه با تغییر صدا در دسترس هستند که می توانند صدای افراد مشهور را تقلید کنند، اگرچه عملکرد آنها به طور

کلی ضعیف است. با این حال، با برنامه Voichap ما، می توانید صدای خود را به صدای هر فرد مشهوری تبدیل کنید، زمانی که بتوانید برخی از گفته های آموزشی از او دریافت کنید. البته این غیرعملی است که واقعاً از یک سلبریتی بخواهیم متن مشخص شده را روی تلفن ما بخواند. با این حال، به راحتی می‌توانیم سخنرانی‌های عمومی آنها را از وب‌سایت‌هایی مانند یوتیوب دانلود کرده و سپس فایل‌های صوتی را استخراج کرده و به عنوان مواد آموزشی سخنران مورد نظر به جملات تقسیم کنیم. کار باقی مانده این است که همان جملات را بیان کنیم و گفته هایمان را به عنوان داده های آموزشی گوینده منبع ضبط کنیم. این فرآیند در شکل ۱۴ نشان داده شده است، جایی که ما رئیس جمهور ترامپ را به عنوان مثال در نظر می گیریم. به طور خلاصه، ابتدا یک هدف جدید ایجاد می شود و فایل های صوتی بارگذاری می شوند که ما آنها را از YouTube برداشتیم. پس از آموزش مدل، می‌توانیم کلماتی مانند «آمریکا را دوباره بزرگ کن» به زبان بیاوریم که گویی رئیس جمهور ترامپ هستیم، در حالی که سیستم تبدیل صدا، فردیت صدا را تغییر می‌دهد و آن را طوری می‌سازد که گویی واقعاً توسط رئیس جمهور ترامپ گفته شده است.

آزمایشی ارزیابی نرم افزار سیستم تبدیل صدا

اثربخشی و عملکرد بلادرنگ سیستم تبدیل صدای تلفن همراه ما، Voichap، با تبدیل صدا در میان سخنرانان زن و مرد مورد ارزیابی قرار گرفت. برخلاف معیار سنتی تبدیل صدا، که همیشه به دنبال تشابه تبدیل بهتر یا کیفیت گفتار بدون توجه به هزینه محاسباتی است، به عنوان یک برنامه کاربردی موبایل، Voichapتلاش می‌کند تا در یک بازه زمانی قابل قبول به بهترین عملکرد اما نه لزوماً بهترین عملکرد برسد. . هدف نهایی ما توسعه یک سیستم تبدیل صدای بلادرنگ در تلفن های همراه برای استفاده روزانه است و تجربه کاربر بیشترین اهمیت را دارد. بدیهی است که در این شرایط اگر کاربر مجبور باشد برای دریافت گفتار تبدیل شده نیم ساعت صبر کند، حتی اگر کیفیت تبدیل بسیار بالا باشد، چندان منطقی نیست.

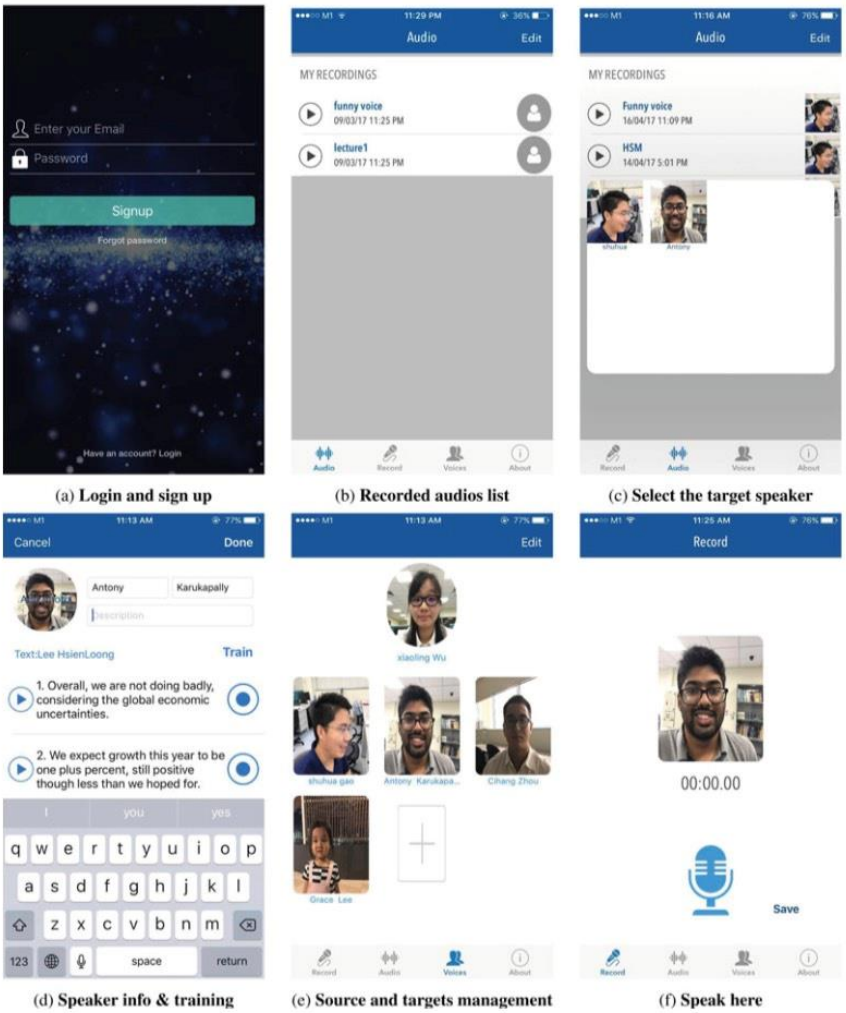
چهار نامزد در محدوده سنی ۲۰ تا ۳۰ سال به عنوان سخنران در آزمون های زیر انتخاب شدند، که شامل یک مرد و یک زن به عنوان منبع و تنظیم یکسان برای هدف بود. بنابراین، چهار جفت گوینده در مجموع بررسی می‌شوند: مرد به مرد(M–M) ، مرد به زن(M–F) ، زن به مرد (F– M) و زن به زن.(F-F)

(1تنظیم پارامتر و بازده زمانی

دو پارامتر اصلی سیستم تبدیل صدای ما، تعداد جملات آموزشی و تعداد اجزای گاوسی در مدلGMM ، یعنی mدر معادله (۷) است. بر اساس نظرسنجی از چندین کاربر، ما دریافتیم که ساخت یک مجموعه رونویسی از ۲۰ جمله با مدت زمان حدود ۴ یا ۵ ثانیه به منظور آموزش، انتخاب خوبی است. اگرچه به طور کلی یک مجموعه آموزشی بزرگتر می تواند به عملکرد بهتر کمک کند، ممکن است کاربر را از خواندن جملات زیاد خسته کند و همچنین می تواند منجر به زمان طولانی آموزش شود. برای تعیین بهترین گزینهm ، عمدتاً باید بین زمان اجرا و

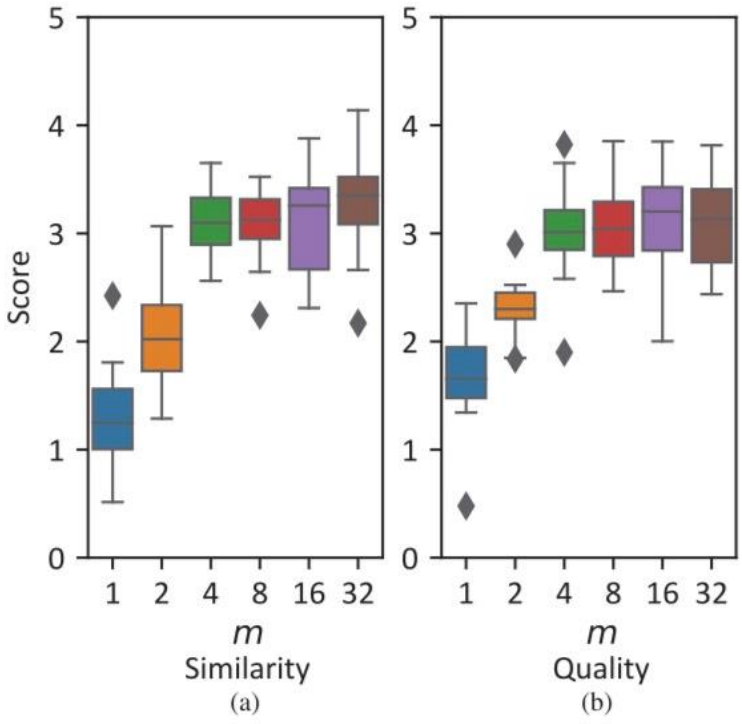
عملکرد تبدیل تعادل برقرار کنیم. با آزمایش مقادیر مختلفm ، زمان تمرین را با توجه به هر m در شکل ۱۵ خلاصه می کنیم.

واضح است که اگر مولفه های گاوسی بیشتری به کار گرفته شوند، مرحله آموزش زمان بیشتری طول خواهد کشید. علاوه بر این، می بینیم که کل زمان آموزش با توجه به تعداد مولفه های گاوسی m رابطه غیرخطی نشان می دهد. از طرف دیگر، میانگین زمان تمرین برای مجموعه تمرینی شامل ۲۰ گفته بین ۲۰ تا ۵۵ ثانیه برای m های مختلف است. مهمتر از همه، از طریق یک آزمون گوش دادن ذهنی از گفتار تبدیل شده از نظر تشابه تبدیل شده به هدف و کیفیت گفتار، که در شکل ۱۶ نشان داده شده است، متوجه می شویم که بهبود کلی یک بار < m 4جزئی است و در نتیجه شنوندگان نظرات قابل مقایسه ای در مورد شباهت و کیفیت گفتارهای تبدیل شده تولید شده توسط مقادیر مختلف m بالای این آستانه ارائه می دهند. بنابراین، از شکل‌های ۱۵ و ۱۶، انتخاب m 4 =به‌عنوان مقدار پارامتر پیش‌فرض سیستم ما معقول است، که می‌تواند به کاهش زمان اجرا کمک زیادی کند، اما بدون اینکه کیفیت گفتار تبدیل شده و شباهت را از جنبه انسانی بدتر کند. ادراک در ارزیابی های عینی و ذهنی زیر، همه ما setm.۴ =



Fi
th
ev
fo

تصویر ۱۴: یک سناریو معمولی چگونه صدای خود را شبیه ترامپ کنیم؟



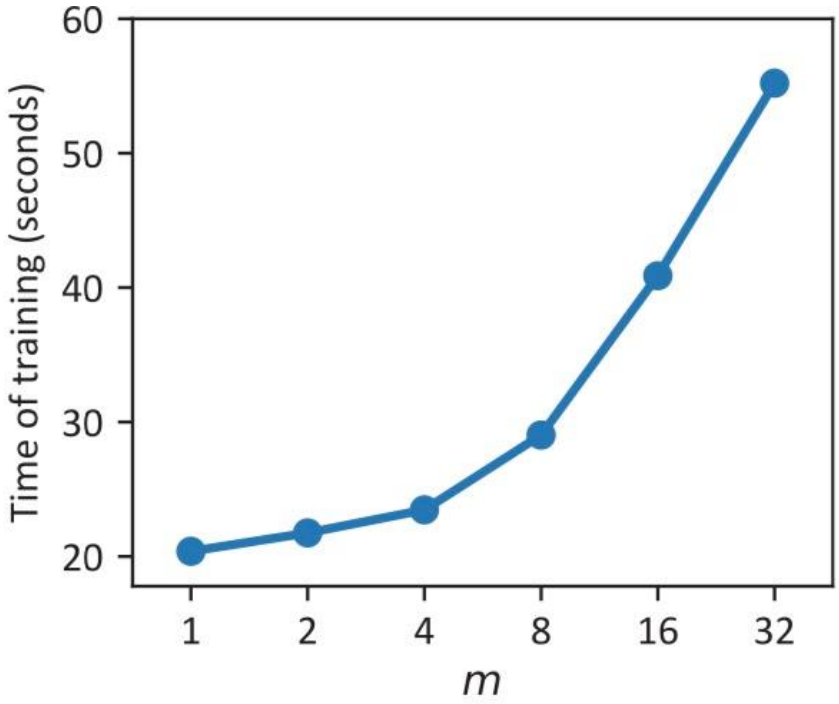
تصویر ۱۴: زمان اجرای مرحله تمرین با توجه به تعداد متفاوت مولفه‌های گاوسی) فاصله اطمینان ۹۵ (اندازه‌گیری شد. مجموعه آموزشی شامل ۲۰ جمله با طول متوسط حدود ۴,۵ ثانیه است و زمان اجرا با تکرار ۱۰ بار اندازه گیری می شود. بردارسازی و موازی سازی ۴ هسته ای فعال هستند.

بیایید اکنون به برنامه تلفن همراه Voichap برگردیم. با استفاده از همان مجموعه آموزشی و مجموعه تست، زمان اجرای برنامه با موازی سازی چند هسته ای و چند رشته ای فعال در آیفون ۷ به شرح زیر اندازه گیری شد:

•مرحله آموزش یک پیکره موازی شامل ۲۰ جمله با طول متوسط حدود ۴ ثانیه تقریباً ۳۶,۵ ثانیه طول می کشد.

•مرحله تبدیل برای ۴,۶ جمله از بلندگوی منبع حدود ۰,۸۵ ثانیه طول می کشد.

این نتیجه انتظارات ما را بر اساس شکل‌های ۹ و ۱۵ برآورده می‌کند. توجه داشته باشید که در مقایسه با رایانه شخصی، توانایی محاسباتی دستگاه آیفون نسبتاً ضعیف است، که به طور اجتناب‌ناپذیری منجر به طولانی‌تر شدن زمان اجرا نسبت به زمان اندازه‌گیری شده با MATLAB در رایانه شخصی ویندوز ۱۰ می‌شود.



تصویر ۱۶: ارزیابی ذهنی سیستم تبدیل صدا با توجه به تعداد اجزای گاوسی m در GMM. از ۱۰ شنونده خواسته شد تا شباهت و کیفیت گفتار را ارزیابی کنند. در اینجا میانگین امتیاز چهار جهت تبدیل منبع-هدف ممکن را نشان می دهد.

Table 1. The MCD of the unconverted source, the traditional GMM and the weighted frequency warping (WFW) method

	No conversion	GMM	WFW
MCD(dB)	7.83	5.85	5.97

(2ارزیابی عینی تبدیل صدا معمولاً دو نوع ارزیابی را می توان برای امتیاز دادن به عملکرد یک سیستم تبدیل صدا استفاده کرد: ارزیابی عینی و ارزیابی ذهنی. برای ارزیابی عینی، پرکاربردترین معیار در ادبیات، اعوجاج-Mel cepstral (MCD)بین گفتار تبدیل شده و گفتار هدف اصلی است [۹، ۱۵، ۱۸ MCD .]توسط محاسبه می شود

$$\mathrm{MCD[dB]} = \frac{10}{\log 10}\sqrt{2\sum_{i=1}^{24}(c_i - \hat{c}_i)^2}, \qquad (18)$$

گیت هاب چیست؟

یکی از بزرگترین انجمن‌های توسعه دهندگان وب در جهان github است. در واقع گیت هاب پلتفرمی است که در آن توسعه دهنگان وب از سراسر جهان در آن گرد هم آمده و با یکدیگر ارتباط و همکاری دارند. در گیت هاب شما به عنوان توسعه دهنده وب می‌توانید پروژه‌های خود را با همکارانتان یا هر فرد دیگری که مایل باشید به اشتراک بگذارید و به صورت مشترک روی یک پروژه کار کنید. به این ترتیب به سادگی می‌توانید نسخه‌های قبلی یک نرم افزار را ارتقا دهید بدون این که تغییر یا اختلالی در نسخه‌های فعلی ایجاد شود.

github کار کردن روی کدها را بسیار ساده کرده است. به کمک این پلتفرم می‌توانید به کوتاه‌ترین و ناپیدا ترین خط کد خود دسترسی پیدا کنید و در صورت لزوم آن را تغییر دهید. اما جذاب‌ترین ویژگی گیت‌هاب این است که به کمک آن می‌توانید با سایر کدنویسان در جهان ارتباط برقرار کنید. تیم بسازید و به‌طور مشترک روی پروژه‌های مختلف کار کنید.

مزایای گیت هاب چیست و چرا از آن استفاده می‌کنیم؟

مزایای گیت‌هاب بسیار زیاد و دلایلی که به خاطر آن از این پلتفرم استفاده می‌کنیم برای هر کد نویسی متفاوت است. اما اولین دلیلی که کد نویسان جهان را مجبور می‌کند به گیت هاب بپیوندند این است که در آن امکان همکاری نرم وجود دارد. همچنین امکان تست و کنترل نسخه دلیل دیگری است که github را برای کد نویسان جذاب کرده است.

مزیت دیگر گیت هاب این است که امکان یادگیری مباحث جدید و زبان‌های برنامه نویسی تازه در آن فراهم است. کافی است کمی روی آن وقت بگذارید تا ببینید که چه دریای وسیعی از دانش و اطلاعات در آن وجود دارد که فرا گرفتن آن‌ها می‌تواند شما را در حوزه کاریتان تبدیل به یک فرد نخبه کند.

این ویژگی که افراد قادرند نسخه خود را با هر کسی که تمایل دارند به اشتراک بگذارند تا مورد بررسی و تحلیل واقع شده و اگر اشکالی در آن وجود دارد رفع شود، جزو جذابیت‌های غیر قابل انکار گیت هاب است. در حال حاضر بسیاری از تیم‌های کد نویسی یا شرکت‌هایی که به‌طور تخصصی در

این زمینه کار می‌کنند عضو github هستند و در این پلتفرم پروژه‌های خود را پیش می‌برند. بنابراین اگر تمایل دارید در این حوزه کار کنید خوب است که زیر و بم گیت هاب را بشناسید و هرچه سریع‌تر عضو آن شوید. چون این می‌تواند روی افزایش قدرت رزومه شما نقش ویژه‌ای داشته باشد.

چگونه از گیت هاب استفاده کنیم؟

ممکن است در ابتدای کار ظاهر github به نظر کمی پیچیده بیاید. اما دانستن چند نکته به شما کمک می‌کند خیلی زود با این پلتفرم ارتباط برقرار کنید. در اینجا به صورا مقدماتی آموزش کار با سایت github را به شما می گوییم. نخسین قدم این است که عضو گیت هاب شوید.

۱- ساخت اکانت گیت هاب

عضویت در گیت‌هاب رایگان است. به راحتی می‌توانید یک حساب کاربری در آن ایجاد کرده و کار خود را آغاز کنید. با این حساب کاربری رایگان به بانک وسیعی از اطلاعات دسترسی خواهید داشت. ضمن آن که می‌توانید از ویژگی ردیابی مشکل در تکه کدهای خود استفاده کنید و امکان به اشتراک گذاری پروژه‌ها و مدیریت آن‌ها نیز به شما داده می‌شود.

تنها اشکالی که می‌توان به حساب کاربری رایگان گیت‌هاب نسبت داد این است که شما اجازه دارید به جز خودتان فقط ۳ نفر دیگر را به پروژه‌ها اضافه کنید. به‌هرحال کار کردن با این نسخه به منظور آشنا شدن با فضای Github و به اشتراک گذاشتن پروژه‌های کوچک به شدت توصیه می‌شود

۲- Git را نصب کنید

گیت هاب روی Git اجرا می‌شود. Git یک سیستم کنترل نسخه است که توسط اسطوره برنامه نویسی " لینوس تورووالد " ایجاد شده است. git ابزار جذابی است که به برنامه نویسان کمک می‌کند تا با یکدیگر همکاری داشته باشند، به صورت مشترک روی یک پروژه کار کنند، نرم افزارها را ارتقا دهند و اشکالات موجود روی نسخه‌های قبلی اپلیکیشن‌ها را برطرف کنند.

وظیفه اصلی Git این است که اشکالات یا تغییرات را ردیابی کند و این امکان را برای تیم‌ها فراهم کند که از راه دور روی یک پروژه مشترک کار کنند.

Git را در گوگل جستجو کنید و آن را روی سیستم خود نصب کنید.

۳- یک Repository یا مخزن ایجاد کنید

قبل از انجام هر کاری در گیت هاب باید یک مخزن یا Repository در آن ایجاد کنید. در فضای github به مخزن repo گفته می‌شود و این کلمه معادل واژه پروژه است. هر repo در واقع فضایی است که در آن هر چیزی که مربوط به یک پروژه است گرد آوری و سازمان دهی می‌شود. چیزهایی مثل تصاویر، اکسل شیت‌ها، ویدئوها و به‌طورکلی هر چیزی که برای راه اندازی پروژه خود به آن نیاز دارید.

اغلب افراد در repo یک فایل read me قرار می‌دهند که حاوی همه اطلاعاتی است که برای پروژه مذبور به آن نیاز است. به این ترتیب هر فرد جدیدی که به یک پروژه بپیوندد با خواندن این فایل در جریان جزئیات پروژه قرار گرفته و می‌تواند کار خود را آغاز کند.

برای ایجاد یک مخزن جدید:

- روی گزینه new repository در گوشه سمت راست بالای صفحه گیت هاب خود کلیک کنید.
- سپس برای این مخزن یک اسم انتخاب کنید و برای آن مختصری توضیحات بنویسید.
- روی کادری که می‌گوید «این مخزن را با یک README راه‌اندازی کنید» تیک بزنید.
- روی گزینه “ ”create repositoryکلیک کنید

۴ - یک شعبه یا Branch ایجاد کنید

پروژه‌های برنامه نویسی همیشه چند وجهی هستند. یعنی هنگام ساختن آن‌ها به نسخه‌های برنامه نویسی زیادی نیاز است. وقتی برای یک پروژه Branch یا شعبه ایجاد می‌کنید این امکان برای شما فراهم می‌شود که چندین نسخه منحصر به فرد یک مخزن را به صورت هم‌زمان ویرایش کنید.

به‌طور پیش فرض هر مخزن دارای یک شاخه به اسم Master است و میتواند چندین زیر شاخه داشته باشد. شما به دلخواه خود می‌توانید روی هر شاخه کار کنید. در نهایت این شاخه‌ها به مخزن اصلی متصل و یک پروژه واحد را تشکیل می‌دهند.

برای ایجاد یک شاخه یا Branch جدید در گیت هاب باید:

- به مخزن جدید خود بروید.
- از منوی کشویی گزینه branch: Master را انتخاب کنید.
- برای شعبه خود یک اسم انتخاب کنید.
- گزینه Create Branch را بزنید.

اکنون می‌توانید روی شعبات مختلف از یک پروژه کار کنید یا آن را با دیگران به اشتراک بگذارید

۵- ایجاد و انجام تغییرات روی یک Branch

حالا اگر می‌خواهید روی یک شعبه تغییراتی ایجاد کنید باید:

- روی نماد شعبه ایجاد شده در مخزن خود کلیک کنید.
- وقتی شعبه مورد نظر باز شد هر تغییری را که لازم است روی آن ایجاد کنید.
- روی نماد مداد در قسمت سمت راست بالا کلیک کنید.
- تغییرات خود را با نوشتن یک پیام commit توصیف کنید و سپس روی " commit changes"کلیک کنید. به هر تغییر ایجاد شده یک commit میگویند.
- هر Commit جزئیات مربوط به تغییرات ایجاد شده روی پروژه را اعلام می‌کند. این commit ها به افراد عضو یک پروژه کمک می‌کنند تا دریابند چرا روی هر پروژه تغییر ایجاد شده است

۶- یک Pull Request ایجاد کنید

Pull Request به منظور ادغام هر شاخه‌ای در شعبه شخص دیگر ایجاد می‌شود. گیت هاب از این امکان به منظور اطلاع دادن به طرف‌های مربوطه در مورد درخواست الحاق تغییرات در شعبه استفاده می‌کند.

هر زمان که یک commit کامل شد می‌توانید درخواست Pull Request را صادر کنید. برای این که فرد مورد نظر شما در جریان ارائه این درخواست قرار بگیرید باید قبل از نام او یک علامت @ درج کنید.

برای ارائه درخواست Pull Request باید :

- به سربرگ Pull Request بروید.
- دکمه new Pull Request را بزنید.
- در کادر example comparisons شاخه‌ای را که ساخته‌اید پیدا کنید.
- مطمئن شوید تغییرات ایجاد شده همان چیزی است که مدنظرتان بوده است.
- درخواست خود را عنوان کنید و تغییرات را به‌طور خلاصه شرح دهید.
- روی گزینه Pull Request کلیک کنید.
- درخواست‌های Pull Request خود را ادغام کنید

ممکن است لازم باشد Pull Request خود را با دیگری ادغام کنید تا دیگران در بخش Master در جریان جزئیات کار و تغییرات آن قرار بگیرند. برای این کار :

- روی گزینه merge pull request کلیک کنید.
- سپس روی گزینه confirm merge را انتخاب کنید.
- شاخه‌ای را که ادغام کرده‌اید، پس از اینکه در Master قرار داده شد حذف کنید.

آیا می توان با گیت هاب در برنامه نویسی کار پیدا کرد؟

در این سایت نمی توان مانند لینکدین کار پیدا کرد ولی بسیاری از شرکت های برنامه نویسی از برنامه نویسان پروفایل گیت هاب را می خواند و بهتر است آن را در رزومه خود داشته باشید

تفاوت گیت هاب و گیت لب

گیت لب یکی از رقبا گیت هاب است. تفاوت بزرگی که این دو سایت با هم دارند این است که در گیت لب می توانید امکانات رایگان بیشتری را به نسبت گیت هاب پیدا کنید.

گوگل کولب چیست؟

کولب کوتاه شده واژه Colaboratoryاست. واژه Colaboratory یعنی آزمایشگاه مشترک، حالا Google Colab که کوتاه شده واژه Google Colaboratoryهست به معنای آزمایشگاه مشترک گوگل هست که شما می تونید به صورت آنلاین از اون استفاده کنید و در مرورگرهای خود کدهای پایتونی رو بنویسید و اجرا کنید. در واقع Goole Colab یک میزبان آنلاین برای پروژه های شما است که امکان پردازش با CPU یا GPU یا TPU را برای شما فراهم می کند. Google Colab می تواند کدهای شما را از گیت هاب و یا گوگل درایو شما فراخوانی یا به آن ها ارسال کند. گوگل کولب یک سرویس ابری برای هوش مصنوعی به‌صورت خاص یادگیر است. بنابراین، مثال‌های این پست **ممکن است** شامل کدهای پایتونی، مباحث شبکه عصبی، یادگیری ماشین، فریمورک پایتورچ، تنسورفلو و سایر مباحث مرتبط باشد.

سلول ها در کولب

هر نوتبوک از مجموعه‌ای سلول (Cell) تشکیل شده است .سلول محلی برای نوشتن کد یا متن هست .بله، در نوتبوک دو نوع سلول متن و سلول کد وجود دارد:

- **سلول کد :**در سلول کد، به‌راحتی می‌توانید کدهای پایتونی بنویسید و اجرا کنید.
- **سلول متن** :در سلول متن، مشابه با یک داکیومنت)ورد یا گوگل داک (می‌توانید توضیح بنویسید. مثلا در نوتبوک مرجع، یک سلول متن مشاهده می‌کنید که شامل متن با هدینگ، بولد، ایتالیک، لینک و تصویر هم هست .یعنی نوتبوک شما می‌تواند یک گزارش کار آنلاین باشد!

چگونه سلول ایجاد کنیم؟

- با استفاده از شورتکات **ctrl+m b** می‌توانید سلول کد بسازید.
- از گزینه Insert می‌توانید سلول متن و کد بسازید.

دستور ها در کولب

- **دستور pwd**: برای اینکه **Current Directory**را ببینید، باید از دستور pwd استفاده کنید. این دستور به شما می‌گوید شما در چه مسیری قرار دارید و به محتویات کدام پوشه یا مسیر دسترسی دارید.

❖ مسیر/content ، مسیر پیش‌فرض است و به همان مسیری که الان در Files می‌بینید، اشاره دارد. چطور مطمئن شویم؟ با دستور بعدی...

- **دستور Is:** اگر می‌خواهید محتویات یک پوشه یا مسیر را ببینید، باید از دستور ls استفاده کنید.
- **دستور cd:** با استفاده از دستور cd می‌توانید مسیر جاری یا Current Directory را به راحتی تغییر دهید.
- **دستور wget:** ا استفاده از دستور wget می‌توانید **فایل‌های مدنظرتان را دانلود** و در نوتبوک از آنها استفاده کنید
- **دستور unzip:** معمولا فایل‌های دانلودی، فایل‌های فشرده هستند و باید از حالت فشرده استخراج شوند. مثلا فایل بالا از نوع zip است **.برای استخراج از حالت فشرده باید از دستور unzipاستفاده کنید**

دستور pip: برای نصب پکیج‌های پایتونی در گوگل کولب کافی است از دستور **pip install!** استفاده کنید. به‌همین راحتی، تنها با افزودن علامت ! می‌توانید از این دستور در گوگل کولب بهره ببرید

C++

در زبان پیشرفته سی پلاس پلاس بر خلاف زبان های سطح پایین از شی گرایی استفاده می شود. به طور مفصل در بخش تعریف کلاس به بیان این موضوع می پردازیم اما در این جا نیاز است بدانیم که ساختار برنامه در C++ از قسمت های زیر تشکیل شده است.

1.مشخص کردن header ها.

ما در برنامه سازی همیشه از صفر شروع نمی کنیم و می توانیم از برنامه ها و توابع و کلاس هایی که از قبل طراحی شده استفاده کنیم. به این منظور ما می توانیم با ذکر نام آن ها در ابتدای برنامه یا به اصطلاح با includeکردن آن ها از آن ها استفاده کنیم. گاهی با include یک فایل، کلاس و کتابخانه می توانیم به تعداد زیادی از توابع و اعضای آن ها دسترسی داشته باشیم. فرضا اگر اسم header ما myheader باشد. نحوه include کردن آن به شکل زیر است:

```
# include
```

2.مشخص کردن نوع namespace که مفهوم آن در آینده توضیح داده خواهد شد.
در ورژن های جدید کامپلرهای استاندارد (iso)می بایست نوع name space را مشخص کنیم. نوع namespaceاستفاده شده در برنامه هایی که ما می نویسیم std می باشد. این کار به صورت زیر انجام می گیرد:

```
using namespace std;
```

3.تعریف یک سری کلاس هایی که پیاده سازی آن ها به عهده خودمان می باشد.
4.تعریف توابعی که در بدنه اصلی برنامه یا همان main قصد استفاده از آن را داریم.
5.و در نهایت main برنامه.
در سی پلاس پلاس تمام دستورات اجرایی در main برنامه پیاده سازی می شود. یعنی در قسمت های قبل فقط می توان گفت که یک سری توابع و کلاس و ... وجود دارد اما برای استفاده از آن ها بایستی که در main برنامه دستورات مربوطه را وارد کنیم. به ترتیب زیر:

```
int main(){
مجموعه دستورات
return 0;
}
```

)البته این نوع تقسیم بندی برای شروع کار می باشد، به طوری که در برنامه های بزرگ کلاس ها را در یک قسمت، توابع را در یک قسمت دیگر و... طراحی می کنند و یا برای نوشتن visual اصلا از main به این شکل استفاده نمی شود، اما برای آموزش از همین تقسیم بندی استفاده می کنیم(.

تقریبا تمام برنامه نویسان بزرگ دنیا اولین برنامه ای که به زبان سی پلاس پلاس می نویسند برنامه ای است که پیغام hello world را چاپ می کند.
این برنامه به صورت زیر می باشد:

```
# include
using name space std;

int main(){
  cout<<"Hello world!";
return 0;
}//end programme
```

در خط اول header را از نوع iostream مشخص کردیم.

iostream:این header تشکیل شده از سه کلمه in out stream بر جریان های (stream) ورودی (in)و خروجی (out)نظارت دارد. منظور از جریا ن ورودی اطلاعاتی است که کاربر با استفاده از توابع خاص خود به کامپیوتر می دهد و جریان خروجی هم اطلاعاتی است که روی صفحه مانیتور نمایش داده می شود.
تابع **cout:**این تابع به منظور چاپ روی صفحه نمایش استفاده می شود. نحوه استفاده از تابعcout به این شکل است که برای چاپ از دو علامت <به صورت پشت سر هم (مانند مثال) استفاده می کنیم و سپس نوع خروجی را در ادامه آن می نویسیم. اگر نوع خروجی در هر مرحله فرق کند می بایست برای جدا کردن آن ها باز از دو علامت <استفاده کنیم. در اینجا برای چاپ یک رشته از کاراکتر آن را در داخل دو گیومه قرار می دهیم.
تابع **cin:**به منظور گرفتن دیتا از کاربر از این تابع استفاده می کنیم. این تابع تا زمانی که به space یا Enterیا tab برسد اطلاعات را از کاربر دریافت می کند. برای دریافت نوع داده از دوعلامت >پشت سر هم استفاده می کنیم. اگر نوع داده ای عوض شد مانند cout دوباره از >>استفاده می کنیم.
به این منظور فرض می کنیم می خواهیم یک عدد از نوع صحیح (که در بخش های بعدی نوع داده را توضیح می دهیم) از کاربر بگیریم. نماد a را به عنوان عدد صحیح در نظر می گیریم.دستور به شکل زیر خواهد بود:

```
cin>>a;
```

وقتی برنامه اجرا می شود کامپیوتر منتظر می ماند تا یک عدد از کاربر بگیرد. فرض می کنیم کاربر عدد ۴ را وارد کرد. اگر بعد ازدستور بالا دستورچاپ را وارد نماییم، عدد ۴ روی نمایشگر چاپ می شود. دستور چاپ آن به شکل زیر می شود چرا که عدد ۴ در نماد a ریخته شده است.

cout<

در خط بعدی نوع namespace را مشخص نموده ایم. و سپس main برنامه را نوشته ایم. در پایان نیز مقدار ۰ را با استفاده از دستور return به تابع بر می گردانیم.

مفهوم comment

در برنامه های بزرگ شما می بایست در نقاط مختلف پیغام هایی را درج کنید که اگر بعد از گذشت مدتی به آنها مراجعه کردید از نحوه کار برنامه آگاهی پیدا کنید. به این دستورات که تنها برنامه نویس آن ها را می بیند و کامپایلر در هنگام کامپایل و اجرای برنامه آنها را نادیده می گیرد comment می گویند. در تمام زبان ها این مفهوم وجود دارد .در سی پلاس پلاس دو راه برای comment گذاری وجود دارد.

۱. گذاشتن :// همانطور که در خط آخر برنامه مشاهده می کنید از جایی که دو علامت / را گذاشتیم تا انتهای خط تبدیل به comment می شود.

۲.روش / و *: گاهی commentشما ممکن است بیش از یک خط باشد به این منظور در ابتدای کامنت */ و در انتهای آن /* را قرار می دهیم. در این صورت هر چیز که میان این دو است به عنوان کامنت شناخته می شود .

```
/* this is for
test*/
```

یک برنامه نویس خوب کسی است که تمام اصول اولیه را رعایت کند. سعی کنید از همین اول که برنامه های شما کوچک است به کامنت گذاری اهمیت بدهید.
حال اگر برنامه را اجرا کنید. پیغام مورد نظر روی صفحه چاپ می شود و سپس صفحه بسته می شود. برای اینکه بتوانید پیغام را مشاهده کنید و تا زدن یک دکمه صفحه بسته نشود می توانید قبل از return 0 دستور system("pause")1 را وارد نمایید. تابع کلیدی system تابعی است که می توان در داخل آن تمام دستورات که در cmd ویندوز موجود می باشد را وارد نماییم. کد به صورت زیر خواهد شد

```
#include
using name space std;

int main(){
    cout<<"Hello world!";
```

```
system("pause");
return  0;
}//end programme
```

حال با اجرای برنامه پیغام مورد نظر چاپ می شود و بعد از آن پیغام press any key to countinue چاپ می شود و تا زمانی که دکمه ای زده نشود برنامه بسته نمی شود.

اگر دقت کنید تمام این نوشته ها در یک خط چاپ می شود و شاید از نظر بیننده جالب به نظر نرسد. می توانید در تابعcout بعد از آوردن رشته دستور end lineرا بدهید و به کامپیوتر بگویید که بعد از چاپ به سر خط بعدی برود. کد به صورت زیر در می آید:

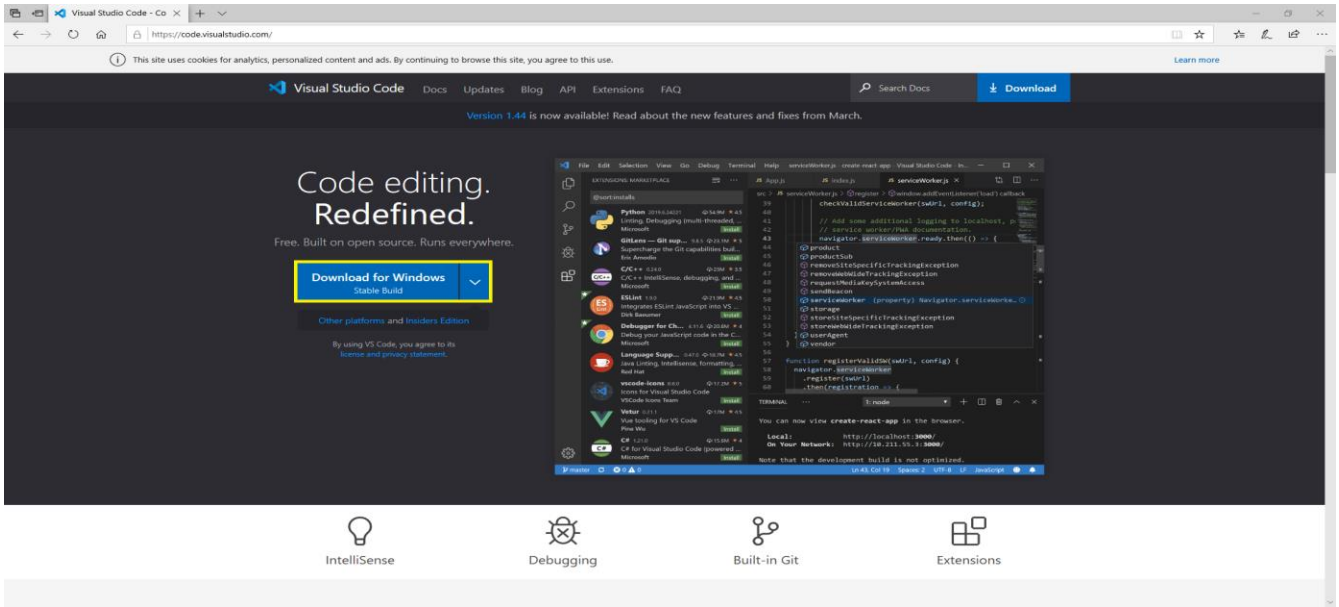
```
#include
using  name space std;

int  main(){
     cout<<"Hello  world!"<<endl;
system("pause");
return   0;
}//end  programme</endl;
```

راهنمای نصب VS CODE

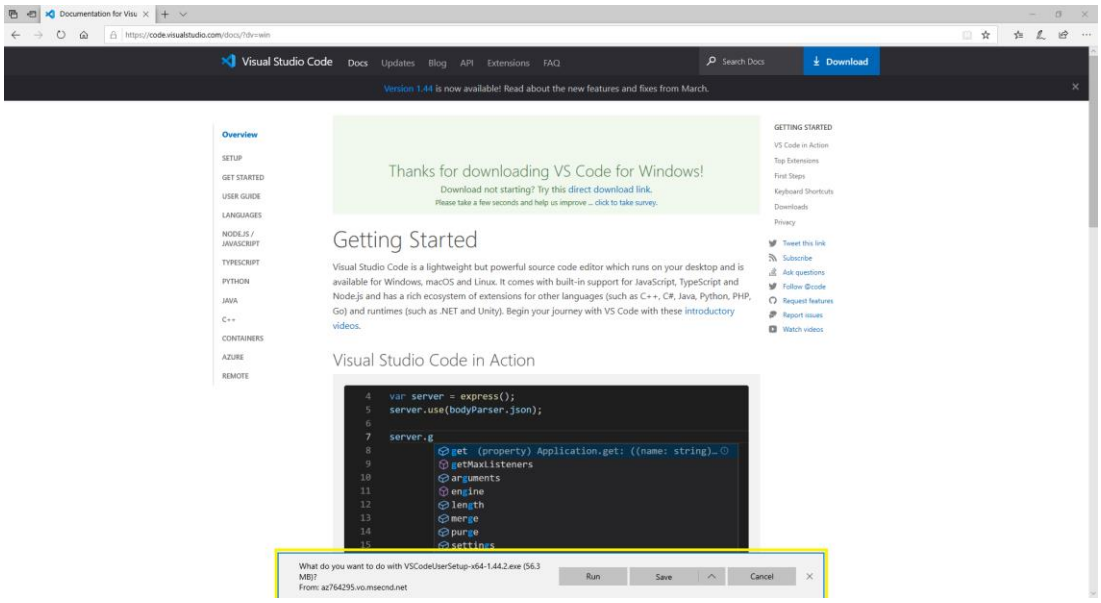
ویژوال استودیو کد یک ویراشگر متن توسط شرکت مایکروسافت برای توسعه برنامه تحت سیستم عامل ویندوز، لینوکس و مک است. این ویراشگر قابلیت هایی نظیر اشکال زدایی ((Debuggingکنترل نسخه (Version control)مانند Git و اتصال به GitHub، برجسته کردن دستور ((Syntax highlighting، تکمیل سازی کد هوشمند ((Intelligent code completionتکه کد های آماده ((Snippets، بازسازی ساختار کد ((Refactoringو مجموعه کاملی از افزونه ((Extensionرا برای تسهیل کار دارا می باشد. در ادامه به نحوه نصب آن می پردازیم.

برای دانلود به وبسایت زیر مراجعه کنید:



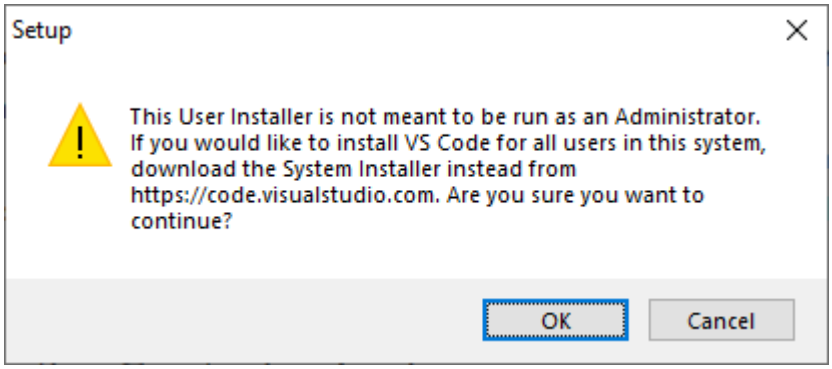
ظاهر وب سایتvscode

گزینه Download را انتخاب کنید.



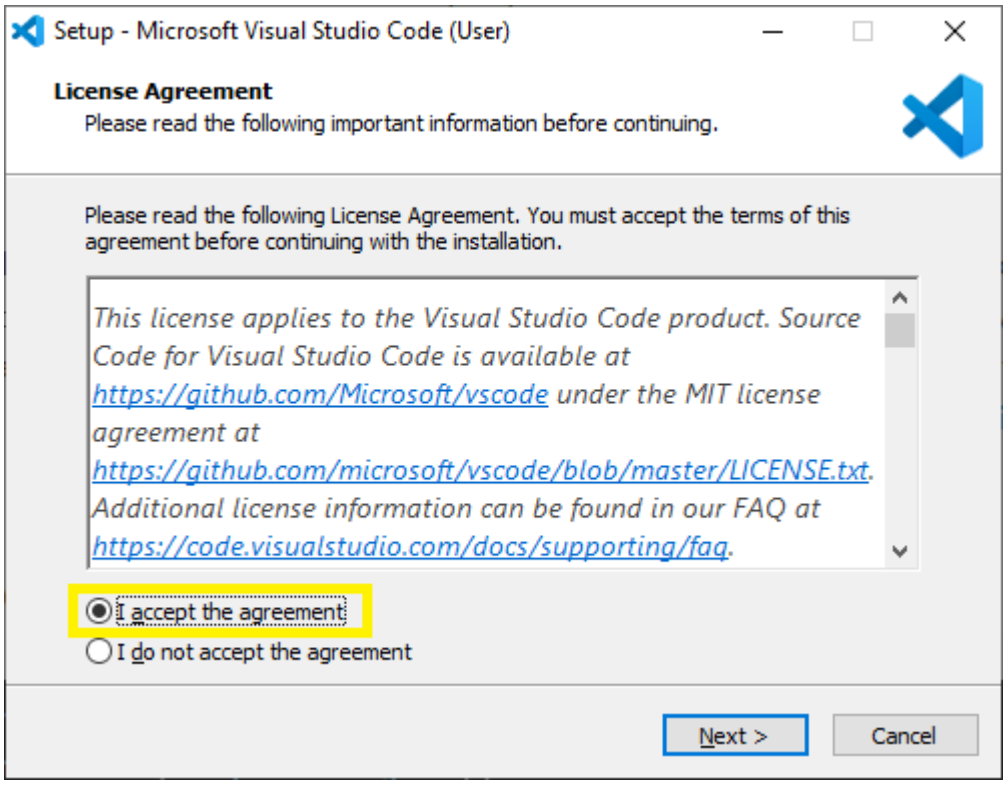
پیام دانلود در مرورگرEdge

صبر کنید تا پیام دانلود نمایان شود و پس از دانلود آن را اجرا کنید.

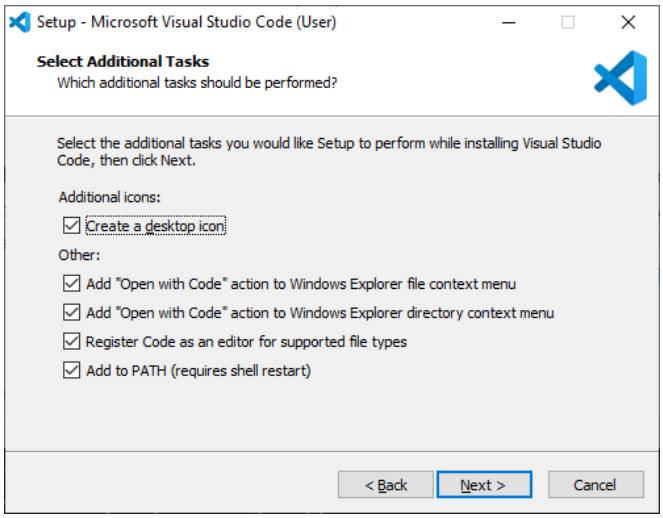


هشدار نصب

در صورت مواجه شدن با همچین پیامی OK را انتخاب کنید. دقت داشته باشید VS Code فقط برای کاربر فعلی در ویندوز نصب می گردد.



توافق نامه



با انتخاب I accept the agreement و سپس دکمه Next توافق نامه را بپذیرید.

محل نصب

در این صفحه می توانید محل نصب را تغییر دهید. پیشنهاد می شود محل نصب را تغییر ندهید.

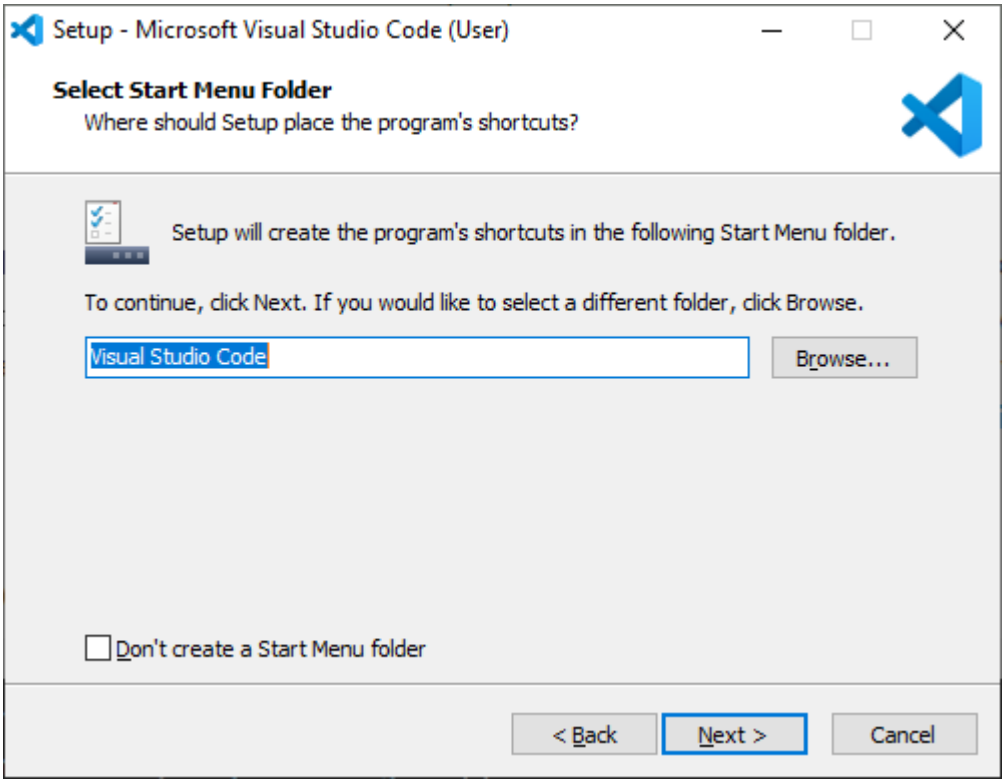
نام پوشه منوی استارت

نام پوشه برنامه در منوی استارت ویندوز را می توانید تعیین کنید. این صفحه حائز اهمیت نیست و می توانید با انتخاب Next از آن عبور کنید.

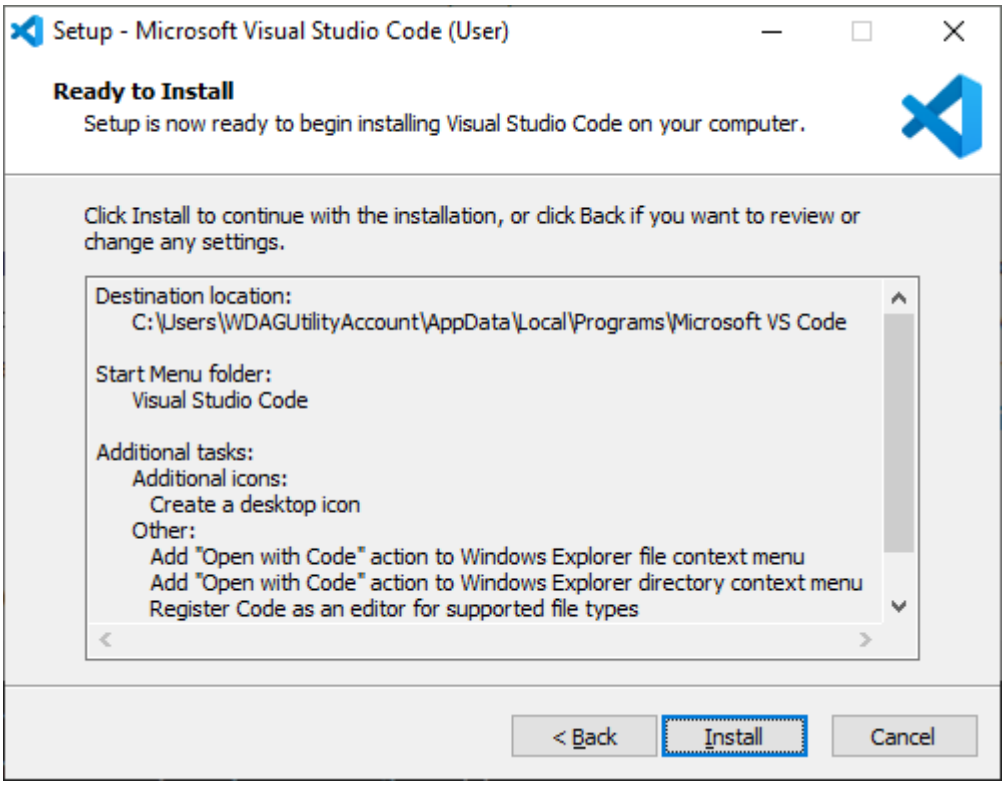
وظایف افزوده

در این صفحه پیشنهاد می کنم تمام موارد را تیک بزنید. توضیح هرکدام:

- گزینه Create a desktop icon بر روی صفحه دسکتاپ شما یک میانبر می سازد.

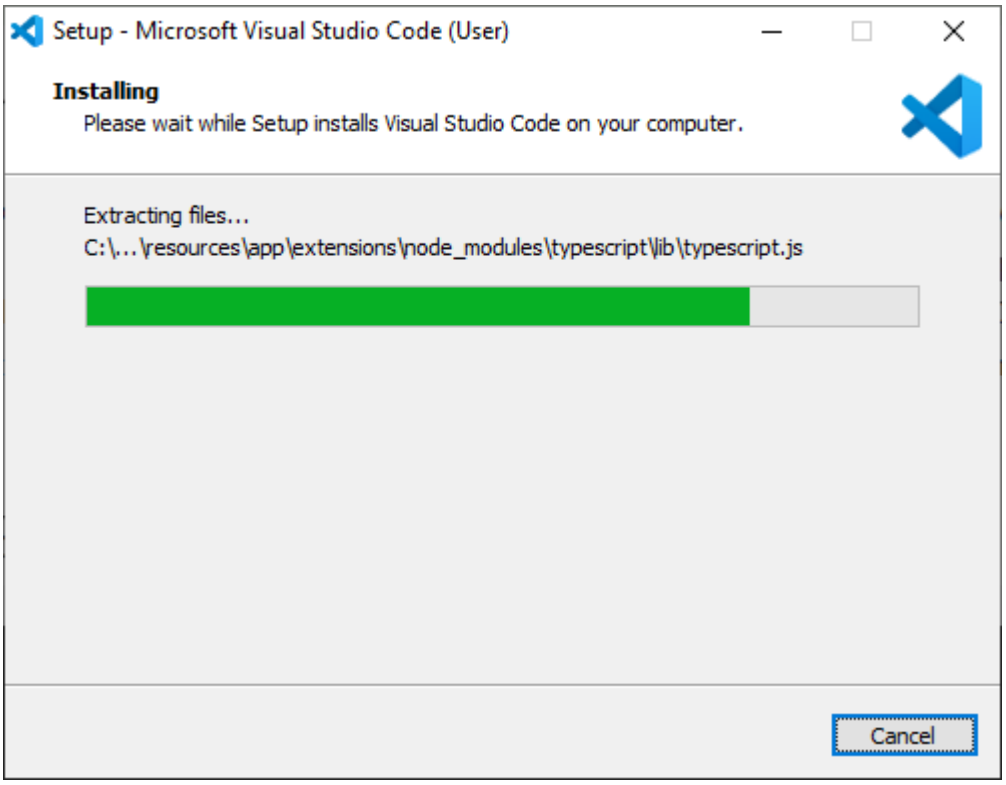


- گزینه Add "Open with Code" action to Windows Explorer file context menu به شما این امکان را می دهد تا با انتخاب فایل و کلیک راست، گزینه Open with code برای ویرایش هر فایلی نمایان شود.
- گزینه Add "Open with Code" action to Windows Explorer directory context menu به شما این امکان را می دهد تا با انتخاب پوشه و کلیک راست، گزینه Open with code برای افزودن پوشه به محیط کار (Workspace)نمایان شود.
- گزینه Register Code as an editor for supported file types تمام فایل های متنی با پسوند پشتیبانی شده در سیستم شما با VS Code باز می شوند. (مانند .cpp و .py)
- گزینه Add to PATH (requires shell restart) عبارت code را به آدرس PATH ویندوز اضافه می کند که به شما امکان اجرای VS Code با دستور code از خط فرمان را می دهد. **این گزینه را حتما فعال** کنید.



آماده نصب

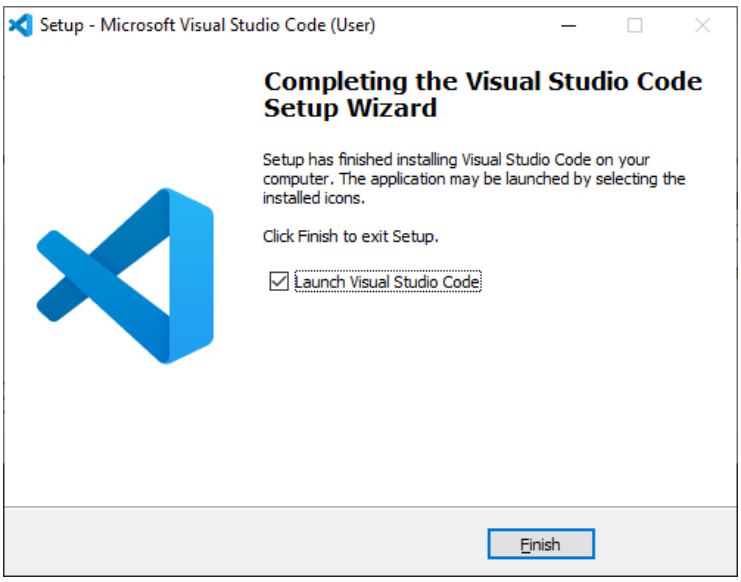
حال برنامه آماده نصب می باشد. Install را انتخاب کنید.



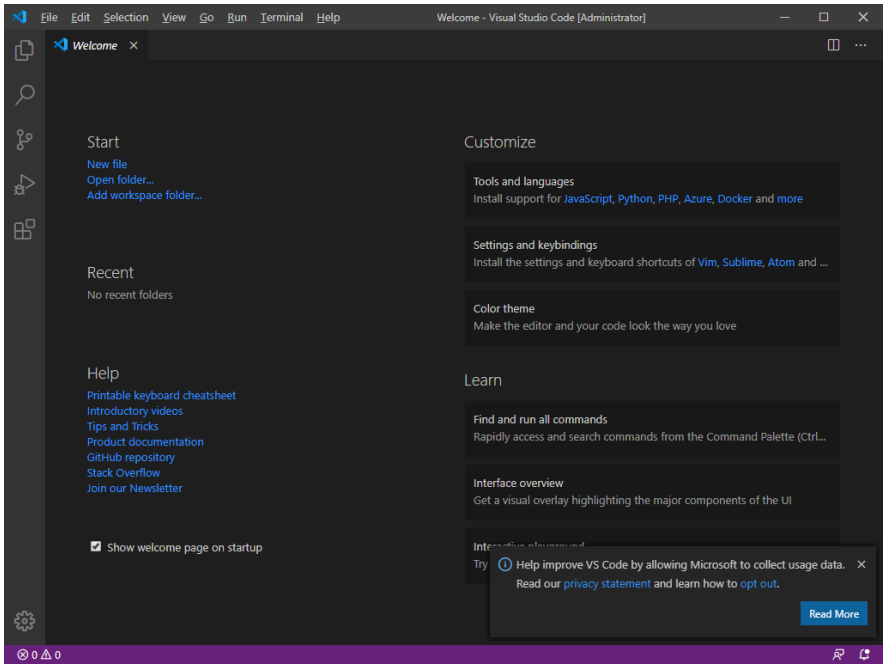
در حال نصب

صبر کنید تا نصب به پایان برسد.

پایان نصب



برنامه VS Code با موفقیت بر روی سیستم شما نصب شده است. با انتخاب Launch Visual Studio Code و سپس Finish آن را اجرا کنید.



ویژوال استودیو کد (**Visual Studio Code**

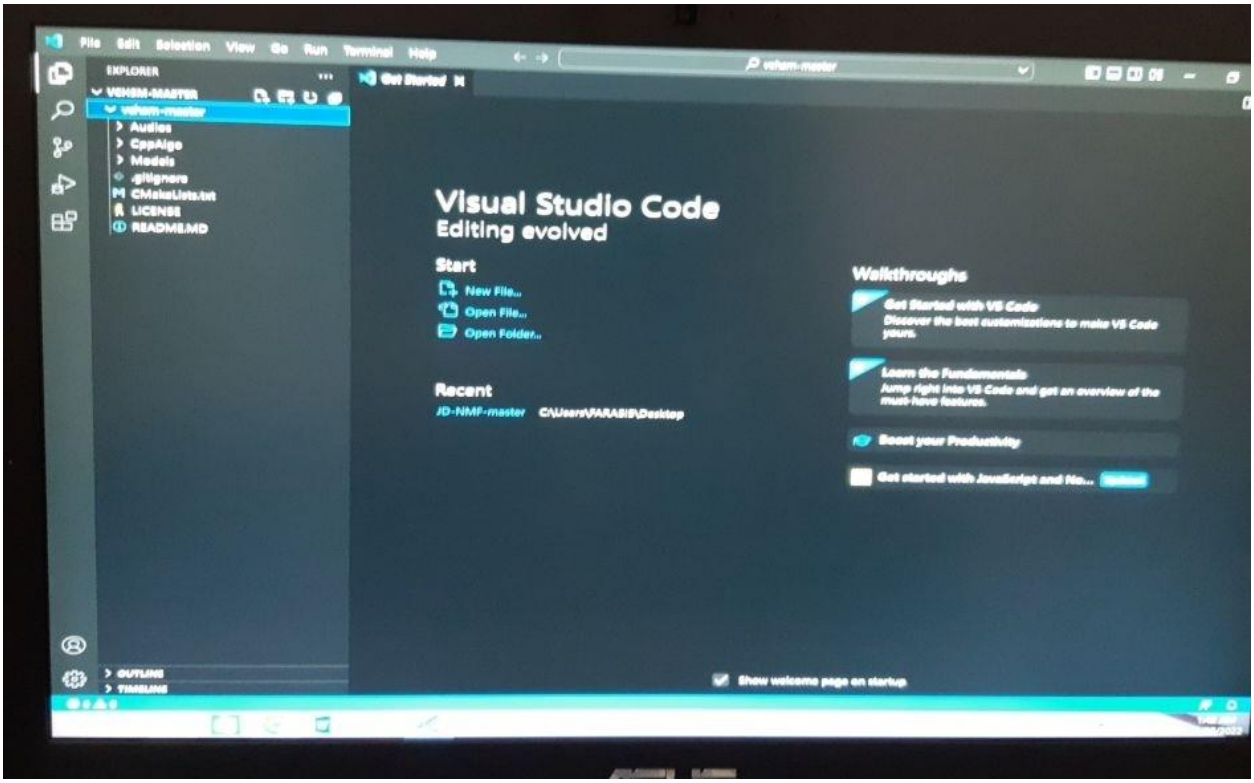
یا به اختصار (VSCodeیک ویرایشگر کد متن باز برای لینوکس، مک و ویندوز می‌باشد که به صورت درونی از تکمیل کد هوشمند (انگلیسی: (intelligent code completion، برجسته سازی نحو (syntax highlighting)، بازسازی کد (انگلیسی: (code refactoring، عیب یابی ((debuggingو تکه کد ها ((snippetsپشتیبانی می کند. ویژوال استودیو کد با C++، نود.جی‌اس((Node.js، فریم ورک الکترون (چارچوب نرم‌افزاری) و دیگر تکنولوژی‌های توسعه وب و بر اساس ویرایشگر کد تحت وب موناکو نوشته شده‌است. حجم کمتر، برخورداری از یک مخزن بزرگ از افزونه‌ها آن را رقیب جدی برای دیگر ویرایشگرها قرار داده است. این نرم‌افزار توسط مایکروسافت توسعه داده شده و هم‌اکنون به‌طور رایگان و)Open Sourceمتن باز) در دسترس است.

ویژوال استودیو کد در تاریخ ۲۹ آوریل سال ۲۰۱۵ توسط **مایکروسافت** در **کنفرانس بیلد** معرفی شد و پس از مدت کوتاهی یک نسخه پیش نمایش از آن معرفی شد. در ۱۸ نوامبر همان سال ویژوال استودیو کد، تحت پروانه ام‌آی‌تی در سایت **گیت‌هاب** (Github.com) منتشر شد.

در نظرسنجی که در سال ۲۰۱۸ توسط وب سایت Stack Overflow انجام گرفت، **ویژوال استودیو کد** به عنوان **محبوب‌ترین ابزار توسعه** با **رای ۳۴,۹ درصد** از **۷۵،۳۹۸ رای** انتخاب شد.

اجرای پروژا در VSCODE

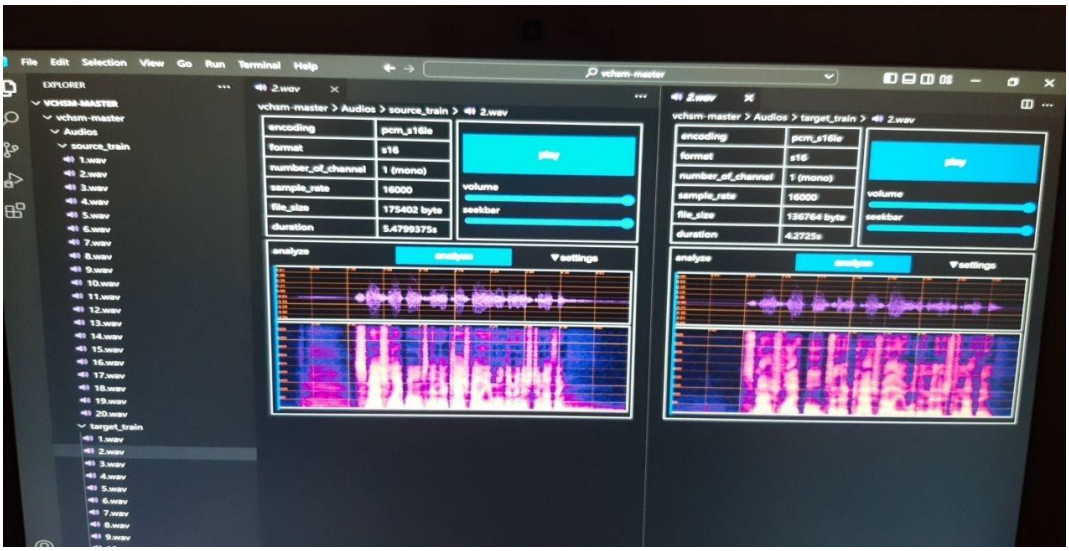
در ابتدا برنامه گیتاب دسکتاب و VSCODE را نصب می کنیم فایل های زیپ مربوط به پروژه را دانلود کرده و از حالت زیپ خارج میکنیم سپس وارد محیط برنامه می شویم.



از روی گزینه OPEN FOLDER فایل مربوط پروژه را پیدا کرده و باز میکنیم در نوار بالا چند گزینه وجود دارد که شامل ....,FILE, EDITE گزینه ی VIWE عبارت PROBLEM را انتخاب می کنیم دبرای ادامه لازم است EXTATION را نصب کنیم و برنامه ی AUDIO PREVIEW را نصب کنیم بعد از نصب برنامه به پروژه برمیگردیم و فایل های ویس را باز میکنیم

با توجه به نصب برنامه مورد نیاز ویس ها بدون هیچ مشکلی پخش می شوند گزینه آنالیز را انتخاب کرده و سپس گزینه PLAYرا انتخاب می کنیم

حال علاوه بر داشتن صدا نمودار هم داریم که یکی از آنها اسپکتوگرام و دیگری طیف موج در نمودار موج یک سری فاصله زمانی داریم که نشان دهنده نقاط سکوت در صدا است و این نقاط دذ اسپکتوگرام هم قابل مشاهده است



زمانی که می خواهیم موج خروجی را مشاهده کنیم متوجه می شویم قسمت های سکوت کامل حذف شده و صدای ما سریع تر نصب می شود طول موج ورودی ما ۵,۴ ثانیه بوده اما طول موج خروجی ما ۴,۲ تغییر کرده و نشان دهنده این است که اطلاعاتی که مفید نبوده را حذف کرده

این برنامه کمک میکند متوجه شویم چه تعییراتی در سیگنال بوجود امده