

## تشخیص اعداد بیان شده ی کاربر

محمدعلی طهوری ۹۳۲۳۰۴۹ - صالح غلام زاده ۹۰۲۳۷۲۴

۱۶ تیر ۱۳۹۶

### ۱ هدف پروژه

ضبط صدای کاربر و پردازش روی آن به منظور تشخیص اعداد گفته شده توسط کاربر و سپس نمایش آن برای کاربر در محیط گرافیکی .

### ۲ ضبط صدا

وقتی کاربر دکمه Play را فشار داد تابع start اجرا می شود در این تابع کلاس processthread فراخوانی شده و یک شی از آن ساخته می شود سپس این thread اجرا می شود .

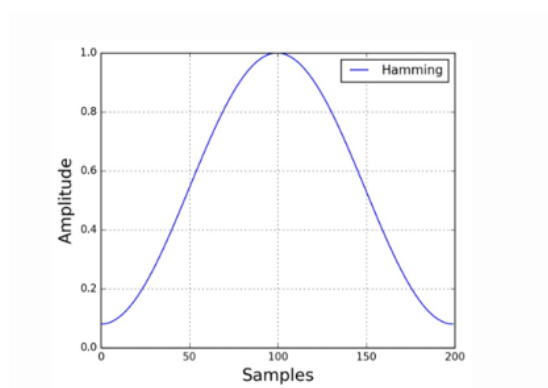
فرمت ذخیره صدا به صورت Wav است همچنین به صورت Monotrack و با فرکانس نمونه برداری ۴۴۱۰۰ سمپل در ثانیه ذخیره می شود .

در تابع run در کلاس processthread با استفاده از ماژول pyaudio ابتدا فرمت ذخیره سازی تنظیم می شود سپس یک لیست خالی به نام audiolist تعریف می شود و زمانی که ضبط صدا شروع شد دیتا داخل لیست audiolist ذخیره می شود و زمانی که کلید stop زده شود به تابع stop رفته و stop\_flag که به صورت پیش فرض False بوده True می شود و ضبط صدا متوقف می شود . سپس این فایل صوتی با نامی که خود ما مشخص کردیم ذخیره می شود .

### ۳ پردازش صوت

در این بخش ابتدا فایل ذخیره شده trim می شود یعنی تک تک اعداد گفته شده جدا می شوند و به صورت فایل مجزا در می آیند و در پوشه Trimmed\_voice ذخیره می شوند . سپس با استفاده از دستورات os بر روی یکی یکی فایل ها تابع feature\_extract از ماژول feature\_extraction اجرا می شود .

در تابع feature\_extract فایل صوتی خوانده می شود و به صورت یک لیست در signal ریخته می شود و فرکانس نمونه برداری نیز در متغیر sample\_rate قرار می گیرد. چون سیگنال صوت در فرکانس های بالا ضعیف است و دامنه کمی دارد با استفاده از pre\_emphasis دامنه سیگنال های فرکانس بالاتر را تقویت می کنیم. سپس سیگنال صوت را به کمک پنجره hamming همان طور که در شکل **شکل ۱** مشاهده می کنید هر ۲۵ میلی ثانیه فریم بندی می کنیم و در لیست frames می ریزیم، پنجره ها با هم ۱۰ میلی ثانیه اشتراک دارند.



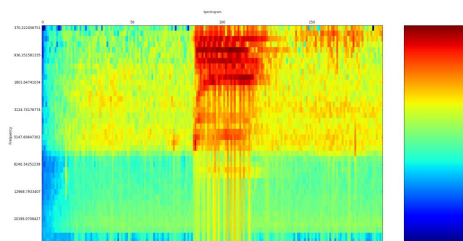
شکل ۱: پنجره hamming

فرمول پنجره hamming نیز در معادله **معادله ۱** نشان داده شده است.

$$\omega[n] = 0.54 - 0.46 \cos\left(\frac{2\pi n}{N-1}\right) \quad (1)$$

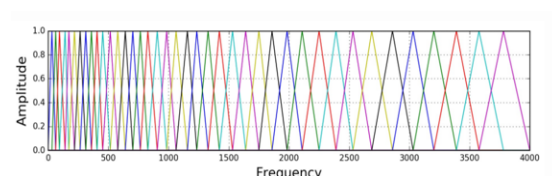
سپس fft را روی frames اجرا می کنیم تا اطلاعات و ویژگی های زمانی و فرکانسی سیگنال مورد نظر استخراج شود. همان طور که در شکل **شکل ۲** قابل مشاهده است spectrogram زمانی و فرکانسی یک سیگنال برای نمونه رسم شده است.

## ۴ نمایش گرافیکی



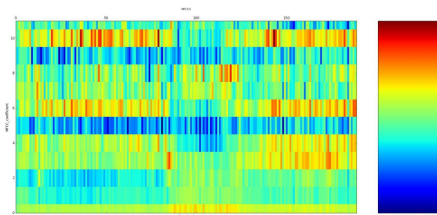
شکل ۲: spectrogram زمانی و فرکانسی

سپس روی خروجی مرحله قبل filter bank را اعمال می کنیم . filter bank برای مدل سازی رفتار غیر خطی گوش انسان است . تعداد این فیلتر بانک ها ۴۰ تا است و به صورت شکل **شکل ۳** می باشد .



شکل ۳: filter banks

سپس ضرایب کپسترال (MFCC) را به دست می آوریم و به عنوان ورودی به classifier می دهیم تا هر سیگنال های صوت را طبقه بندی کند . این ضرایب ۱۲ تا هستند . در نهایت خروجی این MFCC را بر می گردانیم . در شکل **شکل ۴** نمودار ضرایب MFCC بر حسب زمان نشان داده شده است .



شکل ۴: MFCC بر حسب زمان

## ۴ نمایش گرافیکی

با استفاده از تابع uic فایل main.ui که با Qt ساختیم را در متغیر Form ذخیره می کنیم و به عنوان ورودی به کلاس Main\_Class که کلاس اصلی هست می دهیم .

در کلاس اصلی یک figure می سازیم که شامل push button ها و دو Text Edit هست که در یکی وضعیت پردازش نمایش داده می شود و در دیگری اعدادی که کاربر گفته است .