

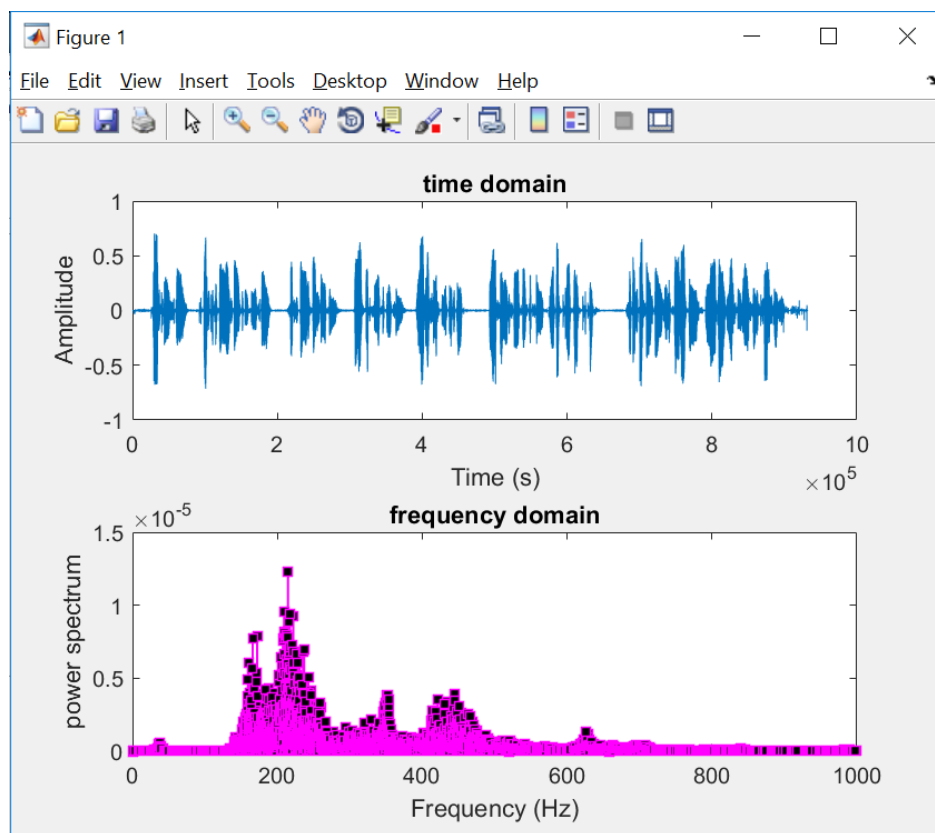
گزارش پروژه پایانی درس سیگنال‌ها و سیستم‌ها

ارمغان سرور 9531807

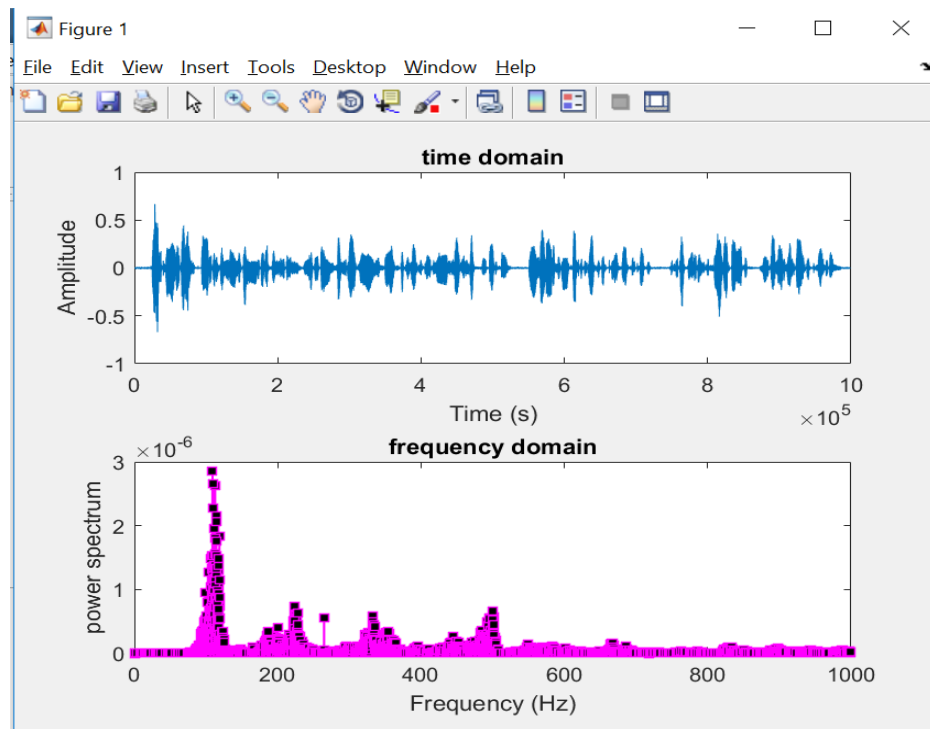
قدم 2

در این مرحله پس از مشاهده ویدئوی بخش اول پروژه نمودار مربوط به سیگنال (داده‌ی صدا) در حوزه زمان و طیف توانی آن در حوزه فرکانس رسم گردید. (در بازه ی 0 تا 1000 برای نمایش). برنامه‌ی این بخش در تابع `plotPowerSpec.m` نوشته شده است. خروجی برای 5 فایل اول به شرح زیر است:

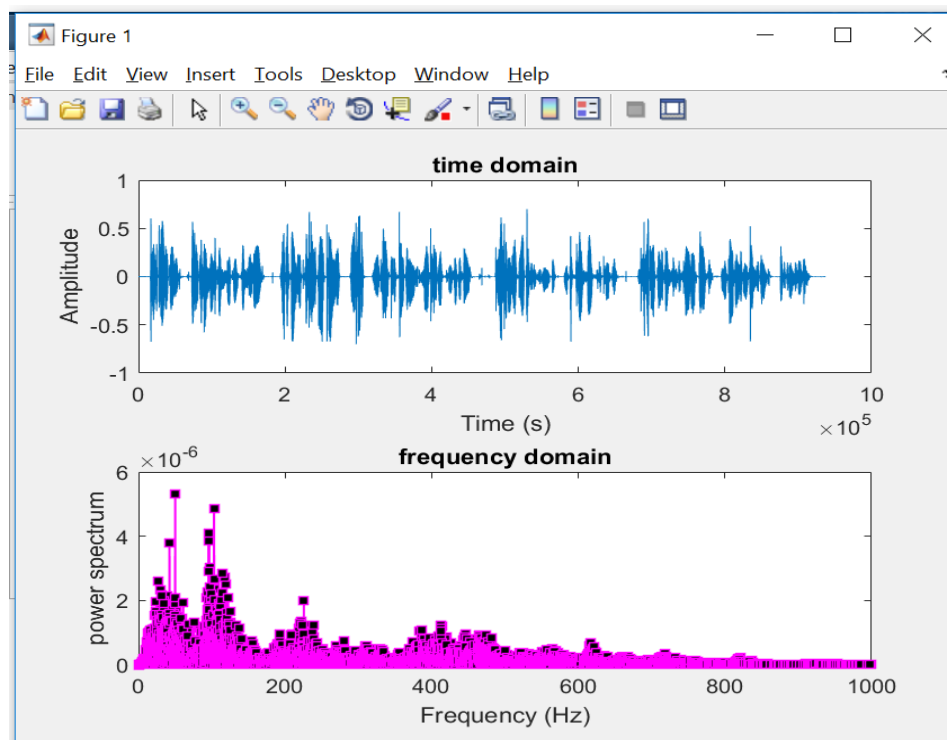
V0.mp3



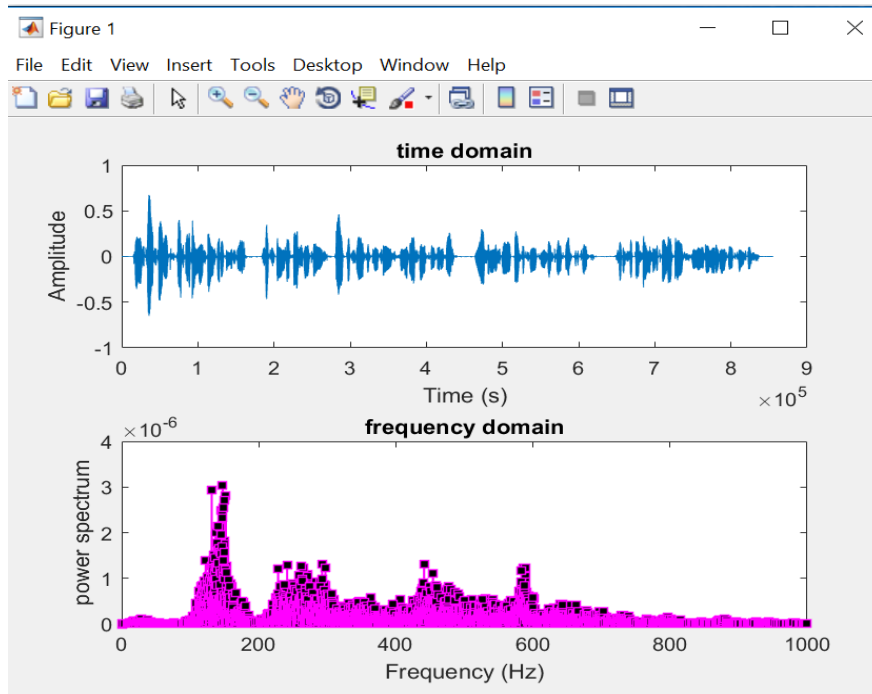
V1.mp3



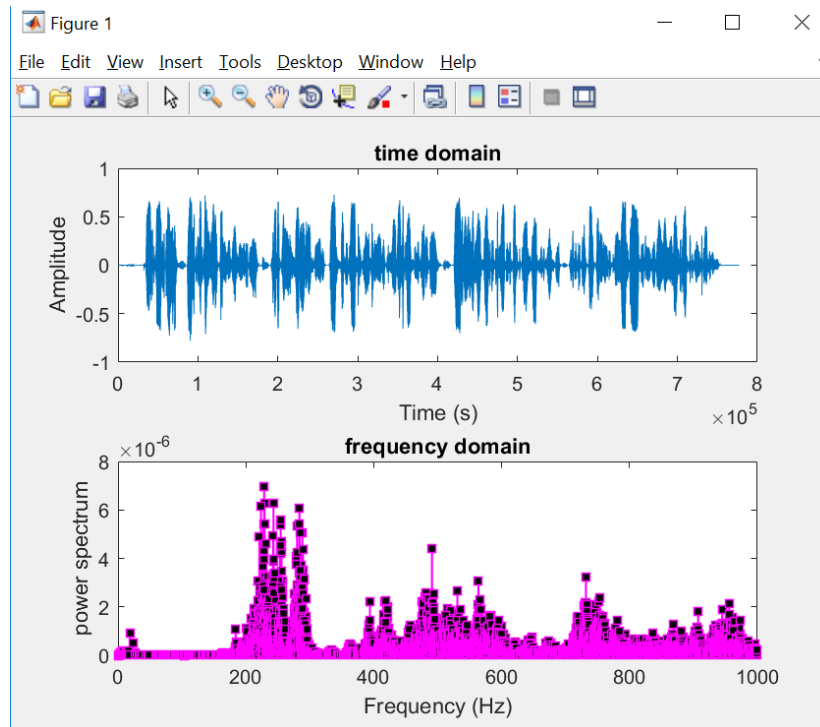
V2.mp3



V3.mp3



V4.mp3



قدم 3

تابع استخراج فرکانس قله در `getPeak.m` پیاده‌سازی شده‌است. برای این بخش باید `index` مربوط به بزرگترین مقدار سیگنال `power spectrum` را بدست می‌آوریم که همان بیشترین فرکانس است.

نهایتاً در `run.m` آن را فراخوانی کردم و بر اساس نزدیکی فرکانس قله هر فایل ورودی به فرکانس‌های 212 و 122 و با استفاده از تابع `movefile` نامشان را تغییر داده و تقسیم‌بندی به دو دسته مرد و زن را انجام دادم.

قدم 4

روش‌های افزایش دقت سیستم تشخیص صدا:

✚ استفاده از **Random Forest classifier**: این طبقه‌بندی یک تخمین‌زن است که می‌تواند `overfitting` را کنترل کرده و دقت پیش‌بینی را افزایش دهد. در واقع ساختاری متشکل از چندین درخت تصمیم‌گیری می‌باشد. برای مثال اگر مجموعه داده `N` تایی به عنوان نمونه که هر کدام `C` بعد (feature) دارند به همراه کلاس مقصودشان داشته باشیم، آن‌ها را به زیرمجموعه‌هایی تقسیم می‌کنیم (مثلاً ماتریس داده‌ها و کلاس‌ها را به زیرماتریس‌هایش تقسیم می‌کنیم) و برای هر کدام از این زیرمجموعه‌ها از یک الگوریتم درخت تصمیم استفاده می‌کنیم. حال اگر بخواهیم یک نمونه جدید را کلاس‌بندی کنیم باید پاسخ هر درخت تصمیم برای آن را بدست آوریم و بیشترین رای بین آن‌ها، کلاس مقصد خواهد بود.

✚ استفاده از **Multi-Layer Perceptron**: این شبکه عصبی از چندین `perceptron` تشکیل شده است و دارای یک لایه ورودی برای دریافت سیگنال، یک لایه خروجی برای تصمیم‌گیری یا پیش‌بینی و تعدادی لایه داخلی که موتور محاسباتی آن هستند می‌باشد. این روش معمولاً برای مسائل `supervised` به کار می‌رود. طی یادگیری الگوریتم، وزن‌ها به مرور تغییر می‌کنند تا خطا به کمترین حالت خود برسد. این شبکه از نوع `feed-forward` است که شامل دو بخش زیر است:

- **Forward pass**: یک دور ورودی از لایه‌ها عبور می‌کند تا به لایه آخر برسد و در آن‌جا با `label` ها مقایسه شود.
- **Backward pass**: با استفاده از `back propagation` تغییرات در مقدار بایاس و وزن‌ها صورت می‌گیرد تا نهایتاً خطا کم شود. می‌توان آن را با روش‌هایی مثل گرادیان انجام داد.

این رفت و برگشت‌ها به قدری تکرار می‌شوند تا مقدار خطا همگرا شود.

✚ استفاده از **Support Vector Machine**: از این الگوریتم یادگیری برای انتقال متغیرها به ابعاد بالا استفاده می‌شود. این ماشین در بهینه‌های محلی متوقف نخواهد شد و از آن جا که مستقیماً با فضای با ابعاد بالا ارتباط نداشته و با ضرب داخلی عمل می‌کند، مشکلات پیچیدگی در محاسبات را نخواهد داشت.

✚ استفاده از **Logistic regression**: در مسائل کلاس بندی و برای تشخیص ارتباط میان یک متغیر وابسته باینری و یک یا چند متغیر مستقل به کار می‌رود. تفاوت آن با linear regression در این است که به جای تشخیص پیوسته، true و false بودن را مشخص می‌کند و برای آن از likelihood استفاده می‌کند.