بسمه تعالی

دانشگاه صنعتی شریف

دانشکده مهندسی برق

فاز نهایی پروژه سیگنال‌ها و سیستم‌ها

نگارنده

امیرمهدی سلیمانی‌فر

شماره دانشجویی

۹۸۱۰۱۷۴۷

استاد درس

دکتر خلج

بهار ۱۴۰۰

پروژه مربوط به شناسایی نت‌های موسیقی از یک صفحه نت و نواختن آن‌ها و یا ذخیره در یک فایل صوتی است. در ابتدا وابستگی‌های مربوط به پروژه توضیح داده می‌شود و سپس روند اجمالی کار کردن کد بررسی خواهد شد. جزئیات بیشتر در خود کد نیز موجود است و قسمت‌های مختلف توضیحات جداگانه‌ای دارند.

وابستگی‌های پروژه که احتمالا نیاز به نصب جداگانه دارند، شامل موارد زیر است:

$OpenCV - python$
$numpy$
$MIDIutil$
$music21$
$pdf2image$
PIL

ورودی برنامه از طریق ترمینال و با دستوری به صورت زیر داده خواهد شد:

$python\ main.py$ "path-to-PDF-of-music-sheet"

نت‌هایی که قرار است مورد بررسی قرار بگیرند در پوشه ***test*** موجود هستند. در ابتدا برنامه با خوانش فایل ***PDF*** و تبدیل آن به تصویر با استفاده از کتابخانه ***pdf2image*** کار خود را شروع می‌کند. مواردی هم قبل از اجرای برنامه ست می‌شوند که مربوط به خواندن فایل‌های ***template*** است که قرار است در آینده برای نمونه برداری مورد استفاده قرار بگیرند.

پس از خوانش تصویر ابتدا با یک فیلتر نویزگیری مقدار نویز را کاهش داده و سپس با استفاده از متد ***Otsu*** که یک متد برای باینری کردن است تصویر را به دو رنگ سفید و مشکی در می‌آوریم. از آنجایی که رنگ در برگه نت نشان‌دهنده چیزی نیست وجود دو رنگ به شناسایی بهتر نت‌ها و خطوط حامل کمک خواهد کرد.

متدی که برای شناسایی خطوط حامل مورد استفاده قرار می‌گیرد به اینصورت است که در ابتدا با گذر عمودی از صفحه تعداد پیکسل‌های در عرض که پشت‌سرهم سیاه هستند و تعداد پیکسل‌های سفید بین دو پیکسل سیاه اندازه‌گیری می‌شود که نشان‌دهنده عرض خط حامل و عرض فاصله بین دو خط حامل است. سپس با توجه به اینکه ۵ خط حامل داریم که در کل عرضی مشخص ایجاد می‌کند. سطرهایی به این طول را جداگانه مورد بررسی قرار می‌دهیم. در مکان‌های درست ۵ خط حامل همگی در عرض مورد بررسی قرار خواهند گرفت. برای شناسایی آنها تعداد پیکسل‌های سیاه در هر ردیف پیکسلی را شمارش می‌کنیم. به این ترتیب یک نمودار هیستوگرام بدست خواهد آمد که باید دارای ۵ قله باشد. با استفاده از ***threshold*** مناسب می‌توانیم این ۵ سطر از پیکسل را شناسایی و آن‌ها را بعنوان خطوط حامل ذخیره کنیم.

تصویری که خطوط حامل شناسایی شده را نشان می‌دهد پس از هر بار اجرا در ***staffDetection.jpg*** و تصویر نت‌ها با حذف خطوط حامل در ***noStaffImage.jpg*** در پوشه ***output*** قابل مشاهده خواهند بود. پس از آن به شناسایی نت‌ها و ذخیره‌آن‌ها بدون در نظرگرفتن مکان آنها نسبت به هم می‌پردازیم. برای این‌کار با استفاده از یک تابع ***matching*** سعی می‌کنیم مناسب‌ترین

مقیاس را برای *template* و تصویر نت انتخاب کنیم تا همه نمونه‌های موجود یافت شوند. تمامی موارد در یک آرایه ذخیره می‌شود. دقت شود که موارد ذخیره شده شی‌های ساخته‌شده از روی یک کلاس هستند که مختصات خود را نیز در بر دارند.

پس از شناسایی نت‌ها و دیگر موارد مانند *barline* در روی هر *staff* با توجه به مختصات آن‌ها در یک آرایه جدید موارد را مرتب می‌کنیم و سپس به بررسی وجود *sharp* و *flat* در میان نت‌ها می‌پردازیم که در صورت وجود نت را دچار تغییر خواهند کرد. البته لازم به ذکر است که علاوه بر نت‌ها با توجه به اینکه *barline* نیز تشخیص داده می‌شود و زمان نت نیز شناسایی می‌شود، می‌توان پس از شناسایی ضرب‌آهنگ نت براحتی آن‌را تبدیل به یک برنامه جامع‌تر برای خواندن و نواختن موسیقی کرد که البته در اینجا فرصت برای انجام آن نبود. همچنین کلید موجود در ابتدای هر *staff* نیز بررسی می‌شود که با توجه به اینکه کلید سل یا فا است نت‌های روی خطوط حامل تفسیر متفاوتی خواهند داشت (بدلیل این عمومی سازی استفاده از آرایه‌ها و نواهای تولید شده‌ای که در این فاز در اختیار ما قرار گرفته بود ممکن نبوده و به همین دلیل از متد دیگری برای ایجاد صوت استفاده شده است که در ادامه توضیح داده می‌شود). این تمایز با تعریف آرایه‌هایی که کد نت‌های مربوط به خطوط حامل هر کلید را شامل می‌شوند امکان‌پذیر است. در اینجا عمومی‌سازی کلید دو انجام نشده است چرا که برای پیانو دو کلید قبلی کافی هستند اما برای نت‌های دیگر سازها می‌توان کلید دو را نیز به برنامه اضافه کرد.

با خواندن و شناسایی نت‌ها که نتایج مربوط به آن در پوشه *output* پس از هربار اجرا قابل مشاهده خواهد بود، برنامه اقدام به بررسی وجود نت‌های چنگ خواهد کرد. نت‌های چنگ جدا با استفاده از *template* مناسب قابل تشخیص خواهند بود. در رابطه با نت‌های چنگ با پرچم‌های متصل به هم به صورت زیر عمل خواهیم کرد. با توجه به اینکه نت چنگ نصف نت سیاه زمان می‌برد براساس قواعد موسیقی حداکثر دو نت چنگ پرچم متصل خواهند داشت. تمامی نت‌های سیاه پشت سر هم را مورد بررسی قرار می‌دهیم، بررسی به این صورت است که از مختصات *template* شناسایی شده مربوط به نت‌های سیاه به اندازه میله نت بالا می‌رویم و سپس خطی از بالای میله دو نت را در نظر می‌گیریم و نقطه وسط آن را می‌یابیم. سپس در محدوده اطراف نقطه وسط به بررسی تعداد پیکسل‌های سیاه می‌پردازیم. اگر تعداد پیکسل‌ها از یک مقدار مشخص *threshold* بیشتر باشد به این معنی است که خطی میان پرچم‌ها وجود داشته و به هم متصل هستند و بنابراین هر دو نت را به نت‌های چنگ تغییر خواهد داد. این تغییر با توجه به اینکه بعد از ذخیره شناسایی نت‌ها روی خطوط حامل انجام می‌شود در تصاویر مشاهده نمی‌شود اما در کد وجود دارد و *template* های مربوط به بخش اول نیز قابل بررسی‌اند.

در این قسمت خروجی شناسایی شده در ترمینال به تفکیک خط و دوباره *barline* نمایش داده می‌شود و سپس با توجه به اینکه از ابتدا هدف ما ذخیره به یک فرمت جامع بود از *MIDI* استفاده می‌کنیم و با استفاده از دو آرایه که در ابتدای کد وجود دارند، نت‌ها را به اعداد استاندارد تبدیل و به یک ورودی *MIDI* می‌دهیم و در نهایت خروجی را که بطور پیش‌فرض با ساز *grand piano* نواخته می‌شود را استخراج می‌کنیم. البته از آنجایی که در نت‌ها ذکر شده که نت مربوط به ساز ویون است با استفاده از کتابخانه *music21* فایل ذخیره شده را دوباره باز می‌کنیم و فرمت آن را به *violin* تغییر می‌دهیم و به این ترتیب خروجی با دو ساز موجود است.

البته ایراداتی نیز در برنامه وجود دارد که نحوه تلاش برای رفع آن توضیح داده شده است. شناسایی نت‌ها با تعیین *threshold* بسیار زمان‌بر است و با توجه به اینکه مبانی یادگیری ماشین در پروژه استفاده نشده است تعیین آن بصورت دستی زمان‌بر است. تلاش شده تا مقادیر مناسبی انتخاب شوند و نتیجه حاصل نیز برای نت‌های داده شده از دقت بالای نود درصد برخوردار است (در تمامی تست کیس‌ها تنها دو یا سه مورد اشتباه وجود دارد). مورد دیگر تشخیص مکان نت‌ها روی خطوط است که با توجه به اینکه تعداد پیکسل‌های خطوط خیلی کمتر از تعداد پیکسل‌های فواصل هستند، با توجه به اینکه شناسایی با *template matching* انجام می‌گیرد امکان دارد که مرکز نمونه شناخته شده از مکان خود کمی تغییر کند که این اتفاق در بسیاری از مواقع منجر به تبدیل

نت روی خط حامل به نت بالایی یا پایینی خود می‌شد. برای رفع آن در فرض خود برای تشخیص عرض خطوط حامل را از هر طرف سه پیکسل اضافه کرده‌ام که مشکل را تا حد خوبی برطرف کرده است.

قسمت بعدی مربوط به نواخته شدن یک یا دو نت با اکتاو بالاتر یا پایین‌تر نسبت به آنچه واقعا هستند می‌باشد که موفق به شناسایی و رفع آن نشدم. با توجه به اینکه این اتفاق تنها در یک یا دو نت رخ می‌دهد مشکل به احتمال زیاد از مقادیر اولیه آرایه‌های مختلف به کار رفته مانند ***convertToMIDI*** و ***convertToPitch*** است.

قسمت دیگر مربوط به شناسایی خطوط حاملی بود که تا انتهای صفحه پیش نرفته‌اند که باعث می‌شد تعداد پیکسل‌های سیاه در سطر کاهش یافته و در نتیجه خطوط شناسایی نشوند. برای اینکار ***threshold*** تا حد معقولی پایین آورده شده است که البته برنامه نویسی جداگانه برای آنها راه‌حل بهتری بود اما زمان بیشتری را می‌طلبید.

در آخر فایل صوتی نواخته شده با توجه به آنکه از تبدیل ***standard*** مورد استفاده ***MIDI*** که در صنعت موسیقی استفاده می‌شود استفاده می‌کند، نواهای تولید شده نیز استاندارد هستند. اگر صداهای نمونه داده شده به گوش زیباتر می‌رسند دلیل آن ایجاد افکت اکو و همچنین کشیده‌تر بودن است در حالی که در صفحه نت ویولن ما کشیدگی نت و یا انتقال از یک نت به دیگری را نداریم (این انتقال باید توسط یک کمان بالای سر نت‌ها نشان داده شود) و بنابراین تشخیص جدا بودن نت‌ها حالت درست‌تری برای نواختن نت است. در برنامه نویسی این پروژه از تفکر شی‌گرا استفاده شده است و برنامه به راحتی قابل بسط به سازهای بیشتر، کلیدهای بعدی و نت‌های دولاچنگ، سه‌لاچنگ و همچنین دینامیک‌های مختلف نواختن ساز است. خواهشمندم این تفکر و عمومی‌سازی را در هنگام مشاهده ایرادات نواختن نت در نظر بگیرید. ایرادات با صرف زمان برای تعیین ***threshold*** مناسب و یا استفاده از یک یادگیری ماشین مناسب قابل رفع خواهند بود و بنیادی نیستند. با تشکر از شما