В наше время можно увидеть широкое распространение аудиоинформации на различных носителях: это и радио, и сеть Интернет, и компакт-диски, и т.д. Набирают популярность различные стриминговые платформы, которые дают возможность прослушивания музыкальных каталогов по подписке, например, Spotify, VK combo и Яндекс.Музыка. В связи с этим возникает множество направлений обработки музыки, одним из которых является распознавание нот. Самым известным приложением, позволяющим решать эту задачу является “Shazam”, с помощью которого миллионы людей по всему миру могут найти нужную им песню лишь по ее отрывку. Принцип его работы заключается в том, что хеши, извлекаемые из записанного трека, сравниваются с имеющимися хешами из базы данных с последующими уточнениями. Однако, в том же “Shazam” можно столкнуться с проблемой сопоставления оригинальной записи и, например, обычного акапельного исполнения. Отсюда может возникнуть задача создания аудио-поисковой системы, то есть системы поиска музыки по фрагменту главной мелодии, напетой или сыгранной на каком-либо инструменте. Музыку можно оценивать по-разному. Наиболее техническим способом является гармонический анализ, который помогает понять инструменты и методы, используемые композитором при написании мелодии. Современный мир позволяет автоматизировать этот процесс, к тому же сделать его быстрым и точным. Так, задача распознавания нот может быть решена комбинацией из гармонического анализа и нейронных сетей за счет знания законов музыки и высоких вычислительных мощностей. Достаточно распознать главную мелодию, чтобы, например, имея банк записей нот различных музыкальных инструментов, суметь заменить звучащий инструмент или голос на необходимый.

Таким образом, целью исследования будем рассматривать разработку метода распознавания нот мелодии путем извлечения основных гармоник. При этом необходимо свести обработку звука к работе с изображением.

Для исследования необходимо выбрать архитектуру нейронной сети. Существует множество различных вариантов. Больше известны сети, использующие сверточные слои. Одна из самых первых – сеть LeNet5. Она включает 7 слоев, в дополнение к входному слою, 3 из них – сверточные. На выходе – слой softmax с 10 возможными значениями, соответствующими цифрам от 0 до 9. Ключевое преимущество LeNet5 - возможность сохранять параметры и результаты вычислений, по сравнению с использованием каждого пикселя в качестве отдельных входных данных для большой многослойной сети. Изображения сильно коррелированы пространственно, поэтому в первом слое не используются пиксели, что означает, что использование в качестве входных свойств отдельных пикселей не даст применить преимущества этих корреляций.

Сеть AlexNet с архитектурой LeNet5, однако у AlexNet больше вложенных сверточных слоев и фильтров на слое. Проблема переобучения решается использованием Dropout вместо регуляризации. Также используются слои пространственного пуллинга с перекрытием. За счет использования ReLU в качестве функции активации AlexNet отличается высокой скоростью обучения. Применением операций сдвига и зеркального отражения производится увеличение количества данных (data augmentation). Также, AlexNet проходит 90 эпох и обучение осуществляется 6 дней на двух видеокартах, из-за чего сеть разделена на две части.

Еще один пример архитектуры – сеть ResNet, использующая остаточные блоки для повышения точности модели. ResNet - это тип нейронной сети, который применяет сопоставление идентификаторов. Это означает, что входные данные для некоторого соля передаются напрямую или в виде ярлыка для какого-либо другого слоя. Наиболее важной концепцией, задействованной в ResNet, является пропускное соединение. Это в основном сопоставление идентификаторов, где входные данные из предыдущего слоя добавляются непосредственно к выходу другого слоя. С помощью сетей ResNet можно обучать очень сложные сети с примерно 1000 слоями.

Также следует упомянуть архитектуру MobileNet, которая является первой мобильной моделью компьютерного зрения TensorFlow. MobileNets - это семейство мобильных моделей компьютерного зрения для TensorFlow, предназначенные для эффективного повышения точности с учетом ограниченных ресурсов для приложений. Это небольшие модели с малой задержкой и низким энергопотреблением, параметризованные для соответствия ограничениям ресурсов для различных вариантов использования. Их можно использовать для классификации, обнаружения, встраивания и сегментации.

Будем придерживаться следующей схемы действий для получения основной гармоники мелодии:

1. Подготовим базу данных для распознавания;
2. На основе базы данных сгенерируем спектрограммы;
3. Сформируем архитектуру нейронной сети для классификации нот;
4. Подадим спектрограммы на вход нейронной сети и обучим модель;
5. Сопоставим полученные номера классов и ноты.

Для распознавания нот в произведениях необходимо сначала найти базы данных. В рассматриваемом случае данные представляют из себя аудиофайлы формата WAV, являющиеся моно-сигналами, при этом для каждого файла должна иметься информация о частоте/высоте тона, времени появления и времени окончания/длительности каждой ноты в секундах. Если в базе даны длительности нот, то их можно легко перевести во время окончания формулой (1). Если вместо частоты дана информация о высоте тона ноты, то будем использовать стандарт midi – формулу (2). Частота мелодий стандартно рассматривается равной 44,1 кГц, то есть 44100 отсчета в секунду. Значит, для перевода времени в отсчеты воспользуемся формулой (3).

В рамках проделанной работы рассматривалось 2 набора данных: «База данных с маркировкой звука флейты для автоматической транскрипции музыки» и «CSD: набор данных детских песен для исследования певческого голоса», при этом вторая база также разделялась на два подкаталога английских и корейских песен. Обработка этих наборов состоит в следующем:

1. Сначала загружаем данные и метки;
2. Переразбиваем данные на маленькие интервалы, пригодные для обучения;
3. После переразбиваем метки согласно данным.

При разбиении аудиофайлов на интервалы время переведем в отсчеты. Получившиеся фрагменты необходимо распределяем на каталоги, соответствующие конкретным нотам. Мы не будем опираться на различие октав, поэтому основываемся на 12 основных полутонах: A (ля), G# (соль-диез), G (соль), F# (фа-диез), F (фа), E (ми), D# (ре-диез), D (ре), C# (до-диез), C (до), B (си), A# (ля-диез). Такое распределение осуществимо из соответствия нот и частот. Математическая формула частоты для всего звукоряда имеет вид (4). Для нахождения количества полутонов, на которое заданная частота отстает от частоты камертона, используется формула (5). Для сведения всех нот к 12 основным, необходимо найти значение по формуле (6). При этом, если , то разворачиваем ноту так, чтобы она оказалась выше ноты A. Это отображается во второй строке системы. Нота, соответствующая заданной частоте без учета октав, определяется из значения z по таблице 1.

Следующий пункт схемы получения основной гармоники – генерация спектрограмм на основе полученной базы данных. Именно спектрограмма будет являться тем изображением, которое мы будем подавать на вход нейронной сети.

Спектрограмма — изображение, показывающее зависимость спектральной плотности мощности сигнала от времени. Спектрограммы применяются для идентификации речи, анализа звуков животных, в различных областях музыки, радио- и гидролокации, обработке речи, сейсмологии и в других областях. Спектрограмма представляет собой матрицу , где соответствует определенной частоте, а каждый из столбцов – спектр фрагмента аудиозаписи. Неформально говоря, элементы спектрограммы характеризуют интенсивность звучания конкретных частот в пределах промежутка времени. Как видно по рисунку 1, наиболее распространенным представлением спектрограммы является двумерная диаграмма: на горизонтальной оси представлено время, по вертикальной оси — частота; третье измерение с указанием амплитуды на определенной частоте в конкретный момент времени представлено интенсивностью или цветом каждой точки изображения.

Одним из способов получения спектрограммы является вычисление оконного преобразования Фурье, представленного формулой (7). Аудиосигнал постоянно меняется, поэтому для упрощения мы предполагаем, что на коротких временных масштабах аудиосигнал статически не сильно меняется. Оконное преобразование Фурье также называют коротким (или кратковременным) преобразованием Фурье (STFT). В качестве весового окна можно использовать различные варианты, но наибольшей популярностью пользуется окно Хэмминга.

Вместо спектра самого сигнала можно использовать спектр его мощности, что мотивировано человеческой ушной улиткой, которая вибрирует в разных местах в зависимости от частоты входящих звуков. В зависимости от расположения в вибрирующей улитке срабатывают различные нервы, сообщая мозгу, что присутствуют определенные частоты. Оценка спектрограммы выполняет аналогичную работу для нас, определяя, какие частоты присутствуют в кадре.

Спектральная оценка спектрограммы содержит много информации, не требуемой для автоматического распознавания нот. В частности, улитка не может различить разницу между двумя близко расположенными частотами. Этот эффект становится более выраженным по мере увеличения частот. Поэтому мы будем брать скопления блоков периодограммы и просуммируем их, чтобы получить представление о том, сколько энергии существует в различных частотных областях. Это выполняется с помощью банков фильтров Мела: по мере увеличения частоты фильтры расширяются, поскольку мы становимся менее обеспокоенными вариациями. Нас интересует только приблизительно, сколько энергии приходится на каждый фрагмент. Шкала мела направлена на то, чтобы имитировать нелинейное восприятие звука человеческих ухом, будучи более различимой на более низких частотах и менее различимой на более высоких частотах. Конвертировать Герцы и Мелы между собой можно с помощью формул (8) и (9). Мы будем использовать банк треугольных фильтров: каждый фильтр имеет треугольную форму с откликом 1 на центральной части и линейно уменьшается к 0, пока не достигнет центральных частот двух соседних фильтров, где отклик равен 0. Моделирование таких фильтров можно осуществить с помощью уравнения (10). На рисунке 2 показан пример банка треугольных фильтров. Как только у нас есть энергии фильтрующего банка, мы берем логарифм их, получая кепстр или мел-кепстральные коэффициенты MFCC — представление сигнала, грубо говоря, в виде особого спектра, из которого с помощью различных фильтраций и преобразований удалены незначительные для человеческого слуха компоненты.

В качестве архитектуры сети рассмотрим предложенную Филипом Корзениовски в диссертации «Harmonic Analysis of Musical Audio using Deep Neural Networks». Такая сеть состоит из 3 скрытых слоев из 512 узлов с функцией активации ReLU и выходного слоя из 12 узлов по количеству полутонов. В качестве функции активации на последнем слое будем использовать softmax. Модель сети представлена на рисунке 3. Как из него видно, в качестве входных данных используются спектрограммы. Для получения спектрограмм 2 подхода: будем сначала использовать спектр сигнала, а потом мел-спектр.

Человеческое ухо не слышит колебания меньше 20 Гц, поэтому граница будет задавать максимальный размер окна. Для частоты дискретизации 44100 отсчетов/сек. Получаем размер окна 2205. Для красоты возьмем 2048 отсчетов, что составляет примерно 21.5 Гц – это и будет первоначальным размером фрейма для оконного преобразования Фурье. Шаг смещения фрейма при этом возьмем равным половине размера кадра. После оконного преобразования Фурье получаем множество коротких спектров, длина которого вычисляется по формуле 11. Множество таких спектров составит одну спектрограмму. На вход нейросети подается несколько соседних спектрограмм. Будем брать, как предлагал Корзениовски, 15 последовательных спектрограмм. Если размер файла не позволяет это осуществить, то будем дополнять спектрограмму последним столбцом до нужного количества. Таким образом, размерность входа для нейросетевой модели – 1025х15.

При получении результатов, номера классов будем сопоставлять с нотой в соответствии с таблице 2.

Вся дальнейшая работа производилась с помощью языка Python.