# Разработка метода автоматической генерации звуков по изображению

*Н.А. Никитин,*
*асп., set.enter@mail.ru,*
*В.Л. Розалиев,*
*доц, к.т.н., vladimir.rozaliev@gmail.com,*
*Ю.А. Орлова,*
*доц, д.т.н., к.п.н., yulia.orlova@gmail.com,*
*ВолгГТУ, г. Волгоград*

В данной работе описан метод для автоматизации процесса создания музыки, путём автоматизированной генерации звуков по изображению. Разработанный метод автоматической генерации звуков по изображению основывается на совместном использовании нейронных сетей и светомузыкальной теории, что позволяет повысить качество выходной музыкальной композиции и снизить роль пользователя. Описана программа для генерации звуков по изображению, для подтверждения эффективности предложенного метода. Также описано тестирование программы.

This paper describes a method for automating the process of creating music by means of automated generation of sounds from an image. The developed method of automatically generating sounds by image is based on the sharing of neural networks and light-music theory, which allows improving the quality of the output musical composition and reducing the role of the user. A program is described for generating sounds by image to confirm the effectiveness of the proposed method. Testing of the program was also described.

## Введение

С тех пор как музыку стали записывать на бумаге в виде нотных знаков, стали появляться оригинальные «способы» ее сочинения. Одним из самых первых методов алгоритмической композиции стал способ сочинения музыки, придуманный Моцартом – «Музыкальная игра в кости» [1]. Первое компьютерное музыкальное произведение – «Illiac Suite for String Quartet» – было создано в 1956 году пионерами применения компьютеров в музыке – Лежарен Хиллер и Леонард Айзексон [2]. В этом произведении использованы почти все главные методы алгоритмической музыкальной композиции: теория вероятностей, марковские цепи и генеративная грамматика.

Развитие компьютерной музыки, в том числе и генерации звуков по изображению, в прошлом веке было сильно ограничено вычислительными ресурсами – покупать и содержать мощные ЭВМ могли позволить себе лишь крупные университеты и лаборатории, а первым персональным компьютерам не хватало вычислительной мощности. Однако в XXI веке, изучением компьютерной музыки может заниматься практически каждый человек. В настоящее время компьютерная музыка может применяться во многих отраслях: создание музыки для компьютерных игр, рекламы и фильмов. Сейчас, для создания фоновых музыкальных композиций в компьютерных играх и рекламе, компании нанимают профессиональных композиторов или покупают права на уже написанные музыкальные произведения. Однако в таком жанре, требования к музыкальной композиции не велики, а значит, данный процесс можно автоматизировать, что позволит компаниям снизить расходы на сочинение композиций. Также, генерацию звуков по изображению можно применить в образовательном процессе [3]. Взаимодействие музыки и изобразительного искусства в процессе интегрированной образовательной деятельности с детьми дошкольного возраста может осуществляться в форме сочетания восприятия произведений музыкального и изобразительного искусства на основе общности их настроения, стиля, жанра, что способствует развитию музыкального восприятия у дошкольников [4].

Наибольших успехов автоматизация процесса написания и создания музыки достигла сравнительно недавно (в последние десятилетия), однако по большей части связана с изучением и повторением различных музыкальных стилей [5]. Поскольку процесс создания музыки сложно формализуем, то для программного (автоматизированного) создания композиций лучше всего подходят искусственные нейронные сети, так как они позволяют выявить связи, которые не видит человек [6]. Помимо этого, для снижения роли пользователя-композитора в генерации музыкальных произведений, было принято решение брать часть музыкальных характеристик с изображения. В связи с этим, целью данной работы является увеличение гармоничности и мелодичности программной генерации звуков по цветовой гамме изображений посредством использования нейронных сетей.

## 1. Метод получения композиции по изображению

Для снижения роли пользователя-композитора в генерации звуков, часть характеристик музыкального произведения получается путём анализа цветовой гаммы изображения. Таким образом, характер полученной музыкальной композиции будет соответствовать входному изображению. Данная особенность делает возможным применение данного подхода для создания фоновых музыкальных произведений в компьютерных играх, рекламе и фильмах.

Ключевыми характеристиками музыкального произведения является его тональность и темп. Именно эти параметры определяются путём анализа цветовой гаммы изображения. Для начала определим соотношение цветовых и музыкальных характеристик [7] (таблица 1).

***Таблица 1***

**Соотношение цветовых и музыкальных характеристик**

| **Цветовые характеристики** | **Музыкальные характеристики** |
|---|---|
| Оттенок (красный, синий, жёлтый…) | Нота (до, до-диез, ре, ре-диез, ми, ми, фа, фа-диез, соль, соль-диез, ля, ля-диез, си) |
| Цветовая группа (тёплый/холодный) | Музыкальный лад (мажор/минор) |
| Яркость | Октава ноты |

| | |
|---|---|
| Насыщенность | Длительность ноты |

Затем, необходимо определить схему соотнесения названия цвета и ноты. На данный момент существует большое количество подобных схем, однако в данной работе была выбрана схема Ньютона.

Как видно из таблицы 1, тональность произведения определяется двумя цветовыми характеристиками – оттенок и цветовая группа, а темп – яркость и насыщенность. Алгоритм определения тональности опирается на анализ изображения и таблицу 1, состоит из 3 шагов и описан ниже.

Шаг 1. Преобразуем входное изображение из цветового пространства RGB в HSV. Данный шаг позволяет преобразовать изображение к более удобному виду, поскольку HSV пространство уже содержит необходимые характеристики – название цвета (определяется по параметру hue), насыщенность (параметр saturation) и яркость (параметр brightness).

Шаг 2. Анализируя в целом изображение, определяем преимущественный цвет.

Шаг 3. Определяем название и цветовую группу преимущественного цвета.

Шаг 4. Согласно таблице 1 и схеме Ньютона определяем тональность произведения (нота и музыкальный лад).

Для определения темпа произведения, необходимо получить яркость и насыщенность (по параметрам saturation и brightness) преимущественного цвета, и рассчитать темп, согласно данным параметрам.

## 2. Выбор нейронной сети для генерации музыкальных композиций

Важной особенностью нейронных сетей прямого распространения (feedforward neural networks) является то, что у данной нейросети есть общее ограничение: и входные и выходные данные имеют фиксированный, заранее обозначенный размер, например, картинка 100×100 пикселей или последовательность из 256 бит. Нейросеть с математической точки зрения ведет себя как обычная функция, хоть и очень сложно устроенная: у нее есть заранее обозначенное число аргументов, а также обозначенный формат, в котором она выдает ответ.

Вышеперечисленные особенности не представляет больших трудностей, если речь идет о тех же картинках или заранее определенных последовательностях символов. Но для обработки любой условно бесконечной последовательности, в которой важно не только содержание, но и порядок, в котором следует информация, например, текст или музыка необходимо использовать нейронные сети с обратными связями – рекуррентные нейронные сети (RNN). В рекуррентных нейросетях нейроны обмениваются информацией между собой: например, вдобавок к новому кусочку входящих данных нейрон также получает некоторую информацию о предыдущем состоянии сети. Таким образом в сети реализуется «память», что принципиально меняет характер ее работы и позволяет анализировать любые последовательности данных, в которых важно, в каком порядке идут значения [8].

Однако большой сложностью сетей RNN является проблема исчезающего (или взрывного) градиента, которая заключается в быстрой потере информации с течением времени. Конечно, это влияет лишь на веса, а не состояния нейронов, но ведь именно в них накапливается информация. Сети с долгой краткосрочной памятью (long short term memory, LSTM) стараются решить вышеупомянутую проблему потери информации, используя фильтры и явно заданную клетку памяти. У каждого нейрона есть клетка памяти и три фильтра: входной, выходной и забывающий. Целью этих фильтров является защита информации. Входной фильтр определяет, сколько информации из предыдущего слоя будет храниться в клетке. Выходной фильтр определяет, сколько информации получат следующие слои. Такие сети способны научиться создавать сложные структуры, например, сочинять тексты в стиле определённого автора или сочинять простую музыку, однако при этом потребляют большое количество ресурсов [9].

Таким образом, для реализации программы автоматизированной генерации музыкальных композиций по цветовой гамме изображений необходимо использовать именно рекуррентные нейронные сети с долгой краткосрочной памятью – RNN LSTM (долгая краткосрочная память – разновидность архитектуры рекуррентных нейронных сетей). Именно данный вид нейронных сетей используется для генерации музыкальных композиций в программе Magenta – это музыкальный проект с открытым исходным кодом от Google, также RNN LSTM используется в программе сочинения композиций в стиле И.С. Баха – BachBot, а также в DeepJaz – система позволяет генерировать джазовые композиции на основе анализа midi файлов [10].

## 3. Выбор метода синтеза звуков

В процессе исследования методов синтеза звуков были рассмотрены и проанализированы наиболее популярные методы синтеза звуков: аддитивный синтез, FM – синтез, фазовая модуляция, сэмплинг, таблично-волновой синтез, линейно-арифметический синтез, субтрактивный синтез и векторный синтез.

Аддитивный синтез очень сложен для реализации, из-за необходимости отдельного контроля громкости и высоты каждой гармоники, которых даже несложный тембр насчитывает десятки.

FM – синтез хорошо применим для синтеза звука ударных инструментов, синтез же остальных музыкальных инструментов звучит слишком искусственно. Главный недостаток FM-синтеза — неспособность при его помощи полноценно имитировать акустические инструменты.

Фазовая модуляция даёт достаточно хороший звук, но сильно ограничена, поэтому редко используется на практике.

Сэмплинг применяется в большинстве современных синтезаторов, так как даёт наиболее реалистичный звук и достаточно прост в реализации.

Таблично-волновой синтез и Линейно-арифметический синтез похожи на семплерный метод, однако данные методы сложны в реализации, поэтому на практике предпочтение отдаётся семплингу, как наиболее простому методу.

Субтрактивный синтез обычно используется совместно с аддитивным, обладает хороших качеством синтеза звуков, однако сложен в реализации.

Векторный синтез используется для получения более богатых и сложных тембров, однако в рамках рассматриваемой системы это не существенно.

В процессе исследования методов синтеза звука, были рассмотрены восемь методов. Каждый из них обладает своими плюсами и минусами, однако для реализации системы был выбран Сэмплинг. Данный метод даёт наиболее реалистичное звучание инструментов, что является важной характеристикой для системы, также данный метод от-

носительно прост в реализации. Недостатком Сэмплинга является ограниченность метода, однако в рамках реализации системы – это не существенно, так как пользователь будет выбирать из ограниченного набора заранее известных инструментов, таким образом не требуется больших возможностей изменения готовых пресетов.

Таким образом, лучшим методом синтеза звука для реализации системы является Сэмплинг.

## 4. Выбор технологии реализации нейронной сети

Для разработки искусственной нейронной сети был выбран язык программирования Python, поскольку данный язык является кроссплатформенным, также данный язык направлен на улучшение продуктивности разработки и читаемости кода. Помимо этого, данный язык широко используется для анализа данных и содержит большое количество библиотек для машинного обучения.

Theano — это расширение языка Python, позволяющее эффективно вычислять математические выражения, содержащие многомерные массивы. Поскольку данная библиотека является низкоуровневой, то процесс создания модели и определения ее параметров требует написания объемного и шумного кода. Однако преимуществом Theano является ее гибкость, а также наличие возможности реализации и использования собственных компонент [11].

TensorFlow — это библиотека с открытым исходным кодом для численного расчета с использованием потоковых графов. Данная библиотека, также, как и Theano является низкоуровневой, а значит процесс разработки сложный. Однако благодаря низкому уровню разработки нейронных сетей можно получить более гибкую модель. Также преимуществом данной библиотеки является большое сообщество и хорошая документация [12].

Lasagne — легковесная обёртка для библиотеки Theano. Программирование с использованием Lasagne достаточно низкоуровневое – необходимо объявить каждый уровень нейронной сети посредством использования модульных строительных блоков над Theano. Lasagne выступает как компромисс между гибкостью Theano и простотой Keras [13].

Keras — это высокоуровневый API для разработки нейронных сетей, написанный на Python и способный работать поверх TensorFlow, CNTK или Theano. Библиотека была разработана с упором на возможность быстрого экспериментирования. Минусом данной библиотеки является небольшая гибкость [14].

MXNet — это система глубокого обучения с открытым исходным кодом, используемая для обучения и развертывания глубоких нейронных сетей. Поскольку MXNet является библиотекой высокого уровня разработка нейронных сетей с использованием MXNet проще и быстрее, чем с использованием Theano или TensorFlow, однако уступает библиотеке Keras за счёт большого числа поддерживаемых языков и больших возможностей для масштабирования, что делает программный код более громоздким [15].

Для проведения сравнения были выделены следующие критерии: гибкость, масштабируемость, поддержка параллельных вычислений, поддержка вычисления на GPU, скорость разработки. Все рассмотренные библиотеки были оценены по представленным выше критериям по пятибалльной шкале, где 0 – минимальное значение критерия, 5 – максимальное. Результаты сравнения библиотек представлены в таблице 2.

***Таблица 2***

**Сравнение библиотек для реализации нейронной сети**

| Параметр | Theano | TensorFlow | Lasagne | Keras | MXNet |
|---|---|---|---|---|---|
| Гибкость | 5 | 4 | 3.5 | 2 | 3 |
| Масштабируемость | 4 | 5 | 4 | 5 | 5 |
| Поддержка параллельных вычислений | 5 | 4 | 5 | 5 | 5 |
| Поддержка вычисления на GPU | 4 | 5 | 4 | 5 | 5 |

Таким образом, можно сделать вывод о том, что для разработки рекуррентной нейронной сети для генерации музыкальных композиций следует использовать библиотеку Keras, поскольку данная библиотека позволяет работать поверх Theano и TensorFlow, используя их преимущества, при этом процесс разработки нейронных сетей с использованием данной библиотеки простой и быстрый, что позволяет создавать прототипы для быстрого экспериментирования.

## 5. Описание разработанной программы

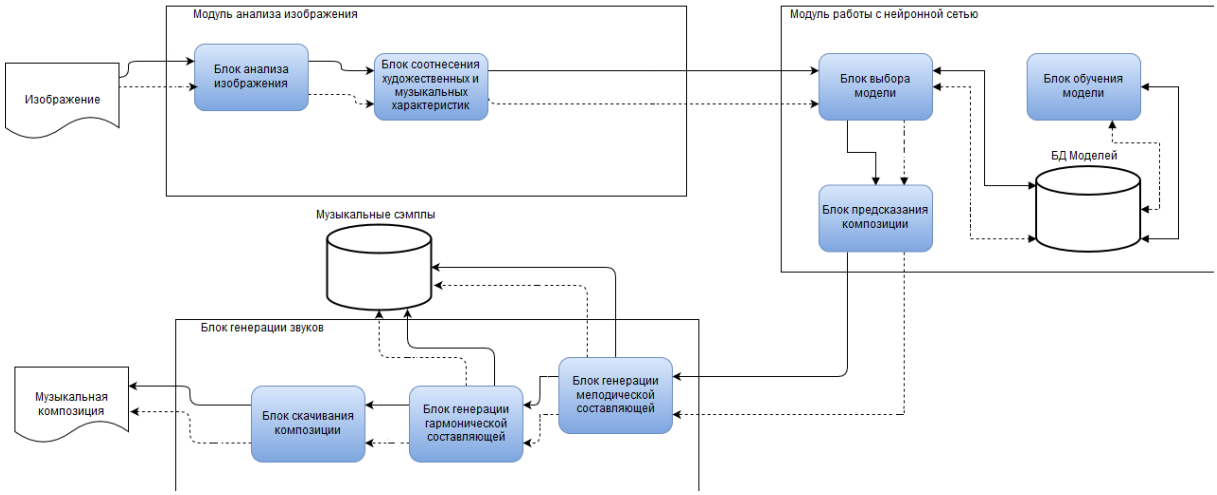


рис. 1 Архитектура программы

Для подтверждения эффективности предложенных алгоритмов и методов, была разработана программа для генерации звуков по цветовой гамме изображений. Данная программа представляет собой веб-сайт, реализованный на языке Python. На вход программа получает изображение, которое пользователь загружает вручную. После получения пути к изображению, программа загружает изображение в память, используя библиотеку OpenCV. Затем происходит конвертация изображения в цветовое пространство HSV. Затем, в процессе анализа изображения, программа преимущественный цвет изображения, по которому определяется тональность и темп композиции. Затем происходит предсказание (достраивание) композиции, согласно полученным характеристикам с помощью нейронной сети.

Архитектура программы представлена на рисунке 1.

## 6. Проведение эксперимента

. Для подтверждения эффективности предложенных алгоритмов генерации звуков по цветовой гамме изображения, была разработана программа на языке Python, с использованием библиотеки Keras. Для синтеза звуков используется метод сэмплинг (sampling). Программа генерации музыкальных композиций с использованием нейронных сетей была обучена на 29 композициях Людвига ван Бетховена. После обучения был составлен набор из десяти тестовых изображений, имеющих различный тип (абстрактные изображения, пейзажи, города и люди). По всем десяти изображениям были получены и сохранены выходные музыкальные композиции. Данные музыкальные композиции были отправлены на анализ 10 экспертам, которые должны были оценить каждое произведение по следующим критериям:

- соответствие характеру изображения (по пяти бальной шкале);
- реалистичность звучания инструмента (фортепьяно или гитара);
- мелодичность композиции;
- качество гармонии (аккомпанемента);
- приятность мелодии для восприятия;
- цельность композиции;
- реалистичность/искусственность композиции.

Данные от каждого эксперта были, обработаны, сведены в таблицу 3 и проанализированы.

***Таблица 3***

**Экспертные оценки композиций**

| Критерий | Среднее значение для всех тестов |
|---|---|
| Соответствие характеру изображения | 4.9 |
| Реалистичность звучания инструмента | 3.9 |
| Мелодичность композиции | 4.4 |
| Качество гармонии | 4.9 |
| Приятность для восприятия | 4.6 |
| Цельность композиции | 4.5 |
| Реалистичность композиции | 4.3 |

Проанализировав оценки всех экспертов и высчитав средние по каждому критерию, можно сделать вывод о том, что фортепьяно на слух экспертов звучит реалистичнее, чем гитара. Также можно сделать вывод о том, что композиция, сгенерированная по абстрактным изображениям, более приятна на слух, чем генерация по пейзажам. В целом общее впечатление от сгенерированных звуков у экспертов положительное. Среди минусов некоторые эксперты выделяют однотипность гармонии, иногда рваность и недостаточную реалистичность произведения, и не достаточную реалистичность гитары.

Делая вывод по каждому критерию можно сказать, что все эксперты оценили на высокий бал соответствие произведения характеру изображения, по второму критерию – инструмент фортепьяно звучит довольно реалистично. Мелодичность композиций разделилась пополам, то есть половина композиций эксперты оценили на высший бал, другую половину на 4, в целом неплохой результат. Качество гармонии также было оценено экспертами на высший бал. Приятность мелодий для восприятия получил 60% высших балов и 40% четвёрок, что говорит о том, что некоторые произведения звучат не вполне реалистично. Реалистичность и цельность композиций в среднем оценено на 4, что является естественным результатом для компьютерной генерации звуков.

## Заключение

В ходе выполнения работы была определена схема соотнесения цветовых и музыкальных характеристик, был проведён обзор типов нейронных сетей и выбран наиболее подходящий тип для генерации музыкальных композиций, была детально описана используемая нейронная сеть, была выбрана технология реализации нейронной сети, был выбран метод синтеза звуков, был проведён эксперимент по оценке гармоничности и мелодичности выходных музыкальных композиций.

В ходе анализа различных типов и архитектур ИНС был сделан вывод о том, что наиболее подходящей сетью для обработки музыкальной информации являются рекуррентные нейронные сети (RNN), а именно сети с долгой краткосрочной памятью (long short term memory, LSTM).

В результате проведения эксперимента, была обучена модель (нейронная сеть) на композициях Баха, а также были сгенерированы композиции по 10 изображениям. Данные композиции были отправлены на анализ экспертам. В результате анализа экспертных оценок можно сделать вывод о том, что программа генерирует достаточно мелодичные композиции, однако сказывается, что модель была обучена на небольшом количестве произведений только одного автора.

*Работа выполнена при поддержке РФФИ (грант 17-07-01601)*

**Литература**


1. Фазылова, Э.Ф. Системы генерации музыки или как автоматизировать искусство? // Молодёжный научно-технический вестник. – 2014. URL: http://sntbul.bmstu.ru/doc/723360.html. (Дата обращения: 20.03.2017).
2. Ariza, C. Two Pioneering Projects from the Early History of Computer-Aided Algorithmic Composition / C. Ariza // Computer Music Journal. – MIT Press, 2012. – №3. – pp. 40-56
3. Черешнюк, И. Р. Алгоритмическая музыкальная композиция и её место в современном музыкальном образовании / И. Р. Черешнюк // Педагогика искусства. – 2015. – № 3. – С. 65-68.
4. Выготский, Л.С. Воображение и творчество в детском возрасте / Л.С. Выготский. – Москва, 2008 // Мышление и речь: сборник / Л.С. Выготский. – Москва: АСТ, 2008. – С. 497-594.
5. D. Cope, Computer Models of Musical Creativity, MIT Press, Cambridge, Mass., 2005.
6. Mazurowski, L. Computer models for algorithmic music composition / L. Mazurowski // Proceedings of the Federated Conference on Computer Science and Information Systems. – Szczecin, Poland, 2012. – pp. 733–737
7. Caivano, J. L., Colour and sound: Physical and Psychophysical Relations, Colour Research and Application, 12(2), pp. 126-132, 1994
8. Sak, H., Senior, A., Beaufays, F. Long Short-Term Memory Based Recurrent Neural Network Architectures for Large Vocabulary Speech Recognition / H. Sak, A. Senior, F. Beaufays // ArXiv e-prints. – 2014
9. Doornbusch, P. Gerhard Nierhaus: Algorithmic Composition: Paradigms of Automated Music Generation / P. Doornbusch // Computer Music Journal. - Volume: 34, Issue: 3. – 2014.
10. Brinkkemper, F. Analyzing Six Deep Learning Tools for Music Generation [Электронный ресурс]. – 2015. - Режим доступа: http://www.asimovinstitute.org/analyzing-deep-learning-tools-music/ (Дата обращения: 03.07.2017).
11. James Bergstra, Olivier Breuleux, Fr´ed´eric Bastien, Pascal Lamblin, Razvan Pascanu, Guil-laume Desjardins, Joseph Turian, David Warde-Farley, and Yoshua Bengio. Theano: a CPU and GPU math expression compiler. In Proceedings of the Python for Scientific Computing Conference (SciPy), June 2010.
12. TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems [Электронный ресурс]. – 2015. – Режим доступа: https://www.tensorflow.org/ (Дата обращения: 11.07.2017).
13. Lasagne - lightweight library to build and train neural networks in Theano [Электронный ресурс]. – 2017. – Режим доступа: http://lasagne.readthedocs.org/ (Дата обращения: 12.07.2017).
14. Keras: The Python Deep Learning library [Электронный ресурс]. – 2017. – Режим доступа: https://keras.io/ (Дата обращения: 12.07.2017).
15. MXNet: A Flexible and Efficient Library for Deep Learning [Электронный ресурс]. – 2017. – Режим доступа: https:// http://mxnet.io// (Дата обращения: 12.07.2017).