

Санкт-Петербургский государственный университет

Кафедра системного программирования

Мирошниченко Егор Дмитриевич

Реализация алгоритма генерации
изображений по сигналам головного мозга

Дипломная работа

Научный руководитель:
д.ф.- м.н., проф. А.Н. Терехов

Рецензент:
ст. преп. Воробьева А. А.

Санкт-Петербург
2020

SAINT-PETERSBURG STATE UNIVERSITY

System Programming

Egor Miroshnichenko

Implementing an algorithm for generating images from brain signals

Graduation Thesis

Scientific supervisor:
professor Andrey Terekhov

Reviewer:
Senior Lecturer Vorobyeva Alice

Saint-Petersburg
2020

Оглавление

Введение	4
1. Постановка задачи	6
2. Обзор решения	7
2.1. Brain2Image	7
2.1.1. Энкодер	8
2.1.2. Декодер	9
3. Сбор данных	10
3.1. Emotiv EPOC+	10
3.2. Программное обеспечение	11
3.3. Предобработка	11
4. Архитектура модели	13
4.1. Энкодер	13
4.2. Декодер	15
5. Оценка результатов	19
Заключение	20
Список литературы	21

Введение

Изучение работы мозга является сложной задачей, которая привлекает внимание исследователей из разных областей. Последние исследования в области нейрофизиологии доказывают, что процессы мозговой активности могут быть выявлены из данных, полученных с помощью функциональной магнитно-резонансной томографии, *электроэнцефалограммы (ЭЭГ)*, магнитоэнцефалографии. А распространение устройств для считывания этих сигналов за пределами стен исследовательских и медицинских центров способствует привлечению энтузиастов, что ускоряет развитие области.

Идея управление машинами с помощью мозга не нова и уже долгое время исследуется, и есть много успешных результатов, особенно в сфере *нейрокомпьютерных интерфейсов (НКИ)*, которые в основном сосредоточены на распознавании образов для прямого управления машинами. Кроме этого, сигналы головного мозга были использованы для обучения интеллектуальных систем для классификации эмоций [8], медицинских задач [7] и т. п. И если эти задачи подразумевают выделение признаков активностей некоторых зон мозга, то задача визуального взаимодействия сильно сложнее и меньше изучена. Однако исследования показали, что активность головного мозга содержит в себе конкретные образы, в зависимости от классов объектов, на которые смотрит человек.

Параллельно с этим быстро растущая область алгоритмов машинного обучения породила множество исследований темы, в последние несколько лет появилось много разнообразных новых подходов, которые показывают хорошие результаты при решении широкого спектра задач. В частности, появились всевозможные генераторы изображений из скрытого пространства признаков. Хорошие результаты были продемонстрированы разнообразными автоэнкодерами и *generative adversarial networks (GAN)*[4], которые находят применение в самых разнообразных местах: от переноса стиля с одного изображения на другое, до подмены лиц на фото и видео.

В связи с этим сейчас возникает всё больше работ на пересечении этих сфер. И область активно развивается.

1. Постановка задачи

Целью данной работы является реализация алгоритма генерации изображений из ЭЭГ, считанных с помощью нейроинтерфеса Emotiv EPOC+.

Для достижения этой цели были поставлены следующие задачи:

- Проанализировать представленное в Brain2Image [6] решение.
- Собрать необходимые данные для обучения алгоритма с помощью шлема Emotiv EPOC+.
- Реализовать вариант алгоритма, использующий GAN.
- Сравнить качество работы генератора изображений с существующим [6].

2. Обзор решения

2.1. Brain2Image

В серии работ [2, 6] авторы представили модель для генерации изображений по электроэнцефалограмме снятой с головного мозга Brain2Image. Сначала в [2] был разработан энкодер, который выделяет из сырой ЭЭГ основные признаки объектов находящиеся изображении, по которым можно сделать выводы о его принадлежности некоторому классу. Они получили хорошие результаты, точность классификации превышает 80%.

Далее в качестве генератора были опробованы два метода: GAN и VAE. Методы показали разные результаты, в частности: изображения полученные с помощью GAN оказались лучше в цветопередаче и четкости границ объектов, но при этом плохо отлавливают структуру объектов, форму; сгенерированные VAE наоборот, содержат больше данных о форме и структуре, но размыты и цвета переданы хуже. Общая схема:

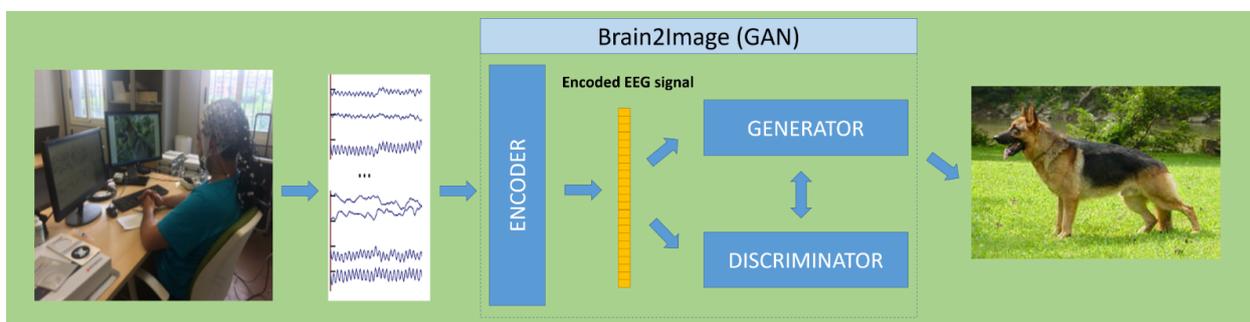


Рис. 1: Схема Brain2Image (из [6])

2.1.1. Энкодер

По результатам [2] в качестве энкодера была выбрана схема, в которой за слоем *long short-term memory (LSTM)* узлов идёт стандартный полносвязный слой для придания пространству признаков нормального распределения, которое требуется на входе GAN.

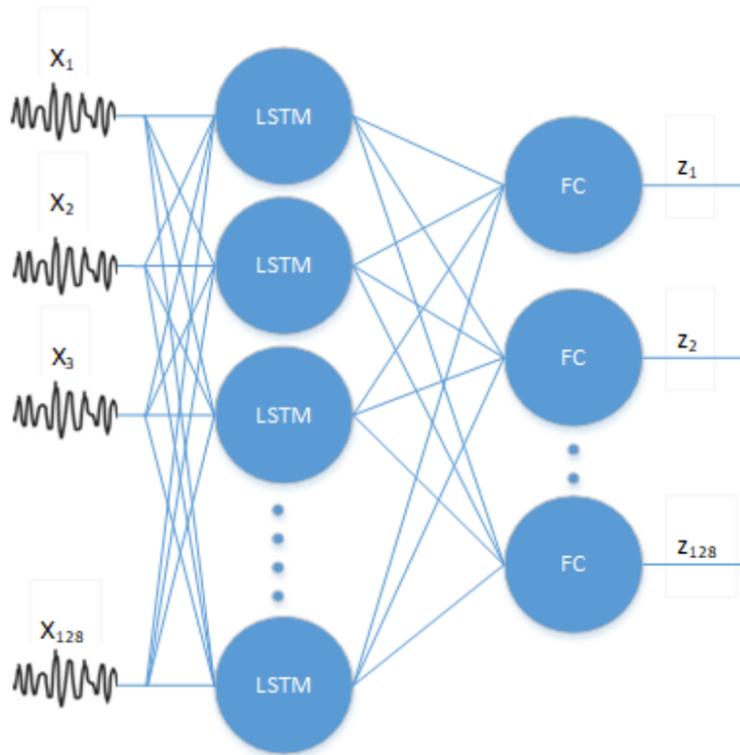


Рис. 2: Схема энкодера (из [6])

2.1.2. Декодер

Декодер представляет собой классический GAN, на вход которого подается вектор признаков из энкодера, к которому конкатенируется вектор шумов. В процессе обучения генератор создает изображения из входных данных, а дискриминатор предсказывает вероятность того что данное изображение соответствует исходному. Таким образом сети обучаются параллельно, соревнуясь друг с другом.

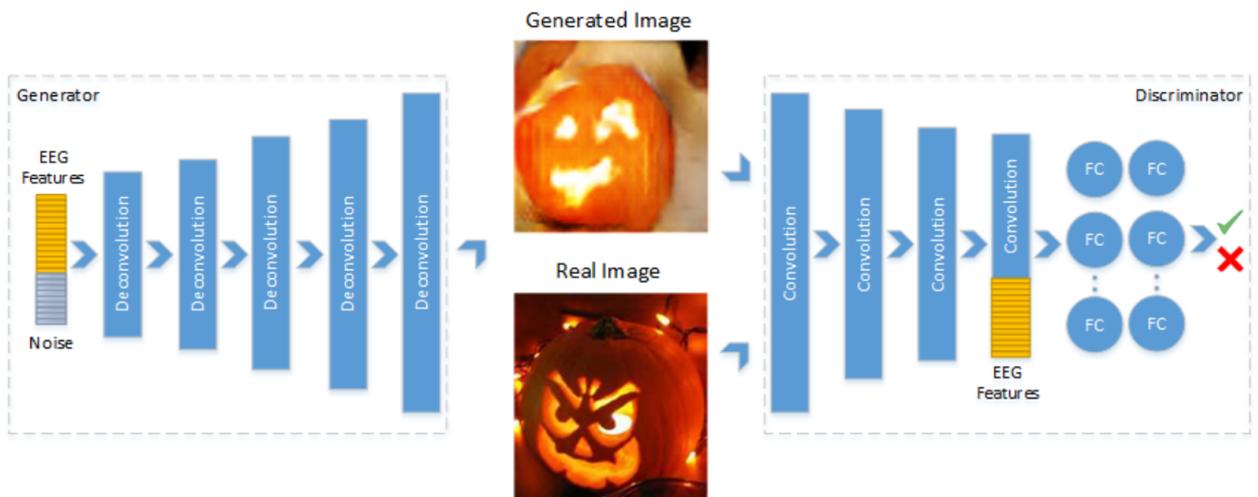


Рис. 3: Схема декодера на основе GAN (из [6])

3. Сбор данных

Первым этапом работы стала организация процесса сбора данных. Для этого были проведены сессии записи ЭЭГ следующего вида: доброволец надевал нейроинтерфес и занимал удобное, расслабленное положение перед монитором для минимизации помех от мышечной активности. Далее запускался показ подготовленного ряда фотографий, во время которого фиксировались показания с датчиков НКИ.

Ряд состоял из 350 фотографий, которые были отобраны из 5 классов (Рис 4) в открытой базе изображений ImageNet [5], 70 штук на категорию. Все изображения одного класса показывались подряд, каждое по 0.6 секунд, а между классами был 10-ти секундный перерыв с нейтральным фоном. При выборе изображений особое внимание уделялось положению главного объекта в кадре и отсутствию ярких, не относящихся к классу предметов, которые могли бы привлечь внимание испытуемых.



Рис. 4: Категории изображений

В эксперименте участвовало четыре физически здоровых человека в нормальном психическом состоянии.

Далее представлено описание технических средств.

3.1. Emotiv EPOC+

Для считывания ЭЭГ использовался нейроинтерфес Emotiv EPOC+, который имеет 14 датчиков (также имеются два референсных для коррекции сигнала), расположенных на поверхности головы. Устройство считывает данные с внутренней частотой дискретизации 2048 Гц, применяет аппаратную фильтрацию помех и понижает частоту дискрети-

зации до значения установленного пользователем (128 или 256 Гц). Данные с каждого датчика кодируются 16 битами и передаются на беспроводной приемник.

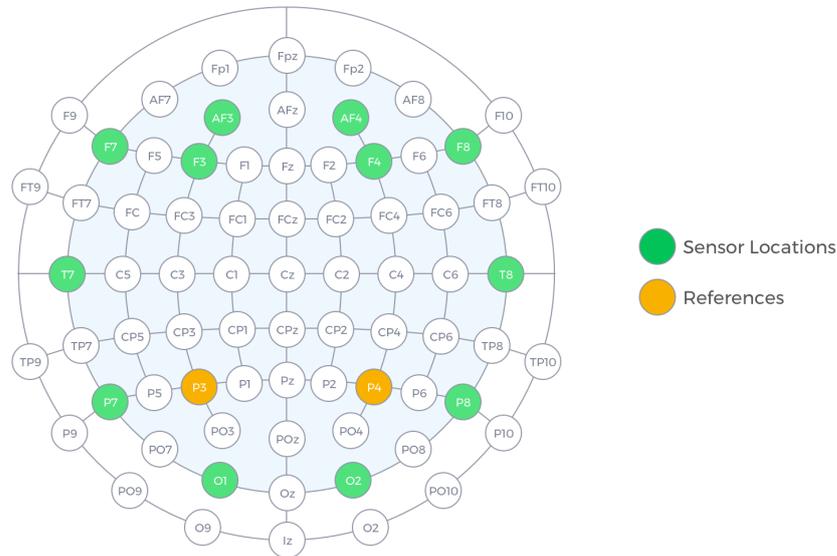


Рис. 5: Положение датчиков Emotiv EPOC+ (из руководства [3])

3.2. Программное обеспечение

Как говорится в работе [9], проприетарное программное обеспечение от компании Emotiv, позволяющее работать с ЭЭГ, требует лицензии, поэтому для снятия показаний была использована библиотека с открытым исходным кодом *SyKit*, которая была модифицирована для добавления таймстемпов к считываемым значениям.

Для настройки положения шлема и контроля за качеством контакта электродов с поверхностью головы во время эксперимента использовалось приложение *EmotivAPP*.

3.3. Предобработка

Полученные в ходе эксперимента сырые данные ЭЭГ были разделены на непрерывные отрезки по таймстемпам показываемых изображений и от начала каждого убраны ещё 40мс для уменьшения влияния предыдущего изображения. После этого все отрезки были выров-

нены по минимальной длине. В итоге получилось 2100 массивов размера (120x14) отмеченных классом и именем изображения. Значения были стандартизованы вдоль каналов (вычитание среднего и деление на стандартное отклонение).

4. Архитектура модели

Модель состоит из двух основных частей: энкодер и декодер-генератор. Задача энкодера — извлечь визуальные признаки из сырой электроэнцефалограммы, а декодера — из этих признаков сгенерировать изображение.

Важным шагом здесь является проверка предположения о том, что в электроэнцефалограмме записанной с помощью нейроинтерфеса ЕРОС+ содержится информация о визуальных стимулах человека, иначе энкодеру будет нечего извлекать. В работах [1, 2] было доказано, что по ЭЭГ, снятым высокоточными устройствами можно успешно классифицировать объект, на который смотрит человек.

Для доказательства этого предположения с текущим оборудованием необходимо было построить рабочий классификатор. Если он построен, то эту же модель без последнего слоя можно использовать, как энкодер.

4.1. Энкодер

В качестве метрики будет использоваться точность ассигасу, определяемая как отношение правильно классифицированных объектов к общему числу объектов:

$$\text{accuracy} = \frac{\text{correct}}{\text{total}}$$

В качестве базовой модели для сравнения взят генератор случайных чисел, который для 5 классов имеет точность 20%.

Известно, что входные данные ЭЭГ являются временными рядами, поэтому в первую очередь были опробованы рекуррентные нейронные сети, в частности LSTM, которые разрабатывались для выделения связей на некотором временном промежутке и успешно решают такие задачи. Были опробованы разные конфигурации моделей: однослойные, многослойные, с добавлением полносвязных слоев. Результаты лучших моделей приведены в таблице 1.

Кроме этого, были опробованы полносвязные нейронные сети, кото-

рые не показали хороших результатов.

Структура	Accuracy
2xLSTM(56)	45.97%
3xLSTM(28)	39.86%
LSTM(56) + Dense(14)	43.01%

Таблица 1: Примеры классификаторов

Таким образом, наилучшую точность показала модель, состоящая из двух идущих подряд LSTM слоев, по 56 узлов в каждом. После оптимизации гиперпараметров ее точность составила 45.97%, результаты отдельно по классам приведены в таблице 2.

Класс	Accuracy
Airliner	43.34%
Sorrel	45.19%
Jack-o'-lantern	48.52%
Panda	53.13%
Anemone fish	39.60%

Таблица 2: Точность по классам

Точность предсказания модели получилась сильно выше, чем базовые 20%, что позволяет принять предположение о том, что в изначальных данных содержится визуальная информация и модель её успешно находит. Это позволяет нам убрать её последний слой и получить энкодер (рис. 6), который извлекает из кусочка электроэнцефалограммы вектор признаков с нужной информацией.

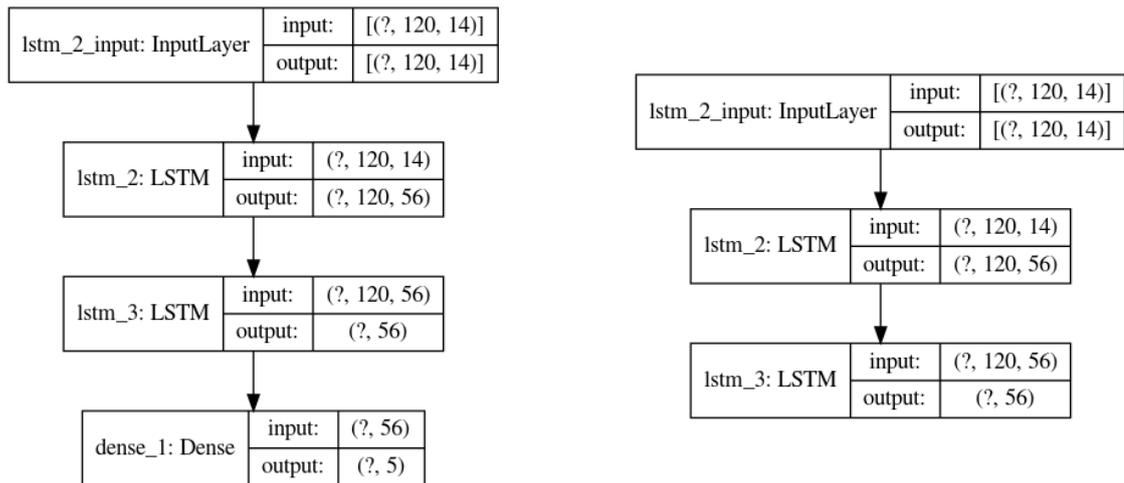


Рис. 6: Выделение энкодера

4.2. Декодер

Задача декодера — восстановить оригинальное изображение по вектору признаков, полученных от энкодера. В качестве декодера был выбран вариант conditional GAN, который и предназначен для генерации изображений обладающих некоторым признаком в зависимости от входящей переменной. GAN представляет собой объединение двух моделей, которые обучаются параллельно, соревнуясь друг с другом. Одна из них — генератор — стремится создать такие изображения, что другая — дискриминатор — решит, что они настоящие. А дискриминатору в свою очередь на вход подаются как настоящие изображения, так и созданные генератором. Таким образом они совершенствуются параллельно до тех пор, пока в итоге не получается генератор, который создает изображения очень похожие на настоящие.

Устройство генератора схожее у всех подобных моделей: сначала исходный вектор расширяется до трехмерного тензора с квадратной стороной. Зачастую это (4, 4). Затем последовательными деконволюционными слоями он расширяется в два раза на каждом шаге, пока не достигнет профиля (64, 64) — целевой размер изображения. После этого по нему проходятся сверткой для придания вида (64, 64, 3). Что и выдается как финальное изображение.

У дискриминатора всё с точностью до наоборот. Он с помощью свер-

ток уменьшает профиль до формы (4, 4), после этого выпрямляет его до вектора и выходной слой делает предсказание о реалистичности входного изображения.

Результаты таких моделей могут выглядеть впечатляюще, но они сложно обучаются, часто расходятся и требуют большого количества изображений. В моем случае данных было мало, даже с учетом всевозможных преобразований исходных изображений, поэтому было принято решение уменьшить целевой размер изображений до 32x32 пикселей.

В ходе экспериментов, модель многократно менялась, были опробованы разные конфигурации. Итоговые схемы основных элементов приведены ниже.

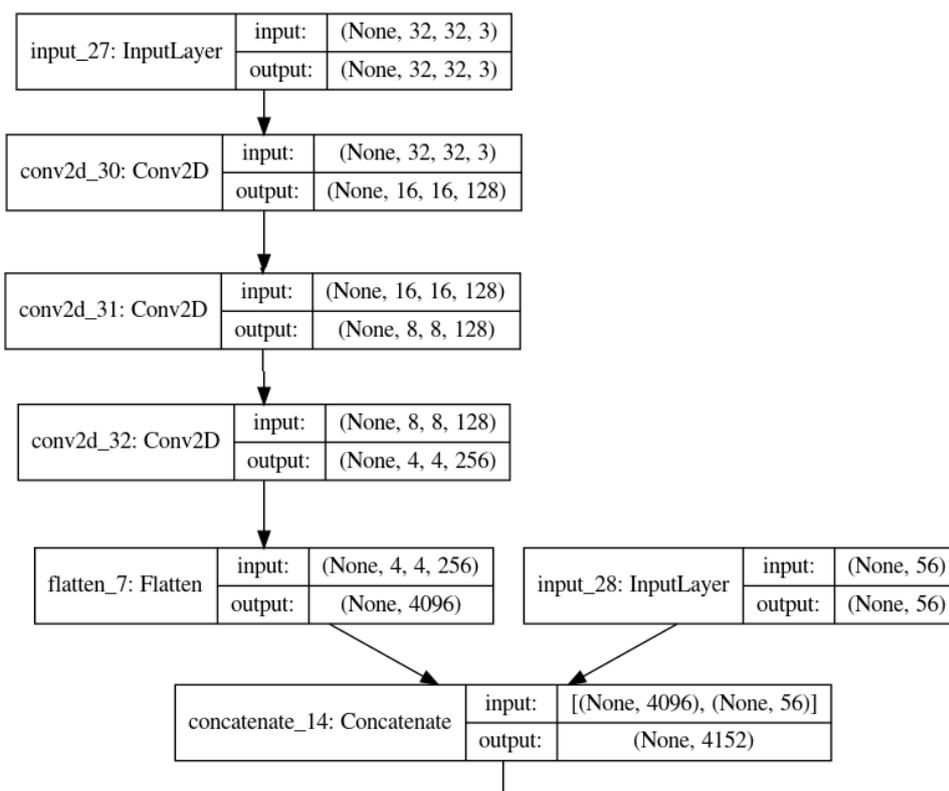


Рис. 7: Основные элементы дискриминатора

На рисунке 7 изображена схема дискриминатора: на вход подается изображение для оценки и вектор, сгенерированный энкодером. Входящее изображение проходит через три сверточных уровня, на каждом из которых понижает размер в два раза. Далее предыдущий результат выпрямляется и к нему присоединяется вектор от энкодера. Далее через

полносвязный слой выдается предсказание о реалистичности изображения.

На рисунке 8 изображен генератор: входящий вектор от энкодера конкатенируется с вектором шумов, ему придается форма трехмерного тензора с профилем (4, 4) и далее проходит через три деконволюционных слоя, в каждом из которых увеличивает профиль в два раза. Последний сверточный слой преобразует вход в цветное изображение.

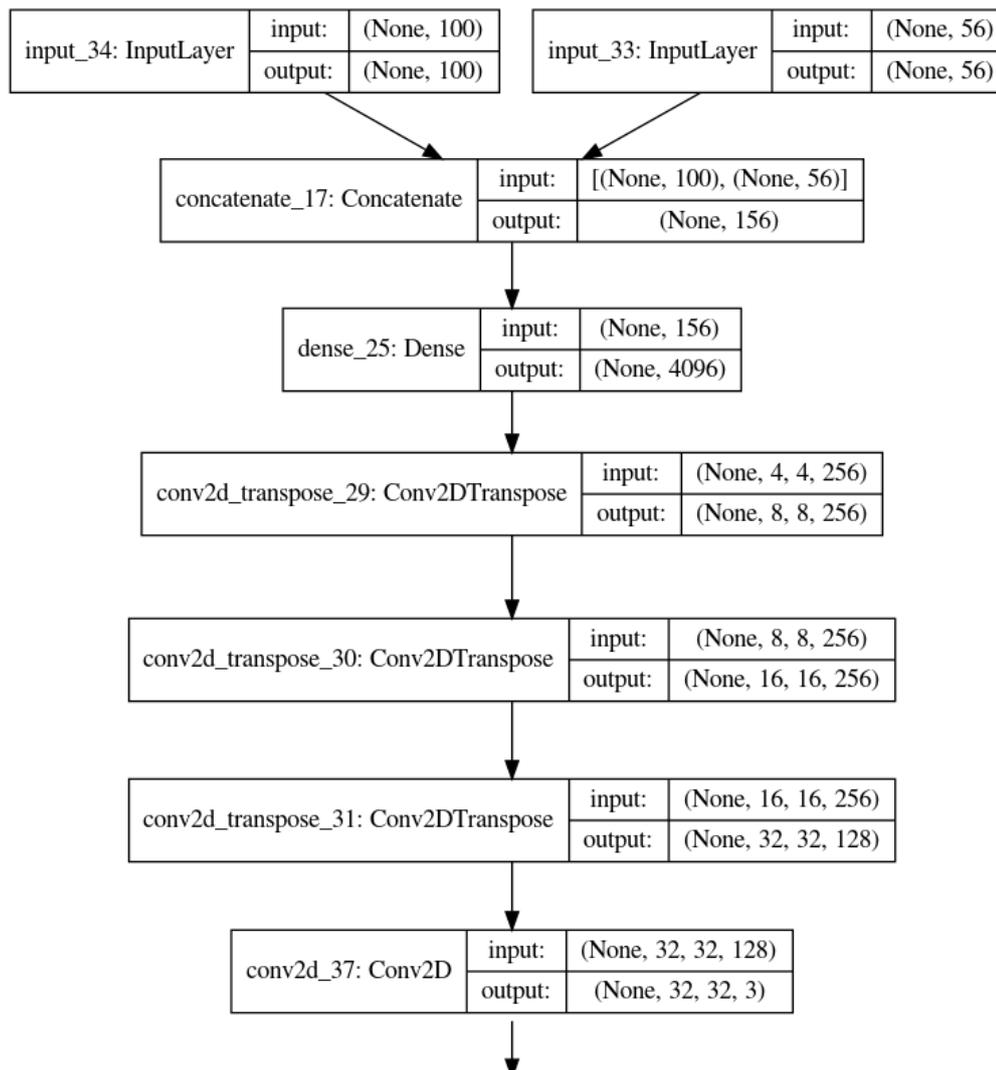


Рис. 8: Основные элементы генератора

За каждым сверточным и деконволюционным слоем следует слой нормализации и активационный с функцией LeakyReLU, которые показывают повышение стабильности обучения.

Процесс обучения подобной модели состоит в пропуске через дискриминатор партии ложных изображений и партии реальных с из-

менением весов. Затем генератором создается партия изображений, для которых дискриминатор вычисляет ошибку. И генератор по этим данным обновляет свои внутренние веса.

С помощью обученного в такой модели генератора удалось получить множество разных изображений. Достаточно много совсем никаких. Лучшие примеры приведены на рисунке 9, на которых переданы некоторые основные черты исходных классов, например цвет. Хотя не столь успешные, как хотелось бы.

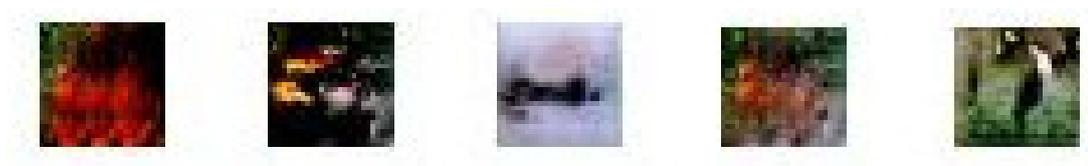


Рис. 9: Тыква, рыбка, самолет, лошадь, панда

5. Оценка результатов

Работа была проделана с использованием бюджетного нейроинтерфейса, открытых программных библиотек и открытых данных. И несмотря на колоссальную разницу в классе устройств, относительно оригинальной статьи [6], и отсутствие большого количества данных, получилось добиться ощутимых результатов. Самое главное, было показано, что в этом направлении можно развиваться и здесь ещё есть множество вариантов для действий: от увеличения количества исходных данных до применения других методов и алгоритмов, не рассмотренных в работе.

Заключение

В ходе работы были получены следующие результаты:

- Организован сбор необходимых данных с помощью устройства Emotiv EPOC+.
- Реализован классификатор для определения класса изображения на которое смотрит человек во время считывания ЭЭГ.
- Реализован генератор изображений из пространства признаков энкодера на основе GAN.

Список литературы

- [1] Blair Kaneshiro Marcos Perreau Guimaraes Hyung-Suk Kim Anthony M. Norcia Patrick Suppes. A Representational Similarity Analysis of the Dynamics of Object Processing Using Single-Trial EEG Classification // Advances in neural information processing systems.
- [2] Concetto Spampinato Simone Palazzo Isaak Kavasidis-Daniela Giordano Mubarak Shah Nasim Souly. Deep Learning Human Mind for Automated Visual Classification // 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR).
- [3] Emotiv. EPOC+ User Manual. — URL: <https://emotiv.gitbook.io/epoc-user-manual/> (online; accessed: 10.05.2020).
- [4] Ian Goodfellow Jean Pouget-Abadie Mehdi Mirza Bing Xu David Warde-Farley Sherjil Ozair Aaron Courville, Bengio. Yoshua. Generative adversarial nets // Advances in neural information processing systems.
- [5] Deng J., Dong W., Socher R. et al. ImageNet: A Large-Scale Hierarchical Image Database. — URL: <http://www.image-net.org/> (online; accessed: 10.05.2020).
- [6] Isaak Kavasidis Simone Palazzo Concetto Spampinato Daniela Giordano Mubarak Shah. Brain2Image: Converting Brain Signals into Images // MM '17: Proceedings of the 25th ACM international conference on Multimedia.
- [7] U Rajendra Acharya S Vinitha Sree G Swapna Roshan Joy Martis- andJasjit S Suri. Automated EEG analysis of epilepsy: a review // Knowledge-Based Systems. — 2013. — Vol. 45. — P. 147–165.
- [8] Yuan-Pin Lin Chi-Hong Wang Tzyy-Ping Jung Tien-Lin Wu Shyh-KangJeng Jeng-Ren Duann, Chen Jyh-Horng. EG-based emotionrecognition in music listening // IEEE Transactions on Biomedical Engineering. — 2010. — Vol. 57. — P. 1798–1806.

- [9] Александрович Евсеев Олег. Разработка интерфейса управления компьютером через нейроинтерфейс // 2019 ВКР кафедры системного программирования СПбГУ. — 2019.