ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ

«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»

Факультет информатики, математики и компьютерных наук

**Васильева Инна Алексеевна**

# ПРИМЕНЕНИЕ МАШИННОГО ОБУЧЕНИЯ ДЛЯ ЗАДАЧ КЛАССИФИКАЦИИ В ИСКУССТВЕ

Выпускная квалификационная работа - МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ

по направлению подготовки 01.04.02 Прикладная математика и информатика

образовательная программа «Интеллектуальный анализ данных»

Рецензент

д-р физико-математических наук, проф.

Калягин Валерий Александрович

Научный руководитель

Доцент кафедры ПМИ

факультета БИиПМ,

к-т физико-математических наук

Бацын Михаил Владимирович

Нижний Новгород, 2018

Оглавление

# Введение

Понятие «машинное обучение» в настоящее время изветно не только специалистам – данная область сыскала впечатляющую популярность благодаря огромному потенциалу, способному упростить жизнь человека. Технологии машинного обучения стали чрезвычайно популярными, благодаря ряду успешных проектов, результатом которых являются известные продукты, такие как: библиотека алгоритмов компьютерного зрения OpenCV с открытым программным кодом, Pagerank - алгоритм ранжирования от корпорации Google, прошумевшее на весь мир приложение Prizma, превращающее фотографии в картины и др. Достижения этой научной дисциплины буквально прорвали технологические и психологические барьеры, которые стояли на пути реализации идеи о возможном более эффективном использовании данных с целью извлечения из них новых знаний.

Развитый и удобный инструментарий и доступность методологии стимулировали творческую фантазию разработчиков в применении методов машинного обучения к решению самых разнообразных и сложных задач. Благодаря масштабному распространению информационно-коммуникационных технологий, методы машинного обучения давно вошли в повседневную жизнь интернет-пользователей. Например, лента новостей социальной сети Facebook использует машинное обучение для составления персонального контента для каждого пользователя, алгоритмы машинного обучения уже много лет помогают фильтровать спам и иденцифицировать взломы аккаунтов пользователей на основании сравнения паттернов поведения обычных зарегистрированных пользователей и пользователей, чьи аккаунты были скомпрометированы или генерировать таргетированную контекстную рекламу с учетом контекста открытой интернет-страницы, анализа пользовательской активности, обработки истории потребления контента и пр.

С помощью машинного обучения компьютеры начинают решать задачи, которые до недавнего времени считались исключительно человеческими. Алгоритмы пишут музыку, сочиняют стихи и рисуют картины. Прогресс в этой области очевиден, всё сложнее отличить произведения искусства, созданные человеком, от сгенерированных компьютеров.

Изображение 1. Проект Neural Style переносит художественный стиль картины «Звёздная ночь» на ночную фотографию Стэнфордского кампуса - алгоритм для объединения одного изображения со стилем другого с использованием свёрточных нейронных сетей.

Заметный рост объема цифровых данных в последнее время значительно облегчает доступ к произведениям искусства для широкой публики. Вместе с тем, все больше и больше усилий прилагается для создания автоматических

решений для обработки изображений, которые облегчают понимание искусства. Эти решения могут быть нацелены на получение качественных и высококачественных цифровых версий изображений [4], или повлиять на различные аспекты, такие как распознавание сцен, виртуальная реставрация, омоложение цвета и т.п. [5]. Другим, более подходящим для конечной цели аспектом является распознавание контекста – отдельное, популярное сейчас направление в компьютерном зрении, чем, в случае данной работы является идентификация направления (стиля) искусства – задача нетривиальная для неспециалистов в области истории искусств.

Согласно словарю, направление (течение, стиль) в искусстве – понятие, отражающие исторически сложившуюся общность художественных признаков в том или ином виде искусства (или одновременно в нескольких искусствах), характерную для разных эпох и народов и обусловленную единством идейно-эстетических устремлений творческого меньшинства. В то время как некоторые произведения искусства четко определены в единое художественное направление, другие находятся в некотором переходном состоянии, поскольку художники любят экспериментировать с новыми идеями, постепенно приходя к созданию нового художественного стиля. Кроме того, в то время как фактические характеристики работ очевидно относят произведение к какому-либо стилю, или даже нескольким, звестны случаи, когда его автор по личным причинам отказывался от классификации таким образом и порождал споры.

Недавно в Метрополитен-музее Нью-Йорка было выпущено более 375 000 картин, посвященных общественному достоянию, которые вскоре будут доступны для индексации (The Met, 2017 [1]). Однако для индексации художественных изображений требуется описание визуального стиля изображения, а также описание содержимого, которое обычно используется

для индексации нехудожественных изображений. Выявление стиля изображения полностью автоматическим способом является сложной проблемой. Действительно, хотя стандартные задачи классификации, такие как распознавание лиц, могут опираться на четко идентифицируемые черты, такие как глаза или нос, классификация визуальных стилей не может опираться на какую-либо определенную функцию. Эта проблема особенно сложна для нерепрезентативной художественной работы.

В этой работе рассматривается проблема вычислительной категоризации оцифрованных картин в художественных стилях. В отличие от других направлений классификации изображений, таких как распознавание сцен или объектов, где существуют большие базы данных и стандарты оценивания, оценка описанных методов осуществлялась на имеющейся в открытом доступе небольшой базе данных картин, разбитых по 18 напавлениям. На этом наборе данных удалось получить хороший результат – 95% точности.

**Цель:** используя методы машинного обучения разработать метод распознавания направления изобразительного искусства, имеющий хорошую точность относительно уже существующих подходов. В идеале – превзойти.

**Задачи:**

- Провести обзор существующих методологий и наборов данных для данной задачи;
- Разработать метод распознавания направления изобразительного искусства;
- Провести вычислительные эксперименты на данных;
- Проанализировать результаты работы метода.

# Глава I. Обзор литературы

Такого рода творческая работа обычно привлекает много внимания в ИИ сообществе, возможно, из-за поднимаемых философских вопросов, или из-за потенциальных сфер применения в качестве приложений. В результате в ряде публикаций обсуждается проблема как генерации того, что принято называть искусством, так и распознавания стиля искусства в разных художественных областях, таких как изобразительное искусство (Gatys et al., 2015) [19] или музыка (Oordet al., 2016; Huang and Wu, 2016) [23].

В области генерации визуального искусства Gatys et al. смогли построить модель определенного стиля живописи, а затем перенести ее на нехудожественные фотографии. С технической точки зрения, генерация «произведений искусства» не очень отличается от распознавания стиля искусства: в обоих случаях первым шагом является точная модель одного или нескольких художественных стилей. Gatys et al. (2015) предлагает взять за основу обучение глубокой сети VGG (Simonyan and Zisserman, 2014) [21] с большим количеством фотографий из определенного художественного стиля. Затем они сгенерировали результрующее изображение – комбинацию стиля с и исходного изображения. В этом и заключается важное отличие от проблемы распознавания художественного стиля - для модели необходимо как можно сильнее отделить стиль от контента, чтобы успешно передать стиль в новый контент. В случае распознавании стиля описание содержимого используется как дополнительная информация (например, изображение человека, скорее всего, появляется в картине импрессиониста, чем в абстрактной живописи).

В нескольких научных исследованиях была рассмотрена проблема распознавания стилей с использованием существующих подходов к компьютерному обучению. Например, Florea et al. (2016) [3] оценивали производительность различных комбинаций популярных признаков

изображения (гистограммы градиентов, пространственных огибающих, дискриминационные названия цветов и т.д.) с различными классификационными алгоритмами (SVM, случайные леса и т.д.). Несмотря на размер набора данных и ограниченное количество классов для прогнозирования, они отметили, что несколько стилей по-прежнему трудно отличить с помощью этих методов. Они также демонстрируют, что добавление дополнительных признаков более не улучшает точность моделей, по-видимому, из-за так называемого проклятия проклятия размерности.

Хотя работа с заранее вычисленными признаками может быть полезна для лучшего понимания поведения классификатора, результирующие классификаторы обычно не достигают высокого уровня точности.

В 2015 году Karayev et al. [18] определил, что большинство систем, предназначенных для автоматического распознавания стилей построены на простейших признаках и позволяют распознавать стили с использованием линейного классификатора, обучаемого на признаках, автоматически извлекаемыми с использованием сверточной нейронной сети (CNN). Для этого была использована CNN AlexNet (Krizhevsky et al., 2012) [22], обученная на ImageNet для распознавания объектов на нехудожественных фотографиях. Не так давно Tan et al. (2016) [20] рассмотрел проблему, используя вариации той же нейронной сети и сумел добиться максимальной производительности с полностью автоматической процедурой с точностью 54,5% по 25 стилям.

Имеющиеся решения для анализа искусства и картин с использованием методов компьютерного зрения базируются на средних и малых базах данных. Резюме таких методов представлено в таблице 1.

| Авторы | Год | Количество направлений | Количество изображений | Тестовая выборка, | Точность, |
|---|---|---|---|---|---|
| | | | | | |
| | | | | 10 | |
| | | | | 20 | |
| | | | | 29,8 | |
| | | | | 20 | |
| | | | | 46,5 | |
| | | | | 10 | |
| | | | | 10 | |
| **Karayev et al. [18]** | | | | - | NA |
| **Tan et al. [20]** | | | | | |
| | | | | | |
| **Полученный результат** | | | | | |

Таблица 1. Сравнение наборов данных

Можно с легкостью заметить, что размер баз данных (и число исследуемых направлений искусств) увеличивался со временем, в то время как процент правильного распознавания направления стабилизировался в диапазоне 50%. Некоторые из наиболее репрезентативных баз данных, используемых для идентификации движения искусства:

- Набор художественных направлений [6]. Изображения, собранные из Web Museum-Paris, разбитые на следующие художественных направления: классицизм, кубизм, импрессионизм, сюрреализм, экспрессионизм.
- Набор художественных направлений [7]. Изображения из различных интернет-источников были разделены на 5 стилей: абстракционизм, импрессионизм, кубизм, поп-арт и реализм.

- Набор данных живописи в разных стилях [8]: изображения, собранные в Интернете, были сгруппированные в абстрактный экспрессионизм, барокко, кубизм, граффити, импрессионизм и ренессанс.
- Набор данных художественных стилей [9]: картины 9 художников были сгруппированы в три направления: импрессионизм, абстрактный экспрессионизм и сюрреализм.
- Набор художественных стилей [10] с изображениями, собранными из набора данных Artchive Fine Art, и сгруппированные в изобразительные стили: ренессанс, барокко, импрессионизм, кубизм, абстракционизм, экспрессионизм и поп-арт.
- Набор данных Paintings-91 [11] с изображениями, собранными в Интернете. В то время как эта база данных больше, чем предыдущие, она содержит только картины художников, работавших каждый в одном художественном направлении. После разметки изображений по направлениям, а не по художникам, результатом стала меньшая база данных, содержащаяся абстрактный экспрессионизм, барокко, конструктивизм, кубизм, импрессионизм, неоклассицизм, поп-арт, постимпрессионизм, реализм, ренессанс, романтизм, сюрреализм и символизм.
- Набор данных художественных стилей [12] является основой предлагаемой базы данных. Мы увеличили этот набор данных, добавив больше изображений для иллюстрации существующих художественных движений и добавив еще 4 новых.
- Набор художественных стилей [13] содержит изображения, собранные из WikiArt и сгруппированные в: абстрактный экспрессионизм, барокко, кубизм, импрессионизм, экспрессионизм, поп-арт, рококо, реализм, ренессанс и сюрреализм.

- Набор данных Pandora (набор оцифрованных картин для распознавания направления искусства) [2] состоит изображений, разбитых на 18 направлений. Именно эти данные легли в основу данной работы.

## Данные Pandora

Florea и др. [2], базируясь на собрании изображений WikiArt, собрали базу данных размеченных по направлениям искусства изображений Pandora и использовали ее в своей работе. В таблице 2 приведены стили и количество изображений на каждый, а на рисунке 1 – примеры представителей каждого стиля.

| **Направление искусства** | **Количество изображений** | **Временные рамки** |
|---|---|---|
| Византийская иконопись | 847 | 500 - 1400 |
| Ранний ренессанс | 752 | 1280 - 1450 |
| Северный ренессанс | 821 | 1497 - 1550 |
| Высокий ренессанс | 832 | 1490 - 1527 |
| Барокко | 990 | 1590 - 1725 |
| Рококо | 832 | 1650 - 1850 |
| Романтизм | 895 | 1770 - 1880 |
| Реализм | 1200 | 1880 - 1880 |
| Импрессионизм | 1257 | 1860 - 1950 |
| Постимпрессионизм | 1276 | 1860 - 1925 |
| Экспрессионизм | 1027 | 1905 - 1925 |
| Символизм | 1057 | 1850 - 1900 |
| Фовизм | 719 | 1905 - 1908 |
| Кубизм | 1227 | 1907 - 1920 |
| Сюрреализм | 1072 | 1920 - 1940 |
| Абстракционизм | 1063 | 1910 - н.в. |
| Наивное искусство | 1053 | 1890 - 1950 |
| Поп-арт | 1120 | 1950 - 1969 |

Таблица 2. Направления искусства, охваченные набором данных Pandora.




Рисунок 1: 18 направлений искусства, представленных в наборе данных.

Данный набор данных был заверен экспертами как в области искусства, так и в области машинного обучения, в результате чего из него были исключены изображения даже с минимальной степенью сомнения, такие как иконопись или фрески на сильно искривленных стенах или изображения с сильно выцветшими красками. Подробнее о нем можно прочитать в [2].

# Глава II. Методология и результаты

## Часть 1. Метод с использованием классических признаков изображения

### Нормализация

В рамках работы с изображениями принято перед вычислением признаков приводить исходные файлы к одному размеру для того, чтобы признаки были наиболее однородными. Однако стандартная практика – сжатие изображений – приводит к тому, что изображения теряют часть информации: края объектов на изображении размываются, а гистограмма цветов становится менее репрезентативной. При этом для определения некоторых стилей, например, импрессионизма, мелкие детали очень репрезентативны. Поэтому для приведения всех изображений к одному размеру $h \times w$ изображение разрезается на необходимое количество фрагментов. Таким образом, из одного исходного изображения размером $H \times W$ после нормализации образуется $\left\lfloor \frac{H}{h} \right\rfloor \cdot \left\lfloor \frac{W}{w} \right\rfloor$ фрагментов размера $h \times w$. Подобное разделение имеет смысл, так как в рассматриваемом массиве данных все изображения выполнены в одном стиле, т.е. полученные фрагменты имеют тот же стиль, что и исходное изображение. Значения $h$ и $w$ определяются как наименьшая высота и наименьшая ширина изображения из набора, соответственно. Таким образом, для каждого изображения будет получен как минимум один фрагмент. Для рассматриваемого набора данных были получены значения $h = 125;\ w = 101.$

### Определение стиля фрагмента

Задача распознавания стиля фрагмента рассматривается как задача классификации, в качестве входных данных выступают сами фрагменты, в качестве класса – стиль. Для распознавания стиля фрагмента в качестве

признаков использовалась гистограмма цветов. Сначала, фрагмент был переведен в формат HSV [14], для каждого пикселя был определён его оттенок (hue) по шкале от 0 до 255. Затем, на основе полученных значений оттенка строится гистограмма, т.е. для каждого значения оттенка вычисляется количество пикселей этого оттенка. Таким образом, для каждого фрагмента был получен вектор размерности 256, который и используется как вектор признаков.

Задача классификации решается с помощью ансамбля решающих деревьев, который был получен с помощью алгоритма Random Forest [15]. Этот метод решения известен как один из наиболее эффективных алгоритмов для задачи определения стиля изображения [3]. Эмпирическим методом были выбраны следующие параметры алгоритма:

| Параметр | Значение |
|---|---|
| **Количество деревьев** | 50 |
| **Критерий оптимизации** | Индекс Джини |
| **Максимальная глубина дерева** | Не ограничена |
| **Минимальное количество объектов для разделения узла** | 2 |
| **Количество рассматриваемых признаков при разделении** | 16 |

Таблица 3. Параметры алгоритма Random Forest [15]

## Использование предсказаний стиля фрагмента для определения стиля исходного изображения

*Без использования дополнительных признаков*

Каждое изображение можно рассматривать как последовательность фрагментов, при этом в зависимости от размера изображения эти последовательности могут иметь разные длины для разных изображений. Для

приведения этих последовательностей к одному формату используется техника паддинга. Последовательность $S$ приводится к последовательности $S$ следующим образом:

$$S_i = \begin{cases} S_i, i \le \min\{|S|, n\} \\ 0, \min\{|S|, n\} < i \le n \end{cases}$$

Таким образом, все последовательности $S$ имеют одинаковую длину – $n$. В рамках рассматриваемой задачи было выбрано $n = 150$, т.к. только 27 изображений имеют больше 150 фрагментов. Соответственно, только они потеряют часть информации при паддинге, при этом длины последовательностей не будут избыточно большими.

Для получения предсказаний о стиле изображения, имея предсказания стиля фрагментов, используется нейронная сеть. Так как нейронная сеть по умолчанию не может работать с категориальными признаками, они приводятся к бинарным следующим образом: предсказание для каждого фрагмента $S_i \in [0,18]$ (значение одного из 18 классов или 0) заменяется на бинарный вектор из восемнадцати признаков $S_{i\,1\times18}$, где

$$S_{i\,j} = \begin{cases} 1, \ S_i = j \\ 0, \ \textit{иначе} \end{cases}$$

Таким образом получается бинарная матрица $S_{150\times18}$, которая затем трансформируется в бинарный вектор $X_{2700}$. Этот вектор идёт на вход нейронной сети. Сама нейронная сеть имеет следующую архитектуру:

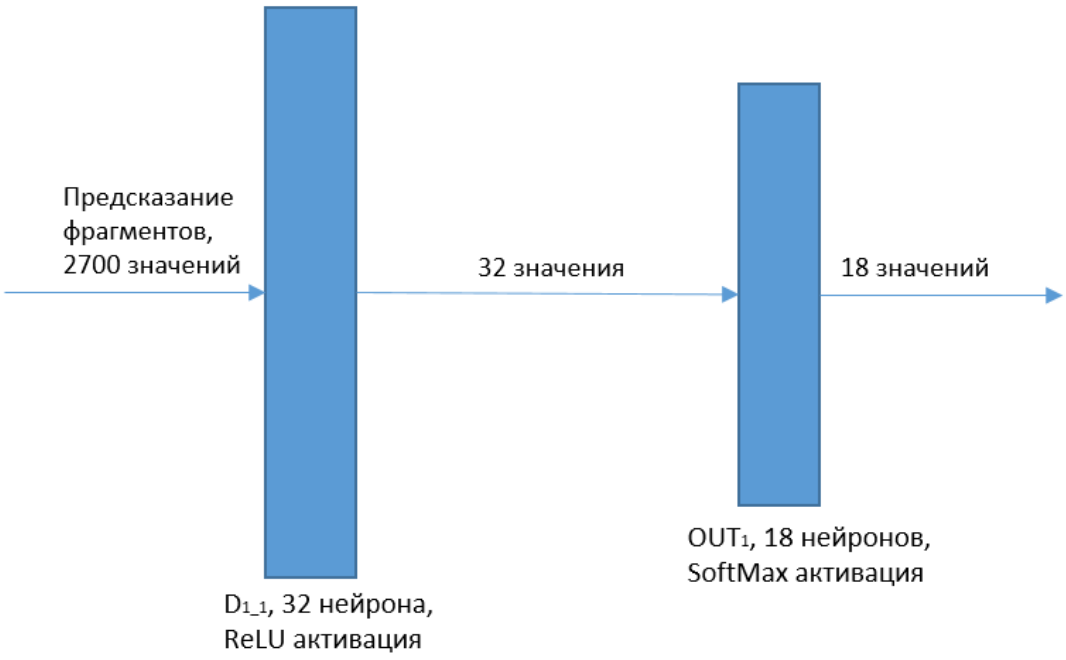


Рис 2. Архитектура нейронной сети, использующей только предсказания фрагментов.

Так как эта сеть решает задачу классификации на 18 классов, выходной слой содержит 18 нейронов, выходное значение каждого из этих нейронов показывает предсказание, соответствует ли изображение стилю, соответствующего этому нейрону.

Приведённая сеть была обучена с помощью алгоритма ADAM [16], в качестве функции потерь использовалась среднеквадратичная ошибка.

*С использованием гистограммы топографических признаков*

В рамках этого подхода наряду с использованием предсказаний для каждого фрагмента в качестве признаков для определения стиля изображения используется гистограмма топографических признаков (HoT) [17]. Эти признаки зарекомендовали себя как одни из наиболее эффективных для определения стиля изображения [1].

Для вычисления HoT изображение приводится к чёрно-белому формату, а затем рассматривается как функция $I:[1..H]\times[1..W]\to\mathbb{R}$, где $H$ - высота изображения, $W$ - его ширина в пикселях, $I(x,y)$ – значение цвета пикселя в координатах $(x,y)$ по черно-белой шкале. Затем вычисляется градиент $\nabla I$ и гессиан $h$ этой функции в каждой точке.

Затем для каждой точки $(i,j)$ вычисляются собственные числа гессиана $\lambda_1(i,j)\le\lambda_2(i,j)$ и направление пространства собственных векторов, соответствующего $\lambda_1(i,j)$. Это направление обозначается как $\theta_\lambda(i,j)$. Затем на основе вычисленных значений строятся следующие гистограммы:

- Жёсткая гистограмма ориентации изгиба поверхности

  Для каждого пикселя, 1 добавляется к величине, соответствующей направлению $\theta_\lambda(i,j)$, если значение меньше собственного значения больше некоторого $T_\lambda$.

$$H_1^H(\theta)=\frac{1}{Z_1}\sum_{\substack{i=[1..H]\\ j=[1..W]}}\Big(\theta_\lambda(i,j)==\theta\Big)\cdot\big(\lambda_2(i,j)>T_\lambda\big)$$

- Мягкая гистограмма ориентации изгиба поверхности

  Аналогично жёсткой гистограмме, только вместо 1 добавляется разница между собственными значениями гессиана.

$$H_2^H(\theta)=\frac{1}{Z_2}\sum_{\substack{i=[1..H]\\ j=[1..W]}}\Big(\theta_\lambda(i,j)==\theta\Big)\cdot\big(\lambda_2(i,j)-\lambda_1(i,j)\big)$$

- Гистограмма меньшего собственного значения

$$H_3^H(k) = \frac{1}{Z_3} \sum_{\substack{i=[1..H] \\ j=[1..W]}} \left( \lambda_2(i,j) \in \left[ (k-1)\frac{M_{\lambda 2}}{N_{bins}}; k\frac{M_{\lambda 2}}{N_{bins}} \right] \right)$$

Где $M_{\lambda 2} = \max \lambda_2(i,j)$, $N_{bins}$ - длина гистограммы.

- Гистограмма разности между собственными значениями

$$H_4^H(k) = \frac{1}{Z_4} \sum_{\substack{i=[1..H] \\ j=[1..W]}} \left( \left(\lambda_1(i,j) - \lambda_2(i,j)\right) \in \left[ (k-1)\frac{M_{\lambda 12}}{N_{bins}}; k\frac{M_{\lambda 12}}{N_{bins}} \right] \right)$$

Где $M_{\lambda 12} = \max(\lambda_1(i,j) - \lambda_2(i,j))$, $N_{bins}$ - длина гистограммы.

- Гистограмма ориентации градиента

$$H_1^G(v) = \frac{1}{Z_5} \sum_{\substack{i=[1..H] \\ j=[1..W]}} \left(\nabla I(i,j) == v\right) \cdot \left( \left\| \nabla I(i,j) \right\| > T_G \right)$$

- Гистограмма нормы градиента

$$H_2^G(k) = \frac{1}{Z_6} \sum_{\substack{i=[1..H] \\ j=[1..W]}} \left( \left\| \nabla I(i,j) \right\| \in \left[ (k-1)\frac{M_G}{N_{bins}}; k\frac{M_G}{N_{bins}} \right] \right)$$

Где $M_G = \max(\left\| \nabla I(i,j) \right\|)$, $N_{bins}$ - длина гистограммы.

В соответствии с [_HOT] были выбраны следующие значения параметров:

$T_\lambda = 0.1,\ T_G = 5,\ N_{bins} = 8.$ $Z_1 - Z_6$ – константы для нормализации гистограммы.

Для решения задачи классификации изображений была использована следующая нейронная сеть:

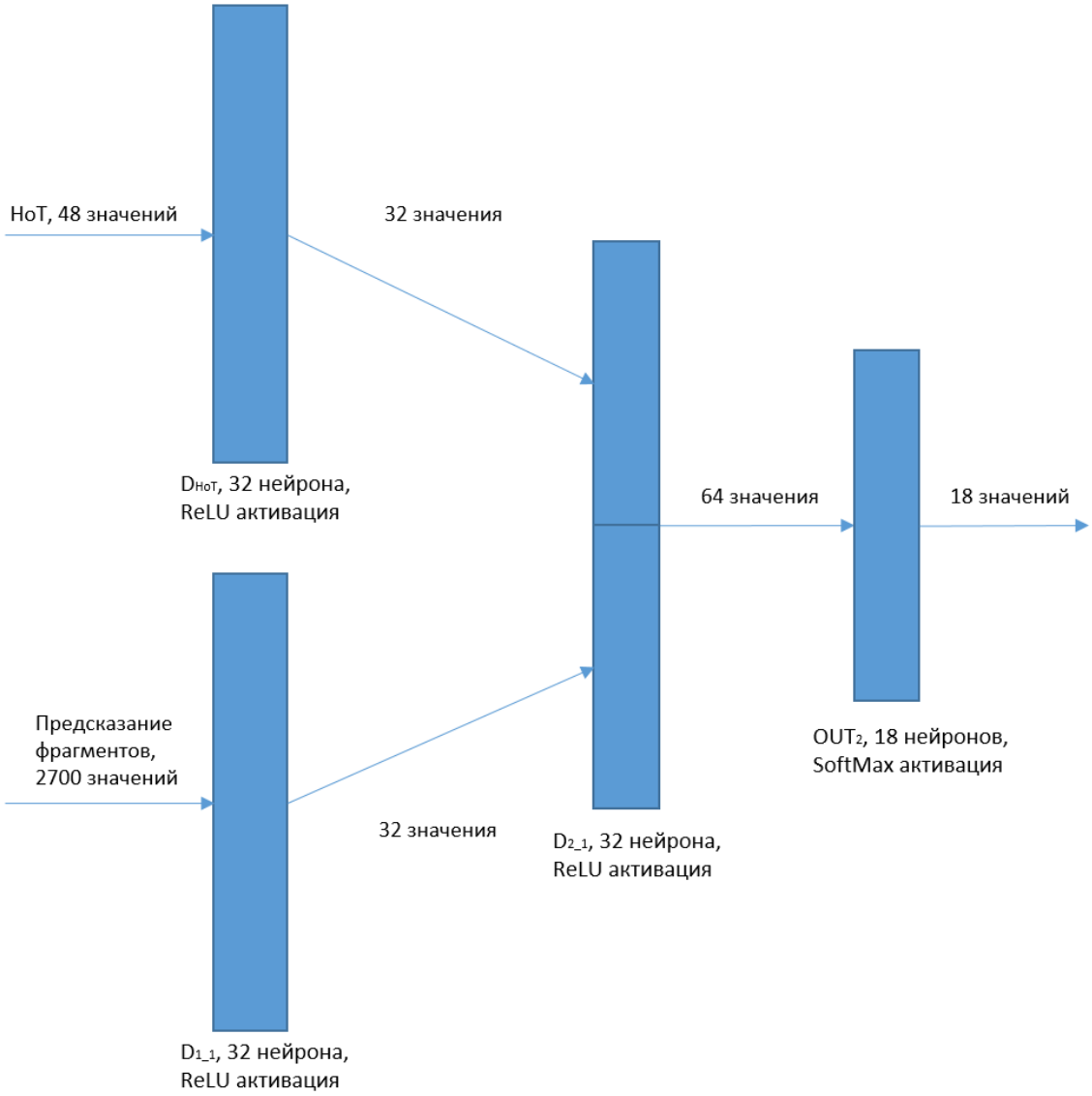


Рисунок 3. Архитектура нейронной сети, использующей предсказания фрагментов и слой $D_{НоТ}$.

Для обучения слоя $D_{HoT}$ была использована отдельная сеть:

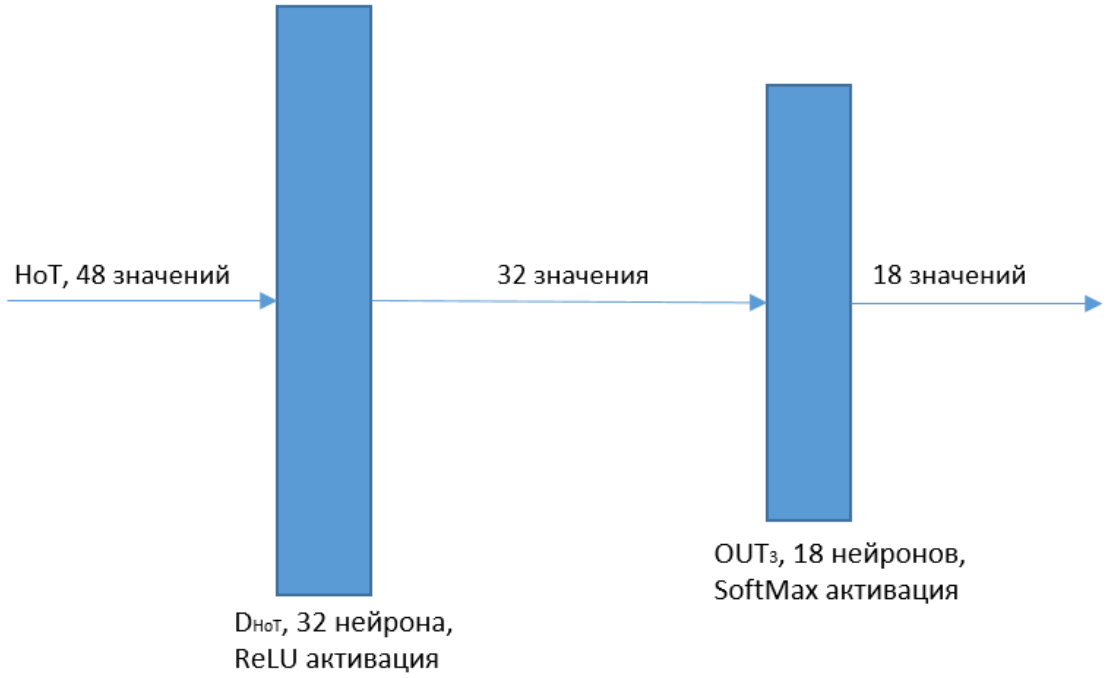


Рисунок 4. Архитектура нейронной сети, для обучения слоя $D_{HoT}$.

После обучения, веса, соответствующие этому слою, были перенесены в нейронную сеть, изображённую на рисунке 3 и зафиксированы. Обе сети были обучены с помощью алгоритма ADAM, в качестве функции потерь использовалась среднеквадратичная ошибка.

## Оценка точности распознавания стиля фрагмента.

Точность распознавания стиля фрагмента была вычислена с помощью 4-кратной кросс-валидации. В среднем была получена точность 22.3%

| Признаки | Точность Random Forest, % |
|---|---|
| **Гистограмма ориентации градиентов (ЧБ) [3]** | 18.4 |

| | |
|---|---|
| **Гистограмма ориентации градиентов (Цветн.) [3]** | 19.6 |
| Гистограмма цвета | **22.3** |
| **Гистограмма топографических признаков [3]** | 29.6 |
| **Локальный бинарный паттерн [3]** | 27.2 |

Таблица 4. Точность Random Forest на разных наборах признаков.

Предложенный набор признаков показал себя достаточно эффективным, при этом это единственный из представленных в таблице наборов, который вычисляется за линейное (зависимое от количества пикселей во фрагменте) время.

## Оценка точности распознавания стиля изображения

Оценка точности распознавания стиля изображения проводилась следующим образом: набор данных (18 тыс. изображений) был разделен на 3 части:

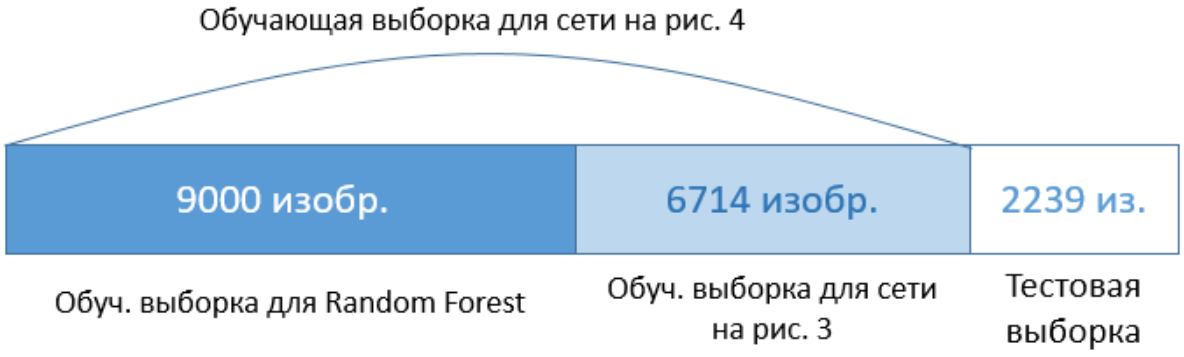


Рисунок 5. Разбиение данных на выборки.

Оценивалась точность предсказаний на тестовой выборке.

| Метод | Точность, % |
|---|---|
| **Только предсказания фрагментов** | 26.3 |
| **Только HoT** | 34.2 |
| **Предсказания фрагментов + HoT** | 36.8 |
| **pHoT+CSD [3]** | 42.3 |

Таблица 5. Сравнение точности представленных методов.

## Часть 2. Метод с использованием популярных архитектур CNN

В данной работе были взяты за основу несколько нейронных сетей, каждая из которых в свое время (а некоторые и до сих пор) являются state-of-the-art методами для ряда задач компьютерного зрения. Претренированные веса для них получены с помощью фреймворков Caffe и Keras. Поверх признаков, полученных с помощью данных архитектур, обучалась логистическая регрессия.

В работы были испробованы следующие архитектуры:

### Alexnet

На вход сети AlexNet [22] подается матрица 227 × 227 × 3. Его структура состоит из пяти сверточных и трех полносвязанных слоев. Сверточные слои имеют соответственно фильтры размером 11 × 11, 5 × 5, 3 × 3 и 3 × 3. Каждый из них соответственно генерирует 96, 256, 384, 384 и 256 карт признаков. Три слоя max-pooling следуют за первым, третьим и пятым сверточными слоями. Таким образом в двух следующих полносвязных слоях имеется 4096 нейронов.

В 2012 году AlexNet значительно превзошла всех своих конкурентов и выиграла состязание, снизив ошибку на Top-5 предсказаниях с 26% до 15,3% (второе место составило около 26,2%).

## VGG-16

VGG-16 [21] имеет входные параметры 224 × 224 × 3, содержит 13 сверток с max-poling слоями на конце и 3 полносвязных слоя. Используется много фильтров 3x3 вместо малого количества больших, что позволяет выделять более низкоуровневые признаки. Количество фильтров увеличивается с уменьшением количества выходов, таким образом на каждом слое достигается приблизительно одинаковое количество параметров.

## GoogLeNet

Победителем конкурса ILSVRC 2014 был GoogLeNet [24](или Inception V1) от Google. Он сократил процент ошибки на Top-5 до 6,67%. Этот результат сравнивали с умением человека решать ту же задачу – точность человека организаторы оценивали дополнительно. Как оказалось, это было довольно непросто, и потребовалось некоторое обучение для того, чтобы превзойти точность GoogLeNet. После нескольких дней дополнительного обучения, эксперт Андрей Карпаты смог достичь уровня ошибок 5,1% на Top-5. Данная топлогия использует CNN основанную на LeNet, но привнесла новый элемент – Inception модуль (отсюда и второе название). Этот модуль основан на нескольких очень маленьких свертках, чтобы резко уменьшить количество параметров. Их архитектура состояла из 22 слоев, но уменьшила количество параметров с 60 миллионов (AlexNet) до 4 миллионов.

## Resnet 50

ResNet [25] имеет входные размеры 224 × 224 × 3. Его архитектура состоит из residual блоков, где каждый блок использует «соединение быстрого доступа» (shortcut). Это соединение может быть простым идентификационным соединением (id-block) или соединением со сверточным слоем (convblock).

В основной части блока используется 3 сверточных слоя с различным количеством фильтров для каждого блока. Первая и третья свертки этой

группы используют размеры фильтров 1 x 1, а вторая – как правило 3 x 3. В конце блока значения функций shortcut и основной части складывается. За каждым сверточным слоем следует слой батч-нормализации.

ResNet-50 начинается с сверточного слоя с размером фильтра 7x7, генерируя 64 фильтра, за которым следует слой батч-нормализации, слой активации и max-pooling слой. Затем идёт 4 группы полносвязных блоков, каждый из которых начинается с блока со сверткой. Каждая группа содержит соответственно 1, 3, 5 и 2 id-блока. ResNet50 содержит в общей сложности 53 сверточных слоя, отсюда и название.

## GoogLeNet v3

[26] Использует ту же идею, что и оригинальный GoogLeNet, но вместо сверток 5x5 и 3X3 используются комбинации сверток меньшего размера для уменьшения количества весов при сохранении того же уровня точности. Благодаря этому, модель быстрее обучается и предуобученные веса получены после обучения на большем количестве эпох.

Сравнительная таблица:

| Топология | Точность, % |
|---|---|
| **Alexnet** | 37,3 |
| **VGG-16** | 93,6 |
| **GoogLeNet** | 64,6 |
| **ResNet-50** | 62,1 |
| **GoogLeNet-V3** | 95,3 |

Таблица 6. Сравнение методов с использованием известных архитектур

## Часть 3. Визуализация данных методами понижения размерности

Для правильной визуализации нам необходимо разместить точки из пространства высокой размерности в пространстве низкой размерности так, чтобы близкие объекты в исходном пространстве были близки в новом, и наоборот, далекие объекты в исходном пространстве были так же далеки в новом.

### Визуализация методом t-SNE

Метод визуализации t-SNE описан в [27]. Чтобы измерить похожесть объектов, введем условную вероятность присутствия в определенном месте одной точки при условии присутствия другой:

$$p_{j|i} = \frac{exp(-\| x_i - x_j \|^2 / 2\sigma_i^2)}{\sum_k \sum_{l \neq k} exp(-\| x_k - x_l \|^2 / 2\sigma_i^2)}$$

, где $\sigma_i^2$ - дисперсия нормального распределения, которое моделирует близость объекта j к объекту i.

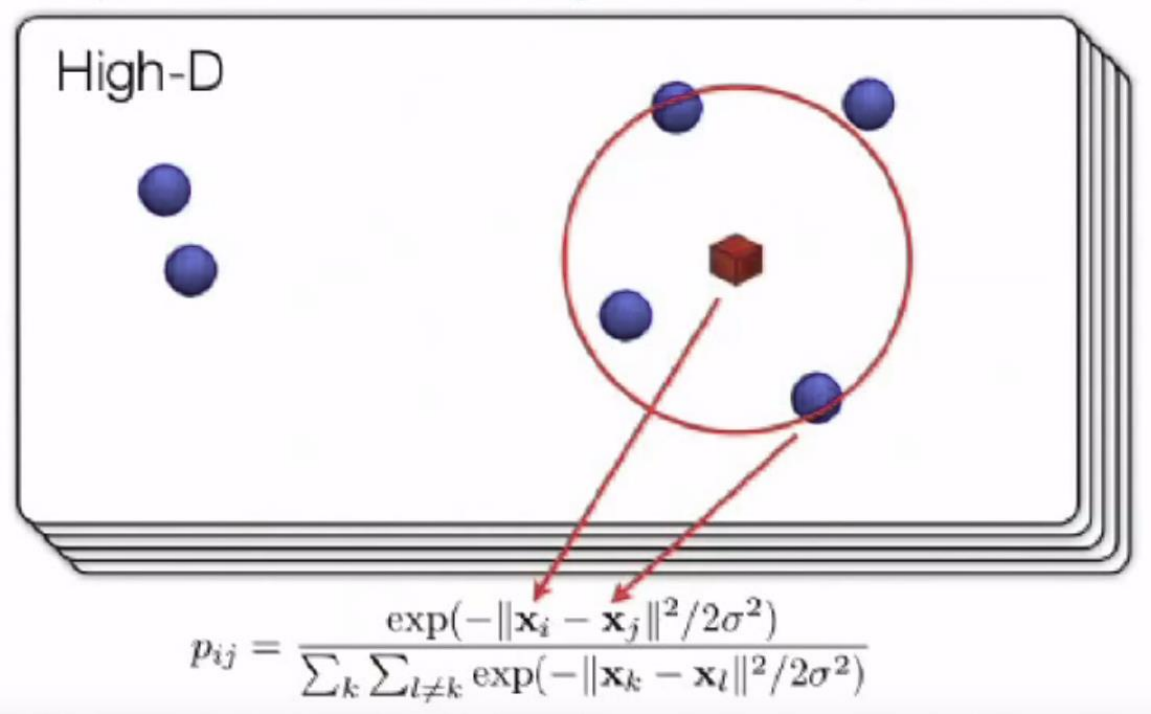


Рисунок 6. Визуальное представление идеи t-SNE.

То же самое сделаем для точек из пространства малой размерности (предполагаем, что $\sigma$ постоянна и равна $\sigma$=1/4 в новом пространстве):

$$q_{j|i} = \frac{exp(-\| y_i - y_j \|^2)}{\sum_{k \neq i} exp(-\| y_i - y_k \|^2)}$$

Это выражение взято из оригинальной версии SNE алгоритма. В алгоритме t-SNE вместо нормального распределения используется t-распределение Стьюдента со степенью свободы 1.

$$q_{j|i} = \frac{(1 + \| y_i - y_j \|^2)^{-1}}{\sum_{k \neq i} (1 + \| y_i - y_k \|^2)^{-1}}$$

Таким образом, мы получили две функции измерения похожести объектов в двух различных пространствах. Т.е., если наша модель идеально представляет

точки в новом пространстве, то p будет равен q. $y_i$ - это те точки, которые мы собираемся сгенерировать.

Введем расстояние Кульбака-Лейблера для измерения расстояния между $P_i$ и $Q_i$, где $P_i$ - условное распределение вероятности всех точек $x_j$ кроме j!=i, а $Q_i$, - условное распределение вероятности всех точек $y_j$ кроме j!=i:

$$C = \sum_i KL(P_i \parallel Q_i) = \sum_i \sum_j p_{j|i} log \frac{p_{j|i}}{q_{j|i}}$$

- функция потерь, которую нам необходимо минимизировать. Для нахождения оптимального решения мы используется метод градиентного спуска. После схождения метода, полученные точки можно отрисовать в двумерном пространстве.

Результат работы алгоритма t-SNE на признаках, полученных с помощью CNN GoogleNet V3 представлен на Рис. 7. Полученная визуализация позволяет увидеть как очевидные зависимости, например, значительное отличие иконописи (*Byzantin_Iconography*) от остальных стилей, так и неочевидные, такие как сильное различие стилей Северного, Раннего и Высокого Реннесанса.

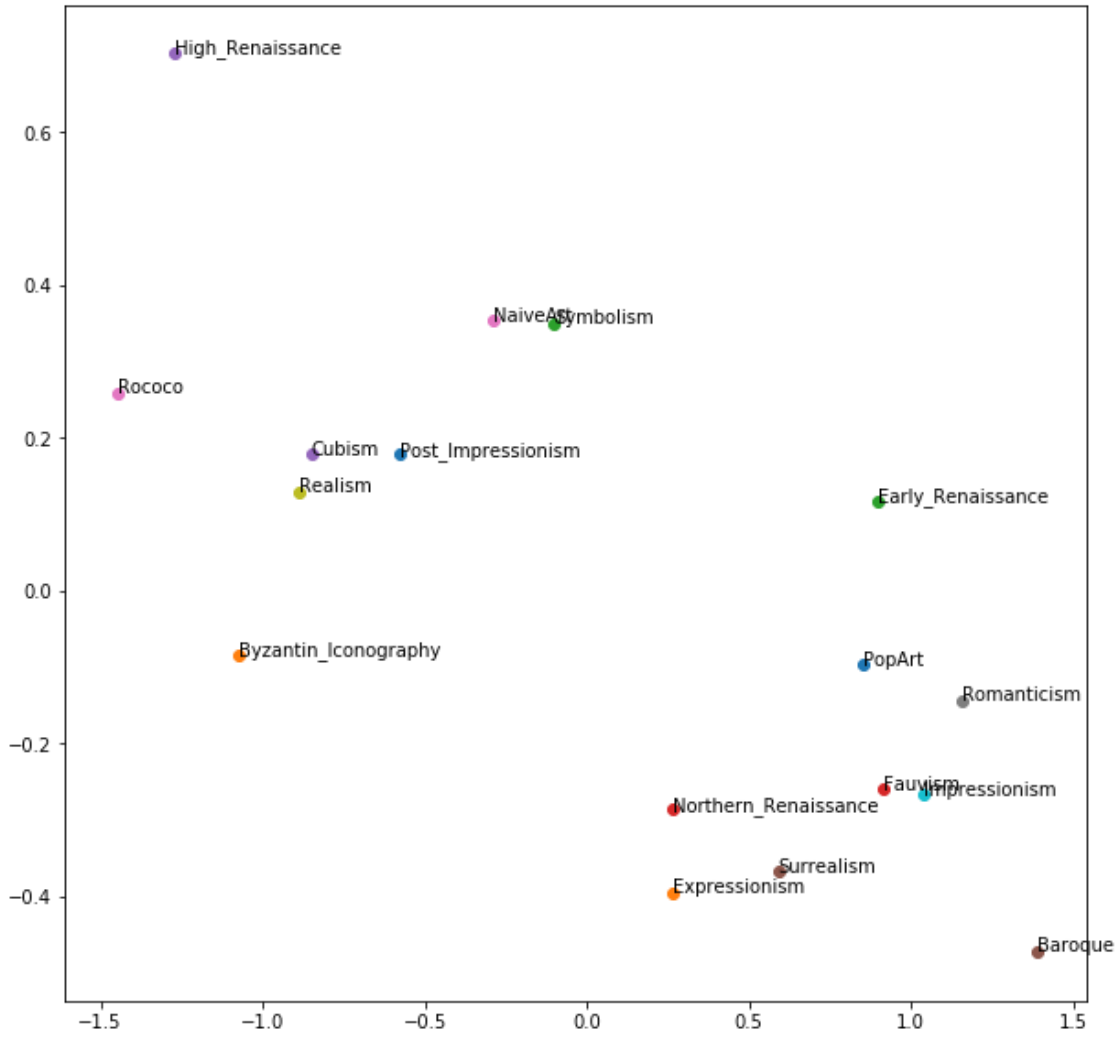


Рисунок 7. Центроиды классов в пространстве пониженной размерности, полученные методом t-SNE.

## Визуализация методом PCA (метод главных компонент)

Метод главных компонент [28] предлагает понижать размерность многомерного пространства, проектируя его точки на подпространство с таким ортонормированным базисом, что ковариация между разными координатами равна нулю. Нахождение такого подпространства выполняется при помощи вычисления матрицы ковариации, а затем её диагонализации при помощи SVD-разложения.

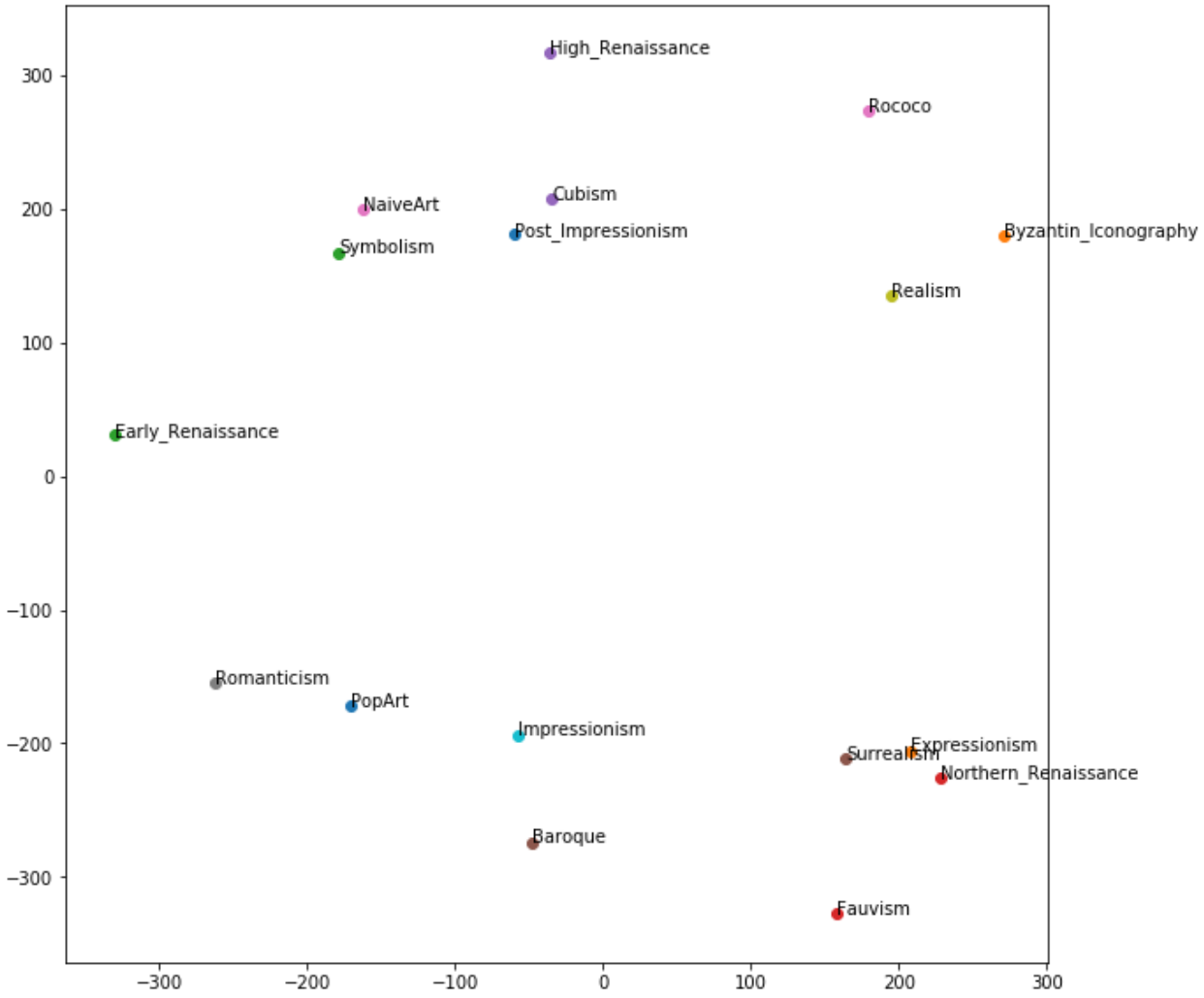


Рисунок 8. Центроиды классов в пространстве пониженной размерности, полученные методом главных компонент.

Полученная визуализация в целом отражает меньше зависимостей, чем визуализация, полученная с помощью t-SNE, однако некоторые выявленные зависимости выделяются более ярко, такие как близость Экспрессионизма, Сюрреализма и Северного Ренессанса, и удалённость Раннего Ренессанса от остальных классов.

# Заключение

Предложенные методы были протестированы на датасете Pandora и включали в себя построение классификатора поверх признаков, полученных классическими методами компьютерного зрения, а также поверх признаков, полученных с внутренних слоёв state-of-the-art нейронных сетей. Среди всех предложенных методов наилучшим образом себя показал классификатор на основе логистической регрессии в сочетании с внутренними признаками сети GoogLeNet v3. Также была произведена визуализация классов данных в контексте результата работы сети.

В целом, можно отметить, что предложенный классификатор, использующий признаки классического компьютерного зрения, конкурентоспособен на фоне других классификаторов, использующих аналогичные признаки. Использование предварительно натренированной GoogLeNet и логистической регрессии дало лучший результат среди опубликованных работ, а предложенная визуализация предоставила новые факты о данных.

# Список литературы

[1] The Met Makes Its Images of Public-Domain Artworks Freely Available through New Open Access Policy. [Электронный ресурс; 7-Февраля-2017]. https://www.metmuseum.org/press/news/2017/open-access

[2] Digital painting database for art movement recognition [Электронный ресурс; 21-апреля-2016]. http://imag.pub.ro/pandora/pandora_download.html

[3] Corneliu Florea, Razvan Condorovici, Constantin Vertan, Raluca Butnaru, Laura Florea, and Ruxandra Vranceanu. Pandora: Description of a painting database for art movement recognition with baselines and perspectives. In Proceedings of the European Signal Processing Conference (EUSIPCO), 2016.

[4] K. Martinez, J. Cupitt, D. Saunders, and R. Pillay. Ten years of art imaging research. Proceedings of the IEEE, 90(1):28–41, 2002.

[5] D. Stork. Computer vision and computer graphics analysis of paintings and drawings: An introduction to the literature. In Proc. of CAIP, pages 9–24, 2009.

[6] B. Gunsel, S. Sariel, and O. Icoglu. Content-based access to art paintings. In Proc. Of ICIP, pages 558–561, 2005.

[7] J. Zujovic, L. Gandy, S. Friedman, B. Pardo, and T.N. Pappas. Classifying paintings by artistic genre: An analysis of features & classifiers. In Proc. of IEEE MMSP, pages 1–5, 2009.

[8] B. Siddiquie, S.N. Vitaladevuni, and L.S. Davis. Combining multiple kernels for efficient image classification. In Proc. of WACV, pages 1–8, 2009.

[9] Lior Shamir, Tomasz Macura, Nikita Orlov, D. Mark Eckley, and Ilya G. Goldberg. Impressionism, expressionism, surrealism: Automated recognition of painters and schools of art. ACM Trans Appl Percept, 7(2):1–17, 2010.

[10] R. S. Arora, and Ahmed Elgammal. Towards automated classification of fine–art painting style: a comparative study. In Proc. of ICPR, pages 3541–3544, 2012.

[11] Fahad Shahbaz Khan, Shida Beigpour, Joost van de Weijer, and Michael Felsberg. Painting-91: a large scale database for computational painting categorization. Mach. Vis. App., 25(6):1385–1397, 2014.

[12] Razvan George Condorovici, Corneliu Florea, and Constantin Vertan. Automatically classifying paintings with perceptual inspired descriptors. J. Vis. Commun. Image. Represent., 26:222 – 230, 2015.

[13] S. Agarwal, H. Karnick, N. Pant, and U. Patel. Genre and style based painting classification. In Proc. of WACV, pages 588–594, 2015.

[14] Joblove G. H., Greenberg D. Color spaces for computer graphics //ACM siggraph computer graphics. – ACM, 1978. – Т. 12. – №. 3. – С. 20-25.

[15] Ho T. K. Random decision forests //Document Analysis and Recognition, 1995., Proceedings of the Third International Conference on. – IEEE, 1995. – Т. 1. – С. 278-282.

[16] Kingma D., Ba J. Adam: A method for stochastic optimization //arXiv preprint arXiv:1412.6980. – 2014.

[17] Florea C., Florea L., Vertan C. Learning pain from emotion: transferred hot data representation for pain intensity estimation //European Conference on Computer Vision. – Springer International Publishing, 2014. – С. 778-790.

[18] Sergey Karayev, Matthew Trentacoste, Helen Han, Aseem Agarwala, Trevor Darrell, Aaron Hertzmann, and Holger Winnemoeller. Recognizing image style. In Proceedings of the British Machine Vision Conference. BMVA Press, 2014.

[19] Gatys L. A., Ecker A. S., Bethge M. A neural algorithm of artistic style //arXiv preprint arXiv:1508.06576. – 2015.

[20] Tan W. R. et al. Ceci n'est pas une pipe: A deep convolutional network for fine-art paintings classification //Image Processing (ICIP), 2016 IEEE International Conference on. – IEEE, 2016. – С. 3703-3707.

[21] Simonyan K., Zisserman A. Very deep convolutional networks for large-scale image recognition //arXiv preprint arXiv:1409.1556. – 2014.

[22] Krizhevsky A., Sutskever I., Hinton G. E. Imagenet classification with deep convolutional neural networks //Advances in neural information processing systems. – 2012. – С. 1097-1105.

[23] Van Den Oord A. et al. Wavenet: A generative model for raw audio //arXiv preprint arXiv:1609.03499. – 2016.

[24] Szegedy C. et al. Going deeper with convolutions. – Cvpr, 2015.

[25] He K. et al. Deep residual learning for image recognition //Proceedings of the IEEE conference on computer vision and pattern recognition. – 2016. – С. 770-778.

[26] Szegedy C. et al. Rethinking the inception architecture for computer vision //Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. – 2016. – С. 2818-2826.

[27] Maaten L., Hinton G. Visualizing data using t-SNE //Journal of machine learning research. – 2008. – Т. 9. – №. Nov. – С. 2579-2605.

[28] Wold S., Esbensen K., Geladi P. Principal component analysis //Chemometrics and intelligent laboratory systems. – 1987. – Т. 2. – №. 1-3. – С. 37-52.