

УДК 004.8

DOI 10.30837/bi.2021.1(96).15

И.И. Евтушенко<sup>1</sup>, А.В. Вечур<sup>2</sup><sup>1</sup>ХНУРЭ, г. Харьков, Украина, ivan.yevtushenko@nure.ua,  
ORCID iD: 0000-0001-6085-4874<sup>2</sup>ХНУРЭ, г. Харьков, Украина, alexander.vechur@nure.ua,  
ORCID iD: 0000-0001-9605-1475

## РАСПОЗНАВАНИЕ ХУДОЖНИКА ПО КАРТИНЕ

Разработан алгоритм определения автора картины с изображением. Задача является сложной, так как сфера искусства традиционно плохо поддается формализации. К тому же, при обработке нужно минимально видоизменять картины, иначе информация о стиле может быть потеряна. Алгоритм основан на современных подходах, а именно на сверточной нейронной сети с архитектурой RegNet. Эта архитектура известна не только своей мощностью, но и интересным механизмом проверки гипотез, который обсуждается в работе. Для обучения используется набор данных «Best Artworks of All Time». В рамках экспериментов проводится обучение с использованием нескольких функций потерь случайных изменений данных, предварительного обучения без учителя, угасающей скорости обучения.

АРХИТЕКТУРА, ГЛУБОКОЕ ОБУЧЕНИЕ, СВЕРТОВАЯ НЕЙРОННАЯ СЕТЬ, ЭПОХА, КАРТИНА, КЛАССИФИКАЦИЯ, НАБОР ДАННЫХ, ОБУЧЕНИЕ, ПЕРЕОБУЧЕНИЕ, ПРОСТРАНСТВО, СТИЛЬ, ФУНКЦИЯ ПОТЕРЬ, ХУДОЖНИК

**Євтушенко І.І., Вечур О.В. Розпізнавання художника за картиною.** Розроблено алгоритм визначення автора картини із зображення. Завдання є складним, так як сфера мистецтва традиційно погано піддається формалізації. До того ж, при обробці потрібно мінімально видозмінювати картини, інакше інформація про стилі може бути втрачена. Алгоритм заснований на сучасних підходах, а саме на згортковій нейронній мережі з архітектурою RegNet. Ця архітектура відома не тільки своєю потужністю, але і цікавим механізмом перевірки гіпотез, який обговорюється в роботі. Для навчання використовується набір даних «Best Artworks of All Time». У рамках експериментів проводиться навчання з використанням кількох функцій втрат випадкових змін даних, попереднього навчання без вчителя, згасаючої швидкості навчання.

АРХІТЕКТУРА, ГЛИБОКЕ НАВЧАННЯ, ЗГОРТКОВА НЕЙРОННА МЕРЕЖА, ЕПОХА, КАРТИНА, КЛАСИФІКАЦІЯ, НАБІР ДАНИХ, НАВЧАННЯ, ПЕРЕНАВЧАННЯ, ПРОСТІР, СТИЛЬ, ФУНКЦІЯ ВТРАТ, ХУДОЖНИК

**Yevtushenko I.I., Vechur O.V. Artist recognition by painting.** An algorithm for determining the author of a painting with an image has been developed. The task is difficult, since the field of art has traditionally been difficult to formalize. In addition, during processing, you need to modify the paintings to a minimum, otherwise information about the style may be lost. An algorithm is based on modern approaches, namely a convolutional neural network with the RegNet architecture. This architecture is known not only for its power, but also for its interesting hypothesis testing mechanism, which is discussed in the work. The data set «Best Artworks of All Time» is used for training. Within the framework of the experiments, training is carried out using several functions of loss of random data changes, preliminary training without a teacher, and a dying learning rate.

ARCHITECTURE, DEEP LEARNING, CONVOLUTIONAL NEURAL NETWORK, ERA, PICTURE, CLASSIFICATION, DATA SET, LEARNING, RETAINING, SPACE, STYLE, LOSS FUNCTION, ARTIST

### Вступление

Сегодня информационные технологии играют огромную роль в мире не только потому что они хорошо изучены, но и потому что они все чаще и чаще автоматизируют рутинную работу людей на предприятиях. Эффективность бизнеса повышается, ведь компьютеры способны работать быстро, точно, дешево и без перерыва, в то время как человек имеет возможность сконцентрироваться на других.

Но с некоторых пор компьютеры умеют решать даже сложные задачи, которые, как казалось ранее, способен решать только мозг человека. Такие случаи принято называть задачами искусственного интеллекта, и их список очень велик: это может быть чтение рукописного текста, управление транспортом, выявление климатических аномалий, общение

с людьми и тому подобное. Они решаются большим количеством математических методов еще с прошлого века, но за последние десять лет особый успех и популярность получили алгоритмы, способные со временем улучшать качество своей работы, анализируя статистику данных, с которыми они работают — алгоритмы машинного обучения. За это время они с научных исследований переросли в реально работающих программные системы, и сегодня почти каждое известное приложение или компания определенным образом использует их для улучшения качества работы.

Особым прорывом в сфере машинного обучения стали нейронные сети, ведь хоть они и были придуманы в середине XX века, только недавно стало понятно, насколько хорошо они способны анализировать

тексты, изображения, звуки и тому подобное. Это дало возможность попробовать использовать их даже в, наверное, наиболее гуманном направлении - искусстве. Нейронные сети обучали генерировать стихи, писать музыку, рисовать картины. Конечно, как и в жизни, большинство их произведений не была оценено, но некоторые из них стали известными в обществе. Безусловно, такие работы имеют, в первую очередь, академический и развлекательный интерес, но иногда они могут позволить более глубоко понять процесс создания шедевров искусства, и на основе этого делать новые интересные выводы, которые не очевидны искусствоведам при визуальном анализе.

В данной работе речь пойдёт о задаче определения автора картины с помощью модели машинного обучения.

### 1. Связанные работы

С точки зрения анализа данных это является случаем классификации, когда имеется некоторое конечное множество классов (художников), и для наблюдений (картин) прогнозируется, к какому из классов и с какой вероятностью оно относится.

Уже на протяжении нескольких лет при решении задач классификации изображений наилучшие результаты демонстрируют нейронные сети. Их архитектура практически ничем не ограничена, за счёт чего вариаций таких алгоритмов существует огромное количество. С одной стороны, это дает возможность добиваться впечатляющих результатов на разных задачах, но с другой стороны, поиски той самой оптимальной архитектуры отнимают большое количество интеллектуальных, вычислительных, материальных и временных ресурсов. Команда из Facebook AI Research весной 2020-ого года попробовала решить эту проблему в своей работе "Designing Network Design Spaces" [1], попытавшись выяснить "рецепт" успешной архитектуры. Наиболее важными в их работе можно считать даже не сами правила проектирования, которые авторам удалось выработать, а подход, которым им удалось их получить, так как он

хорошо обобщается на методы проведения других научных исследований. В итоге для сообщества это стало не статьёй с очередной архитектурой, а целым пособием по проведению быстрых и эффективных экспериментов. Поэтому в научном сообществе эта работа ценится по сей день.

Изначально авторы берут некоторую простую базовую архитектуру, похожую на известный ResNet [2], которая имеет большое количество параметров, таких как глубина, ширина, количество блоков и так далее, каким-то образом влияющие на ее качество, и назвали её AnyNet. Они трактуют множество этих параметров как некоторое пространство, из которого мы случайным образом выбираем архитектуру для тренировки. Задачей работы является сузить это пространство параметров до такого, в котором вероятность получить хорошую архитектуру будет максимальной.

Для того, чтобы сравнивать два пространства, был принят следующий критерий: из каждого выбирается по 500 случайных архитектур, после чего каждая из них обучается в течении 10 эпох на решении задачи ImageNet [3]; затем для обоих пространств составляется функция распределения успешных архитектур, скорость роста которой демонстрирует качество пространства; на основе сравнений этих функций принимается статистически значимое решение о том, лучше ли новое пространство. На рис. 1 слева продемонстрирован процесс последовательных переходов к более узким пространствам, а справа их функции распределения.

Таким образом, большое количество выбранных архитектур позволяет быть уверенным, что улучшение не есть случайным, при этом их запуски можно делать параллельно, а тот факт, что они короткие и быстрые, не мешает им обобщаться на полномасштабные тренировки. Это подтверждается тем, что таким методом авторам удалось сузить пространство архитектур до достаточно хорошего для того, чтобы затем обучить полноценную архитектуру, названную RegNet, и получить лучшие результаты для задачи

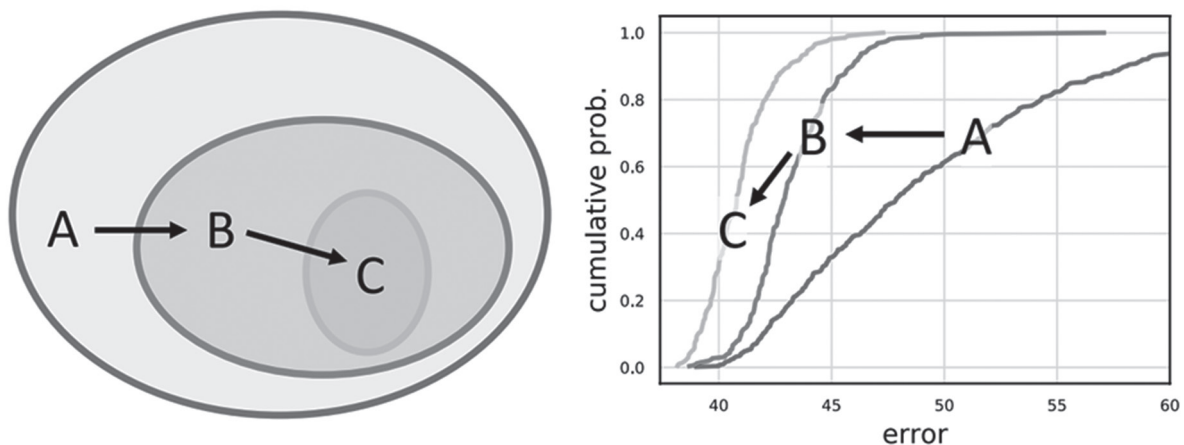


Рис. 1. Процесс перехода между пространствами архитектур

ImageNet на момент публикации статьи. Вдобавок к этому были сформулированы и интерпретированы правила, которые позволили прийти к успеху. Через короткое время вышли новые архитектуры, которым удалось улучшить этот результат, но подобный принцип проведения экспериментов стал повсеместно использоваться исследователями.

Важным преимуществом нейронных сетей является тот факт, что в ходе тренировки они обучаются формировать полезные сжатые представления наблюдений. Сжатие происходит в архитектурах неизбежно с целью экономии вычислительных ресурсов, а полезная информация собирается для решения финальной задачи. Помимо того, что именно возможность выучивать такие представления делает данный вид алгоритмов столь выразительным, часто всё эти же представления могут быть использованы для решения каких-то других задач. Однако, на практике сеть не всегда обучается формировать хорошие представления, так как она может иметь не совсем удачную функцию потерь, страдать проблемой переобучения, несбалансированности классов и тому подобное. Одним из самых эффективных методов навязывания нейронной сети полезных представлений является использование многозадачного обучения, хорошо описанного в статье “An Overview of Multi-Task Learning in Deep Neural Networks” [4]. Суть подхода заключается в том, что, как ни странно, если мы обучаем алгоритм решать сразу несколько разных задач, а не одну, то в итоге он способен обучиться решать каждую из них более эффективно. Происходит это главным образом по той причине, что как раз в ходе решения большого количества задач сеть вынуждена сохранить как можно больше полезной информации в своём внутреннем представлении данных, и может оказаться, что как раз этой информации не хватало для успеха в отдельной задаче.

Отдельно стоит отметить, что задача классификации изображений является методом обучения с учителем, в которых для тренировки алгоритма нужны не только наблюдения, но и правильно указанные классы для них. Плюс этого в том, что такие методы очень хорошо изучены и широко используются, но не для каждой задачи есть возможность найти достаточно большие объемы данных. Эта проблема актуальна и в нашем исследовании, так как не у всех художников физически есть много картин, доживших до сегодняшних дней хоть в каком-то виде. Поэтому всё больше исследований проводится в области самообучения, когда задача обучения подбирается таким образом, что наблюдения сами для себя становятся размеченными данными. Одним из наиболее успешных примеров таких работ является “Unsupervised Representation Learning by Predicting Feature Decoupling” [5], в рамках которой входное

изображений поворачивается на определенный угол, а алгоритм должен спрогнозировать угол наклона. Это заставляет нейронную сеть выучить хорошие представления наблюдений, и при этом им совсем не нужна разметка. После предварительного обучения сети таким подходом можно на уже небольшом наборе размеченных данных продолжить обучать её под финальную задачу. Это может помочь получить более качественные результаты за счет того, что в ходе предварительного обучения алгоритм смог сформировать полезные представления для значительно большего объема данных, и всё, что ему остаётся, это адаптировать их для классификации художников.

## 2. План работы

В рамках данной работы предлагается обучить собственную нейронную сеть для решения задачи распознавания художника по картине. Её архитектура унаследует RegNet, предварительно обученный для решения задачи ImageNet. Но сам процесс обучения будет заметно отличаться и потребует большого количества экспериментов, так как картины художников требуют особого вида обработки.

В частности, планируется проверить гипотезы по оптимальному разрешению входного изображения (112x112/224x224/448x448), попробовать различные методы оптимизации (SGD, Adam [6], LAMB [7]), различные методы изменения скорости обучения (константный, затухающий, косинусный [8]), методы регуляризации для борьбы с переобучением (Dropout [9], Регуляризация Тихонова, Модификация данных). Также для борьбы с несбалансированностью классов стоит попробовать использовать в качестве функции потерь не только стандартную для классификации кросс-энтропию, но и взвешенную, где веса классов обратно пропорциональны их частоте, и DiceLoss [10], где большой вес имеют сложные для модели примеры.

Важными экспериментами будут попытки использовать многозадачное обучение и предварительное самообучение для получения лучших представлений и, соответственно, качества. Гипотезы предлагается проверять последовательно, а для их проверки обучать каждую модель в течение 20 эпох и сравнивать их качество по метрике `macro f1-score`. В отличие от точности, она хорошо учитывает несбалансированность классов, поэтому если модель будет совсем плохо работать для какого-то художника, то это не останется незамеченным. При этом точность также будет озвучиваться для лучшей интерпретации экспериментов.

В работе используется язык Python, а наиболее важной библиотекой — PyTorch [11], что является одним из наиболее популярных решений сегодня. Обучение алгоритмов будет проходить на компьютере с GPU для ускорения вычислений.

### 3. Данные

Преимуществом поставленной задачи является тот факт, что данные, необходимые для тренировки модели распознавания — это картины и их авторы. Эти данные можно найти в интернете в открытом доступе, для их сбора не придется проводить или генерировать наблюдения, что является частой проблемой анализа данных в других отраслях, например, в медицине.

С другой стороны, есть проблема, связанная с тем, что для качественной работы нейронных сетей тренировочных данных нужно много. При этом мы никак не сможем собрать больше картин художника, чем дожило до сегодняшних дней, поэтому данный вопрос решается не на уровне сбора данных, а на уровне проектирования и обучения модели.

Одним из популярных способов сбора данных из интернета выглядит следующим образом: находится веб-источник, на котором есть множество необходимых нам данных, например, интернет-галерея, после чего создаётся скрипт, который рекурсивно посещает разные страницы веб-сайта и скачивает изображения из него, попутно вылавливая имя указанного там же автора. Естественно, такие данные требуют последующего осмотра, так как в работе программы могли быть допущены ошибки, но в ситуациях, когда другого выхода нет, этот метод очень полезен.

К счастью, сообщество исследователей в области анализа данных очень активно делится множеством полезных ресурсов, от чего выигрывают все. Одной из самых известных площадок для их коммуникации является Kaggle [12]. Главной целью этого сайта всегда было и остаётся проведение онлайн-соревнований по машинному обучению, но со временем здесь также появились форум, курсы и возможность публиковать свои наборы данных. По запросу удалось найти набор данных “Best Artworks of All Time” [13]. В нём

содержится 8446 картин 50 разных художников. На рис. 2 видна проблема несбалансированности классов, так как несмотря на то, что большая часть художников имеет от 50 до 200 картин, у Джексона Поллока их всего 24, что почти в 40 раз меньше, чем у Ван Гога, имеющего 877 картин в этом наборе. Это может вызвать проблемой, так как алгоритм выучит одних лучше других.

Преимуществом этого набора данных является тот факт, что помимо разметки авторов для картин здесь присутствует дополнительная информация об авторах: их национальность, годы жизни и жанр. То есть мы можем для картины помимо самого автора прогнозировать эти переменные, реализуя концепцию многозадачного обучения. Благодаря этому у алгоритма будет возможность понимать, что хоть картины Клода Моне, Эдуарда Мане и Микеланджело относятся к разным авторам, первые два имеют что-то общее — национальность, жанр и приблизительные годы жизни, а третий отличается от них. Ровно как знание этого делает искусствоведа более профессиональным, так и нейронная сеть от него становится более эффективной.

Итого, распределения в наборе данных соответствуют реальности, данных в нём достаточно, и у них хорошая разметка, так что как основной источник информации он подходит для исследования. Но дополнительно нам хотелось бы иметь ещё больший набор неразмеченных данных для проверки эффективности методов предварительного самообучения. Эти данные не обязательно должны быть из того же распределения изображений, но при этом максимально приближены к предметной области. Всё на том же Kaggle удалось найти данные “Wiki-Art: Visual Art Encyclopedia” [14], в котором содержится 96000 изображений. Для них, конечно, не известны авторы, ведь не все из них являются живописью, но при

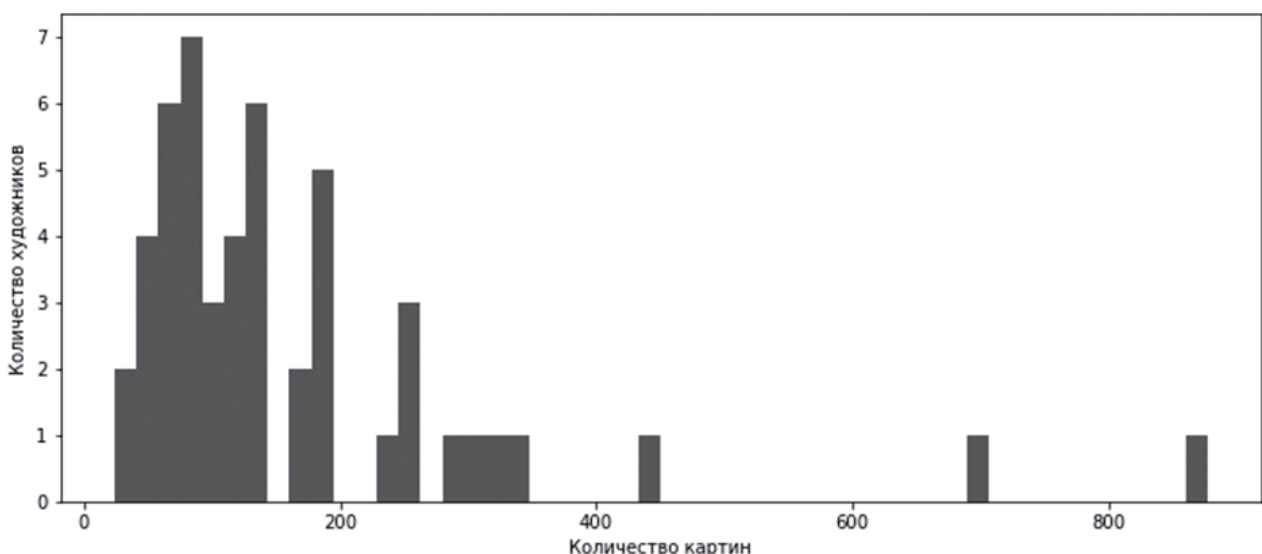


Рис. 2. Распределение количества картин художников



этом они разбиты на 14 категорий — ландшафты, портреты, животные, морские пейзажи и так далее. Эти метки также интересно попробовать использовать при предварительном самообучении в рамках концепции многозадачного обучения.

#### 4. Эксперименты

Очевидно, что в моделях машинного обучения главным являются данные и то, какие преобразования мы делаем над ними перед тем, как подать на вход алгоритму. Часто полезно делать эти преобразования случайными для того, чтобы повысить разнообразие данных. Среди таких преобразований мы предлагаем делать поворот картины по вертикальной оси, так как очень маловероятно, что это мешает узнать художника, и при этом повышает разнообразие изображений в 2 раза. Также все картины должны быть одного размера, чтобы нейронная сеть, во-первых, могла адаптироваться к конкретным масштабам шаблонов в картинах, а во-вторых, мы имели возможность обучить её не по одной картинке в один момент времени, а пакетами, которые эффективно складываются в одну многомерную матрицу. Для этого предлагается сначала масштабировать все картины таким образом, чтобы по своей меньшей стороне они имели размер 256 пикселей, но при этом они сохраняли свое исходное соотношение сторон. Затем из полученной картины вырезается случайный квадрат 224x224 (стандартный размер изображений для классификационных архитектур), и эта случайность также повышает разнообразие наших данных. Естественно, такие преобразования нужны только во время обучения, в режиме тестирования достаточно отмасштабировать картину по тому же принципу и вырезать квадрат 224x224 из ее центра.

Как уже упоминалось ранее, архитектурой нейронной сети в ходе экспериментов являлась RegNet. Для неё есть предварительно обученные на данных ImageNet веса, начинать тренировку с которых часто является более быстрым решением, чем со случайной инициализации. В рамках первого эксперимента в качестве алгоритма оптимизации мы использовали Adam с параметрами по умолчанию, часто это дает хорошие результаты благодаря адаптивности этого алгоритма под ход обучения. Функция потерь для классификации стандартная — кросс-энтропия. Длительность обучения — порядка 15 минут на 20 эпох. Как было упомянуто в разделе данных, метрикой для проверки гипотез является `task0 f1-score`.

В итоге такая первичная конфигурация тренировки сразу получает достойные результаты — `f1-score` равен 0.655, точность — 72.5%. Стоит отметить, что модель достаточно сильно переобучена, так как на тренировочной выборке ее точность превышает 99%.

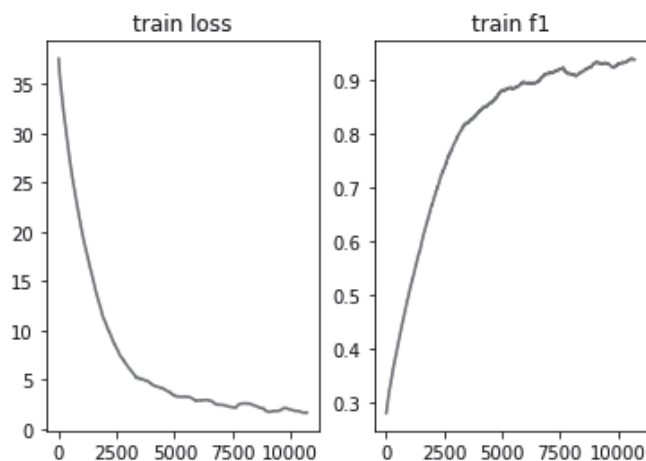


Рис. 3. Визуализация процесса обучения: пошаговые изменения значения функции потерь (слева) и метрики `f1-score` (справа)

Для борьбы с этим эффектом хорошей идеей часто является усилить случайные преобразования в данных, чтобы повысить их разнообразие и усложнить алгоритму задачу. В качестве таких преобразований были попытки использовать добавление нормального шума, перевороты по горизонтали, преобразование цвета, перемасштабирования и так далее. Оказалось, что они не дают прирост, а даже ухудшают метрики на несколько процентов. Скорее всего, это связано с тем, что в картинах очень важны мелкие детали, которые могут быть испорчены подобными преобразованиями, и в итоге художник идентифицируется ещё хуже. Была попытка увеличить разрешение входного изображения с 224 до 448 для того, чтобы терять меньше деталей, но в итоге это привело только к замедлению тренировки. При разрешении 112 наблюдается сильное падение качества. В результате изначальная конфигурация преобразований данных оказалась самой эффективной.

Также причиной переобучения может быть тот факт, что у некоторых художников настолько мало картин, что нейронная сеть просто их запоминает, не обобщаясь на те, которые она не увидела. Попыткой устранить эту проблему было использование больших весов для таких классов, чем для популярных, либо же использование `DiceLoss` вместо кросс-энтропии, которая сильнее штрафует примеры, на которых алгоритм ошибается. Оба эти метода вызвали нестабильность в процессе обучения, из-за чего при их использовании значение метрики только ухудшалось. Такой же неудачный эффект приносили попытки использовать `Dropout` и регуляризацию Тихонова.

В рамках экспериментов с процессом оптимизации выяснилось, что менять алгоритм Adam смысла нет, так как SGD работает быстро, но сходится гораздо хуже, а LAMB работает дольше, но при этом финальные результаты дает приблизительно те же. А вот усложнение способа изменения скорости обучения

оказалось эффективным — уменьшая его в 0.7 раз каждую эпоху удалось добиться f1-score 0.695, а точности — 75.5%.

Следующий шаг для улучшения — многозадачное обучение. Мы воспользовались доступными метками о жанре, национальности и годах жизни. Жанры принимают 31 различных значений, национальность — 17, предсказания обоих этих переменных являются решением все той же задачи классификации, функцией потерь для которой традиционно взята кросс-энтропия. Для лет жизни сначала было вычислено среднее между годом рождения и годом смерти, чтобы прийти к одному численному показателю, затем они были масштабированы в промежуток от 0 до 1, и наконец нейронной сети была поставлена задача регрессии, решаемая путем минимизации средне-квадратичного отклонения между предсказаниями и метками. В результате алгоритм научился предсказывать жанр с точностью 76.5%, национальность с точностью 80%, год со средней ошибкой в 100 лет, и, что самое интересное, стал ещё лучше решать задачу идентификации художника. F1-score вырос до 0.722, а точность стала 77.75%. Примечательно, что при таком подходе алгоритм получает хорошее качество на более ранних эпохах. Скорее всего, это связано с тем, что модель получает больше градиентов с разных задач, и каждый шаг обучения становится более информативным.

Финальный эксперимент — предварительное самообучение. Для него было использовано больше преобразований над данными, чем в задачи распознавания автора картины, так как в данном случае они менее критично могут повлиять на целевую метку, но архитектура и параметры тренировки сохранились теми же. Алгоритм за 8 эпох, которые из-за большого количества данных продлились 6 часов, был обучен распознавать один из четырёх возможных углов поворота с точностью 67%, а категорию — с точностью 70%. При этом последующее обучение этой модели для решения финальной задачи привело к f1-score 0.65, а точности 71.5%. Это неплохие результаты, но они заметно хуже, чем предыдущие. На самом деле, это связано с тем, что в предыдущих экспериментах мы также неявно использовали подход предварительного обучения, когда взяли параметры модели, которые были тщательно сформированы для качественного решения задачи ImageNet. В то время как в данном эксперименте мы самостоятельно обучили модель на собственных данных в режиме самообучения, что, конечно, лучше, чем начинать обучение со случайной инициализации, но всё же хуже, чем веса для ImageNet, благодаря чему этот набор данных остается самым популярным в компьютерном зрении по сей день. Однако, перспективы подходов самообучения гораздо шире, так как они способны

использовать необъемлемый набор данных.

### Выводы

По итогу данной работы удалось разработать модель машинного обучения, основанную на сверточных нейронных сетях, которая распознаёт, какому из 50 художников принадлежит данная картина. Ключевой метрикой для выбора оптимальной модели была f1-score. Финальная модель использует архитектуру RegNet, оптимизатор Adam с затухающей скоростью обучения, многозадачное обучение и модификацию данных в виде случайных вырезаний и вертикальных поворотов. Несмотря на попытки устранить переобучение, добавление различных методов регуляризации делают обучение нестабильным и лишь ухудшают метрики, поэтому финальная модель всё равно страдает от этой проблемы. В последующих исследованиях можно попробовать решить эту проблему, например, повышением разнообразия данных, или использованием ансамблей алгоритмов. Эксперимент с предварительным обучением через методы самообучения показал, что они способны помочь модели получать неплохие результаты, но пока что методы обучения с учителем остаются более предпочтительным решением, так как их качественно обученные параметры есть в открытом доступе для большинства современных архитектур.

Итоговое качество модели — метрика F1-score 0.722, точность — 77.75%. С большинством картин она справляется без проблем, например, на рис. 4 изображена картина Клода Моне, которая была правильно идентифицирована с вероятностью 99%.



Рис. 4. Легко идентифицированная картина Клода Моне

Если считать точность не по наиболее вероятному художнику, а по тройке, предложенной моделью, то точность возрастает до 91.5%. Так уже правильно идентифицируются и более сложные изображения, например, одна из картин Питера Пауля Рубенса (см. рис. 5). Она была отнесена к её автору с вероятностью 26%, в то время как к Эжену Делакруа — с вероятностью 41%. Интересно, что если посмотреть на работы Делакруа, то они действительно похожи на данную по стилистике, но при этом в тренировочном наборе данных их всего 45, поэтому модель плохо изучила этого художника и ошибочно отнесла картину к нему.



**Рис. 5. Правильно идентифицированная картина Питера Пауля Рубенса по топ-3 предсказаниям**

Тем не менее, итоговый алгоритм всё же имеет проблемы с некоторыми картинами, на которых так и не удалось получить качественный прогноз. К примеру, такой стала картина всё того же Питера Пауля Рубенса, изображенная на рис. 6.



**Рис. 6. Неправильно идентифицированная картина Питера Пауля Рубенса**

Согласно предсказанию, картина принадлежит ему с вероятностью менее 1%, в то время как Альбрехту Дюреру — с вероятностью 47%, Сандро Боттичелли — 16%, Иерониму Босху — 10%. Эти три художника жили и творили в конце XIV — начале

XV века в направлении Ренессанс, в то время Рубенс вел деятельность примерно на 100 лет позднее, и был уже представителем Барокко. Каждый из этих художников имеет около 150 картин в используемом наборе данных, что не так уж и мало. Это демонстрирует несовершенство модели, ведь несмотря на высокую точность в среднем иногда ее работу сложно интерпретировать.

Дополнительно модель способна прогнозировать жанр с точностью 76.5% (90.4% на топ-3), а также национальность с точностью 80% (94.4% на топ-3).

#### Список литературы:

- [1] I. Radosavovic, R. P. Kosaraju, R. Girshick и др. Designing Network Design Spaces, 2020. URL: <https://arxiv.org/pdf/2003.13678.pdf>.
- [2] K. He, X. Zhang, S. Ren, J. Sun. Deep Residual Learning for Image Recognition, 2015. URL: <https://arxiv.org/pdf/1512.03385.pdf>.
- [3] O. Russakovsky J. Deng H. Su и др. ImageNet Large Scale Visual Recognition Challenge, 2015. URL: <https://arxiv.org/pdf/1409.0575.pdf>.
- [4] Sebastian Ruder An Overview of Multi-Task Learning in Deep Neural Networks, 2020. URL: <https://arxiv.org/pdf/1706.05098.pdf>.
- [5] S. Gidaris, P. Singh, N. Komodakis Unsupervised Representation Learning by Predicting Image Rotations, 2018. URL: <https://arxiv.org/pdf/1803.07728.pdf>.
- [6] D. P. Kingma и J. L. Ba Adam: A Method for Stochastic Optimization. URL: <https://arxiv.org/pdf/1412.6980.pdf>.
- [7] Y. You J. Li S. Reddi и др. LARGE BATCH OPTIMIZATION FOR DEEP LEARNING: TRAINING BERT IN 76 MINUTES. URL: <https://arxiv.org/pdf/1904.00962.pdf>.
- [8] I. Loshchilov и F. Hutter SGDR: STOCHASTIC GRADIENT DESCENT WITH WARM RESTARTS. URL: <https://arxiv.org/pdf/1608.03983.pdf>.
- [9] N. Srivastava G. Hinton A. Krizhevsky и др. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. URL: <https://jmlr.org/papers/volume15/srivastava14a/srivastava14a.pdf>.
- [10] X. Li X. Sun Y. Meng и др. Dice Loss for Data-imbalanced NLP Tasks. URL: <https://arxiv.org/pdf/1911.02855.pdf>.
- [11] Документация PyTorch: <https://pytorch.org/docs/stable/index.html>.
- [12] Сайт Kaggle: <https://www.kaggle.com>.
- [13] Набор данных Best Artworks of All Time: <https://www.kaggle.com/ikarus777/best-artworks-of-all-time>.
- [14] Набор данных WikiART: <https://www.kaggle.com/ipythonx/wikiart-gangogh-creating-art-gan>.

*Поступила в редколлегию 16.04.2021*