ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»

Факультет информатики, математики и компьютерных наук

**Программа подготовки бакалавров по направлению
Компьютерные науки и технологии**

Пестов Лев Евгеньевич

КУРСОВАЯ РАБОТА

Интерактивный конструктор моделей искусственного интеллекта с поддержкой мультимодальных задач.
Реализация пайплайна обучения глубоких нейросетей

Научный руководитель
старший преподаватель НИУ
ВШЭ - НН

Саратовцев Артём Романович

Нижний Новгород, 2025г.

Структура работы

1. Введение

Развитие искусственного интеллекта (ИИ) и его интеграция в различные сферы деятельности привело к значительному увеличению числа приложений, использующих глубокие нейронные сети. Однако процесс разработки таких моделей по-прежнему остается сложной и ресурсоемкой задачей, требующей от разработчика не только понимания архитектуры нейросетей, но и глубоких знаний в области обработки данных, настройки гиперпараметров и оптимизации вычислительных ресурсов. Это существенно ограничивает доступность технологий машинного обучения для широкой аудитории, в частности для исследователей, предпринимателей и специалистов, не обладающих достаточным уровнем подготовки в данной области.

Данная работа направлена на создание интерактивного конструктора моделей искусственного интеллекта, который позволит пользователям, не имеющим значительного опыта в разработке ИИ, самостоятельно обучать нейросети на своих данных под определенные классы задач. В отличие от существующих решений, сервис ориентирован на адаптацию моделей к малым объемам данных, что особенно актуально в условиях ограниченности ресурсов и сложности сбора крупных размеченных датасетов.

Существенной частью системы является интеграция с облачными вычислительными мощностями Yandex Cloud, что позволяет пользователям запускать обучение моделей удалённо, без необходимости наличия высокопроизводительного оборудования. Впоследствии обученные модели могут быть загружены обратно в сервис для тестирования либо экспортированы в виде предобученных весов с преднаписанным кодом для внедрения в сторонние проекты.

Целью исследования является сравнение различных архитектур моделей и их обучение, с учетом того, что ресурсы ограничены, а также создание скриптов с динамическим изменением гиперпараметров обучения и генеральной совокупности данных для подачи на бекенд-часть сервиса.

Задачи проекта:

1. Провести аналитический обзор литературы и выбрать лучшие архитектуры моделей под разные домены машинного обучения, также рассмотреть способы аугментации данных.

2. Выбрать способ обучения моделей, а также реализовать полное/fine-tuning обучение разных архитектур.

3. Сравнить полученные результаты полученные на разных данных/методах обучения и сделать оптимальный выбор модели под задачи сервиса.

2. Разработка пайплайна обучения нейросетей для задачи классификации изображений

2.1. Теоретические основы и архитектуры нейросетей для классификации изображений

Задача классификации изображений является одной из основных в компьютерном зрении и представляет собой процесс экстракции высокоуровневых и низкоуровневых признаков в один или несколько классов. Для решения таких задач в последние годы наиболее эффективными оказались глубокие нейронные сети, в частности, сверточные нейронные сети (CNN) и трансформеры (ViT).

Сверточные нейронные сети (Convolutional Neural Networks, CNN) - это класс глубоких нейронных сетей, специально разработанных для обработки изображений. Их ключевой особенностью является использование операции свертки вместо обычного матричного умножения хотя бы в одном из слоев. CNN произвели революцию в области компьютерного зрения, предоставив эффективный способ автоматического извлечения признаков из изображений.

Операция свертки - Фильтр перемножает числа своей матрицы и матрицы картинки, далее они суммируются в одно число, процесс итеративно продолжается и тем самым получается новая матрица изображения.

Архитектура **AlexNet** — первая глубокая сверточная нейронная сеть, которая значительно превзошла предыдущие подходы к распознаванию изображений. Предложенная в 2012 году [1], она стала прорывом в задачах классификации изображений, впервые показав возможности глубоких нейронных сетей превзойти классические методы машинного обучения на соревновании ImageNet. AlexNet состоит из 5 сверточных слоёв и 3 полносвязаных слоёв.

VGGNet — продолжение развития сверточных нейросетей, архитектура сети предложенная Оксфордским университетом в 2014 году [2]. Основное различие между AlexNet и VGG заключается в размере фильтров, глубине сети и количестве параметров. AlexNet использует более крупные фильтры, например, 11×11 в первом сверточном слое, тогда как VGG применяет исключительно небольшие 3×3 фильтры, этот приём значительно увеличил глубину сети. Младшая модель VGG 11 имеет 9 сверточных и 2 полносвязаных слоя, и содержит в себе более 130 миллионов параметров, в то время как у AlexNet их всего 60. Это сделало VGGNet тяжеловесными и требовательными к вычислительным ресурсам.

Архитектура **Inception** (GoogLeNet) - её ключевой особенностью являются Inception-блоки, которые параллельно применяют сверточные операции с разными размерами фильтров (1×1, 3×3, 5×5) вместе с операциями max-pooling. Это позволяет сети извлекать признаки на разных уровнях детализации одновременно [3]. Архитектура активно использует сверточные слои 1×1 для уменьшения размерности данных, что помогло снизить вычислительную сложность и количество параметров.

Каждая новая архитектура увеличивалась в размере, и поэтому появилась проблема, что градиенты на последних слоях были совсем маленькими, и это не давало нормально обновить веса модели. **ResNet** (Residual Networks) решила проблему увеличения глубины сетей [4]. Авторы ResNet использовали "остаточные блоки"с соединениями быстрого доступа (skip connections), которые позволяют градиентам эффективно распространяться через многие слои. Входные данные передаются по дополнительному соединению в обход следующих слоев и добавляются к полученному результату, это соединение не добавляет дополнительные параметры в сеть, поэтому ее структура не усложняется.

Появилась необходимость использования нейронных сетей на мобильных устройствах, и в 2017 году исследователями из Google была представлена архитектура **MobileNet** [5]. Основным отличием MobileNet является использование глубоких отделимых сверточных слоев (depthwise separable convolutions), которые разделяют стандартную свертку на две операции - глубинную свертку (depthwise convolution), которая выполняет фильтрацию, применяя один фильтр для каждого входного канала и поточечную свертку (pointwise convolution) с фильтром 1×1, которая выполняет комбинирование выходных каналов. Этот подход значительно снижает вычислительную сложность и количество параметров по сравнению с обычными сверточными слоями. MobileNetV1 содержит около 4,2 млн параметров и показывает 70,6% top-1 на соревновании ImageNet при значительно меньших вычислительных затратах по сравнению с предыдущими сетями.

Авторы обнаружили, что для достижения оптимальной производительности необходимо сбалансированное масштабирование 3 измерений сети: глубины (количества слоев), ширины (количества каналов), разрешения (размера входного изображения). В отличие от предыдущих подходов, которые обычно масштабировали только одно из этих измерений, авторы предлагают составное масштабирование (compound scaling). Из существующего метода под названием «Neural Architecture Search» [6] для автоматического создания новых сетей и своего собственного метода масштабирования авторы получают новый класс моделей под названием **EfficientNet** [7]. EfficientNet-B0 содержит около 5,3 млн параметров

и достигает точности 77,1% top-1 на ImageNet. Более крупные версии демонстрируют еще более высокую точность при контролируемом увеличении количества параметров: например, EfficientNet-B7 достигает 84,4% с более чем 60 млн параметров, что является State-Of-The-Art разработкой до сих пор.

Обращая внимание на архитектуры основанных на трансформерах, стоит упомянуть **Vision Transformer** (ViT), предложенный в 2020 году [8], адаптирует архитектуру трансформера для изображений. ViT разбивает изображение на последовательность непересекающихся патчей, преобразует их в эмбеддинги и подает эту последовательность на вход стандартного трансформера. ViT демонстрирует впечатляющие результаты при обучении на больших наборах данных (ViT-L/16 получил 85.30% top-1 точности после предобучения на датасете JFT-300M), но уступает CNN на меньших датасетах (ViT-B/16 показал слабые результаты при обучении только на ImageNet-1k без предварительного обучения на больших наборах данных, 79,9% top-1 точности на ImageNet, что уступало современным CNNs) и требует значительных вычислительных ресурсов для обучения.

Также важно отметить семейство моделей **YOLO** (You Only Look Once), разработанных специально для детекции объектов в изображениях в режиме реального времени [9]. В отличие от обычных CNN, которые часто используются для классификации изображений или извлечения признаков, YOLO объединяет в себе процессы детекции объектов и их классификации, обрабатывая изображение целиком за один проход. Традиционно многие системы сначала выделяют регионы интереса, а затем отдельно классифицируют содержимое этих регионов. В случае YOLO всё это происходит за один шаг - сеть делит изображение на сетку и для каждой ячейки одновременно предсказывает координаты прямоугольника и вероятность того, что в нём находится объект определённого класса. Это объединение позволяет существенно ускорить задачу классификации. **YOLOv11** [10] является одной из последних моделей этой серии, конкурируя с классическими CNNs.

Обоснование выбора архитектур для работы

Для задач класса *Few-Shot Learning* - необходимо выбрать такие архитектуры, которые обеспечат баланс между точности, скорости обучения и возможности обучаться на небольших данных. На основе проведенного анализа современных архитектур, особенно учитывая требование адаптации к малым объемам данных, для экспериментов выбраны две архитектуры:

1. EfficientNet

- Эффективность использования параметров - отличные метрики при относительно небольшом количестве параметров
- Масштабируемость - семейство моделей от B0 до B7 позволяет выбрать оптимальный баланс между точностью и сложностью
- Меньшая требовательность к вычислительным ресурсам - можно эффективно использовать облачную инфраструктуру Yandex Cloud и быстрый локальный инференс
- Простая поддержка и создание пайплайна для обучения, так как EfficientNet уже есть в базовых классах PyTorch.

2. YOLO

- Универсальность - может использоваться как для классификации, так и для обнаружения объектов. (Удобно для развертывания в дальнейших задач проекта)
- Высокая скорость работы - однопроходная архитектура обеспечивает быстрый инференс, но и обучение.
- Точность — последние версии (YOLOv11) показывают конкурентные результаты по сравнению с другими современными моделями.
- Масштабируемость — доступны варианты разного размера (nano, small, medium, large, xlarge)

2.2. Аугментации данных и реализация процесса обучения моделей

Список использованной литературы

[1] Alex Krizhevsky, Ilya Sutskever и Geoffrey E Hinton. "Imagenet classification with deep convolutional neural networks". В: *Advances in neural information processing systems*. Т. 25. 2012. URL: https://papers.nips.cc/paper_files/paper/2012/hash/c399862d3b9d6b76c8436e924a68c45b-Abstract.html.

[2] Karen Simonyan и Andrew Zisserman. "Very Deep Convolutional Networks for Large-Scale Image Recognition". В: *arXiv preprint arXiv:1409.1556* (2014). URL: https://arxiv.org/abs/1409.1556.

[3] Christian Szegedy и др. "Going deeper with convolutions". В: *arXiv preprint arXiv:1409.4842* (2014). URL: https://arxiv.org/abs/1409.4842.

[4] Kaiming He и др. “Deep residual learning for image recognition”. В: *arXiv preprint arXiv:1512.03385* (2015). URL: https://arxiv.org/abs/1512.03385.

[5] Andrew G Howard и др. “MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications”. В: *arXiv preprint arXiv:1704.04861* (2017). URL: https://arxiv.org/abs/1704.04861.

[6] Barret Zoph и Quoc V Le. “Neural Architecture Search with Reinforcement Learning”. В: *arXiv preprint arXiv:1611.01578* (2017). URL: https://arxiv.org/abs/1611.01578.

[7] Mingxing Tan и Quoc Le. “EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks”. В: *Proceedings of the 36th International Conference on Machine Learning*. PMLR, 2019, с. 6105—6114. URL: https://proceedings.mlr.press/v97/tan19a/tan19a.pdf.

[8] Alexey Dosovitskiy и др. “An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale”. В: *arXiv preprint arXiv:2010.11929* (2020). URL: url=%7Bhttps://arxiv.org/abs/2010.11929%7D.

[9] Joseph Redmon и др. “You Only Look Once: Unified, Real-Time Object Detection”. В: *arXiv preprint arXiv:1506.02640* (2015). URL: https://arxiv.org/abs/1506.02640.

[10] Rahima Khanam и Muhammad Hussain. “YOLOv11: An Overview of the Key Architectural Enhancements”. В: *arXiv preprint arXiv:2410.17725* (2024). URL: https://arxiv.org/pdf/2410.17725.