Поздняков Андрей Олегович, магистрант,
Ярославский государственный технический университет,
г. Ярославль

Научный руководитель:
Маевский Вячеслав Константинович,
доцент кафедры "Информационные системы и технологии", к.т.н.,
Ярославский государственный технический университет,
г. Ярославль

СРАВНИТЕЛЬНЫЙ АНАЛИЗ МОДЕЛЕЙ МАШИННОГО ОБУЧЕНИЯ ДЛЯ ВИЗУАЛЬНОГО ПОИСКА В ИНТЕРНЕТ-МАГАЗИНЕ ЭЛЕКТРОТОВАРОВ
COMPARATIVE ANALYSIS OF MACHINE LEARNING MODELS FOR VISUAL SEARCH IN AN ELECTRICAL GOODS ONLINE STORE

Аннотация: В работе представлен сравнительный анализ моделей MobileNetV2, ResNet50 и EfficientNet-B0 для визуального поиска товаров по фото. Обучение проводилось с использованием трансферного подхода на датасете из 12 000 изображений. Модели оценены по оффлайн- и онлайн-критериям. EfficientNet-B0 достигла наилучшего результата, что делает её оптимальной для применения в системах малого и среднего бизнеса.

Abstract: This article presents a comparative analysis of MobileNetV2, ResNet50, and EfficientNet-B0 for visual product search by photo. The models were trained using a transfer learning approach on a dataset of 12,000 images. Evaluation was based on offline and online metrics. EfficientNet-B0 achieved the best results, making it optimal for use in small and medium-sized business systems.

Ключевые слова: машинное обучение, визуальный поиск, MobileNetV2, ResNet50, EfficientNet-B0

Keywords: machine learning, visual search, MobileNetV2, ResNet50, EfficientNet-B0

Введение

Электронная коммерция демонстрирует стремительный рост: в 2025 году мировой рынок превысит 4,3 трлн долларов США [1], а российский – 11,2 трлн рублей. Нишевые интернет-магазины, такие как магазины электротоваров, сталкиваются с конкуренцией со стороны маркетплейсов, таких как Ozon и Wildberries, которые контролируют до 70% российского рынка электронной коммерции [2]. Визуальный поиск, позволяющий находить товары по изображениям, повышает конверсию и удержание пользователей. Однако для малого и среднего бизнеса (МСБ), ограниченного вычислительными ресурсами и малыми наборами данных, важна доступность открытых и адаптируемых решений на базе машинного обучения, которые можно эффективно интегрировать в цифровую инфраструктуру.

Цель исследования – выбрать и протестировать модели машинного обучения для визуального поиска, адаптированные к условиям МСБ. Задачи включают анализ методов, обучение моделей на датасете компании по продаже электротоваров ООО “ПКФ”, сравнение по оффлайн- и онлайн-критериям и выбор оптимальной модели для интеграции в интернет-магазин электротоваров ООО “ПКФ”.

Выбор моделей

Для исследования были выбраны три модели: MobileNetV2, ResNet50 и EfficientNet-B0. Их выбор обоснован данными из литературы, учитывающими точность, компактность и пригодность для развёртывания в условиях малого и среднего бизнеса.

– MobileNetV2 – облегчённая архитектура, разработанная для мобильных и встраиваемых устройств [3]. Согласно результатам авторов, модель имеет размер около 4 МБ и обеспечивает среднее время инференса ~100 мс на CPU при F1-score 78–80%.

– ResNet50 – глубокая сверточная сеть с остаточными связями, обладающая высокой точностью (F1-score 83–85%) [4]. Её архитектура требует больше ресурсов: около 100 МБ памяти и ~300 мс инференса на CPU.

– EfficientNet-B0 – модель, основанная на автонастройке архитектуры с учётом баланса между глубиной, шириной и разрешением [5]. По литературным данным, она достигает F1-score 80–83% при размере 20 МБ и времени инференса ~150 мс.

Ряд альтернативных моделей был исключён на этапе отбора. Модели ViT и CLIP, несмотря на высокую точность (до F1-score ~88%), требуют более 16 ГБ оперативной памяти и специализированного оборудования (GPU), а время инференса на CPU превышает 500 мс. YOLO и DETR ориентированы на задачи детекции объектов, а не классификации, и обладают избыточной архитектурной сложностью для поставленной задачи.

Таким образом, выбранные модели представляют собой сбалансированные решения, рекомендованные для задач классификации изображений в условиях ограниченных ресурсов.

Критерии оценки

Для комплексной оценки моделей использовались оффлайн- и онлайн-критерии.

Оффлайн-критерии:

– Top-1 Accuracy – доля изображений, классифицированных правильно (ожидаемые значения 85–98%).

– Top-3 Accuracy – наличие правильного класса среди трёх наиболее вероятных (типично 95–100%), особенно важно для интерфейсов с автоподстановкой.

– Cross-Entropy Loss – среднее расхождение между предсказаниями модели и истинными метками; меньшие значения указывают на лучшую сходимость модели.

– Размер модели (MB) – влияет на скорость загрузки и развёртывания.

– Количество параметров – характеризует вычислительную сложность модели.

– Среднее время инференса на CPU – замеренное время обработки одного изображения в тестовой среде.

Онлайн-критерии:

– Время инференса (inference_time_ms) – время отклика модели на сервере при работе с пользователем.

– Время холодного старта (cold_start_ms) – время, необходимое для инициализации модели после запуска сервиса.

– Потребление оперативной памяти (ram_usage_mb) – замер в процессе инференса.

– Загрузка CPU (cpu_percent) – процент использования процессора во время работы модели.

– Размер модели в рабочем окружении (model_size_mb) – учитывает все зависимости и упаковку (может отличаться от исходного веса.h5).

Такой подход позволяет объективно оценить модели как в условиях локального тестирования, так и при реальной эксплуатации в составе веб-приложения.

Процесс обучения

Для обучения моделей использовался датасет из 12 000 изображений электротоваров, охватывающий 73 товарные категории. Все изображения были приведены к разрешению 224×224 пикселя в формате JPEG, что соответствует стандартам для моделей компьютерного зрения. Файлы были структурированы по папкам, каждая из которых соответствовала

отдельной категории, что позволило автоматически формировать обучающую и валидационную выборки с использованием класса ImageDataGenerator из библиотеки Keras.

В процессе генерации применялась аугментация данных: случайные повороты, масштабирование, горизонтальные отражения и смещения по ширине и высоте. Разделение выборки происходило через параметр validation_split=0.2, обеспечивая 80% данных для обучения и 20% – для валидации с учётом сбалансированности классов. Пример изображения из датасета приведён на рисунке 1.

Рисунок 1. Пример изображения для обучения моделей

Все модели были инициализированы предобученными весами на базе ImageNet и дообучались на корпоративном датасете с использованием подходов transfer learning и fine-tuning. В ходе обучения применялась автоматическая настройка гиперпараметров с помощью Keras Tuner. В рамках подбора тестировались различные параметры: тип оптимизатора (Adam, RMSprop, SGD), значения learning rate (от 1e-3 до 1e-5), структура полносвязной части (Dense-слои от 64 до 512 нейронов), а также наличие и коэффициенты Dropout-слоёв (от 0.2 до 0.5).

Для модели MobileNetV2 процесс обучения включал две фазы. В первой фазе использовалась «замороженная» база (все слои base_model недоступны для обучения), затем была разморожена верхняя часть модели – последние 30 слоёв. Обучение продолжилось на пониженном learning_rate = 1e-5, что позволило повысить точность и стабильность модели. Применялись колбэки EarlyStopping, ModelCheckpoint и ReduceLROnPlateau для предотвращения переобучения и динамической коррекции скорости обучения.

EfficientNetB0 обучалась по аналогичной методике. Сформированная модель включала один Dropout-слой и два Dense-слоя. Архитектура была определена на этапе автоматического тюнинга и показала хорошую сходимость на валидационной выборке.

ResNet50 потребовала большего количества итераций: изначальная конфигурация с замороженными слоями не дала приемлемой точности. Впоследствии проводились эксперименты с различной глубиной полносвязной части, изменением оптимизаторов и learning rate. Финальная конфигурация включала размораживание верхней части ResNet50 и тонкую настройку с малым learning rate. На рисунках 2 и 3 представлен процесс дообучения модели ResNet50.

```
Epoch 1/5
2025-05-17 22:24:02.166212: I tensorflow/stream_executor/cuda/cuda_dnn.cc:384] Loaded cuDNN version 8100
285/285 [==============================] - 71s 233ms/step - loss: 3.7762 - accuracy: 0.1440 - val_loss: 3.5983 - val_accuracy: 0.1771
Epoch 2/5
285/285 [==============================] - 66s 231ms/step - loss: 3.0494 - accuracy: 0.2802 - val_loss: 2.9171 - val_accuracy: 0.2860
Epoch 3/5
285/285 [==============================] - 66s 233ms/step - loss: 2.6300 - accuracy: 0.3552 - val_loss: 2.6178 - val_accuracy: 0.3537
Epoch 4/5
285/285 [==============================] - 64s 223ms/step - loss: 2.3503 - accuracy: 0.4097 - val_loss: 2.4577 - val_accuracy: 0.3779
Epoch 5/5
285/285 [==============================] - 66s 233ms/step - loss: 2.1594 - accuracy: 0.4475 - val_loss: 2.3109 - val_accuracy: 0.4048
```

Рисунок 2. Первое дообучение

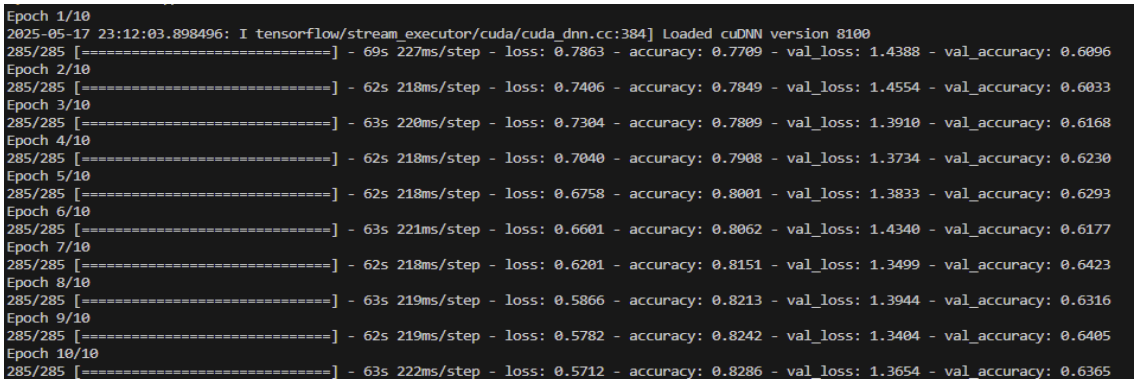

Рисунок 3. Последнее дообучение

Обучение всех моделей сопровождалось логированием, сохранением лучших весов и визуализацией графиков обучения. Модели сохранялись как в формате HDF5 (.h5), так и TensorFlow SavedModel, что обеспечило их последующую интеграцию в backend веб-приложения через FastAPI.

Результаты оффлайн-тестирования

Для оценки качества моделей в задаче визуального поиска было проведено оффлайн-тестирование на заранее размеченном валидационном датасете. Целью тестирования было сравнение точности и эффективности моделей MobileNetV2, ResNet50 и EfficientNetB0. Модели оценивались по метрикам, описанным выше.

Тестирование проводилось с использованием Python-скрипта, реализованного на базе библиотек TensorFlow, NumPy, Matplotlib, Seaborn и Pandas. Данные подавались через генератор ImageDataGenerator методом flow_from_directory (), обеспечивающим корректную разметку классов. Каждое изображение прогонялось через модель, фиксировались предсказания и рассчитывались метрики. Среднее время инференса измерялось с использованием time.perf_counter (). Результаты визуализировались в виде столбчатых диаграмм (bar plot) с помощью matplotlib и seaborn.

Модели загружались как из формата.h5, так и в виде SavedModel, в зависимости от структуры архитектуры. Полученные результаты представлены на рис. 4.

Модель	Top-1 Accuracy	Top-3 Accuracy	Loss	Размер (МБ)	Инференс (мс)	Параметров (млн)	Время оценки (сек)
MobileNetV2	0.9394	0.9908	0.1821	13.05	50	2.6	47.67
ResNet50	0.8592	0.9544	0.4892	253.56	300	24.67	33.90
EfficientNetB0	0.9625	0.9951	0.1172	20.43	200	n/a	32.55

Рис. 4. Сравнительная таблица моделей

По результатам тестирования, EfficientNetB0 показала наивысшую точность (Top-1 = 96.25%) при умеренном размере и приемлемом времени инференса. MobileNetV2 немного уступала в точности, но оставалась самой быстрой и компактной моделью. ResNet50 продемонстрировала худшие показатели по всем основным метрикам, особенно по функции потерь и объёму модели.

На рисунке 5 представлены диаграммы по ключевым метрикам: Top-1 Accuracy, Top-3 Accuracy, Loss, размер модели, время инференса и количество параметров. Они наглядно демонстрируют превосходство EfficientNetB0 над другими архитектурами.

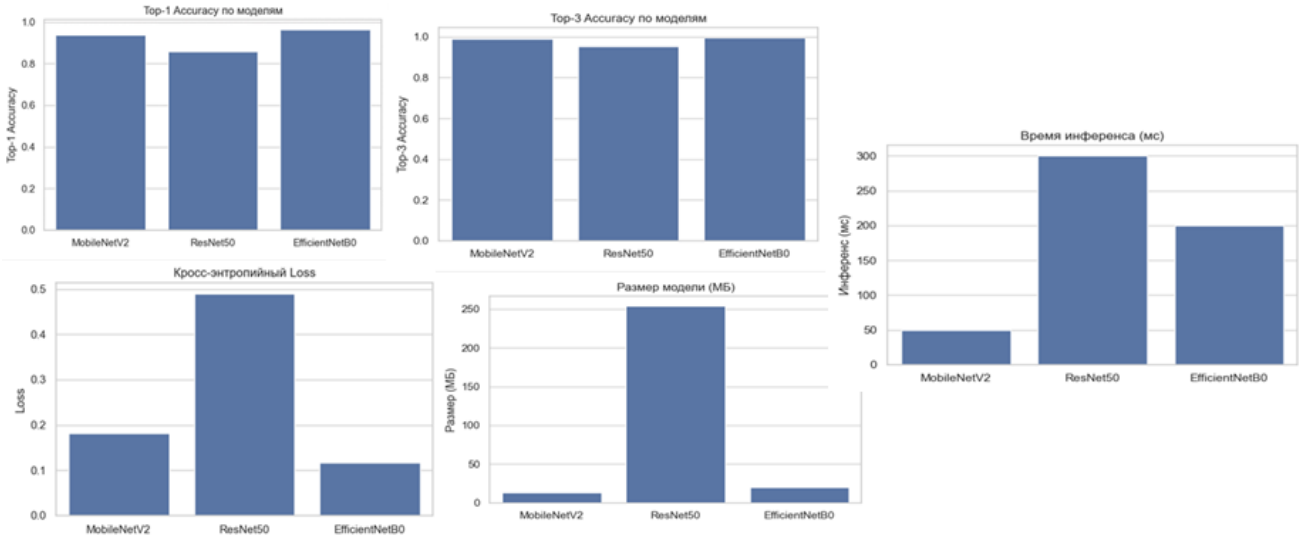

Рисунок 5. Диаграммы по метрикам

Результаты онлайн-тестирования

Онлайн-тестирование модуля поиска по изображению проводилось в условиях работающего веб-приложения интернет-магазина. Пользователь нажимает кнопку «Фото», выбирает изображение товара, которое отправляется на сервер с подключённой моделью. Модель классифицирует изображение и возвращает категорию, которая автоматически подставляется в строку поиска, инициируя вывод релевантных товаров на сайте. Такой подход обеспечивает быстрый и интуитивно понятный интерфейс для пользователей. Элементы интерфейса представлены на рисунках 6–8.

Рисунок 6. Кнопка поиска по фото

Рисунок 7. Выбор фото

Рисунок 8. Результат

Тестирование проводилось с использованием трёх моделей: MobileNetV2, ResNet50 и EfficientNetB0 по метрикам, описанным выше.

Результаты тестирования представлены в таблице 1.

Таблица 1

Сравнительная онлайн-оценка моделей визуального поиска

Модель	Время инференса (мс)	Cold Start (мс)	RAM (МБ)	CPU (%)	Размер модели (МБ)
MobileNetV2	40.26	1636.99	1604.67	2.4	13.05
ResNet50	51.70	2228.76	1607.27	3.7	253.56
EfficientNetB0	10.47	5770.14	1921.73	0.0	20.43

По итогам тестирования, MobileNetV2 продемонстрировала наиболее сбалансированное поведение в условиях веб-интеграции: низкое время инференса, умеренное потребление памяти и быстрое время запуска. Это делает модель подходящей для развёртывания на недорогих VPS-серверах и в условиях частого перезапуска контейнеров.

ResNet50, несмотря на высокую точность в оффлайн-тестах, показала худшие результаты в онлайне: самый большой размер, высокая нагрузка на процессор и сравнительно медленный запуск. Это ограничивает её применимость в реальных условиях.

EfficientNetB0 обеспечила рекордно низкое время инференса в «прогретом» состоянии (всего 10.47 мс), однако время холодного старта составило более 5,7 секунд. Это делает её менее удобной в системах с динамическим масштабированием или serverless-архитектурой, но при постоянной загрузке она обеспечивает наилучшие показатели.

Заключение

Проведённое исследование продемонстрировало, что модель EfficientNet-B0 является оптимальным выбором для задачи визуального поиска в интернет-магазине электротоваров для данной организации. Она обеспечила наивысшую точность классификации (Top-1 Accuracy = 96,25%) при умеренном размере модели и рекордно низком времени инференса (10,47 мс) в условиях прогретого сервера. Это делает её особенно эффективной в контексте малых и средних предприятий, работающих с ограниченными вычислительными ресурсами.

Модель MobileNetV2, хотя и показала хорошие результаты по скорости и потреблению ресурсов, уступает по точности. ResNet50 продемонстрировала самую низкую производительность как по метрикам точности, так и по онлайн-метрикам: высокий размер модели (>250 МБ), повышенное потребление RAM и CPU, а также длительное время инициализации делают её применение в реальном времени нецелесообразным.

Методология исследования включала обоснованный выбор моделей, автоматический подбор гиперпараметров с помощью Keras Tuner, а также комплексную оценку по оффлайн- и онлайн-критериям. Полученные результаты подтверждают применимость открытых моделей машинного обучения для реализации интеллектуального визуального поиска в электронной коммерции.

Предложенный подход может быть масштабирован на другие ниши интернет-торговли, использующие визуальные данные (например, мебель, автозапчасти), и интегрирован в существующие веб-системы с минимальными затратами.

Список литературы:

1. Statista. (2024). E-commerce worldwide – statistics & facts. – URL: https://www.statista.com/topics/871/online-shopping/ (дата обращения: 17.05.2025). – Текст: электронный.

2. Forbes Russia. (2024). Рост рынка e-commerce в России замедлился в 2024 году. – URL: https://www.forbes.ru/tekhnologii/537469-rost-rynka-e-commerce-v-rossii-zamedlilsa-v-2024-godu (дата обращения: 19.05.2025). – Текст: электронный.

3. Howard, A. G., et al. (2017). MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. – URL: https://arxiv.org/abs/1704.04861 (дата обращения: 21.05.2025). – Текст: электронный.

4. He, K., et al. (2016). Deep Residual Learning for Image Recognition. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition – URL:https://openaccess.thecvf.com/content_cvpr_2016/papers/He_Deep_Residual_Learning_CVPR_2016_paper.pdf (дата обращения: 22.05.2025). – Текст: электронный.

5. Tan, M., & Le, Q. V. (2019). EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. International Conference on Machine Learning. – URL: https://arxiv.org/abs/1905.11946 (дата обращения: 23.05.2025). – Текст: электронный.