

Разработка метода выбора моделей машинного обучения распознавания объектов на изображениях с применением логики продукционных правил

А.А. Сметанин¹, А.В. Духанов¹, М.Ю. Герасимчук¹

¹Национальный исследовательский университет ИТМО,
197101, Россия, Санкт-Петербург, Кронверкский пр., д. 49

Аннотация

На сегодняшний день технологии в области фото- и видеообработки данных значительно продвинулись, что позволило алгоритмам распознавания и классификации объектов достигать точности выше 90 %. Такие прорывные решения стали широко использоваться в личных и профессиональных целях. Вместе с тем расширение областей применения данных решений привело к увеличению влияющих факторов и их вариаций. От таких факторов зависит качество получения и обработки изображений, включая динамику и деформацию объектов в кадре, что усложняет практическое применение соответствующих алгоритмов. Настоящая статья предлагает подход к разработке рекомендательной системы для выбора моделей машинного обучения в целях решения широкого спектра задач, связанных с обнаружением объектов в кадре изображения, полученных в различных условиях. Данный подход основан на механизме продукционных правил, которые формируются по результатам выполненных ранее исследований, а также анализа научных источников (статьи, репозитории научных программных библиотек и пр.). Результатом функционирования разрабатываемой системы является не просто перечень моделей глубокого обучения с указанием их релевантности к применению в задаче пользователя и априорными оценками значений метрик качества. Предлагаемый подход позволяет формировать предложения по созданию конвейеров машинного обучения и рекомендаций по установке и использованию соответствующих программных библиотек.

Ключевые слова: модели и конвейеры машинного обучения, детектирование объектов, продукционные правила, рекомендательная система.

Цитирование: Сметанин, А.А. Разработка метода выбора моделей машинного обучения распознавания объектов на изображениях с применением логики продукционных правил / А.А. Сметанин, А.В. Духанов, М.Ю. Герасимчук // Компьютерная оптика. – 2025. – Т. 49, № 5. – С. 826-834. – DOI: 10.18287/2412-6179-CO-1583.

Citation: Smetanin AA, Dukhanov AV, Gerasimchuk MY. Development of a methodology for selecting machine learning models for recognizing moving objects in a video stream based on production rules. Computer Optics 2025; 49(5): 826-834. DOI: 10.18287/2412-6179-CO-1583.

Введение

В настоящее время объем и скорость генерации данных стремительно растет, что приводит к повышению значимости машинного обучения (МО) в широком разнообразии классов задач и их параметров. В данном контексте использование конвейеров МО и ансамблей моделей/архитектур становится актуальным. На текущий момент разработано множество наукоемких решений в части детектирования и(или) распознавания объектов как на статических изображениях, так и на видео, в том числе в режиме реального времени с заданными параметрами реактивности [1, 2]. В то же время конечному пользователю, например, специалисту по данным, сложно подобрать нужное решение для его ситуации, которая характеризуется открытым набором параметров с соответствующими множествами допустимых значений. Одним из развивающихся направлений в данной области является разработка рекомендательных систем (РС), которые на основе имеющихся наборов данных (НД) способны находить подходящее решение для конкретных задач [3, 4].

Целью настоящего исследования является разработка метода автоматизации формирования рекомендаций по выбору моделей глубокого обучения в зависимости от обучающего набора данных и требований к детектированию объектов в видеопотоке. Обучающий набор данных характеризуется такими свойствами, как объем, баланс по классам, разрешение изображений, размеры объектов заданного класса, условия получения изображений и другие. Требования к детектированию включают диапазоны точности, вычислительной реактивности распознающей системы, параметры вычислительной среды и другие.

При выборе моделей мы учитываем результаты решения задач детектирования объектов, полученные в прошлых исследованиях на различных наборах данных. Такие результаты аккумулируются в специальной базе знаний.

Разработка и реализация метода автоматизации формирования рекомендаций позволит специалистам компьютерного зрения и смежных областей ускорить разработку соответствующих решений. Также специалисты смогут за одно и то же время исследовать несколько

архитектур глубокого обучения, применимых к основной и смежным задачам.

В оставшейся части статьи будет представлен обзор релевантных исследований, на основе которого будет уточнена основная задача исследования. Затем в параграфе «Метод автоматизации формирования рекомендаций» будет рассмотрен принцип формирования продукционных правил и алгоритм получения рекомендаций. В параграфе «Проведение экспериментальных исследований и результаты» представлены результаты проверки разработанной методики на различных НД с применением метрик для оценки точности и скорости моделей МО. Для проведения эксперимента в качестве виртуальной экспериментальной установки на языке программирования Python разработано программное средство, а также была подготовлена и наполнена база знаний продукционных правил. Статья завершается заключением и информацией о будущих исследованиях.

1. Обзор существующих научно-технических решений

Обзор нацелен на представление ассортимента современных архитектур глубокого обучения для решения задач детектирования объектов, который быстро расширяется. Также рассмотрены работы и решения в области автоматизации формирования конвейеров компьютерного зрения на основе автоматического машинного обучения (AutoML). Обзор завершается качественным сравнительным анализом выявленных решений.

Детектирование, распознавание движущихся объектов в видеопотоке является одной из фундаментальных задач в различных областях [1]. Методы детектирования объектов, основанные на глубоком обучении, широко применяются в различных областях, например, распознавании лиц, мониторинге дефектов продукции, отслеживании качества работы, оценивании эмоционального состояния человека и многих других. Одними из самых популярных архитектур, используемых для этих задач, являются YOLO (You Only Look Once) [5]. Эти архитектуры адаптируются к различным размерам объектов и экономично используют ресурсы, что делает их популярными для встраиваемых систем детектирования [5, 6]. Для задач распознавания объектов различного масштаба подходит нейросетевая архитектура SSD (Single Shot MultiBox Detector), она представляет собой одну нейронную сеть, способную прогнозировать разные размеры и классы объектов с разных слоев в сверточной архитектуре, что позволяет эффективно обнаруживать объекты разных масштабов [7]. Также одними из эффективных архитектур являются рекуррентные сверточные сети (R-CNN) [8, 9].

Для объективного сравнения моделей детектирования объектов широко используются две ключевые метрики: *mean Average Precision* (mAP) для оценки

точности и кадры в секунду (FPS) для оценки скорости. Метрика mAP учитывает точность и полноту модели, что позволяет оценить её способность правильно классифицировать и детектировать объекты. Показатель FPS, в свою очередь, отражает скорость работы модели и демонстрирует, сколько кадров в секунду она способна обрабатывать. Для задач детектирования в реальном времени скорость имеет важнейшее значение, поскольку задержки в обработке могут негативно сказаться на общей производительности системы. Использование FPS в качестве метрики позволяет оценить, насколько эффективно модель справляется с потоками данных в условиях ограниченных вычислительных ресурсов, что особенно актуально для встраиваемых систем [10].

С ростом числа нейронных архитектур специалистам становится сложнее быстро выбирать наилучшие архитектуры для создания эффективных решений. В результате возникли фреймворки AutoML, реализующие автоматизированные методы выбора подходящих/оптимальных решений с использованием собственных алгоритмов [11]. Методы автоматического машинного обучения охватывают разнообразные подходы, включая эволюционные алгоритмы, оптимизацию параметров и адаптивные исследования пространства параметров. Такие подходы позволяют более эффективно исследовать пространство параметров и архитектур, обнаруживая оптимальные решения для разнообразных задач машинного обучения.

В табл. 1 представлен общий сравнительный анализ современных решений как в России, так и за рубежом. В данной работе предложен метод, который отличается от существующих подходов AutoML. Основными преимуществами предлагаемого решения являются уникальный метод рекомендаций моделей, основанный на пользовательском опыте, собранном в базе знаний, и механизме продукционных правил. В числе прочих особенностей: открытый исходный код, возможность работы в офлайн-режиме, гибкая настройка параметров.

Предлагаемое решение позволяет повысить эффективность специалистов за счет сокращения временных затрат на рутинные процессы, глубже погрузиться в свое исследование на творческом уровне.

2. Метод автоматизации формирования рекомендаций

Для формирования композиций моделей МО и подходящего конвейера, нацеленных на решение конкретных задач пользователей, мы прибегаем к использованию механизма продукционных правил. Он базируется на создании набора правил, основанных на знаниях, полученных в результате проведения экспериментов с нейросетевыми архитектурами на различных наборах данных. Эти правила позволяют предоставлять рекомендации, адаптированные под конкретные

задачи и требования. Тем самым, при наличии релевантного пользовательского опыта, отраженного в базе знаний, метод обеспечивает предоставление реко-

мендаций, пригодных для высокоточного решения вышеуказанных задач идентификации объектов в разнообразных областях применения.

Табл. 1. Наборы данных, используемые для составления базы знаний

Параметр	Предлагаемое решение	Robo Flow	H2O AutoML	Azure Custom Vision	Google AutoML Vision	Fedot Industrial	LAMA
Поиск оптимальной модели	+	+	+	+	+	+	+
Решение задач распознавания объектов	+	+	+	+	+	+	-
Открытый исходный код	+	-	-	-	-	+	+
Возможность настройки модели и набора данных	+	-	+	+	+	+	+
Возможность офлайн-использования	+	-	-	-	-	+	+
Веб-интерфейс	+	+	+	+	+	-	+
Страна	РФ	США	США	США	США	РФ	РФ

2.1. Общая схема формирования рекомендаций

Процесс формирования рекомендаций начинается с анализа исходных данных пользователя и его требований. Данные преобразуются в список параметров, который затем обрабатывается алгоритмом DLAS-RA (рис. 1). Алгоритм использует продукционные правила для формирования рекомендаций, которые подбираются на основе знаний, полученных из предыдущих экспериментов.



Рис. 1. Общая блок-схема алгоритма формирования рекомендаций

Подробная схема описания алгоритма представлена на сайте проекта (<https://saaresearch.github.io/>) и в репозитории [12].

2.2. Механизм продукционных правил

Исследовательский опыт отражен в параметрах задачи детектирования объектов и набора данных, а также результатах ее решения: выбранная модель, значения заданных метрик. Иными словами, имеются причинно-следственные связи «какой результат вытекает, если ...». Поэтому предлагаемый метод основан на логике продукционных правил. Данные правила определяют связь между свойствами задачи, набора

данных, диапазонов метрик качества и рекомендациями в выборе модели. Для формирования условной части правила используется перечень параметров, а в качестве выхода выступает модель машинного обучения, отвечающая входному списку параметров (алгоритм 1).

```

Если (
    (<параметр 1> = <значение 1>) и
    ...
    (<параметр M> = <значение M>)
)
то
    (модель обнаружения объектов)
  
```

Алгоритм 1. Шаблон продукционного правила

Множество таких правил является базой знаний, на основе которой формируется рекомендация – подбирая наиболее подходящие правила в зависимости от требований и набора данных в задаче детектирования. Процесс составления таких правил для решения задач выбора моделей машинного обучения распознавания движущихся объектов описан в пункте 3.2.

2.3. Алгоритм рекомендаций DLAS-RA

2.3.1. Схема алгоритма

Алгоритм рекомендации архитектуры глубокого обучения (DLAS-RA – Deep Learning Architecture Selection Recommendation Algorithm) основан на интеграции параметров набора данных X (1), аннотации изображений B (2) и требований пользователя U (3) для формирования рекомендации по выбору моделей машинного обучения (глубокого обучения).

X определяется кортежем параметров набора данных (1):

$$X(s, b, d, br, e, wb, nc, ni), \quad (1)$$

где:

s – размер фотографии в наборе (ширина и высота);
 b – коэффициент баланса распределения изображений;
 d – количество уникальных цветов в изображении и имеет ограниченное число 256;
 br – средняя интенсивность пикселей в градациях серого, показывает общую яркость изображения;
 e – энтропия изображений в наборе показывает степень неупорядоченности распределения цветовых пикселей (энтропия измеряется в битах на пиксель и представляет степень случайности распределения цветовых пикселей)[13];
 wb – баланс белого;
 nc – количество классов, представленных в наборе;
 ni – количество изображений в наборе.

Зададим параметры аннотаций B , которые определяются лейблами для изображений в наборе:

$$B(s, ai, ar, no), \quad (2)$$

$s(min, max, avg)$ – размер рамки параметров определяют размеры объектов на изображении (ширина и высота ограничивающей рамки);
 ai – усредненное количество пересечений рамок;
 ar – усредненное соотношение высоты и ширины рамок;
 $no(min, max, avg)$ – количество объектов на одном изображении.

U определяется кортежем параметров требований пользователя (3):

$$U(a, s, r), \quad (3)$$

a – ожидаемая точность;
 s – ожидаемая скорость;
 r – режим работы (GPU – CPU).

В качестве метрик качества представлены: средний показатель $mAP@0,5$ учитывает качество обнаружения для всех классов и является одной из ключевых метрик для оценки эффективности алгоритмов обнаружения объектов. Метрика $mAP@0,95$ учитывает качество обнаружения объектов для каждого класса с высоким порогом IoU (Intersection over Union) в 0,95. Это позволяет оценить точность обнаружения объектов с более строгими критериями сравнения границ. Эти метрики позволяют оценить, насколько точно и полно модель распознает объекты, а также насколько она достоверно классифицирует их. Для метрики скорости работы оценивали время выполнения модели на графическом процессоре (GPU) и центральном процессоре (CPU), чтобы определить, какая модель обладает наилучшей производительностью с точки зрения скорости обработки данных. Кроме того, мы учитывали частоту кадров в секунду (FPS), которая указывает на количество кадров, обрабатываемых моделью в единицу времени.

На основе векторов X, B, U формируется массив параметров, на основе которого формируется рекомендация, а именно выбирается наиболее подходящее правило из базы знаний. В качестве алгоритм выбора нужного правила используется метод k -ближайших соседей (рис. 2).

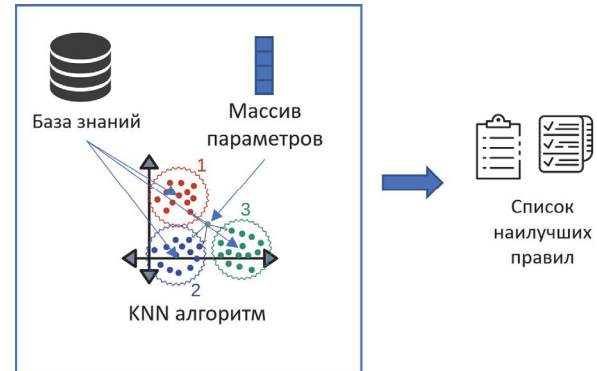


Рис. 2. Общая схема алгоритма DLAS-RA

2.3.2. Математическое описание алгоритма

Математически алгоритм формулируется следующим образом.

В состав входных данных алгоритма включены характеристики входного набора данных (массив изображений) X_{in} , B_{in} и требования пользователя U_{in} . Обозначим такие данные через C_{in} :

$$C_{in} = (X_{in}, B_{in}, U_{in}). \quad (4)$$

Обозначим банк продукционных правил:

$$BPR = \{R_1, R_2, \dots, R_n\}, \quad (5)$$

где BPR (5) – банк продукционных правил, R_j – j -е, продукционное правило, $j \in 1, n$. Каждое правило R_j сформировано для вектора $C_j = (X_j, B_j, U_j)$. Иными словами, каждое правило получено путем применения соответствующего конвейера машинного обучения с архитектурой (модели) глубокого обучения ML_j . Применим алгоритм kNN (k -ближайших соседей) для поиска оптимального подходящего правила из банка. Чтобы использовать указанный алгоритм, для каждого правила вычисляется метрика близости к заданным данным C_{in} (4): $\delta(C_{in}, C_j)$ – расстояние между векторами C_{in} и C_j в Евклидовом пространстве. Результат работы алгоритма DLAS-RA формулируется через отображение: $kNN(k): C_{in} \rightarrow (ML_{out}, RS)$, где $kNN(k)$ – алгоритм k ближайших соседей (k – параметр алгоритма – число выбираемых «лучших» соседей); ML_{out} – выбранная архитектура глубокого обучения с точки зрения $kNN(k)$; $RS = \{R_{i_1}, \dots, R_{i_k}\}$ – выбранные с применением алгоритма $kNN(k)$ «лучшие» k соседи, $R_{i_l} \in BPR$, $l \in \overline{1, k}$.

Обратим внимание, кроме лучшей с точки зрения $kNN(k)$ архитектуры глубокого обучения, имеем список правил RS , которые соответствуют ближайшим к $C_{in} = (X_{in}, U_{in})$ векторам по метрике δ . Это означает, что, кроме указанного алгоритмом архитектуры ML_{out} ,

может быть одна или несколько архитектур $ML_{il} = \{ML_{i1}, \dots, ML_{ik}\}$, $l \in 1, k$, которые соответствуют более близким к C_{in} векторам C_{il} по отношению к C_{out} . Здесь C_{out} – вектор, соответствующий рекомендованной архитектуре глубокого обучения ML_{out} , C_{il} – вектор, соответствующий архитектуре, следуемой по правилу $R_{il} \in RS$, $l \in 1, k$. Таким образом, кроме основной архитектуры ML_{out} , полученной с помощью $kNN(k)$, мы можем предложить альтернативные архитектуры, соответствующие в том числе более близким к C_{in} векторам. Окончательное решение по выбору архитектуры принимает разработчик.

3. Проведение экспериментальных исследований и результаты

Для оценки эффективности разработанной методики мы провели эксперимент, которые состоит из двух стадий. На первой стадии мы вывели продукционные правила, опираясь на НД, которые приведены ниже. Вторая стадия была нацелена на оценку эффективности методики на нескольких режимах: когда по заданному НД и значениям параметров возможно подобрать однозначную модель (одно правило с весом единица) или когда приходится подбирать несколько правил с заданными весами и давать рекомендации как по отдельному, так и совместному их применению. Чтобы реализовать вышеобозначенные шаги, был разработан виртуальный экспериментальный стенд (ВЭС) в форме репозитория GitHub [12].

3.1. Программная реализация. Фреймворк ODRS

Для обеспечения более удобного и эффективного взаимодействия с пользователем нами был реализован фреймворк ODRS (Object Detection Recommendation System). В основе фреймворка лежит предложенный метод автоматического формирования рекомендаций. Инструменты фреймворка позволяют

получать конвейеры МО по заданным наборам данных и параметрам пользователя, а также обучать предложенные модели по средствам API. Для взаимодействия в ODRS доступен веб-интерфейс, представляющий собой удобную платформу, где пользователи могут загружать размеченные наборы данных и получать подробные рекомендации с соответствующей информацией [14]. В качестве примера получения рекомендаций через API фреймворка ODRS были проведены запуски программных сценариев на наборе данных Aerial Maritime Drone Dataset [15] (снимки с дронов). Полный эксперимент представлен в разделе Wiki проекта ODRS на GitHub [12].

3.2. Формирование базы знаний – набора продукционных правил (шаг 1)

Для составления базы знаний рекомендаций были выбраны различные открытые бенчмарки (табл. 2), которые представляют собой стандартные наборы данных и метрики для оценки производительности моделей рекомендаций. Для проведения экспериментов были подобраны наборы данных, оптимально подходящие для решения разнообразных задач в данной области исследования. В основном уделено внимание средним (до 15000 изображений) и небольшим (до 1000 изображений) наборам данных, учитывая, что у большинства пользователей может отсутствовать достаточное количество ресурсов для формирования крупного НД. В таблице представлен фрагмент списка выбранных наборов данных, а также их краткие характеристики.

Оценка моделей производилась на основе различных критериев, представленных в п. 2.1.

Мы также использовали современные и высокоэффективные модели машинного обучения семейства YOLO, RNN-сети и сеть SSD. Результаты экспериментов для набора №1 (табл. 3) представлены в табл. 2, полная версия таблицы экспериментов и ссылки на наборы доступны в нашем репозитории [12].

Табл. 2. Наборы данных, используемые для составления базы знаний

Название набора	Краткое описание	Характеристики данных			
		Размер	Баланс	Количество классов	Количество изображений
WARP	Набор промышленных изображений твердых коммунальных отходов	1920×1080	94	28	5948
Aerial-maritime	Размеченные снимки доков, лодок, подъемников, автомобилей и других объектов с беспилотного летательного аппарата	800×600	64	5	1016
Food	Размеченные изображения еды	256×256	85	7	802
Plantdoc	Снимки поверхностей больных и здоровых растений	416×416	95	30	2567
Website Screenshot	Скриншоты с различных популярных вебсайтов с размеченными элементами (картинки, кнопки, текст и другие)	1024×768	77	8	4824

Табл. 3. Результаты экспериментов по обучению моделей МО на выбранных НД

Модель	Набор_№	mAP_50	mAP_95	FPS_GPU	FPS_CPU
yolov5l	1	0,503	0,386	50	4
yolov5m	1	0,444	0,333	61	7
yolov5n	1	0,452	0,321	80	18
yolov5s	1	0,52	0,402	79	13
yolov5x	1	0,512	0,404	42	2
yolov7x	1	0,555	0,435	15	4
yolov7	1	0,53	0,41	15	6
yolov7-tiny	1	0,473	0,344	14	12
yolov8x6	1	0,536	0,433	35	2
yolov8x	1	0,478	0,388	40	2
yolov8s	1	0,51	0,382	72	11
yolov8n	1	0,466	0,351	75	17
yolov8m	1	0,512	0,409	57	5
Faster-vgg16	1	0,461		29	4
SSD	1	0,467	0,435	45	4

По итогам проведения экспериментов из данных о наборах и метриках была сформирована база знаний, содержащая сведения о метриках качества моделей обнаружения объектов на различных наборах данных. На основе данных из базы знаний был сформулирован набор из 75 правил для составления рекомендации выбора моделей. Продукционные пра-

вила позволяют полу- и (или) автоматически выбирать наилучшую модель для конкретной задачи на основе имеющихся данных и «банка» моделей. Полный список правил представлен в нашем репозитории. Пример фрагмента некоторых правил, составленных на основе набора данных №1, представлен в табл. 4.

Табл. 4. Пример продукционных правил, сформированных на основе базы знаний

Размер	Баланс	Количество классов	Количество изображений	mAP50	mAP_95	FPS_GPU	FPS_CPU	Модель
1920×1080	94	28	5948	0,512	0,404	42	2	Yolov5x
1920×1080	94	28	5948	0,555	0,435	15	4	Yolov7x
1920×1080	94	28	5948	0,53	0,41	15	6	Yolov7
1920×1080	94	28	5948	0,473	0,344	14	12	Yolov7-tiny
1920×1080	94	28	5948	0,536	0,433	35	2	Yolov8x6

Для обучения наилучшего классификатора KNN было произведено несколько экспериментов. Для достижения наилучшего результата набор правил был разделен на две подвыборки – обучающую и тестовую, соответственно составляющие 80 % и 20 % от общего объема данных. Кроме того, было произведено создание дополнительных 30 правил на основе всего набора данных с целью расширения обучающего контекста для повышения точности алгоритма предсказания. Эксперименты проводились на ЭВМ с видеокартой NVIDIA A5000 и центральным процессором 12th Gen Intel(R) Core(TM) i3-12100, с размером батча обучения 32. В качестве признаков были выбраны атрибуты, выделенные желтым цветом (табл. 3), а в качестве предиктора выбрано поле модель. В процессе обучения алгоритма KNN удалось достичь высокого значения точности в размере 89 % на тестовой выборке для рекомендательного алгоритма. Этот результат свидетельствует о хорошей способности алгоритма предоставлять рекомендации, точно соответствующие предпочтениям пользователей или другим критериям, используемым в процессе обучения.

3.3. Оценка работы метода выбора моделей машинного обучения (шаг 2)

Чтобы оценить качество сформированных правил, был выбран набор условий, каждый из которых содержал изображения и аннотации из заданного НД и список соответствующих параметров требований к искомой модели. В данной серии нет условий, которые участвовали в формировании продукционных правил. Для каждого условия с помощью предложенного алгоритма были получены рекомендации в виде списка наиболее подходящих моделей. Модели из списка рекомендаций сравнивались со всеми моделями из базы знаний для оценки эффективности рекомендаций. Эксперименты проводились на расширенном наборе данных WaRP, содержащем 25000 размеченных изображений для 27 классов объектов твердых коммунальных отходов (далее – ТКО) и собранном совместными усилиями компаниями Insystem и AIRI, а также Университетом ИТМО [16 – 18]. Набор данных представлен изображениями отсортированного мусора с различных сортировочных заводов из Ленинградской и Московской областей, а также из Ставропольского края.

Целью эксперимента было получить рекомендации при помощи предложенной в статье методики, после чего сравнить их с обученными на расширенном наборе данных WaRP моделями, использованными при формировании продукционных правил, для оценки эффективности методики в решении реальной задачи. Первый эксперимент заключался в поиске наиболее точной модели распознавания (по метрике mAP50), а второй – в поиске компромиссной максимально быстрой модели на GPU (по метрике FPS_GPU), имеющей высокую точность (по метрике mAP50).

Эксперимент 1: на вход системе был предоставлен расширенный набор данных WaRP и конфигурационный файл с требованиями максимальной точности. Основной целью эксперимента было получение наиболее точной модели. В результате работы рекомендательной системы были получены следующие модели: YOLOv5-x, YOLOv8-x, YOLOv5-m.

Эксперимент 2: мы предоставили системе конфигурационный файл, в котором были установлены требования к максимальной скорости и высокой точности. Целью данных экспериментов было определение наилучшей компромиссной модели, которая сочетает в себе высокую скорость работы. В результате работы рекомендательной системы были получены следующие модели: YOLOv7-tiny, YOLOv8-n, YOLOv5-s.

Далее были проведены подсчеты точности и скорости для всех моделей.

Результаты экспериментов показаны на графике (рис. 3). Синим цветом выделены наиболее точные модели, а зеленым – наиболее быстрые.

По результатам проведенных экспериментов наиболее эффективной моделью является YOLO v5-x – по полученным значениям метрик она продемонстрировала высокую скорость и точность распознавания объектов. На втором месте по эффективности оказалась модель YOLO v5-m, на третьем месте – YOLO v8-x. Из предсказаний системы и проведенных экспериментов следует, что система действительно выдает обоснованные начальные приближения в выборе МО для обнаружения объектов. Полученные данные были использованы для пополнения базы знаний рекомендательной системы. Из сравнительного графика (рис. 4) вытекают следующие выводы. При выборе наиболее быстрой модели рекомендательная система показала высокие результаты. Действительно, были выбраны наиболее оптимизированные модели с учетом скорости работы. Однако при рассмотрении баланса между скоростью и точностью результаты экспериментов показали, что модель YOLOv8s является наиболее подходящей для обеспечения высокой точности предсказаний.

3.4. Дискуссия

Для оценки эффективности разработанного метода были проведены эксперименты на различных открытых и полуоткрытых платформах, предоставляющих

возможность создания моделей для обнаружения объектов. В качестве платформ для исследования были выбраны Roboflow, на которой зарегистрировано более 250 тысяч пользователей, и Fedot Industrial, насчитывающая свыше 1000 пользователей. Дополнительно эксперименты проводились с использованием набора данных WaRP. На графике (рис. 4) представлены результаты, демонстрирующие точность и скорость моделей, предложенных этими фреймворками.

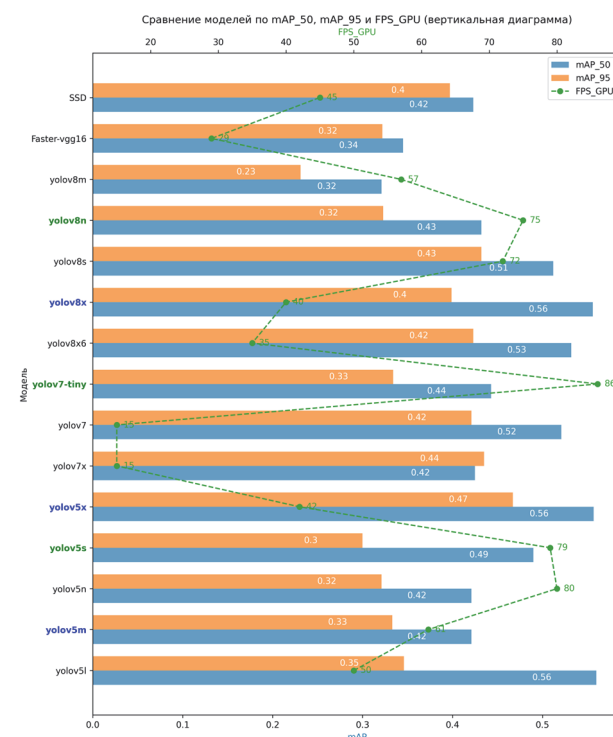


Рис. 3. Сравнительный график моделей глубокого обучения по скорости и точности

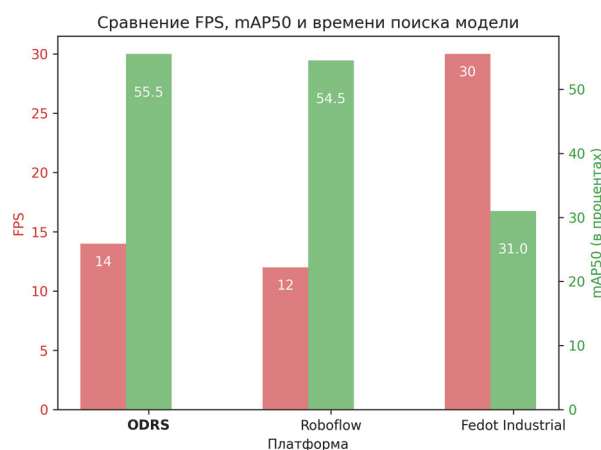


Рис. 4. Сравнительный график AutoML фреймворков по скорости, точности и времени поиска

Фреймворки ODRS и Roboflow продемонстрировали схожие результаты как по скорости, так и по точности модели. В свою очередь, фреймворк Fedot Industrial позволил найти модель с высокой скоростью, но существенно уступающую по точности. На основе

проведённых экспериментов платформа ODRS показала конкурентоспособные результаты среди современных открытых платформ.

Заключение

В рамках данной работы предложен новый метод автоматизированного выбора моделей машинного обучения для детектирования объектов на изображениях, основанный на механизме продукционных правил. Основное преимущество предложенного подхода заключается в его способности интегрировать накопленные знания о существующих архитектурах нейросетей и их производительности на различных наборах данных. Это позволяет значительно сократить время на подбор оптимальных моделей и повысить точность решений, что особенно актуально для задач реального времени.

Ценность данной работы заключается в разработке системы рекомендаций, использующей продукционные правила, которая адаптируется под конкретные требования пользователя и набор данных. В отличие от существующих решений, данная система не только предоставляет рекомендации по выбору моделей, но и предлагает конвейеры для автоматизации их обучения и использования, что делает ее уникальной в контексте применения в компьютерном зрении.

Практическая значимость результатов проявляется в возможности применения предложенной системы в различных областях, таких как промышленная автоматизация, медицинская диагностика, системы безопасности и интеллектуальные транспортные системы. Благодаря возможности гибкой настройки и работы в офлайн-режиме, система может быть внедрена в условиях ограниченных ресурсов, что повышает ее ценность для широкого круга пользователей. Планы на будущее включают расширение базы данных продукционных правил за счет добавления новых наборов данных, таких как медицинские изображения и изображения дронов для мониторинга окружающей среды. Кроме того, планируется расширить набор параметров для формирования рекомендаций с использованием больших языковых моделей, которые помогут извлекать текстовые описания изображений и расширять пространство анализируемых признаков.

Благодарности

Данная работа была поддержана Министерством образования и науки, договор FSER-2024-0004.

References

- [1] LeCun Y, Bengio Y, Hinton G. Deep learning. *Nature* 2015; 521(7553): 436-444. DOI: 10.1038/nature14539.
- [2] Butz MV. Towards strong AI. *KI – Künstliche Intelligenz* 2021; 35(1): 91-101. DOI: 10.1007/s13218-021-00705-x.
- [3] Adomavicius G, Tuzhilin A. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Trans Knowl Data Eng* 2005; 17(6): 734-749. DOI: 10.1109/TKDE.2005.99.
- [4] Tu NV, Cuong LA. A deep learning model of multiple knowledge sources integration for community question answering. *VNU Journal of Science: Computer Science and Communication Engineering* 2021; 37(1). Source: <<https://jcsce.vnu.edu.vn/index.php/jcsce/article/view/295/128>>. DOI: 10.25073/2588-1086/vnuesce.295.
- [5] Diwan T, Anirudh G, Tembhurne JV. Object detection using YOLO: challenges, architectural successors, datasets and applications. *Multimed Tools Appl* 2023; 82(6): 9243-9275. DOI: 10.1007/s11042-022-13644-y.
- [6] Chen W, et al. YOLO-face: a real-time face detector. *Vis Comput* 2021; 37(4): 805-813. DOI: 10.1007/s00371-020-01831-7.
- [7] Liu W, et al. SSD: Single shot multibox detector. In Book: Leibe B, Matas J, Sebe N, Welling M, eds. *Computer vision – ECCV 2016. 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I*. Cham, Switzerland: Springer International Publishing AG; 2016: 21-37. DOI: 10.1007/978-3-319-46448-0_2.
- [8] He K, et al. Mask R-CNN. *IEEE Trans Pattern Anal Mach Intell* 2020; 42(2): 386-397. DOI: 10.1109/TPAMI.2018.2844175.
- [9] Chaudhuri A. Hierarchical modified fast R-CNN for object detection. *Informatica (Slovenia)* 2021; 45(7): 67-82. DOI: 10.31449/inf.v45i7.3732.
- [10] Chen W, et al. A review of object detection: Datasets, performance evaluation, architecture, applications and current trends. *Multimed Tools Appl* 2024; 83(24): 65603-65661. DOI: 10.1007/S11042-023-17949-4.
- [11] He X, Zhao K, Chu X. AutoML: A survey of the state-of-the-art. *Knowl Based Syst* 2021; 212: 106622. DOI: 10.1016/j.knosys.2020.106622.
- [12] saaresearch/ODRS 2025. Source: <<https://github.com/saaresearch/ODRS>>.
- [13] Chen X, et al. No-reference color image quality assessment: from entropy to perceptual quality. *EURASIP J Image Video Process* 2019; 2019(1): 77. DOI: 10.1186/s13640-019-0479-7.
- [14] saaresearch/ODRS_WEB. 2025. Source: <https://github.com/saaresearch/ODRS_WEB>.
- [15] Aerial Maritime Drone. 2025. Source: <<https://www.kaggle.com/datasets/ammarnas-sanalhajali/aerial-maritime>>.
- [16] InSystem – Intelligent Systems AI. 2025. Source: <<https://insystem.io/>>.
- [17] iTMO. 2025. Source: <<https://en.itmo.ru/>>.
- [18] AIRI. AIRI Institute. 2025. Source: <<https://airi.net/?force=en>>.

Сведения об авторах

Сметанин Артём Алексеевич, 1998 года рождения, окончил бакалавриат в 2020 по направлению «Информатика и вычислительная техника», окончил магистратуру в 2022 году по направлению «Прикладная математика и информатика» в университете ИТМО. В настоящий момент аспирант и инженер университета ИТМО, область научных интересов включает компьютерное зрение, глубокие нейронные сети и рекомендательные системы. E-mail: artem_smetanin@niuitmo.ru

Духанов Алексей Валентинович, д.т.н., доцент, 1977 года рождения, окончил в 1999, 2001 Владимирский государственный университет по специальностям «Прикладная математика и физика» и «Финансовый менеджмент». В 2016 года прошел профессиональную переподготовку в Московской школе управления «Сколково» по программе «Лидеры изменений для глобальных университетов»; кандидатская диссертация была защищена в 2003 году по специальности 05.13.12 Системы автоматизации проектирования, докторская – в 2016 году по специальности 05.13.06 Автоматизация и управление технологическими процессами и производствами. В настоящее время работает в Университете ИТМО на должности ведущий научный сотрудник Исследовательского центра «Сильный искусственный интеллект в промышленности», доцентом факультета цифровых трансформаций. Область научных интересов: искусственный интеллект и объектно-ориентированный подход в образовании, в исследовании операций, компьютерном зрении. E-mail: dukhanov@itmo.ru

Герасимчук Михаил Юрьевич, 2000 года рождения, окончил бакалавриат в 2022 по направлению «Информатика и вычислительная техника», окончил магистратуру в 2024 году по направлению «Программная инженерия» в университете ИТМО. В данный момент инженер университета ИТМО, область научных интересов включает метаалгоритмы с применением машинного обучения, интеллектуальный анализ растровых изображений, оптимизация нейронных сетей, интеллектуальные рекомендательные системы. E-mail: mishagerasimchuk@itmo.ru

ГРНТИ: 28.23.29

Поступила в редакцию 02 июля 2024 г. Окончательный вариант – 03 октября 2024 г.

Development of a methodology for selecting machine learning models for recognizing moving objects in a video stream based on production rules

A.A. Smetanin¹, A.V. Dukhanov¹, M.Y. Gerasimchuk¹

¹National Research University of ITMO,
Kronverksky pr-t 49, St. Petersburg, 197101, Russia

Abstract

To date, technologies in the field of photo-video data processing, algorithms for recognizing and classifying objects in images, have become more accurate, often exceeding the accuracy threshold of 90%. Such technological breakthroughs have led to extensive use of these innovations for professional and personal purposes. However, the variation of factors such as the conditions for obtaining and processing images, the dynamics and deformation of objects in the frame, can complicate the practical application of these algorithms. The presented scientific research focuses on the development of a recommendation system for choosing optimal machine learning models in order to solve a wide range of tasks related to object recognition. The principle of forming recommendations is generated on the basis of production rules, which are developed taking into account experimental data and analysis of academic sources. The result of the proposed system is not just a list of models indicating their relevance, but also proposals for the creation of machine learning pipelines and recommendations for the installation and use of appropriate program libraries. The current article describes a methodology for automating the formation of recommendations, including a priori estimation of metric values in the context of object classification tasks in images.

Keywords: machine learning algorithms, object recognition, recommendation system, production rules, machine learning pipelines.

Citation: Smetanin AA, Dukhanov AV, Gerasimchuk MY. Development of a methodology for selecting machine learning models for recognizing moving objects in a video stream based on production rules. *Computer Optics* 2025; 49(5): 826-834. DOI: 10.18287/2412-6179-CO-1583.

Acknowledgements: This research is financially supported by the Ministry of Science and Higher Education, agreement FSER-2024-0004.

Author's information

Artem Alekseevich Smetanin, (b. 1998), graduated with a bachelor's degree in 2020 in Computer Science and Computer Engineering, graduated with a master's degree in 2022 in Applied Mathematics and Informatics at ITMO University. Currently, he is a graduate student and engineer at ITMO University, his research interests include computer vision, deep neural networks and recommendation systems. E-mail: artem_smetanin@niuitmo.ru

Alexey Valentinovich Dukhanov, D.Sc., Associate Professor, born in 1977, graduated in 1999 and 2001 from Vladimir State University with degrees in Applied Mathematics and Physics, and Financial Management (specialist). In 2016, he completed professional retraining at the Moscow School of Management Skolkovo under the program "Leaders of Changes for Global Universities". His Ph.D. thesis was defended in 2003 in the field of computer aided design systems (specialty 05.13.12), and his doctoral thesis was defended in 2016 in the field of Automation and Control of Technological Processes and Productions (specialty 05.13.06). Currently, he works at ITMO University as a lead researcher at the Research Center "Strong Artificial Intelligence in Industry" and as an associate professor at the Faculty of Digital Transformations. His research interests include artificial intelligence and object-oriented approach in education, operations research, and computer vision. E-mail: dukhanov@itmo.ru

Mikhail Yurevich Gerasimchuk, (b. 2000), graduated with a bachelor's degree in 2022 in Computer Science and Computational Engineering, graduated with a master's degree in 2024 in Software Engineering at ITMO University. Currently an engineer at ITMO University, his research interests include meta-algorithms using machine learning, intelligent analysis of raster images, optimization of neural networks, intelligent recommendation systems. E-mail: mishagerasimchuk@itmo.ru

Received July 02, 2024. The final version – October 03, 2024.