



ИСКУССТВЕННЫЙ ИНТЕЛЛЕКТ И КОМПЬЮТЕРНОЕ ЗРЕНИЕ. РЕАЛИЗАЦИЯ ПРАКТИКО-ОРИЕНТИРОВАННОГО ПОДХОДА

К.В. Розов

Новосибирский государственный педагогический университет, Россия

Аннотация. В статье рассмотрены особенности реализации практико-ориентированного подхода к построению учебных заданий для освоения технологий искусственного интеллекта, в частности, компьютерного зрения. Предлагается применение упражнений в форме мини-проектов, предполагающих использование изученного ранее материала для решения относительно небольших практических задач. Для успешного выполнения такого упражнения студенту необходимо не только выбрать подходящие методы и средства, но и задействовать свои навыки программирования. Приведены примеры разработанных автором заданий по теме «Компьютерное зрение», включенных в учебный курс «Технологии искусственного интеллекта в образовании» для студентов Новосибирского государственного педагогического университета, чьи профили подготовки связаны с информационными технологиями.

Ключевые слова. Технологии искусственного интеллекта, компьютерное зрение, практико-ориентированное обучение, мини-проекты, профессиональная подготовка.

ARTIFICIAL INTELLIGENCE AND COMPUTER VISION. IMPLEMENTATION OF A PRACTICE-ORIENTED APPROACH

K. V. Rozov

Novosibirsk State Pedagogical University, Russia

Abstract. The article discusses the features of implementing a practice-oriented approach to constructing educational tasks for studying artificial intelligence technologies, in particular computer vision. It is proposed to use exercises in the form of mini-projects, which involve the use of previously studied material to solve relatively small practical problems. To successfully complete such an exercise, the student must not only choose appropriate methods and tools, but

also use his programming skills. Examples of tasks developed by the author on the topic “Computer Vision” are given, which are included in the training course “Artificial Intelligence Technologies in Education” for students of the Novosibirsk State Pedagogical University, whose training profiles are related to information technology.

Keywords. Artificial intelligence technologies, computer vision, practice-oriented training, mini-projects, professional training.

Введение

Современный мир претерпевает быстрые изменения в различных областях науки и технологий, и образование не остаётся в стороне от этого процесса. Педагогические вузы стремятся адаптироваться к новым вызовам, предоставляя возможность студентам получить актуальные знания и компетенции, необходимые для успешной карьеры в современном информационном обществе.

Одной из ключевых областей искусственного интеллекта (ИИ) является компьютерное зрение, достижения в которой вносят всё больший вклад в развитие экономики, промышленности, медицины и т.д. Технологии компьютерного зрения трансформируют наше представление о возможностях компьютеров и машинного обучения, и, как показывает опыт автора, вызывают интерес у обучающихся, как школьников, так и студентов, к изучению ИИ в целом. Вероятно, это происходит потому, что, как и другая широкая область ИИ – обработка естественного языка, включающая распознавание и синтез речи, компьютерное зрение формирует более живой, интеллектуальный образ компьютера в глазах человека. Компьютерное зрение в педагогике информатики, по мнению автора, может выступать не только объектом изучения, но и средством обучения программированию, которое, подобно разработке видеоигр, может эффективно способствовать формированию и развитию учебной мотивации за счёт высокой наглядности и интерактивности. Обучающийся, выполняющий практическую работу по компьютерному зрению или обработке естественного языка, будто оживляет компьютер своими руками, при этом получаемые результаты обладают высокой наглядностью. В то же время практическое знакомство с технологиями ИИ показывает значимость естественного интеллекта – современному ИИ при всех его преимуществах недостаёт понимания контекста происходящих вокруг событий, эмпатии, способности к распознаванию невербальных сигналов и др., особенно если речь идёт о какой-либо единой «интеллектуальной» компьютерной системе, а не об отдельных методах и их программных реализациях.

Исследователями и действующими преподавателями вузов справедливо отмечается противоречие «между необходимостью освоения технологий искусственного интеллекта, в частности, компьютерного зрения, школьниками, учителями, студентами, преподавателями вузов и недостатком методических материалов для этого освоения» [1, с. 127]. В статье будут рассмотрены практические аспекты внедрения технологий компьютерного зрения в учебный процесс педагогического вуза в качестве объекта изучения и средства обучения программированию. Цель работы – предложить рекомендации и ресурсы (примеры учебных задач), которые могут помочь педагогическим университетам эффективно внедрять этот важный элемент современного образования в профессиональную подготовку студентов профилей, связанных с информационными технологиями.

Содержание предметной подготовки студентов педагогического вуза в области компьютерного зрения

В Новосибирском государственном педагогическом университете осуществляется подготовка инженерных и педагогических кадров физико-математических и информационно-технологических профилей в области ИИ в рамках дисциплины «Технологии искусственного интеллекта в образовании». Автор статьи апробировал представленные ниже материалы по теме «Компьютерное зрение» в процессе работы с группами студентов 2-3 курсов следующих направлений и профилей подготовки: 09.03.02 Информационные системы и технологии, Информационные системы и технологии в образовании; 44.03.05 Педагогическое образование (с двумя профилями подготовки), Информатика и Иностранный (английский) язык.

С использованием произвольных статических изображений в формате .jpg/.png и веб-камер студенты изучают возможности библиотек алгоритмов компьютерного зрения: OpenCV, MediaPipe и др. Учебные материалы размещены в электронном курсе преподавателя в системе управления обучением Moodle и представляют собой иллюстрированный практикум в текстовом формате, последовательно раскрывающий назначение и параметры изучаемых функций и методов. Студентам предлагается пошагово использовать фрагменты кода из представленных листингов и экспериментировать со значениями параметров. Таким образом обучающиеся формируют необходимую базу знаний и приобретают навыки написания программ на языке Python. Преподаватель акцентирует внимание как на соблюдении общепринятых правил оформления кода, так и на структурах данных, формируемых и возвращаемых той или иной функцией (методом), и интерпретации содержащейся в них информации.

Примеры основного содержания учебных материалов по OpenCV:

- загрузка, вывод и сохранение изображения;
- работа с размерами и фрагментами изображения;
- компоненты цвета, цветовые модели, цветовые каналы изображения;
- методы размытия изображения;
- поиск и выделение объектов по цвету и контуру;
- обнаружение людей с помощью дескриптора признаков гистограммы направленных градиентов (HOG) и метода опорных векторов (SVM);
- обнаружение объектов с помощью каскадов Хаара;
- обучение каскадного классификатора Хаара;
- применение глубоких нейронных сетей (модуль DNN)
 - обнаружение лиц с помощью res10_300x300_ssd
 - прогнозирование класса объекта с помощью DenseNet-121
 - классификация нескольких объектов с помощью MobileNet SSD
 - классификация нескольких объектов с помощью YOLOv5
- работа с видеопотоком (веб-камера, видеоролик);
- трекинг объектов;

Примеры основного содержания учебных материалов по MediaPipe:

- распознавание ключевых точек лица;
- распознавание ключевых точек кистей рук;
- распознавание ключевых точек тела человека.

Примеры учебных заданий для самостоятельной работы студентов

После знакомства с многообразием методов работы с изображениями и возможностями специализированных программных библиотек обучающимся предлагается выполнить ряд упражнений в форме мини-проектов. Термин «упражнение» здесь означает необходимость активного применения ранее полученных знаний и приобретённых умений для формирования устойчивых навыков без необходимости поиска дополнительной информации за пределами учебных материалов курса. При этом поисковая деятельность имеет место, но заключается прежде всего не в поиске дополнительных методов и средств (что не запрещается), а в поиске способов действия и решения возникающих проблем. Под «мини-проектом» понимается проектная деятельность с существенным ограничением по времени (примерно 1-2 занятия). Подобно полноценному проекту результатом является законченный продукт, в данном случае программный. При этом выполняются требования к современному проекту как к практической иллюстрации изученного материала, изложенные коллективом авторов в работе [2]:

критерий центральности – студенты знакомятся и изучают основные понятия дисциплины;

ведущий (основополагающий) вопрос – студенты сталкиваются с проблемами и ограничениями ИИ;

конструктивность исследования – студенты продумывают дизайн программы с точки зрения взаимодействия её структурных элементов, осуществляют принятие решений, поиск проблем, построение моделей, что трансформирует и конструирует их знания и навыки;

автономия – самостоятельность и ответственность студентов;

реализм – решение реальных задач (проблем).

Тем не менее, необходимо подчеркнуть, что при использовании упражнений в форме мини-проектов при изучении технологий ИИ, в том числе компьютерного зрения, на первый план выходит сам процесс работы над программой (проектом). Ему придаётся большее значение, чем итоговому результату, т.е. продукту.

Ниже представлены примеры разработанных автором упражнений.

1. Используя каскады Хаара или нейронную сеть для обнаружения лиц, написать программу для автоматического размывания всех обнаруженных лиц на фото и видео (рис. 1).

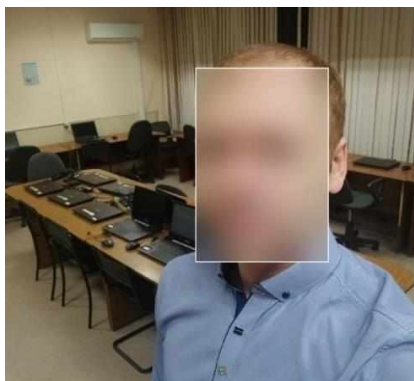


Рисунок 1 – Результат работы программы для автоматического размывания лиц

2. Написать программу для подсчёта количества треугольников, кругов и прямоугольников на изображении с учётом внутренних контуров. Каждой из разновидностей фигур назначить свой цвет. Вывести на экран (в окне) и сохранить в файл изображение с выделенными на нём фигурами с помощью контуров соответствующего цвета и указанным количеством обнаруженных треугольников, кругов и прямоугольников (рис. 2).

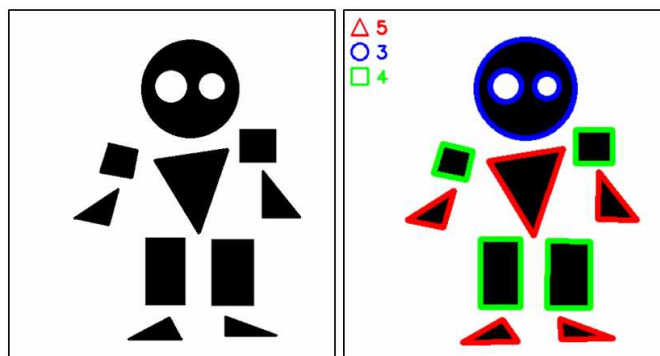


Рисунок 2 – Результат работы программы для подсчёта количества фигур

3. Написать программу для распознавания зелёного и красного цветов на фотографиях светофора. На изображении выводить название распознанного цвета и круг соответствующего цвета в области света светофора (рис. 3).

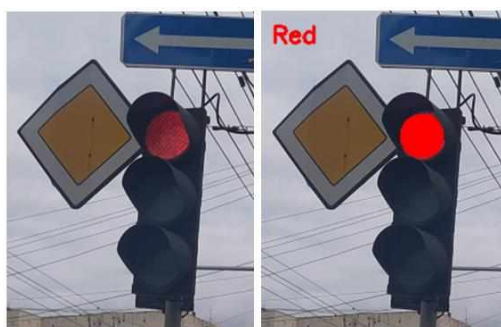


Рисунок 3 – Результат распознавания красного сигнала светофора

Если распознано несколько цветов, то выводить цвет, занимающий наибольшую площадь (рис. 4).



Рисунок 4 – Распознан и отмечен именно зелёный сигнал светофора

Для обнаружения геометрического центра скопления точек одного цвета используйте код для вычисления моментов изображения (новый материал):

```

moments = cv2.moments(filter_color, 1)
m01 = moments['m01']
m10 = moments['m10']
m00 = moments['m00']
if m00 > 100:
    x = int(m10 / m00)
    y = int(m01 / m00)

```

4. Написать программу для определения количества очков, которое выпало на верхней грани игрального кубика. На изображении выделить область верхней грани с помощью прямоугольной рамки, а над рамкой вывести количество очков (рис. 5). Каждая фотография содержит только один кубик (находящийся примерно на одном расстоянии от камеры), при этом кроме верхней грани видны одна или две других. Проверить корректность работы программы для всех шести граней. Для всех изображений должны применяться одни и те же настройки параметров для распознавания (вносить изменения в программу для каждой отдельной фотографии запрещено).

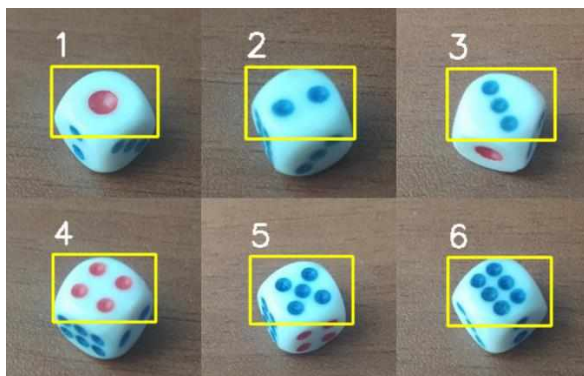


Рисунок 5 – Результат распознавания количества выпавших очков на кубике

5. Написать программу для автоматической сортировки изображений с различными классами объектов на них по отдельным каталогам (для каждого класса объектов свой каталог). Изображения берутся из одного общего каталога и содержат не менее 2-х классов для сортировки (например, автомобили и самолёты).

Для перебора изображений в заданном каталоге можно использовать модуль «os» для работы с операционной системой и следующий код:

```
# каталог с изображениями для сортировки
directory = 'Images/for_sort'
# перебор файлов
for filename in os.listdir(directory):
    # если файл имеет одно из перечисленных расширений
    if filename.endswith(('jpg', 'jpeg', 'png')):
        # загружаем изображение для дальнейшей обработки
        img = cv2.imread(directory+'/'+filename)
```

6. Написать программу для определения нарушений дистанции между прохожими (например, во время эпидемии) в реальном времени (в видеоролике). Для обнаружения людей в кадре использовать HOG и SVM детектор или нейронную сеть. Дистанцию между людьми, считающуюся нарушением, можно задать фиксированной (без учёта расстояния до камеры). Людей, не нарушающих и нарушающих дистанцию, соответственно выделять ограничивающими рамками различных цветов. С помощью

точки (закрашенного круга) выделять центр каждой рамки. Если люди находятся близко друг к другу, точки центров их рамок должны быть соединены линией (рис. 6).



Рисунок 6 – Результат обнаружения людей и нарушения дистанции в видеоролике

7. Написать программу с использованием трекера для рисования пути движения выбранного объекта (рис. 7). При нажатии на пробел очищать путь на экране.

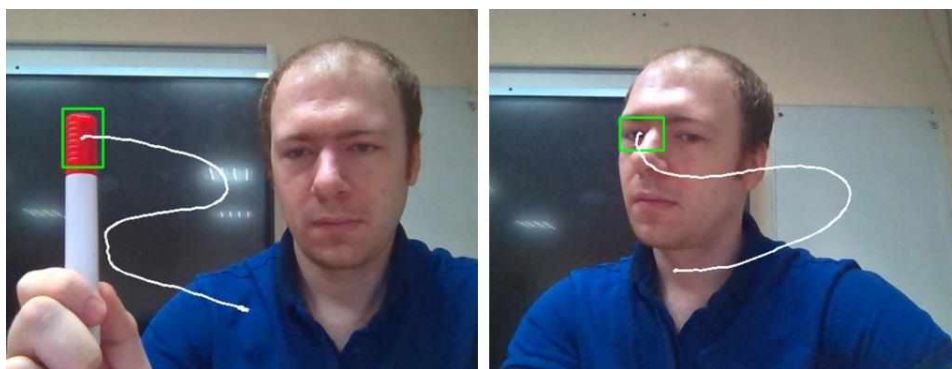


Рисунок 7 – Трекер отслеживает колпачок маркера (слева) и глаз (справа)

8. Написать программу для управления курсором компьютера с возможностью виртуальных щелчков левой кнопкой мыши при помощи пальцев руки (рис. 8). Для управления мышью используйте библиотеку PyAutoGUI.



Рисунок 8 – Средний палец служит для перемещения курсора, а сгибание указательного пальца для щелчка левой кнопкой мыши

9. Написать программу для управления курсором компьютера с возможностью виртуальных щелчков левой кнопкой мыши при помощи глаз.

Перемещение одного из глаз – движение курсора, моргание вторым глазом – щелчок мышью (рис. 9).

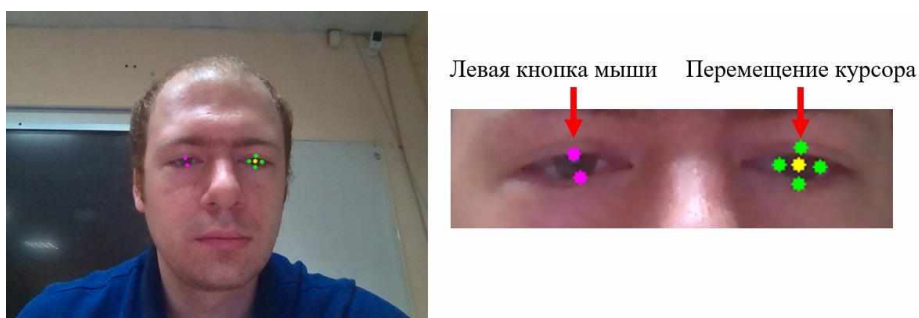


Рисунок 9 – Управление компьютерной мышью с помощью веб-камеры и глаз

Обсуждение результатов

Представленные выше учебные задания не требуют

- больших вычислительных мощностей, т.е. могут быть выполнены даже на сравнительно слабом персональном компьютере;
- загрузки и предобработки больших наборов данных (экономия места на диске и времени подготовки к работе);
- длительного ожидания для получения результата (например, обучение свёрточных нейронных сетей на больших наборах изображений, даже уже размеченных или рассортированных по каталогам, может занимать несколько часов).

Безусловно, сбор, обработка, разметка данных являются неотъемлемыми операциями, выполняемыми специалистами в области современных технологий ИИ для решения реальных задач. Именно «большие данные» и искусственные нейронные сети, обученные на них, позволили ИИ стать перспективным направлением развития современных компьютерных технологий и приносить огромную пользу в самых разных областях деятельности человека. Студентов важно познакомить с алгоритмами и средствами отбора, разметки данных, архитектурами нейронных сетей и средствами их построения (библиотеки TensorFlow, Keras, PyTorch и др.). Но в условиях ограниченного количества академических часов, выделенных на изучение ИИ в педагогическом вузе, по мнению автора, не следует строить все или большую часть практических работ на прохождении полного цикла разработки интеллектуальной системы, а сосредоточиться на разнообразии рассматриваемых методов и средств. Частично перечисленные недостатки практических заданий для студентов с применением построенных и обученных самостоятельно нейронных сетей может компенсировать наличие достаточно мощных компьютеров в вузе или работа в облачном сервисе Google Colab. Но с учётом времени «простоя», когда нейронная сеть обучается, а студент вынужден ждать, и её низкими обобщающими возможностями на выходе, если от этих временных затрат

отказаться (что не позволит раскрыть её как полезную технологию), выполнение таких заданий следует осуществлять вне аудиторных занятий.

Разработанные автором практико-ориентированные упражнения дают возможность обучающемуся на основе имеющихся у него «деталей» собрать полностью функциональную программу, решающую пусть и относительно небольшую, но реальную задачу за время, зависящее прежде всего от его собственных способностей и мотивации. Они направлены, с одной стороны, на повторение и закрепление пройденного материала, непосредственно связанного с технологиями ИИ, с другой, на развитие навыков программирования, углубление знаний о работе с различными структурами данных в Python и продуктивную/творческую деятельность. Каждое упражнение в форме мини-проекта ставит перед обучающимся проблему: как применить приобретённые мной знания и умения/навыки, чтобы получить итоговый продукт согласно заданным требованиям? Выход за рамки рассмотренного в учебном курсе материала при этом будет только приветствоваться. Дополнительный код в формулировке задания (представленный в примерах упражнений 3 и 5) можно скрывать в зависимости от уровня подготовки конкретного обучающегося (группы обучающихся), чтобы дать возможность полностью самостоятельно найти решение или способ успешно внедрить в свой код уже готовый фрагмент. Такие учебные задачи позволяют также оценить алгоритмическую культуру и культуру программирования студента (оптимальное построение алгоритма по количеству используемой памяти, количеству необходимых операций; логически обоснованное разделение программы на подпрограммы; присвоение осмысленных имен идентификаторам и др.), осуществлять их коррекцию.

Заключение

Важно отметить, что для учебной задачи, подобной представленным выше, сложно организовать автоматическую проверку, в отличие, например, от задачи на разработку модели машинного обучения и проверку точности её предсказательных способностей на наборе тестовых данных с выводом результата в текстовом (табличном) формате, характерной для олимпиад по ИИ или курсов повышения квалификации в онлайн-формате, где необходим строго формализованный подход к оценке результатов деятельности. Поэтому роль живого преподавателя в оценке работы студента, его компетентность оказываются очень значимыми при использовании учебных задач в проектной форме. Часто может оказаться более важной оценка самого подхода к решению задачи студентом, в том числе «умение отбирать из множества возможных вариантов оптимальные, наиболее приемлемые или просто удовлетворительные

теоретические или практические решения», что приоритетно при решении творческих технических задач [3, с. 30], а не качество работы программы здесь и сейчас с точки зрения точности распознавания объектов или иных критериев. Программный код, полученный в результате, может быть доработан в будущем и/или использован студентом в качестве части более крупного проекта по ИИ; использоваться им в профессиональной деятельности в качестве шаблона готового алгоритма или набора алгоритмов в процессе подготовки школьников к соответствующим олимпиадам и конкурсам, проектной деятельности по информатике. При этом, если речь идёт о подготовке будущего педагога для школы (основного и среднего общего образования), его код должен быть доступным для понимания не только учителем, но и его учеником.

Список библиографических ссылок (на языке оригинала)

1. Баранова Е.В., Солодкая М.А. Цифровой образовательный ресурс «Основы компьютерного зрения» для будущих учителей информатики. *Методика преподавания в современной школе: проблемы и инновационные решения*. Материалы российско-узбекского образовательного форума по проблемам общего образования. Под научной редакцией С.В. Тарасова. Санкт-Петербург, 2022. С. 126-134.

2. Каракозов С.Д., Рыжова Н.И., Королева Н.Ю., Филимонова Е.В. Подготовка бакалавров педагогического образования к реализации проектно-исследовательской деятельности в условиях цифровизации школы. *Вестник Российского университета дружбы народов. Серия: Информатизация образования*. 2021. 18(2):115-127. DOI: 10.22363/2312-8631-2021-18-2-115-127

3. Корчажкина О.М. Информационный подход при освоении школьниками технологий искусственного интеллекта. *Педагогика информатики*. 2022; 1-2:27-43.

References (на английском языке)

1. Baranova E.V., Solodkaya M.A. Cifrovoy obrazovatel'nyj resurs «Osnovy komp'yuternogo zreniya» dlya budushchih uchitelej informatiki [Digital educational resource “Fundamentals of computer vision” for future teachers of computer science]. *Metodika преподаvaniya v sovremennoj shkole: problemy i innovacionnye resheniya*. Materialy rossijsko-uzbekskogo obrazovatel'nogo foruma po problemam obshchego obrazovaniya. Pod nauchnoj redakciej S.V. Tarasova. Sankt-Peterburg, 2022. P. 126-134. (In Russian)

2. Karakozov S.D., Ryzhova N.I., Koroleva N.Y., Filimonova E.V. Podgotovka bakalavrov pedagogicheskogo obrazovaniya k realizacii proektno-issledovatel'skoj deyatelnosti v usloviyah cifrovizacii shkoly [Teaching of bachelors of pedagogical education

to the implementation of design and research activities in the conditions of digitalization of the school]. *RUDN Journal of Informatization in Education*. 2021. 18(2):115-127. (In Russian). DOI: 10.22363/2312-8631-2021-18-2-115-127

3. Korchazhkina O.M. Informacionnyj podhod pri osvoenii shkol'nikami tekhnologij iskusstvennogo intellekta [Information approach in the development of artificial intelligence technologies by schoolchildren]. *Pedagogy of Computer Science*. 2022; 1-2:27-43. (In Russian)