

РАЗРАБОТКА ПРОГРАММНОГО СРЕДСТВА ДЛЯ ИДЕНТИФИКАЦИИ НОМЕРНЫХ ЗНАКОВ ТРАНСПОРТНЫХ СРЕДСТВ НА ОСНОВЕ МЕТОДОВ КОМПЬЮТЕРНОГО ЗРЕНИЯ

Яхшибоев Р.Э., Очилов Т.Д., Сиддиков Б.Н.

Ташкентский университет информационных технологий имени Мухаммада ал-Хоразмий

Аннотация – В данной статье рассмотрено разработка программного средства для идентификации номерных знаков транспортных средств на основе методов компьютерного зрения. Автоматическое распознавание номерных знаков (АРНЗ) играет важную роль в интеллектуальных транспортных системах, которые не только экономят труд, но и повышают эффективность.

В этой ограничении смягчаются рассмотренным эффективным методом для АРНЗ, который состоит из трех основных этапов: 1) предварительная обработка изображений, 2) расположение номерных знаков и 3) распознавание номерных знаков и символов. Целью шага 1 - является удаление шума с изображений и выделение краев изображений. Шаг 2 - извлечь номерной знак из изображения. Шаг 3 - сегментировать и распознавать числа и символы.

Ключевые слова – компьютерная зрения, искусственный интеллект, идентификация, номер, знак, сегментация.

ВВЕДЕНИЕ

В Постановлении Президента Республики Узбекистан от 17 февраля 2021 года «О мерах по созданию условий для оперативного внедрения технологий искусственного интеллекта» ПК-4996 внесен следующий пункт «В сфере транспорта: в процессе контроля локомотивов использование технологий искусственного интеллекта для контроля за их движением и предупреждения машинистов в опасных ситуациях, анализа движения общественного транспорта и определения его оптимальных маршрутов, а также контроля движения вагонов и пробок [1].

Для достижения цели исследования были поставлены следующие задачи:

- Проведение сравнительного анализа алгоритмов идентификации объектов на изображениях
- Разработка методов обучения и методов предварительной обработки для автоматической идентификации номерных знаков транспортных средств.
- Построение сверточной нейронной сети для автоматической идентификации номерных знаков транспортных средств и повышения ее точности.

Компьютерное зрение — это междисциплинарная область, изучающая, как компьютеры могут получать высокоуровневое понимание цифровых изображений или видео. С инженерной точки зрения он стремится понять и автоматизировать задачи, которые может выполнять зрительная система человека [2,3,4].

Задачи компьютерного зрения включают методы захвата, обработки, анализа и понимания цифровых изображений, а также извлечения многомерных данных из реального мира для создания цифровой или символьной информации. Понимание в этом контексте относится к преобразованию визуальных образов (ввод на сетчатку) в представление мира, которое имеет смысл для мыслительных процессов и вызывает соответствующие действия. Это понимание изображения можно рассматривать как извлечение символической информации из данных изображения с использованием моделей, построенных с использованием геометрии, физики, статистики и теории обучения.

Научная дисциплина компьютерного зрения занимается теорией искусственных систем, получающих информацию из изображений. Данные изображения могут принимать различные формы, такие как видеопоследовательность, виды с нескольких камер, многомерные данные трехмерного сканирования или медицинского сканирования. Технологическая дисциплина компьютерного зрения стремится применить свои теории и модели к построению систем компьютерного зрения [14,15,16].

Подобластями компьютерного зрения являются реконструкция сцены, обнаружение объектов, обнаружение событий, видеонаблюдение, обнаружение объектов, оценка 3D-позы, обучение, индексирование, оценка движения, визуальное сопровождение, моделирование 3D-сцены, а также восстановление изображения.

АНАЛИЗ АЛГОРИТМОВ МАШИННОГО ОБУЧЕНИЕ

В наше время такие понятия, как машинное обучение или искусственный интеллект, развиваются буквально повсеместно, и это не зря, ведь такие технологии на самом деле обеспечивают удивительные достижения в области вычислений. На самом деле, алгоритмы машинного обучения предлагают бесчисленное множество приложений, включая автоматическое определение номерных знаков транспортных средств [17,18]. Следующие строки дают краткое описание этого типа технологии.

- По типу задачи машинного обучения:

- Проблема классификации: при заданном наборе образцов, каждый из которых соответствует определенному классу или категории данного дискретного набора, такие задачи направлены на присвоение каждой выборке

правильной метки класса. Чтобы быть более конкретным, прямым применением этого может быть идентификация животных путем анализа различных изображений.

- Проблема регрессии: этот второй тип решает проблему связывания соответствующих числовых значений или переменных с различными входными выборками. При этом, например, можно измерить расстояние до цели на основе характеристик формы.

- Проблема ассоциации или кластеризации: цель проблем ассоциации или кластеризации состоит в том, чтобы сгруппировать набор экземпляров на основе сходства атрибутов. Так, например, их можно использовать при сегментации изображений.

• Согласно порядку работы алгоритма машинного обучения:

-Контролируемый: такие алгоритмы машинного обучения характеризуются знанием результатов на этапе обучения, как если бы они обучались в присутствии учителя. На самом деле обучающие данные алгоритмов машинного обучения с учителем помечаются классом или известным значением, чтобы они могли учиться на нем и, следовательно, соответствующий класс или метка, когда они позже сталкиваются с тестовым набором, могут оценить значение.

-Неуправляемый: в отличие от предыдущего типа, неуправляемые алгоритмы машинного обучения не имеют никакой информации о результатах на этапе обучения, другими словами, их обучающие данные не размечены. В этом случае целью алгоритмов является выявление неизвестных закономерностей или групп данных путем самостоятельного обучения, чего они достигают, выполняя внутреннюю самооценку по некоторым критериям.

Искусственные нейронные сети. Искусственные нейронные сети (ИНС) представляют собой еще один метод контролируемого машинного обучения. Они получили свое название от реальных нейронных сетей, пучков нейронов и синапсов, обнаруженных в человеческом мозгу, показанных на рис. 1. В целом, они представляют собой мощную, гибкую и шумную систему распознавания образов, сочетающую в себе запоминание и обобщение. Следовательно, их можно использовать для решения множества задач, включая классификацию, регрессию и ассоциацию или кластеризацию [5,6,7]. Однако в этом разделе основное внимание уделяется вопросу классификации применительно к ADRAA.



Рисунок 1: Вид биологической нейронной сети

Они были разработаны примерно в 1950-х годах такими учеными, как Уоррен МакКаллох, Уолтер Питтс или Фрэнк Розенблатт, которые пытались создать вычислительную систему, вдохновленную человеческим мозгом, что привело к рождению искусственных нейронных сетей. Для моделирования искусственных нейронов, основных узлов мозга, они разработали концепцию персептрона и использовали сигналы для моделирования связей между ними, называемых синапсами.

Персептрон — это очень простая математическая функция, которая выдает выходное значение из входных данных посредством серии вычислений. Как видно из общей схемы, показанной на рис. 2, каждая компонента n -мерного входного вектора взвешивается и суммируется, добавляя значение перекоса, и в результате получается функция активации, обеспечивающая выход персептрона. больше вектор, но только двоичное значение, то есть «0» или «1».

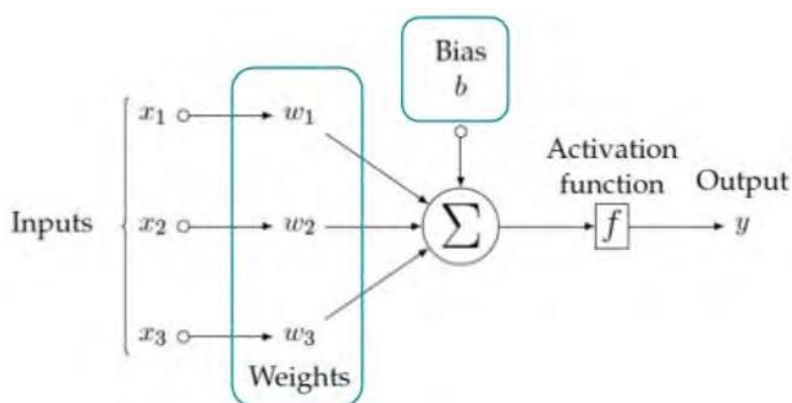


Рисунок 2: Общая схема персептрона

Синий прямоугольник на рис. 2 выделяет параметры, определяющие поведение персептрона, а именно вес и смещение, поэтому важно установить соответствующие значения для обоих. Веса — это действительные числа, которые представляют важность каждого входа для выхода. Что касается

смещения, его можно описать как меру легкости персептрона произвести «1». Чем больше смещение, тем легче получить «1» на выходе.

Однако компонентом, который фактически обеспечивает вывод персептрона, является функция активации. Это предопределенная функция, такая как сигмовидная функция, функция выпрямления или функция знака, которая обеспечивает плавный переход при изменении входных значений, так что небольшое изменение на входе вызывает небольшое изменение на выходе. В заключение, общее поведение воспринимающего может быть оценено математически в уравнении 1.

Например, пусть f будет функцией указателя:

$$y = \text{sign} \left(\sum_{i=0}^n x_i w_i + b \right) \quad (1)$$

Хотя их истинная сила заключается в использовании их в связанных комбинациях, персептроны могут распознавать простые закономерности самостоятельно. Например, они могут по отдельности представлять простые функции, такие как И, ИЛИ, НЕ-И или НЕ-ИЛИ, то есть любую функцию классификации, входы которой линейно разделимы. Однако они не могут представлять нелинейные разделимые функции, такие как исключающее ИЛИ. Чтобы достичь этого, многоуровневые сети персептронов, известные как многослойные персептроны (MLP), составляют популярный класс ИНС с прямой связью.

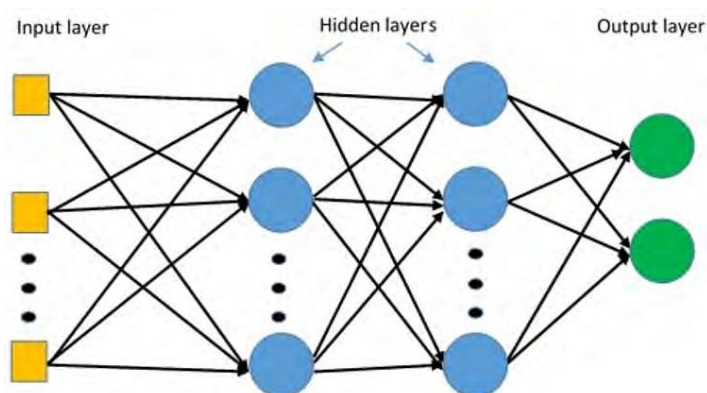


Рисунок 3: Общая схема многослойных персептронов

На рис. 3 показано, что многослойный персептрон состоит из N слоев по M персептронов, каждый из которых полностью связан графически с персептронами следующего слоя. Другими словами, каждый узел уровня N

получает результаты всех M узлов уровня $N-1$. Таким образом, входом в сеть становится вектор числовых признаков, описывающих примеры классификации, а выходом сети является вектор, представляющий результаты классификации. Примером этого выходного вектора для классов А, В и С может быть: $\{1, 0, 0\}$, $\{0, 1, 0\}$, $\{0, 0, 1\}$.

Таким образом, при разработке MLP следует выделить два основных аспекта. Этими аспектами являются количество слоев в сети, которых должно быть не менее трех, и количество скрытых узлов, обычно определяемое экспериментальным путем. Кроме того, как было сказано выше, вес персептронов является основным фактором корректной работы системы. В случае многослойной сети, если функция активации дифференцируема, можно указать ее значение. Кроме того, функция активации также должна обеспечивать выходной сигнал, представляющий собой нелинейную комбинацию входных данных. Обычно используемой функцией, удовлетворяющей обоим условиям, является сигмовидная функция, соответствующая выражению, показанному в уравнении 2.

$$\sigma = 1/(1 + e^{-x}) \quad (2)$$

Используя этот тип функции, алгоритм градиентного спуска MLP можно использовать для обучения нейронных сетей изучению нелинейных функций посредством обратного распространения, что лучше всего подходит для задач классификации. Он так называется потому, что вносит поправки для уменьшения ошибки в обратном направлении, то есть от выходного слоя через каждый скрытый слой к первому скрытому слою. Основные этапы, выполняемые этим методом, следующие.

1. Инициализация весов для малых случайных значений
2. Распространение входных данных
3. Распространение ошибки с обновлением весов и смещений.
4. Состояние завершения, когда ошибка сведена к минимуму или выполнено достаточное количество итераций.

РЕЗУЛЬТАТЫ

Понимание и сбор необходимых данных: нужно данные для создания системы идентификации номерных знаков. Для этого надо собрать как можно больше изображений номерных знаков.

-Маркировка изображений: для маркировки изображений я использовал инструмент аннотирования изображений LabelImg. Надо загрузить labelImg с GitHub и следовать инструкциям по установке пакета.

После открытия графический интерфейс предложит и щелкнет CreateRectBox, чтобы нарисовать прямоугольную рамку, как показано ниже, и сохранить результат в XML.

-Анализ данных из XML: пример файлов .xml можно увидеть ниже.

После того, как завершается процесс тегирования, необходимо предварительно обработать данные. Поскольку вывод метки представляет собой XML, нам нужны данные в формате массива, чтобы использовать их в процессе обучения. Для этого получаем полезную информацию от метки, которая представляет собой диагональные точки прямоугольного прямоугольника или граничные точки x_{min} , u_{min} , x_{max} , u_{max} соответственно.

Это доступно в формате XML. Итак, нам нужно извлечь данные и сохранить их в любом удобном формате, здесь я буду конвертировать пороговые данные в CSV, а затем конвертировать их в массив с помощью Pandas. Давайте посмотрим, как анализировать данные с помощью Python [8,9,10].

-Анализ данных из XML и преобразование их в формат CSV: во-первых, позвольте мне загрузить все библиотеки, которые я хочу, в этом проекте сразу. Также использована библиотека Python xml.etree для анализа данных из XML, а также для импорта pandas и globs. Использование glob позволяет получить все XML-файлы, созданные ранее во время тегирования.

-Проверка данных: до сих пор мы выполняли ручной процесс, важно проверить, являются ли данные, которые получаем, действительными или нет. Он только проверяет, правильно ли выглядит ограничительная рамка для данного изображения.

Обработка данных. Чтение данных: это очень важный шаг, в этом процессе мы берем каждое изображение и преобразуем его в массив с помощью OpenCV и изменяем размер изображения до 224 x 224, что является размером по умолчанию для предварительно обученной модели трансферного обучения.

После этого нормализуется изображение, просто разделив на максимальное число, так как мы знаем, что максимальное число для 8-битного изображения равно $28 - 1 = 255$. Это потому, что мы делим наше изображение на 255,0. Метод уменьшения массива до максимального значения называется нормализацией (Min-Max Scaler). Нам также необходимо нормализовать наши теги. Потому что выходной диапазон для модели глубокого обучения должен быть между 0 и 1. Чтобы нормализовать метки, нам нужно разделить диагональные точки на ширину и высоту изображения. И, наконец, значения в списке python [11,12,13].

-Создано папки `train` и `test`: на следующем шаге мы преобразуем список в массив с помощью `Numpy`.

Линия передачи модели обнаружения объекта (конвейер). Прогноз: это последний этап обнаружения объекта. На этом этапе мы все объединяем и делаем прогноз для данного изображения.

-Нормализация результата

-Ограничивающая рамка: теперь мы рисуем линию границы в верхней части изображения. Я просто хочу представить две диагональные точки. Давайте воспользуемся этими точками и нарисуем прямоугольную коробку.

-Создание пайплайна: Теперь что нам делать, давайте соберем все воедино и создадим функцию. И представьте это в конце. Наш вывод возвращает координаты изображения и ограничивающей рамки.

Оптическое распознавание символов (OCR). `Tesseract OCR`: программное обеспечение для оптического распознавания символов (OCR) используется для извлечения текста из изображения. `Tesseract OCR` имеет API `Python` и является открытым исходным кодом. Во-первых, мы установим его. Это очень просто и зависит от вашей операционной системы. Руководство по установке и загрузки можно найти здесь.

`EasyOCR`, как следует из названия, представляет собой пакет `Python`, который позволяет разработчикам компьютерного зрения легко выполнять оптическое распознавание символов.

- Когда дело доходит до OCR, `EasyOCR` — это самый простой способ использовать оптическое распознавание символов:

- Пакет `EasyOCR` можно установить с помощью одной команды `pip`.

- Зависимости от пакета `EasyOCR` минимальны, что упрощает настройку среды разработки OCR.

-После установки `EasyOCR` требуется только один оператор импорта для импорта пакета в ваш проект.

-Оттуда требуется две строки кода для выполнения OCR — одна для инициализации класса `Reader`, а другая для распознавания изображения с помощью функции чтения текста.

`EasyOCR` реализован с использованием `Python` и библиотеки `PyTorch`. Если у вас есть графический процессор с поддержкой `CUDA`, библиотека глубокого обучения `PyTorch` может значительно ускорить распознавание текста и OCR. На момент написания этой статьи `EasyOCR` может распознавать текст на 58 языках, включая английский, немецкий, хинди, русский и другие! Разработчики `EasyOCR` планируют в будущем добавить дополнительные языки.

Моя рекомендация начать установку `EasyOCR` — следовать руководству `pip install opencv` с важной оговоркой:

Убедитесь, что вы устанавливаете `opencv-python`, а не `opencv-contrib-python` в своей виртуальной среде. Кроме того, если у вас есть оба этих пакета в одной среде, это может привести к неожиданным последствиям. Если у вас установлены оба, `pip` вряд ли будет жаловаться, поэтому будьте осторожны и проверяйте с помощью команды `pip Freeze`.

-Ограничения Pytesseract: Tesseract работает лучше всего, когда есть четкая сегментация фонового текста. На практике такие установки очень трудно гарантировать.

Существуют различные причины, по которым Tesseract не может обеспечить хорошее качество вывода, например, фоновый шум. Чем лучше качество изображения (размер, контрастность, яркость), тем лучше результат распознавания. Для улучшения результатов OCR требуется некоторая предварительная обработка, изображения должны быть соответствующим образом масштабированы, иметь максимально возможную контрастность изображения, а текст должен быть выровнен по горизонтали. Tesseract OCR очень мощный, но имеет следующие ограничения.

-Извлечение текста в важной области изображения: сначала мы загружаем наше изображение и преобразуем его в массив. Мы разрезаем наш ограниченный прямоугольник с его координатами.

Очевидно, мы не получили тот текст, который вы хотели, но вы можете получить по крайней мере 90% информации. Это всего лишь пример, и опять же, больше информации, чем лучшего прогноза. Мы подойдем к этому позже. Вот что я понимаю: во-первых, у нас не так много данных для решения этой проблемы, и я добавил почти идентичные наборы данных от других кагглеров, недавно опубликованных в этой теме.

Во-вторых, мне не нравится эта модель, но весь проделанный процесс дал нам возможность понять концепцию, мы создадим другую модель, используя YOLO, и посмотрим, как она сравнится. наши результаты. Во втором посте Tesseract я объяснил некоторые из его основных моментов, но предварительная обработка изображений может быть другой темой, и для этого может даже потребоваться встроить в него ИИ. Итак, теперь я хочу показать вам, как создать простую паутину, а следующим шагом мы начнем с новой модели.

Были получены следующие результаты:

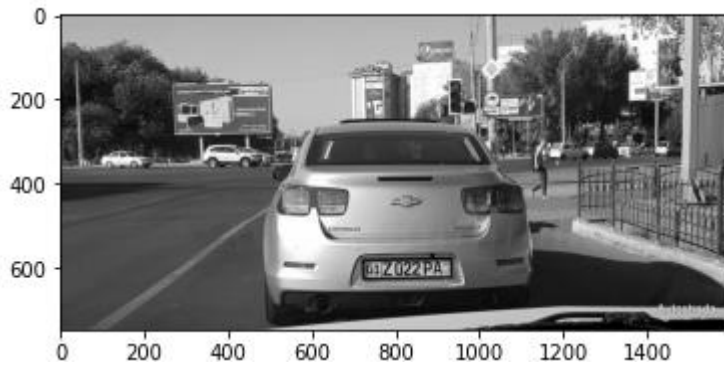


Рисунок 4. Обучение изображение как серое и размытое.

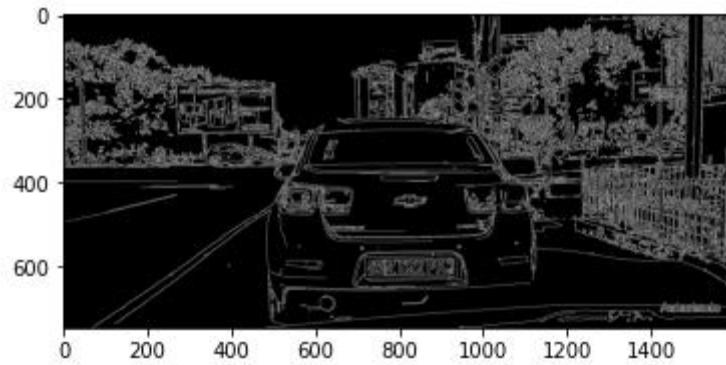


Рисунок 5. Отфильтрование изображение и определите края.

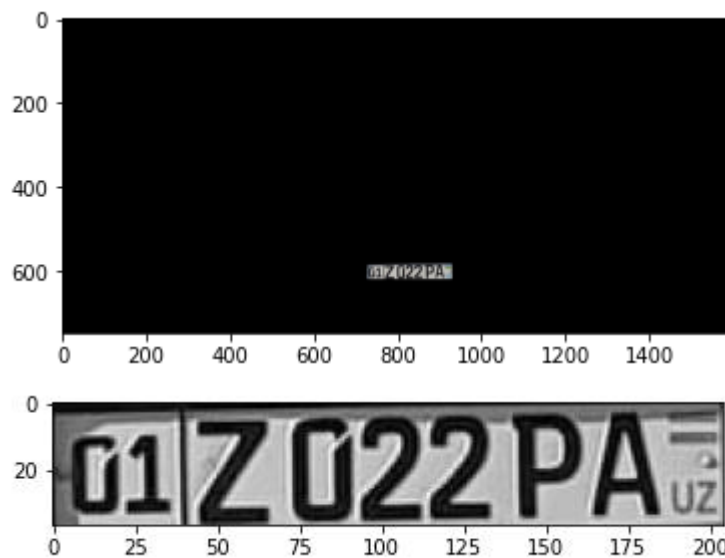


Рисунок 6. Поиск контуры и нанесение маски

`[([[1, 0], [204, 0], [204, 37], [1, 37]], '017022PAu', 0.5281677982497428)]`



Рисунок 7. Окончательный результат.

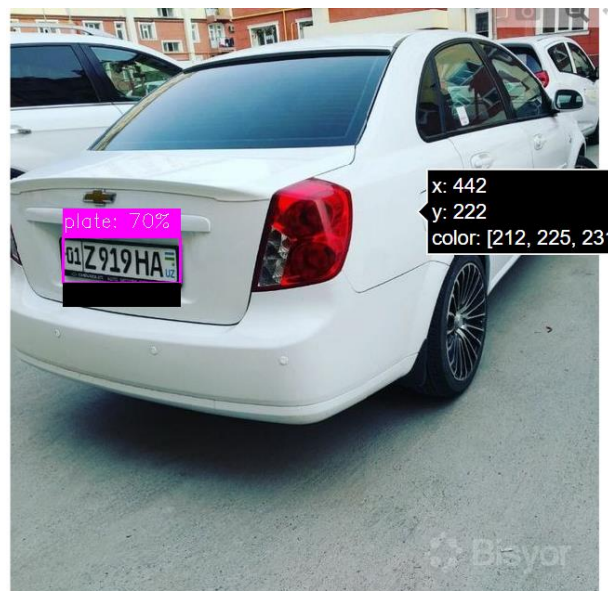


Рисунок 8. Определение критической области через Yolo.

ЗАКЛЮЧЕНИЕ

В заключение следует сказать, что в результате данной статьи был разработан программный инструмент для автоматической идентификации номерных знаков автомобилей с помощью компьютерного зрения и достигнуты следующие результаты:

- проведен сравнительный анализ алгоритмов идентификации объектов на изображениях;
- сформирована выборка для определения ALPR и разработаны методы предварительной обработки;
- Построена сверточная нейронная сеть для ALPR и повышена точность модели за счет настройки гиперпараметров;
- Результаты показали, что нейронная сеть, обученная с помощью этой системы, способна работать с почти 90-95-процентной точностью при распознавании номерных знаков на изображениях с низким разрешением.

СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУР

1. O‘zbekiston Respublikasi Prezidentining 2019-yil 30-avgustdagi “O‘zbekiston Respublikasida kosmik faoliyatni rivojlantirish to‘g‘risida”gi PF-5806-sonli Farmoni.
2. O‘zbekiston Respublikasi Prezidenti Shavkat Mirziyoyevning 2018 yil 28-dekabrdagi Oliy Majlisga Murojaatnomasi
3. Sirojiddin Komolov, Sherzod Rahmatov: Sun'iy intellekt asoslari. Mashinaviy o‘qitish, 2019-yil.
4. Эркин ва фаровон, демократик Ўзбекистон давлатини биргаликда барпо этамиз. Ўзбекистон Республикаси Президенти лавозимида киришиш тантанали маросимида бағишланган Олий Мажлис палаталарининг қўшма мажлисидаги нутқ /Ш.М. Мирзиёев. - Тошкент: «Ўзбекистон», 2016. -56 б.
5. Khamzaev J. et al. DRIVER SLEEPINESS DETECTION USING CONVOLUTION NEURAL NETWORK //CENTRAL ASIAN JOURNAL OF EDUCATION AND COMPUTER SCIENCES (CAJECS). – 2022. – Т. 1. – №. 4. – С. 31-35.
6. Yaxshiboyev R. DEVELOPMENT OF A SOFTWARE AND HARDWARE COMPLEX FOR PRIMARY DIAGNOSTICS BASED ON DEEP MACHINE LEARNING //CENTRAL ASIAN JOURNAL OF EDUCATION AND COMPUTER SCIENCES (CAJECS). – 2022. – Т. 1. – №. 4. – С. 20-24.
7. Yaxshiboyev R. DEVELOPMENT OF A MODEL OF OBJECT RECOGNITION IN IMAGES BASED ON THE «TRANSFER LEARNING» METHOD //CENTRAL ASIAN JOURNAL OF EDUCATION AND COMPUTER SCIENCES (CAJECS). – 2022. – Т. 1. – №. 4. – С. 36-41.
8. Yaxshiboyev R., Yaxshiboyeva D. ANALYSIS OF ALGORITHMS FOR PREDICTION AND PRELIMINARY DIAGNOSTICS OF GASTROENTEROLOGICAL DISEASES //CENTRAL ASIAN JOURNAL OF EDUCATION AND COMPUTER SCIENCES (CAJECS). – 2022. – Т. 1. – №. 2. – С. 49-56.
9. Yaxshiboyev R. E. et al. FORECASTING GROUNDWATER EVAPORATION USING MULTIPLE LINEAR REGRESSION //Galaxy International Interdisciplinary Research Journal. – 2021. – Т. 9. – №. 12. – С. 1101-1107.
10. Yakhshibaev R. et al. Development of a mathematical model for balancing the level and device for remote monitoring of groundwater parameters //2021 International Conference on Information Science and Communications Technologies (ICISCT). – IEEE, 2021. – С. 1-4.

11. Djumanov J. et al. Mathematical model and software package for calculating the balance of information flow //2021 International Conference on Information Science and Communications Technologies (ICISCT). – IEEE, 2021. – С. 1-6.
12. Gafurjonovich M. V., Yaxshivayevich E. E., Erkinbayevna Y. D. Technologies of Organization of Practical Lessons in the Modern Education System //European Multidisciplinary Journal of Modern Science. – 2022. – Т. 4. – С. 332-336.

Интернет-ресурсы

13. <https://www.upgrad.com/blog/ultimate-guide-to-object-detection-using-deep-learning/>
14. <https://towardsdatascience.com/deep-learning-for-object-detection-a-comprehensive-review-73930816d8d9>
15. <https://bdtechtalks.com/2021/06/21/object-detection-deep-learning/>
16. <https://www.toyota-boshoku.com/global/teambreakthrough/technology/001/>
17. <https://www.theworldcounts.com/challenges/consumption/transport-and-tourism/cars-impact-on-the-environment/story>
18. https://www.ccohs.ca/oshanswers/safety_haz/electrical.html#:~:text=Never%20touch%20the%20vehicle%20or,may%20be%20shocked%20or%20electrocuted.