

Особенности разработки. В качестве архитектуры для данного проекта была выбрана следующая модель: *Any client + serverless*. Выбор данной модели для разработки примечателен многочисленными плюсами для масштабирования, сопровождения и управления системой:

- максимальная эластичность. Быстрое масштабирование от нуля до тысяч параллельно работающих функций;
- полная абстракция от операционной системы или любого софта, использующегося для выполнения приложения. Вам неважно, запускаются ли ваши *Serverless* приложения на *Linux*, *Windows* или *custom OS*. Все, что вас волнует, это способность платформы выполнять *Python/Java/Ruby/YouNameIt* код и сопутствующие библиотеки для этого ЯП;
- при правильном проектировании функций легче построить слабо связанную архитектуру, при которой ошибка в одной функции не скажется на работоспособности всего приложения;
- ниже порог входа для новоприбывших. Понять «наносервис» из 100–500 строк (а это и есть обычный размер функции в *Serverless*) для нового разработчика в команде гораздо проще, чем понять *legacy* проект с миллионом строк и сложных связей.

Технологии для построения базовой версии модели. В качестве основных технологий для построения веб-решения будут использованы фреймворки *angular* (как удобное по структуре решение для построения *UI*) и *firebase platform* (как идеальный инструмент связки веб-решения на *node.js* и облачной платформы *google cloud*).

В качестве основных технологий для построения *back-end* составляющей системы будут использованы инструменты *google cloud* платформы:

- *Cloud-functions* – для реализации наносервисов, соответствующих *serverless* модели.
- *Firebase-ml* – для реализации технологий нейронных сетей, вспомогательного инструмента для таргетной рекламы, рекомендаций и умного добавления медикаментов.
- *Pub-sub* – для реализации *message query* между наносервисами и компонентами системы.
- *Firebase/firestore* – *nosql* бд для хранения данных.
- *Security* – для защиты пользовательских данных.
- *Pay-api* – для обработки и осуществления транзакций, используя платформу *google*.
- *Cloud-storage* – для хранения открытых данных пользователей.
- *Firebase-analytics* – для составления аналитики по всей системе.

ИСПОЛЬЗОВАНИЕ КОМПЬЮТЕРНОГО ЗРЕНИЯ ДЛЯ РЕШЕНИЯ ПРОБЛЕМ РАСПОЗНАВАНИЯ ЖЕСТОВ

М. А. Рябиков

Учреждение образования «Гомельский государственный технический университет имени П. О. Сухого», Республика Беларусь

Научный руководитель В. С. Захаренко

С развитием персонального компьютера, также наблюдалась эволюция пользовательского интерфейса. От текстового командного до графического интерфейса: от простой клавиатуры до мыши, геймпада, электронной ручки, видеокамеры и т. д.

Эти устройства стали удобными и естественными для среднестатистического пользователя персонального компьютера. На сегодняшний день необходимо создание более мощных и удобных способов взаимодействия человека с компьютерной системой.

В качестве одной из возможности обеспечения комфортного взаимодействия с компьютером применяется виртуальный шлем. Комплекс *Kinect*, разработанный компанией *Microsoft*, позволяет пользователю взаимодействовать с серией игровой консоли *Xbox* без помощи игрового контроллера через устные команды, позы тела и показываемые объекты или рисунки.

Видеокамера является недорогим и удобным устройством для ввода информации, которое может служить эффективным каналом связи при реализации человеко-машинного взаимодействия. Современные достижения в технологии компьютерного зрения и высокая производительность компьютерной техники делают отслеживание и распознавание жестов в режиме реального времени перспективным направлением исследования с возможностью широкого применения.

На сегодняшний день существуют разные подходы к решению задачи распознавания жестов. Большинство этих подходов воспринимают в целом, и пытаются извлечь соответствующее математическое описание из большого количества обучающих примеров. Такие подходы анализируют положения головы, угол ее наклона, не раскладывая их на составные элементы, применение которых могло бы упростить сложность жестов.

В результате этого большинство существующих подходов характеризуются недостаточной скоростью, точностью, надежностью и ограниченным количеством распознанных жестов.

В настоящее время существует множество различных методов обнаружения объекта на изображении. Эти методы можно разделить на три основные группы:

- скелетные методы;
- методы на основе 3D-модели объекта;
- методы на основе 2D-модели объекта.

В скелетных методах исследуется контур силуэта: обычно отыскиваются углы, выступы, впадины и другие точки с высокими значениями кривизны. Для получения информации о форме контура применяются различные представления границы объекта [1].

В методах 3D-модели руки представляют в виде сложных трехмерных поверхностей и классифицируют с помощью нейронных сетей.

Метод 2D-распознавания схож с предыдущим, но использует двумерное изображение вместо объемных моделей.

Каждый из методов имеет преимущества и недостатки. Так как при разработке алгоритма объектом обнаружения на изображении является рука, то первый метод можно исключить из-за неудобства использования контура для определения конкретного жеста. Недостатком метода на основе 3D-модели является его ресурсоемкость. Построение 3D-модели, обучение нейронной сети и ее использование могут потребовать значительных ресурсов, поскольку не стоит забывать, что для использования данного метода требуются камеры с возможностью определения глубины изображения [1].

Одним из перспективных методов распознавания образов считается метод Виоли–Джонса – это алгоритм, который позволяет обнаруживать объекты на изображениях в реальном времени. Данный метод становится основополагающим для поиска

объектов на изображении в реальном времени в большинстве существующих алгоритмов распознавания и идентификации. Также данный алгоритм является одним из лучших по соотношению эффективности распознавания и скорости работы.

Основные принципы, на которых основан метод Виолы–Джонса:

– возможность представления изображения в интегральном виде, что позволяет вычислять быстро необходимые объекты;

– использование признаков Хаара, т. е. признаков цифрового изображения, используемых в распознавании образов, с помощью которых происходит поиск нужного объекта;

– применение алгоритма бустинга для выбора наиболее подходящих признаков для искомого объекта на данной части изображения, процедуры последовательного построения композиции алгоритмов машинного обучения, когда каждый следующий алгоритм стремится компенсировать недостатки композиции всех предыдущих алгоритмов;

– использование каскадов признаков для быстрого отбрасывания окон, где не найден объект [1].

Использование каскада Хаара имеет следующие плюсы:

– описывают те знания о классе объектов, которые трудно выделить на конкретном числе обучаемых данных;

– устойчивость к смене освещения, даже если это локальная смена освещения, устойчивость к шумам (примитивы представляют собой простейший полосовой фильтр);

– если примитивы были не очень маленькие, то сильно устойчивее корреляции при изменении масштаба (размер примитивов при этом не будет влиять на точность, если обход с маленьким шагом);

– если признаки на большом изображении рассчитать заранее и при сдвиге окна поиска брать уже посчитанные и актуальные для него, то поиск будет значительно быстрее корреляции (необходимо сравнить меньшее количество элементов);

– такие системы работают гораздо быстрее, чем системы, работающие напрямую с пикселями.

Для выбора наиболее подходящих признаков для искомого объекта на части изображения используют алгоритм бустинга. Данный метод позволяет усилить простые классификаторы путем комбинирования примитивных «слабых» классификаторов в один «сильный». Под термином «сила» классификатора в данном случае подразумевается эффективность решения задачи классификатора.

На сегодняшний день уже существуют большое количество библиотек, которые облегчают работу с взаимодействием искусственных нейронных сетей. Большое преимущество этих библиотек состоит в том, что у них уже реализованы алгоритмы по распознаванию, классификации данных. Наиболее популярные библиотеки:

– *TensorFlow*;

– *PyTorch*;

– *Keras*;

– *MXNet*;

– *MediaPipe*;

– *FingerPose*.

Сравнение характеристик библиотек по распознаванию жестов представлено в таблице.

Сравнение характеристик библиотек по распознаванию жестов

Название библиотеки	Преимущества	Недостатки
<i>TensorFlow</i>	Хорошая документация. Мощные средства для мониторинга моделей. Обслуживание моделей	Основная поддержка только на языке <i>Python</i>
<i>PyTorch</i>	Процесс создания модели достаточно прост и прозрачен	Основная поддержка только на языке <i>Python</i>
<i>Keras</i>	Достаточно маловесная	Есть ограничения по использованию методов распознавания жестов
<i>MXNet</i>	Достаточно быстрый, гибкий и эффективный в вопросах работы с алгоритмами глубокого обучения	Не популярен в научном сообществе
<i>MediaPipe</i>	Встроенные быстрые выводы и обработка <i>ML</i>	Не поддерживает низкоуровневую архитектуру

Следовательно, исходя из сравнений библиотек, было принято решение взять библиотеку *MediaPipe* для решения задач по распознаванию жестов. Данный выбор был сделан, поскольку эта библиотека имеет хорошие алгоритмы для отслеживания движения части тел человека и у нее хорошо реализованы алгоритмы классификаций рук и других частей тел.

Библиотека использует конвейер *ML*, состоящий из нескольких моделей, работающих вместе. Модель обнаружения ладони, которая работает с полным изображением и возвращает ориентированную границу руки. Модель ориентира руки, которая работает с обрезанной областью изображения, определенной детектором ладони, и возвращает высокоточные 3D-ключевые точки руки.

Для определения начального расположения рук существует модель детектора одного снимка, оптимизированная для мобильного использования в реальном времени, подобно модели обнаружения лиц в *MediaPipe Face Mesh*.

Более того, ладони можно моделировать с помощью квадратных ограничительных рамок (якорей в терминологии *ML*), игнорируя другие соотношения сторон, что позволяет сократить количество якорей в 3–5 раз. Также используется кодирующий-декодирующий экстрактор признаков для более глубокого понимания контекста сцены.

С помощью вышеперечисленных методов мы достигли средней точности в 95,7 % при обнаружении ладоней.

Литература:

1. Гонсалес, Р. Цифровая обработка изображений / Р. Гонсалес, Р. Вудс. – М. : Техносфера, 2005. – 1072 с.
2. Зайцева, Г. Л. Жестовая речь. Дактилология : учеб. для студентов высш. учеб. заведений / Г. Л. Зайцева. – М. : ВЛАДОС, 2004. – 192 с.
3. Форсайт, Д. Компьютерное зрение. Современный подход / Д. Форсайт. – М. : Вильям, 2004. – 928 с.
4. Хуанг, Т. С. Быстрые алгоритмы в цифровой обработке изображений / Т. С. Хуанг. – М. : Радио и связь, 1984. – 224 с.
5. Шапиро, Л. Компьютерное зрение / Л. Шапиро, Д. Стокман. – М. : БИНОМ. Лаб. знаний, 2006. – 752 с.