УДК 004.932.72'1
Серко А. С.
Научный руководитель: ассистент Монтик Н. С.

ТЕХНОЛОГИЯ РАСПОЗНАВАНИЯ ЖЕСТОВ MEDIAPIPE

Введение

В последние годы распознавание жестов стало широкодоступным благодаря современным технологиям и программному обеспечению, такому как камеры глубины, 3D-сканеры, машинное обучение и нейросети. Кроме того, оно стало намного более востребованным в различных областях, включая обучение и образование, медицину, промышленность, развлечения и т. д. В данной статье будет рассмотрен один из примеров реализации распознавания жестов и основная концепция этой технологии: технология Mediapipe.

В первую очередь следует заметить, что Mediapipe основана на конвейерах восприятия, или пайплайнах, которые представляют собой аудио, видео и данные временных рядов. Многие приложения для машинного зрения начинаются с получения изображений и данных, затем обрабатывают эти данные, выполняют некоторые этапы анализа и распознавания и, наконец, выполняют действие. Для наглядности ниже представлен алгоритм работы конвейера (см. рисунок 1).

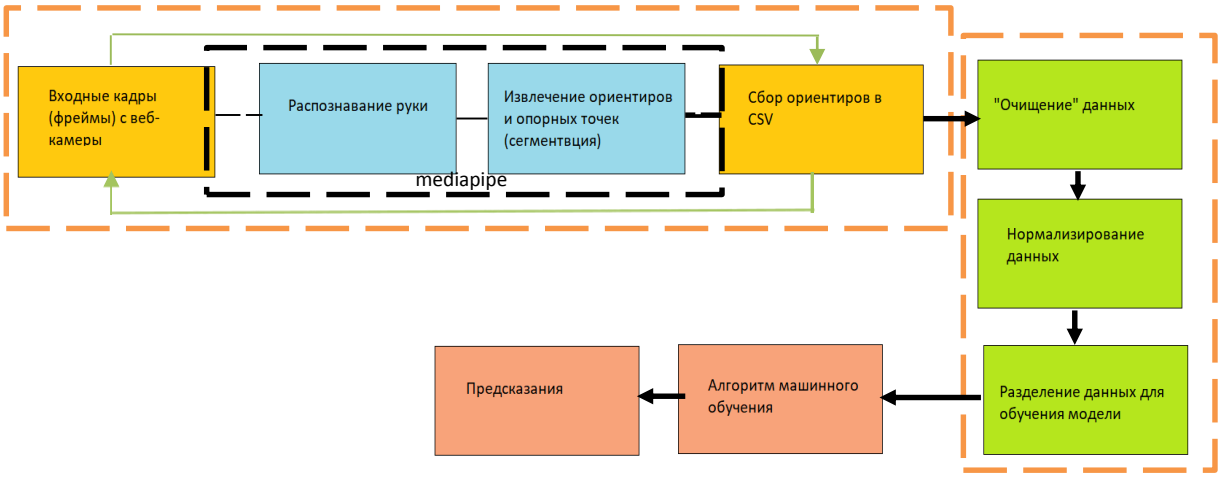


Рисунок 1 – Алгоритм работы конвейера

Необходимо обратить внимание на то, что для распознавания жестов рук используется библиотека Mediapipe Hands, которая является высококачественным решением для отслеживания рук и пальцев. Она использует машинное обучение (ML) для определения 21 3D-ориентира руки всего по одному кадру. Нельзя не дать определение вышеупомянутому конвейеру компьютерного зрения, который является фундаметом всей технологии.

Изначальные функциональные возможности Mediapipe [1]:

- распознавать лица и делать Face Mesh;
- определять радужные оболочки глаза: зрачки и контуры глаза;
- находить руки, ноги, определять позы;
- сегментировать волосы и селфи;
- запускать модель обнаружения и отслеживать ее предсказания;
- мгновенно идентифицировать движения;
- Objectron: определять 3D-объекты по 2D-изображениям;
- KNIFT: сопоставлять признаки на основе шаблонов;
- AutoFlip: конвейер автоматической обрезки видео.

Языки программирования, использованные для разработки Mediapipe: C++, Java, Objective-C .

Основные компоненты MediaPipe:

1. Пакет: базовая единица передачи данных называется “пакетом”. Он состоит из числовой метки времени и общего указателя на неизменяемую полезную нагрузку.

2. Граф: обработка происходит внутри графа, который определяет пути передачи пакетов между узлами. Граф может иметь любое количество входных и выходных данных, а также разветвляться или объединять данные.

3. Узлы: узлы – это места, где выполняется основная часть работы графа. Их также называют “калькуляторами” (по историческим причинам), и они производят или потребляют пакеты. Интерфейс каждого узла определяет количество входных и выходных портов. Могут быть нескольких типов [1]:

3.1 Калькуляторы предварительной обработки – это семейство калькуляторов для обработки изображений и мультимедиа.

3.2 Калькуляторы вывода обеспечивают встроенную интеграцию с TensorFlow и TensorFlow Lite для вывода ML.

3.3 Калькуляторы постобработки выполняют задачи постобработки ML, такие как обнаружение, сегментация и классификация. Tensor To Landmark – это калькулятор постобработки.

4. Утилитарные калькуляторы – это семейство калькуляторов, выполняющих конечные задачи, такие как аннотирование изображений.

5. Потоки: поток – это соединение между двумя узлами, которое передает последовательность пакетов с возрастающими временными метками.

Обучение

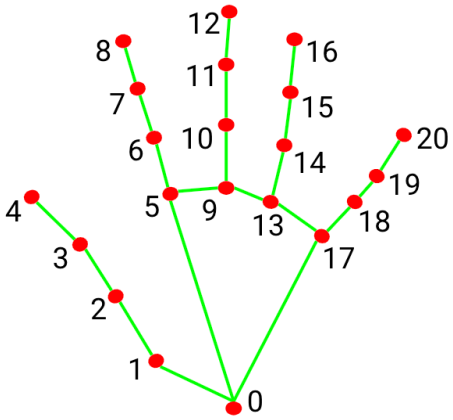
Для обучения компьютерной модели американскому языку жестов (American Sign Language) потребовалось решить следующие задачи:

- изучение алгоритмов распознавания жестов, основанных на методах машинного обучения с использованием опорных точек для распознавания кистей рук и жестов в режиме реального времени;
- разработка и реализация алгоритма и программного обеспечения для распознавания алфавита на основе жестов, используя ***следующие технологии***: **Google Mediapipe Hands, NumpPy, Pandas, Matplotlib, OpenCV2; Python; TensorFlow/Keras и Scikit-Learn**. Документация Mediapipe [2], [3].

Для распознавания жестов в основе лежит конвейер ML, состоящий из двух моделей, работающих независимо друг от друга:

- модель обнаружения ладони;
- модели ориентиров.

Модель распознавания ладони обеспечивает точно обрезанное изображение ладони. Сначала обучается детектор ладони, который оценивает ограничивающие рамки вокруг жестких объектов, таких как ладонь и кулаки, что проще, чем обнаружение рук со сцепленными пальцами. После того как функция распознавания ладони прошлась по всему кадру изображения, на экране появляются последующие модели ориентиров для рук. Эта модель точно локализует 21 трехмерную координату сустава руки (т. е. оси x, y, z) внутри обнаруженных областей кисти (см. рисунок 2). Сама же модель способна отображать координаты частично видимой руки.



0. WRIST
1. THUMB_CMC
2. THUMB_MCP
3. THUMB_IP
4. THUMB_TIP
5. INDEX_FINGER_MCP
6. INDEX_FINGER_PIP
7. INDEX_FINGER_DIP
8. INDEX_FINGER_TIP
9. MIDDLE_FINGER_MCP
10. MIDDLE_FINGER_PIP
11. MIDDLE_FINGER_DIP
12. MIDDLE_FINGER_TIP
13. RING_FINGER_MCP
14. RING_FINGER_PIP
15. RING_FINGER_DIP
16. RING_FINGER_TIP
17. PINKY_MCP
18. PINKY_PIP
19. PINKY_DIP
20. PINKY_TIP

Рисунок 2 – Модель ориентиров [1]

Ориентиры определяются по следующему алгоритму:

1. Функциональная модель обнаружения ладоней и кистей рук передается нашему набору данных.

2. Учитывая набор данных по американскому языку жестов, имеется алфавит от A до Z. Выполняется обнаружение руки и в результате получается 21 ориентир (см. рисунок 3).

3. Полученные ориентиры затем сохраняются в файле формата CSV.

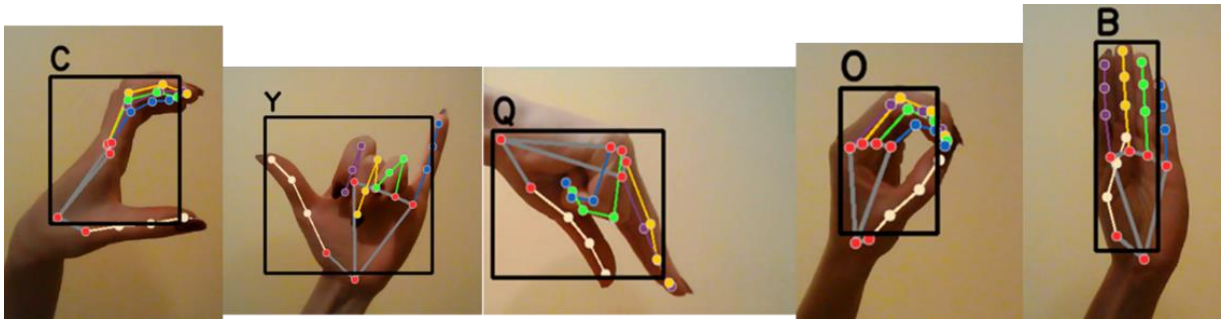


Рисунок 3 – Тестирование распознавания набора жестов

Для очистки и нормализации данных, как и на этапе 1, мы рассматриваем только координаты x и y от детектора, на этом этапе каждое изображение в наборе данных пропускается через этап 1 для сбора всех точек данных в одном файле. Затем этот файл просматривается с помощью библиотечной функции Pandas, чтобы проверить наличие любых записей с нулевыми значениями. Иногда из-за размытого изображения детектор не может обнаружить руку, что приводит к нулевому вводу в набор данных. Нам нужно удалить эти записи из набора данных. Следовательно, необходимо очистить эти точки или нулевые записи, иначе это приведет к смещению при создании прогностической модели. Строки, содержащие эти нулевые записи, ищутся по индексам и удаляются из таблицы. После удаления ненужных точек мы нормализовали координаты x и y, чтобы они вписывались в нашу систему. Затем файл данных подготавливается для разделения на два набора, а именно на обучающий и проверочный. 80 % данных сохраняется для обучения нашей модели с различными функциями оптимизации и потерь, в то время как 20 % данных зарезервировано для проверки модели.

Результаты и их обсуждение. Оценка недостатков и достоинств

Реализованное приложение с точностью 87 % определяет букву алфавита. Недостатком полученного приложения являются высокие требования к вычислительным мощностям компьютера, а также повышенные требования к четкости видеоряда.

Заключение

В статье описана методика обработки скелетного представления руки для распознавания жеста, реализация и подготовка собственного дата-сета и тестирование полученной модели. Точность обученной модели составляет около 90 %.

Список цитированных источников

1. Introduction to MediaPipe [Электронный ресурс] – Режим доступа: https://learnopencv.com/introduction-to-mediapipe/ – Дата доступа: 21.05.2023.

2. On-device machine learning for everyone [Электронный ресурс] – https://developers.google.com/mediapipe – Дата доступа: 21.05.2023.

3. Gesture recognition task guide [Электронный ресурс] – https://developers.google.com/mediapipe/solutions/vision/gesture_recognizer – Дата доступа: – 21.05.2023.