УДК 543.257.2

РАСПОЗНАВАНИЕ ЖЕСТОВЫХ КОМАНД НА ОСНОВЕ ИНТРУМЕНТА MEDIAPIPE И НЕЙРОСЕТИ LSTM

Б. А. Ассанович[1], Н. Н. Бич[2], А.К. Пушкина [3]

[1]*Гродненский государственный университет имени Я Купалы, Ожешко, 22, 230000, г. Гродно, Беларусь,* bas@grsu.by
[2]*Гродненский государственный университет имени Я Купалы, Ожешко, 22, 230000, г. Гродно, Беларусь,* nadaliya@mail.ru
[3]*Гродненский государственный университет имени Я Купалы, Ожешко, 22, 230000, г. Гродно, Беларусь,* nadaliya@mail.ru

Представлена методика реализации распознавания жестовых команд в видеопотоке, основанная на выделении ключевых точек кистей рук с использованием инструмента MediaPipe и распознавании жестов за счет обучения и классификации на основе рекуррентной нейронной сети LSTM и позволившая достичь обученной модели около 90% на собственных данных.

Ключевые слова: Жестовые команды; ключевые точки; MediaPipe; LSTM.

RECOGNITION OF GESTURE COMMANDS BASED ON MEDIAPIPE INSTRUMENT AND LSTM NEURAL NETWORK

B. B. Assanovich[a], N.N. Bich[b], A.K. Pushkina[c]

[a] *Grodno State University named after Ya Kupala, Ozheshko, 22, 230000, Grodno, Belarus*, bas@grsu.by
[b] *Grodno State University named after Ya Kupala, Ozheshko, 22, 230000, Grodno, Belarus*, nadaliya@mail.ru
[c] *Grodno State University named after Ya Kupala, Ozheshko, 22, 230000, Grodno, Belarus*
Corresponding author: nadaliya@mail.ru

A technique for implementing the recognition of gesture commands in a video stream is presented, based on the selection of key points of the hands using the MediaPipe tool and gesture recognition through training and classification based on the LSTM recurrent neural network, which made it possible to achieve a trained model of about 90% on its own data.

Keywords: Gesture commands; key points; MediaPipe; LSTM.

Введение

Жестовые команды находят широкое применение в системах управления. Существует ряд методик детектирование статических и динамических жестов на основе цветовых моделей. Однако, при изменении осве-

щения возникают как пропуски фиксации жестов, так и неверная их идентификация.

Обзор литературных источников [1–11] по теме исследования показал, что программные алгоритмы, умеющие распознавать в видеопотоке человеческую ладонь с жестами и координировать ими, в настоящее время активны, но нуждаются в узконаправленном аппаратном обеспечении (мощных графических процессорах) или сложны в реализации как интерактивные программы для мобильных систем из-за ограничений платформ.

Целью данного исследования является разработка программного интерфейса HMI (от англ. Human-Machine interface – интерфейс Человек-Машина) для распознавания жестов в видеопотоке для выполнения команд, согласно распознанному жесту.

Для достижения данной цели решались следующие задачи:

- изучение алгоритмов распознавания жестов, основанных на методах машинного обучения с использованием ключевых точек для распознавания кистей рук на изображении или с использованием самого изображения;

- разработка и реализация алгоритма и программного обеспечения для распознавания жестовых команд на основе следующих программных компонентов: Google Mediapipe Hands, numpy, pandas, matplotlib, seaborn, opencv2; Python; tensorflow/Keras и sklearn, рекуррентная нейронная сеть LSTM.

Новизна исследования заключается в использовании совокупности специализированных программных инструментов и библиотек для сбора данных и создании своего датасета жестов, использования интерполяции при пропусках жестов в кадрах, создание модели на основе нейросети, обучения ее и получения точности распознавания 89% на тестовой выборке.

1. Методология исследования / теоретические основы

Алгоритм лежащий в основе инструмента Mediapipe выполняет обработку видеокадров с найденными ключевыми точками кисти путем пересчета 2D координат проекции ладони руки в 3D координаты на основе предположения о виртуальной камере, в которой плоскость расположена на расстоянии $Z=f$

$$\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \sim \begin{pmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}, \quad (1)$$

где x, y координаты элементов плоского изображения, а X, Y, Z абсолютные 3D координаты камеры.

При известной матрице преобразования камеры вычисляются как значения координат ключевых точек, так и их тепловые карты, содержащие вероятности смещения этих значений относительно координат центра теплового облака x_c, y_c

$$H_k(x, y) = \exp\Big(- \big(\frac{(x - x_c)^2}{2\sigma^2} + \frac{(y - y_c)^2}{2\sigma^2}\big)\Big),$$

Далее была разработана программная реализация системы согласно структурной схеме (рисунок 1).

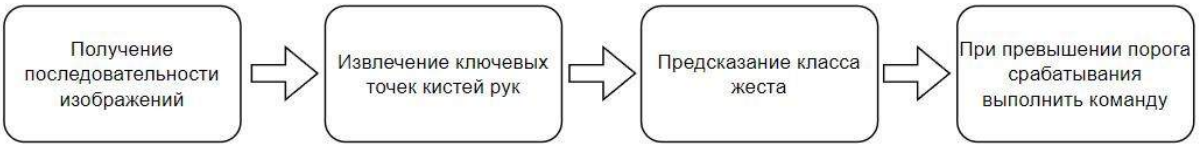

Рисунок 1 – Структурная схема системы

Для извлечения ключевых точек кистей рук реализована функция, на вход которой принимаются: объекты классов Hands (фреймворка MediaPipe), Drawing, DrawingStyles из MediaPipe, набор жестов, количество записей и их размер, путь сохранения файлов, минимальная граница детектирования (по умолчанию 0.5 или 50%), флаг установки статического режима (по умолчанию False).

В случае отсутствия ключевых точек кистей руки на изображении массив координат заменяется массивом, наполненным нулями.

В начале сбора из библиотеки cv2 получается объект, указанный на этапе конфигурации камеры. Далее начинается итерация по указанным жестам. Для каждого жеста итерируется запись, для каждой записи итерируется кадр указанное количество раз.

В каждой итерации кадра проверяется успешность получения изображения с камеры. В случае неудачи, попытка повторяется. Далее происходит вычисление времени с получения предыдущего кадра и вычисляется количество кадров в секунду (FPS).

На основе полученных данных ключевые точки кистей рук и связи между ними отображаются на изображении. Также на изображении выводится информация о текущем жесте, номере записи.

По окончании итераций окна закрываются, и работа функции завершается.

Также, для «холостой» работы алгоритма, разработана функция, работающая аналогично, за исключением сохранения результата. Эту функцию можно назвать демонстрирующей.

Благодаря инструменту QtDesigner и библиотеке PyQt5 сбор данных имеет следующий интерфейс (рисунок 2).

Рисунок 2 – Окно сбора данных приложения

Собранные данные представляют собой набор из 6 жестов (Вверх, Вниз, Влево, Вправо, Назад, ОК), по 30 записей, по 40 кадров, в сумме 7200 .npy файлов (Библиотеки Numpy для Python), который состоит из 126 чисел с плавающей точкой, описывающих 3D-координаты двух рук по 21 ключевой точке.

Коллекция обнаруженных/отслеженных рук, где каждая рука представлена в виде списка из 21 ориентира руки, и каждый ориентир состоит из **x, y, z**. Где **x** и **y** нормализуются от 0.0 до 1.0 по ширине и высоте изображения соответственно. А **z** представляет глубину ориентира, причем глубина на запястье является началом координат, и чем меньше значение, тем ближе ориентир к камере. Величина **z** вычисляется примерно того же масштаба, **x** что и **y.**

Далее происходит разделение данных по координатам на наборы X, Y и Z соответственно. Для визуализации реализована функция, принимающая координаты ключевых точек кисти руки, связи ключевых точек, флаги сохранения фиксированного масштаба координатных осей (fixed_axes) и динамического изменения точки зрения (dynamic_view). В зависимости от значения флага save результат сохраняется. В конечном

итоге с помощью библиотеки MatPlotLib получаем визуализацию каждого отдельного кадра.

Исходя из объёма и формата имеющихся данных (6 жестов, по 30 записей, по 40 кадров каждая) была сконфигурирована модель рекуррентной нейронной сети LSTM. На протяжении всей структуры модели основными функциями активации являются так называется функция ReLU и функция Softmax, выполняющая активацию в суммирующем финальном слое.

Реализованное приложение имеет следующий интерфейс (рисунок 3):

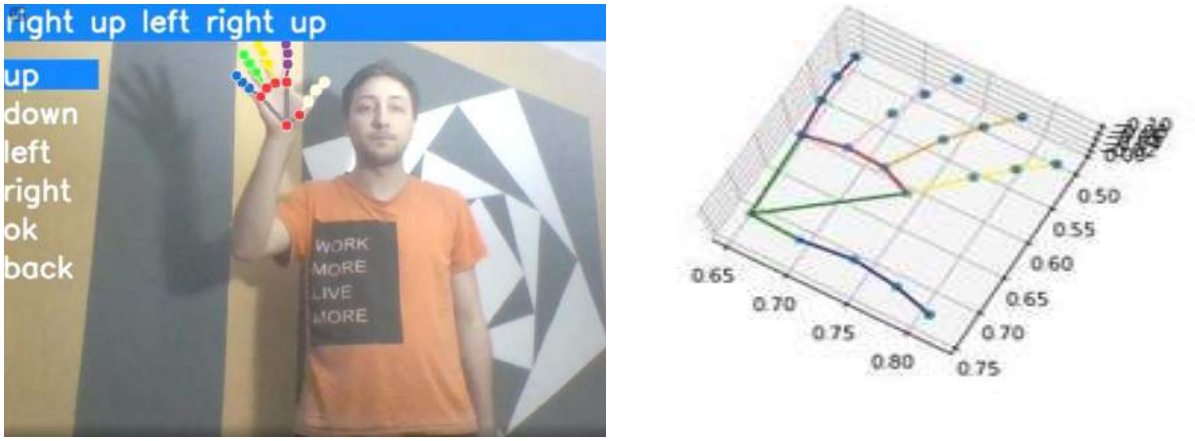

Рисунок 3 - Пример выполнения жестовых команд

2. Результаты и их обсуждение

Реализованное приложение с точностью 88.(8)% определяет команду. Минусом разработки являются баги, выявленные на момент тестирования приложения требования к вычислительным способностям компьютера и чёткости и скорости камеры.

Заключение

В статье описана методика обработки скелетного представления руки для распознавания жеста, а реализация как подготовки собственного датасета, так и тестирование модели нейросети LSTM для распознавания жестовых команд. Полученная точность обученной модели составила около 90%.

Библиографические ссылки

1. Krzysztof R. Classification Algorithm for Person Identification and Gesture Recognition Based on Hand Gestures with Small Training Sets // Sensors. 2020. № 20(24). P. 7279. DOI: 10.3390/s20247279.

2. Guillaume D., Wang X., Fabien M., Jie Y. Deep Learning for Hand Gesture Recognition on Skeletal Data // 13th IEEE Conference on Automatic Face and Gesture Recognition (FG'2018).2018. № 13(15). P. 106–113. DOI: 10.1109/FG.2018.00025; hal-01737771.
3. Sriram S.K., Nishant S. Gestop: Customizable Gesture Control of Computer Systems // 8th ACM IKDD CODS and 26th COMAD (CODS COMAD 2021). 2021. № 26(58). P. 405–409. DOI:10.1145/3430984.3430993.
4. Chenyang L., Xin Z., Lufan L., Lianwen J., Weixin Y. Skeleton-based Gesture Recognition Using Several Fully Connected Layers with Path Signature Features and Temporal Transformer Module // The Thirty-Third AAAI Conference on Artificial Intelligence (AAAI-19). 2019. № 33(1053). P. 8585–8593. DOI: 10.1609/aaai.v33i01.33018585.
5. Ryumin D., Kagirov I., Ivanko D., Axyonov A., Karpov A.A. Automatic detection and recognition of 3D manual gestures for human-machine interaction // ISPRS International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences. 2019. № 42(12). P. 179-183. DOI: 10.5194/isprs-archives-XLII-2-W12-179-2019.
6. Grif M.G., Kondratenko Y.K. Development of a software module for recognizing the fingerspelling of the Russian Sign Language based on LSTM // International Conference on IT in Business and Industry. 2021. № 2032. P. 012024. DOI: 10.1088/1742-6596/2032/1/012024.
7. Zhang F., Bazarevsky V., Vakunov A., Tkachenka A., Sung G., Chang C. MediaPipe Hands: On-device Real-time Hand Tracking // CVPR Workshop on Computer Vision for Augmented and Virtual Reality. 2020. № 20(15). P. 77-81. DOI: 10.48550/arXiv.2006.10214.
8. Caputo A., Giachetti A., Soso S., Pintani D., D'Eusanio A., Pini S. SHREC 2021: Track on Skeleton-based Hand Gesture Recognition in the Wild // Computers & Graphics. 2021. № 99(4). P. 50-62. DOI:10.1016/j.cag.2021.07.007.
9. Yasen M., Jusoh S. A systematic review on hand gesture recognition techniques, challenges and applications // PeerJ Computer Science. 2019. № 5(6). P. 218-248. DOI: 10.7717/peerj-cs.218.
10. Sarma D., Bhuyan M. K. Methods, Databases and Recent Advancement of Vision Based Hand Gesture Recognition for HCI Systems: A Review // SN Computer Science. 2021. № 2(436). P. 1-40. DOI: 10.1007/s42979-021-00827-x.
11. Huang G., Tran S., Bai Q., Alty J. Hand gesture detection in tests performed by older adults // Computer Vision and Pattern Recognition. 2021. № 2110(1). P. 1146. DOI: 10.48550/arXiv.2110.14461.