НАУЧНО-ТЕХНИЧЕСКИЙ ВЕСТНИК ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ
март–апрель 2024 Том 24 № 2 http://ntv.ifmo.ru/
SCIENTIFIC AND TECHNICAL JOURNAL OF INFORMATION TECHNOLOGIES, MECHANICS AND OPTICS
March–April 2024 Vol. 24 No 2 http://ntv.ifmo.ru/en/
ISSN 2226-1494 (print) ISSN 2500-0373 (online)

doi: 10.17586/2226-1494-2024-24-2-241-248

ViSL One-shot: generating Vietnamese sign language data set

Khanh Dang[1]✉, Igor A. Bessmertny[2]

[1,2] ITMO University, Saint Petersburg, 197101, Russian Federation

[1] dangkhanhmta.2020@gmail.com✉, https://orcid.org/0009-0009-5882-7653
[2] bessmertny@itmo.ru, https://orcid.org/0000-0001-6711-6399

Abstract

The development of methods for automatic recognition of objects in a video stream, in particular, recognition of sign language, requires large amounts of video data for training. An established method of data enrichment for machine learning is distortion and noise. The difference between linguistic gestures and other gestures is that small changes in posture can radically change the meaning of a gesture. This imposes specific requirements for data variability. The novelty of the method lies in the fact that instead of distorting frames using affine image transformations, vectorization of the sign language speaker's pose is used, followed by noise in the form of random deviations of skeletal elements. To implement controlled gesture variability using the MediaPipe library, we convert to a vector format where each vector corresponds to a skeletal element. After this, the image of the figure is restored from the vector representation. The advantage of this method is the possibility of controlled distortion of gestures, corresponding to real deviations in the postures of the sign language speaker. The developed method for enriching video data was tested on a set of 60 words of Indian Sign Language (common to all languages and dialects common in India), represented by 782 video fragments. For each word, the most representative gesture was selected and 100 variations were generated. The remaining, less representative gestures were used as test data. The resulting word-level classification and recognition model using the GRU-LSTM neural network has an accuracy above 95 %. The method tested in this way was transferred to a corpus of 4364 videos in Vietnamese Sign Language for all three regions of Northern, Central and Southern Vietnam. Generated 436,400 data samples, of which 100 data samples represent the meaning of words that can be used to develop and improve Vietnamese sign language recognition methods by generating many variations of gestures with varying degrees of deviation from the standards. The disadvantage of the proposed method is that the accuracy depends on the error of the MediaPipe library. The created video dataset can also be used for automatic sign language translation.

Keywords

Vietnamese sign language, Indian Sign Language, sign language recognition, MediaPipe, coordinate transformation, vector space, random noise, GRU-LSTM, one-shots, data augmentation

For citation: Dang Khanh, Bessmertny I.A. ViSL One-shot: generating Vietnamese sign language data set. *Scientific and Technical Journal of Information Technologies, Mechanics and Optics*, 2024, vol. 24, no. 2, pp. 241–248. doi: 10.17586/2226-1494-2024-24-2-241-248

УДК 004.932.72'1, 004.852

ViSL One-shot: генерация набора данных вьетнамского языка жестов

Хань Данг[1]✉, Игорь Александрович Бессмертный[2]

[1,2] Университет ИТМО, Санкт-Петербург, 197101, Российская Федерация

[1] dangkhanhmta.2020@gmail.com✉, https://orcid.org/0009-0009-5882-7653
[2] bessmertny@itmo.ru, https://orcid.org/0000-0001-6711-6399

Аннотация

Введение. Разработка методов автоматического распознавания объектов в видеопотоке, в частности распознавания жестового языка, требует больших объемов видеоданных для обучения. Устоявшимся методом обогащения данных для машинного обучения является искажение и зашумление. Отличие языковых жестов от других жестов состоит в том, что небольшие изменения позы могут радикально менять смысл жеста. Это накладывает специфические требования к вариативности данных. **Метод.** Новизна метода состоит в том, что

вместо искажений кадров с помощью афинных преобразований изображений используется векторизация позы сурдодиктора с последующим зашумлением в виде случайных отклонений элементов скелета. Для реализации управляемой вариативности жестов с помощью библиотеки MediaPipe жест преобразуется в векторный формат, где каждый вектор соответствует элементу скелета. Далее выполняется восстановление изображения фигуры из векторного формата. Достоинством предложенного метода является возможность управляемого искажения жестов, соответствующего реальным отклонениям поз сурдодиктора. **Основные результаты.** Разработанный метод обогащения видеоданных протестирован на наборе из 60 слов индийского языка жестов (общего для всех языков и диалектов, распространенных на территории Индии), представленных 782 видеофрагментами. Для каждого слова выбран наиболее репрезентативный жест и сгенерировано 100 вариаций. Остальные, менее репрезентативные жесты, использованы в качестве тестовых данных. В результате получена модель классификации и распознавания на уровне слов с использованием нейронной сети GRU-LSTM с точностью выше 95 %. Метод апробирован на наборе данных из 4364 видео на вьетнамском языке жестов для трех регионов Северного, Центрального и Южного Вьетнама. Сгенерировано 436 400 образцов данных, из которых 100 образцов представляют значения слов, которые могут использоваться для разработки и совершенствования методов распознавания языка жестов на вьетнамском языке за счет генерации множества вариаций жестов с разной степенью отклонения от эталонов. **Обсуждение.** Недостатком предложенного метода является зависимость точности от ошибки библиотеки MediaPipe. Создаваемый набор видеоданных может также использоваться для автоматического сурдоперевода.

Ключевые слова

вьетнамский язык жестов, индийский язык жестов, распознавание языка жестов, MediaPipe, преобразование координат, векторное пространство, GRU-LSTM, обогащение данных

Ссылка для цитирования: Данг Х., Бессмертный И.А. ViSL One-shot: генерация набора данных вьетнамского языка жестов // Научно-технический вестник информационных технологий, механики и оптики. 2024. Т. 24, № 2. С. 241–248 (на англ. яз.). doi: 10.17586/2226-1494-2024-24-2-241-248

Introduction

Artificial intelligence technology used to solve sign language processing problems, such as sign language recognition or sign language interpretation, has deep humanistic significance in order to reduce communication distances between people with hearing impairments and society. The main problem here is the lack of representative training datasets. Popular and published sign language video datasets include: Word-Level American Sign Language [1, 2], including 2000 American Sign Language signs; RWTH-PHOENIX-Weather 2014T [3] — dataset of German Sign Language. The most representative dataset is the Indian Sign Language dataset [4] which contains 4287 videos depicting 263 words with 15 different word categories that will be further used to test the method developed by the authors.

It should be noted that all available video data with gestures is formed in the form of a sign language dictionary and is intended primarily for training deaf people and sign language interpreters and cannot be used for training automatic gesture recognition systems because the number of samples is too small and will lead to overfitting [5]. To create a dataset for use in a deep learning model for a sign language recognition task, you can use the following methods:

— Manual method involving a large number of actors who speak sign language, with different body sizes at different ages. This method produces very good data sets, but requires a lot of time and effort. There is currently no published sign language dataset that meets the above standards.
— Automatic method based on Generative Adversarial Network (GAN), including discriminatory and generative networks [6], where the discriminatory network is responsible for trying to distinguish real data from fake ones, and the generative network generates fake data, and the goal is to generate data that is most similar to the real ones, which makes the discriminator indistinguishable [7]. GAN technology allows you to generate another face from the original face [8], various human poses [9], learn modeling from individual images and videos [10], and create new data samples containing invisible objects [11]. These techniques implement affine transformations and do not reflect the real variability of sign language gestures where the deflection angles of skeletal elements and movement trajectories play a decisive role.
— The method for generating simulations of poses and gesture forms uses calculations of vector coordinate transformations in space and the addition of random noise based on the features of frame coordinate points extracted from the MediaPipe library. This method generates a large number of data samples, reduces computation time, and is not complex but very efficient. The characteristic of this method is that data samples are generated from an initial data sample (one-shots). The details of the method are presented in the next section.

Presentation of the Research Problem

Extracting coordinate features of gestures in sign language videos using the MediaPipe library

MediaPipe[1] is a fast, compact and powerful solution for solving artificial intelligence problems, such as object detection, hand landmark detection, gesture recognition, image generation, etc. [12–16]. In our study, the MediaPipe library will be used to extract the coordinates of objects associated with sign language gestures. Fig. 1, *a*, *b* describes the location coordinates of the characteristic

[1] Available at: https://developers.google.com/mediapipe (accessed: 22.11.2023).

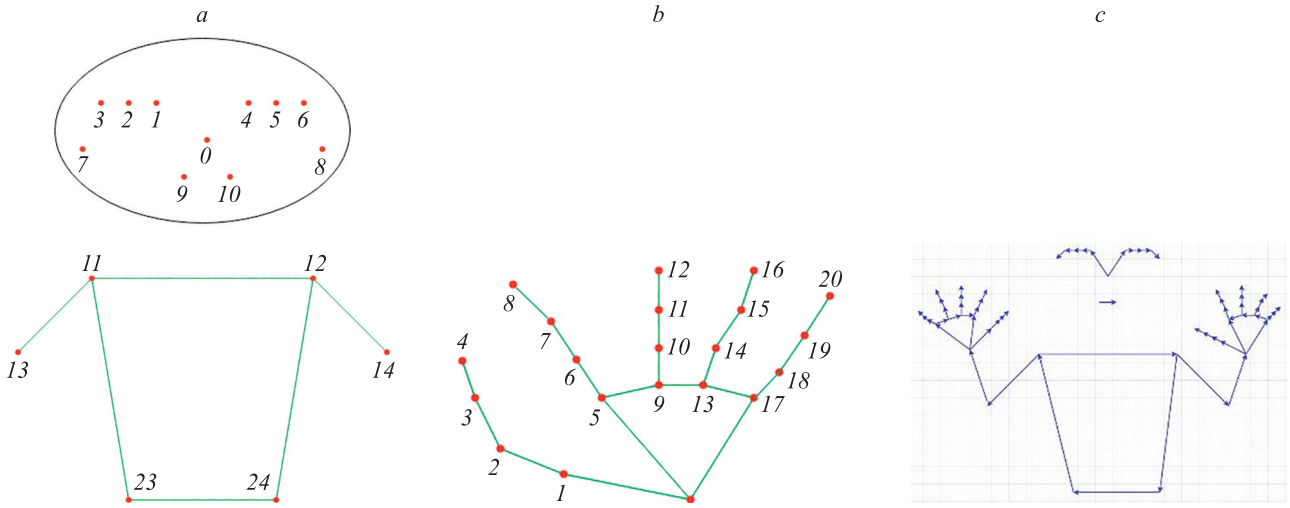

Fig. 1. Extraction of coordinate features of sign language gestures in video (*a*, *b*) and representation in vector space (*c*).
1–24 — skeleton key points

points of interest to us. Fig. 1, *c* converts coordinate objects into vector space.

We can summarize the main research problem of creating a dataset from one-off data as follows (Fig. 2).

The input sign language video sample will be processed on every frame. At time t, corresponding to the i^{th} frame, the MediaPipe library will extract the coordinates $[x, y, z]$, where the X, Y coordinates are local to the region of interest and range from [0.0, 255.0]. The Z coordinate is measured in "image pixels" as the X and Y coordinates and represents the distance relative to the subject's hip plane which is the origin of the Z axis. A K-means clustering model is used to ensure consistency in the number of frames characterizing gestures in a sign language video, and to set the input parameters for a neural network model processing time series data. After obtaining data frames typical of a video, we proceed to compute a vector coordinate transformation in space to generate video samples that mimic the original standard video. The problem with generating multiple video samples that mimic the original standard video is the following.

Available data. A sign language dictionary with the meaning of each word provided by a sign language expert. We consider these to be standard sign language gestures. To resemble the process of simulating real-life activities, the original sample video was named "teacher video". The created video samples are video simulators.

Task. It is necessary to create videos of imitators imitating the teacher's gestures.

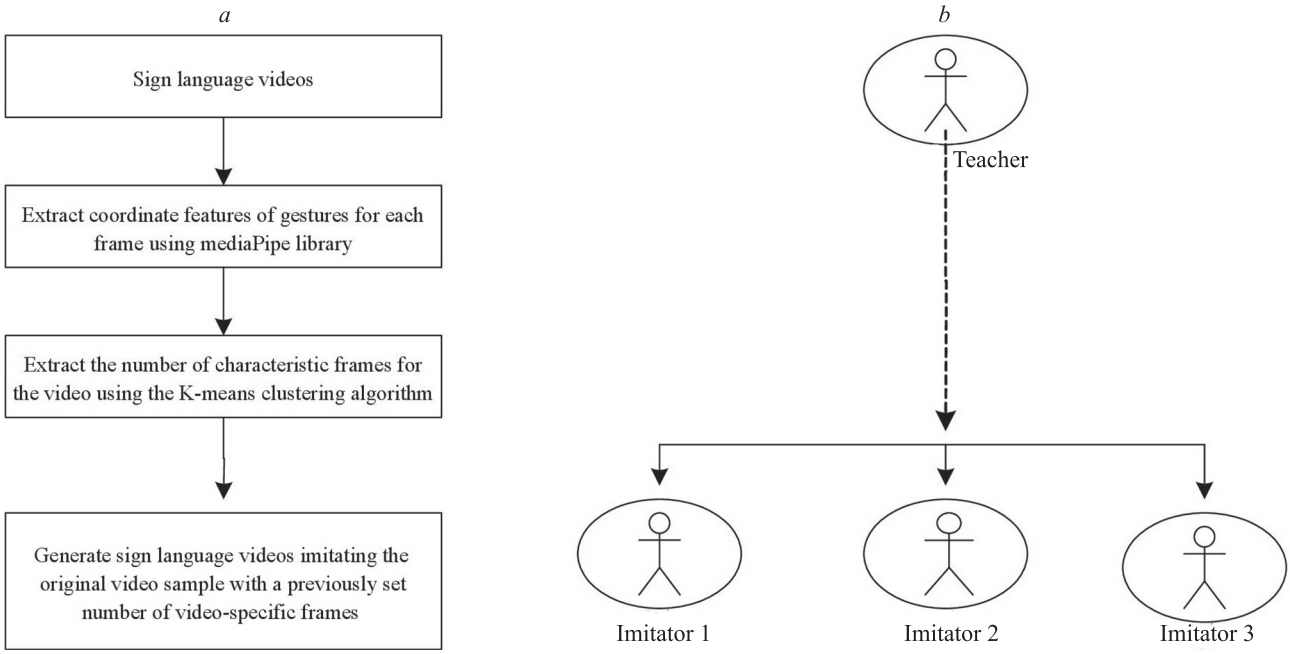

Fig. 2. Summary of the process of generating sign language data samples from an input video sample (*a*) and simulating the generation of gestures that imitate the teacher's gestures (*b*)

Data samples have the following two important properties:

Property 1. Different physical parameters of sign language speakers;

Property 2. Different amplitudes of deviations of skeletal elements.

Based on the above two properties, to generate a data sample according to property 1, we will use the method of calculating and transforming coordinates in vector space. To generate a data sample according to property 2, we need to add random noise to the characteristic coordinate points obtained after building properties 1 & 2. In the next section, we will analyze and present this method in detail.

Generating multiple samples of sign language video data from the original sample

Generating imitator gesture samples based on the original standard gesture samples by computing coordinate transformations in vector space

Let $\mathbf{A}_t$ be the point space of gesture feature locations in the sign language video, extracted at the t^{th} frame in the input video. $\mathbf{\Phi}$ is the selected association map that turns 2 points into vector $\mathbf{A}_{i,j}$.

We can then write in general form:

$$\mathbf{\Phi}: \mathbf{A}_t \times \mathbf{A}_t \to \mathbf{V}_t,$$

where $\mathbf{V}_t$ is the vector space at the t^{th} frame.

According to the rule for selecting linked coordinate points in the vector space at the t^{th} frame, every 2 consecutive points in the index order set in the MediaPipe library are listed in Fig. 1, *a* will form a vector. Then, choosing k consecutive points will represent a geometric feature of a sign language gesture. In fact, this geometric feature characterizes the movement tendency of an object in space.

The $(\mathbf{v}_{t'}^0, \mathbf{v}_{t'}^1, \mathbf{v}_{t'}^2, \ldots, \mathbf{v}_{t'}^k \in \mathbf{V}_{t'})$ are vectors in the vector space $\mathbf{V}_t$, and $(\mathbf{v}_t^0, \mathbf{v}_t^1, \mathbf{v}_t^2, \ldots, \mathbf{v}_t^k \in \mathbf{V}_t)$ are vectors in the vector space $\mathbf{V}_{t'}$ at the t and t'^{th} frames extracted from the given standard sample video.

There exist the mappings $\mathbf{F} = (f_{tt'}^0, f_{tt'}^1, f_{tt'}^2, \ldots, f_{tt'}^k)$ with linear transformations:

$$f^i(\mathbf{v}_t^i) \mapsto \mathbf{v}_{t'}^i.$$

Similar to the sign language gesture video of the generated imitator, we also have: $(\mathbf{u}_t^0, \mathbf{u}_t^0, \mathbf{u}_t^0, \ldots, \mathbf{u}_t^0 \in \mathbf{U}_t)$ which are vectors in the vector space $\mathbf{U}_t$, and $(\mathbf{u}_{t'}^0, \mathbf{u}_{t'}^0, \mathbf{u}_{t'}^0, \ldots, \mathbf{u}_{t'}^0 \in \mathbf{U}_{t'})$ are vectors in the vector space $\mathbf{U}_{t'}$ at the t^{th} frame and t' are generated based on the t^{th} and t'^{th} frames of the input standard video. There exist the mappings $\mathbf{G} = (g_{tt'}^0, g_{tt'}^1, g_{tt'}^2, \ldots, g_{tt'}^k)$ with linear transformations: $g^i(\mathbf{u}_t^i) \mapsto \mathbf{u}_{t'}^i$.

We need to generate a video of the imitator with the number of frames and frame order being the same as the typical number and frame order of the teacher's video. Considering at the t^{th} frame, choose n consecutive vectors in one frame of the teacher's video and the imitator video to form a geometric shape. When having the same viewing angle and the same linear transformation, the property of distance between two points is preserved. So, to compare the geometrical similarity of the teacher and the imitator, we will compare the angular deviation between pairs of corresponding vectors. The error in geometric similarity is calculated according to the following formula:

$$£ = \left[1 + \frac{abs((\mathbf{v}_t^0, \mathbf{v}_t^1) - (\mathbf{u}_t^0, \mathbf{u}_t^1))}{(\mathbf{v}_t^0, \mathbf{v}_t^1)}\right] \times$$

$$\times \left[1 + \frac{abs((\mathbf{v}_t^1, \mathbf{v}_t^2) - (\mathbf{u}_t^1, \mathbf{u}_t^2))}{(\mathbf{v}_t^1, \mathbf{v}_t^2)}\right] \ldots$$

$$\ldots \left[1 + \frac{abs((\mathbf{v}_t^{n-1}\mathbf{v}_t^n) - (\mathbf{u}_t^{n-1}\mathbf{u}_t^n))}{(\mathbf{v}_t^{n-1}\mathbf{v}_t^n)}\right],$$

where $(\mathbf{v}_t^0, \mathbf{v}_t^1)$ is the angle between two vectors ($\mathbf{v}_t^0$ and $\mathbf{v}_t^1$); $abs((\mathbf{v}_t^0, \mathbf{v}_t^1) - (\mathbf{u}_t^0, \mathbf{u}_t^1))$ is the absolute value of the angle difference between $(\mathbf{v}_t^0, \mathbf{v}_t^1)$ and $(\mathbf{u}_t^0, \mathbf{u}_t^1)$.

Maximum geometric similarity is achieved when $(\mathbf{u}_t^i, \mathbf{u}_t^{i+1}) = (\mathbf{v}_t^i, \mathbf{v}_t^{i+1})$ then:

$$\mathbf{u}_t^{i+1} = \lambda \mathbf{v}_t^{i+1}, \lambda > 0. \quad (1)$$

Calculate the value of coefficient λ. According to the property of preserving the length relationship between vector $\mathbf{u}_t^i$ and $\mathbf{v}_t^i$ we have:

$$\frac{|\mathbf{u}_t^i|}{|\mathbf{u}_t^{i+1}|} = \frac{|\mathbf{v}_t^i|}{|\mathbf{v}_t^{i+1}|}, \quad (2)$$

where $|\mathbf{u}_t^i|$ is the length of vector $\mathbf{u}_t^i$.

From (1) and (2) we can calculate:

$$\lambda = \frac{|\mathbf{u}_t^i|}{|\mathbf{v}_t^i|}.$$

Adding a noise to each point after the calculation

The task is performed in three-dimensional coordinate space *Oxyz*. Adding noise is intended to bring the calculation results closer to reality. Noise is the error between the MediaPipe library point detection prediction accuracy, the similarity between the simulator and the original standard gesture, and the error due to perspective changes. Noise will be added to the point $M(x, y, z)$ and the point $M'(x + \varepsilon, y + \varepsilon, z + \varepsilon)$ will be formed, where ε = random (–σ, σ), σ is the threshold value of adding random noise. A larger threshold value means a larger discrepancy.

Testing and results

To evaluate the effectiveness of the proposed method, we build a word-level sign language recognition model with a set of input data in the form of videos in Indian Sign Language and a Gated Recurrent Unit-Long Short-Term Memory (GRU-LSTM) neural network [17] used for sign language recognition in time series. The choice of the Indian data set (the same for all languages and dialects in India) for testing is due to its greatest representativeness.

The MediaPipe library was used to extract the coordinate features of the data points. The data set is published on the website zenodo.org[1].

We randomly select 60 words to test. Each segment is a video of one ISL sign recorded by deaf students from St. Louis School for the Deaf, Adyar, Chennai. For each word there are 15–20 data samples. For each word, we take only one data sample corresponding to one video. The remaining input videos are used as the test set. We consider the video selection as a given standard gesture.

Training data set. We use the MediaPipe library to extract 75 sign language gesture features for each frame. Each object point is given as x, y, z coordinates, so the data size is $75 \times 3 = 225$. 20 frames will be extracted from each video using K-means clustering method. We generate 100 data samples for each word. To generate data samples for a word, we include images to obtain the initial source frame size for the simulator. Then we calculate the vector transformation in space to create the next frames. In addition, we also set random scale values to increase the amount of data in the model. So the size of the training dataset is (6000, 20, 225).

Test dataset. The MediaPipe library is used to extract sign language features from videos and is stored as a numpy file. Since the number of videos collected varies, the total number of videos we tested was 782 videos of 60 Indian Sign Language words. Test data set size: (782, 20, 225).

We use the GRU neural network to train a model to recognize Indian Sign Language at the word level. The GRU-LSTM neural network is a compressed variation that improves the computation speed faster than the LSTM neural network [18–20]. When processing time series data in a GRU network structure, the deleted element helps to capture short-term dependencies in the time series, and the update element helps to capture long-term dependencies in the time series. The training and testing results are shown in Fig. 3 and Fig. 4.

The graph shows a big difference in the changes in the values of the training data and test data. This is not overfitting. The reason is that the test set data is very small compared to the training set data, in addition, the data in the test set is not evenly distributed in quantity. During training, we choose the batch size value to be 256. This

[1] Available at: https://zenodo.org/records/4010759 (accessed: 22.11.2023).

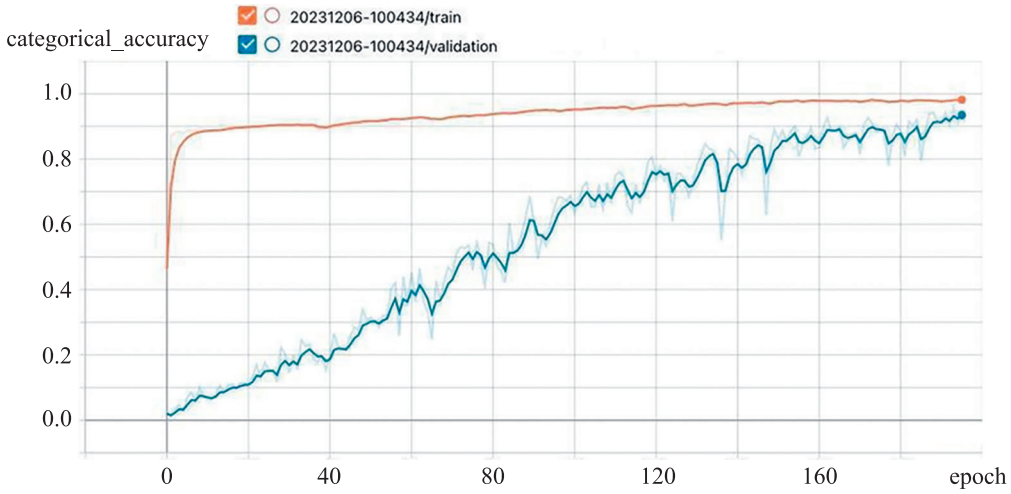

Fig. 3. Accuracy of the model training process over each epoch

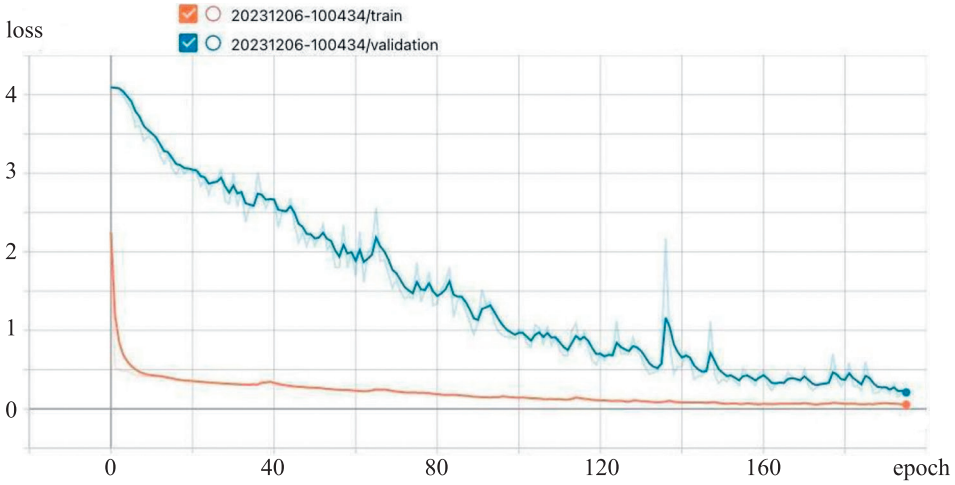

Fig. 4. Value of loss over each epoch during model training

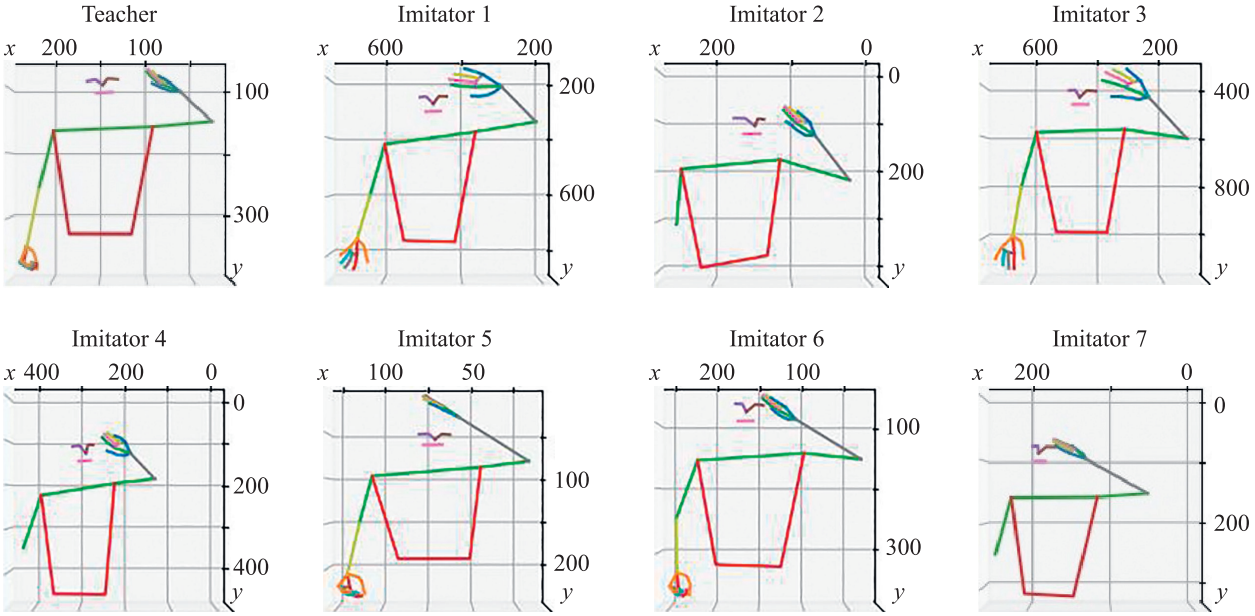

Fig. 5. Example showing different samples for the same gesture in the Vietnamese sign language data set

value is much larger than the number of class samples in the test set. The result of the model accuracy evaluation is that 745 videos were correctly classified out of a total of 782 input videos, which is equivalent to 95.26 %. Detailed information about the training process is provided in the following link[1].

The method tested in this way allows us to generate multiple data samples for the Vietnamese Sign Language (ViSL) dataset. The Vietnamese Sign Language dictionary contains 4364 words, each word contains a sample of data. We enriched the dataset by generating 100 data samples from each word. The Vietnamese Sign Language dataset contains 436,400 samples describing the meanings of 4,364 words in Vietnamese Sign Language, the data samples are stored as numpy files. Fig. 5 example shows different samples for the same gesture "Hello!" in the Vietnamese sign language data set.

Vietnamese Sign Language dataset and code downloaded and updated here[2].

[1] Available at: https://github.com/DangKhanhITMO/oneshotsViSL/blob/main/ViSL_v2.ipynb (accessed: 22.11.2023).

[2] Available at: https://github.com/DangKhanhITMO/oneshotsViSL (accessed: 06.12.2023).

Conclusion and discussion

In this study, the MediaPipe library was used to vectorize sign language gestures in videos which then became the basis for generating similar data samples. The advantage of this method is that it does not require large computing resources or computer configuration. Using the method described above, it is possible to create an unlimited number of data samples of different sizes, which corresponds replacing people when creating many data samples of different sizes and ages. In this study, we only consider data generation for sign language. This approach can be extended to other tasks related to human actions and postures. The disadvantage of this method is that determining the additional noise for the calculation process requires several samples to adjust the error threshold accordingly. The accuracy of the problem solution largely depends on the similarity of the geometric position of the input image compared to the first frame in the input video after selecting the function and depends on the error of the MediaPipe library. Future research direction: creating a machine translation application for Vietnamese Sign Language based on the created dataset. In addition, the above method, combined with automatic pose generation, can generate realistic videos from input sample images and videos.

References

1. Li D., Yu X., Xu C., Petersson L., Li H. Transferring Cross-domain Knowledge for Video Sign Language Recognition. *Proc. of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 6204–6213. https://doi.org/10.1109/cvpr42600.2020.00624
2. Li D., Opazo C.R., Yu X., Li H. Word-level deep sign language recognition from video: A new large-scale dataset and methods comparison. *Proc. of the IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2020, pp. 1448–1458. https://doi.org/10.1109/WACV45572.2020.9093512

Литература

1. Li D., Yu X., Xu C., Petersson L., Li H. Transferring Cross-domain Knowledge for Video Sign Language Recognition // Proc. of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). 2020. P. 6204–6213. https://doi.org/10.1109/cvpr42600.2020.00624
2. Li D., Opazo C.R., Yu X., Li H. Word-level deep sign language recognition from video: A new large-scale dataset and methods comparison // Proc. of the IEEE Winter Conference on Applications of Computer Vision (WACV). 2020. P. 1448–1458. https://doi.org/10.1109/WACV45572.2020.9093512

3. Camgoz N.C., Hadfield S., Koller O., Ney H., Bowden R. Neural sign language translation. *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7784–7793. https://doi.org/10.1109/CVPR.2018.00812
4. Sridhar A., Ganesan R.G., Kumar P., Khapra M. INCLUDE: A large scale dataset for indian sign language recognition. *Proc. of the 28th ACM International Conference on Multimedia*, 2020, pp. 1366–1375. https://doi.org/10.1145/3394171.3413528
5. Ying X. An overview of overfitting and its solutions. *Journal of Physics: Conference Series*, 2019, vol. 1168, no. 2, pp. 022022. https://doi.org/10.1088/1742-6596/1168/2/022022
6. Creswell A., White T., Dumoulin V., Arulkumaran K., Sengupta B., Bharath A. Generative adversarial networks: An overview. *IEEE Signal Processing Magazine*, 2018, vol. 35, no. 1, pp. 53–65. https://doi.org/10.1109/MSP.2017.2765202
7. Gupta K., Singh S., Shrivastava A. PatchVAE: Learning local latent codes for recognition. *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 4745–4754. https://doi.org/10.1109/CVPR42600.2020.00480
8. Karras T., Aila T., Laine S., Lehtinen J. Progressive growing of GANs for improved quality, stability, and variation. *Proc. of the ICLR 2018 Conference Blind Submission*, 2018.
9. Ma L., Jia X., Sun Q., Schiele B., Tuytelaars T., Van Gool L. Pose guided person image generation. *Proc. of the 31st Conference on Neural Information Processing Systems (NIPS 2017)*, 2017.
10. Sushko V., Gall J., Khoreva A. One-shot GAN: Learning to generate samples from single images and videos. *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2021, pp. 2596–2600. https://doi.org/10.1109/CVPRW53098.2021.00293
11. Li J., Jing M., Lu K., Ding Z., Zhu L., Huang Z. Leveraging the invariant side of generative zero-shot learning. *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 7394–7403. https://doi.org/10.1109/CVPR.2019.00758
12. Madrid G.K.R., Villanueva R.G.R., Caya M.V.C. Recognition of dynamic Filipino Sign language using MediaPipe and long short-term memory. *Proc. of the 13th International Conference on Computing Communication and Networking Technologies (ICCCNT)*, 2022. https://doi.org/10.1109/ICCCNT54827.2022.9984599
13. Adhikary S., Talukdar A.K., Sarma K.K. A vision-based system for recognition of words used in Indian Sign Language using MediaPipe. *Proc. of the 2021 Sixth International Conference on Image Information Processing (ICIIP)*, 2021, pp. 390–394. https://doi.org/10.1109/ICIIP53038.2021.9702551
14. Zhang S., Chen W., Chen C., Liu Y. Human deep squat detection method based on MediaPipe combined with Yolov5 network. *Proc. of the 2022 41st Chinese Control Conference (CCC)*, 2022, pp. 6404–6409. https://doi.org/10.23919/CCC55666.2022.9902631
15. Quiñonez Y., Lizarraga C., Aguayo R. Machine learning solutions with MediaPipe. *Proc. of the 11th International Conference on Software Process Improvement (CIMPS)*, 2022, pp. 212–215. https://doi.org/10.1109/CIMPS57786.2022.10035706
16. Ma J., Ma L., Ruan W., Chen H., Feng J. A Wushu posture recognition system based on MediaPipe. *Proc. of the 2nd International Conference on Information Technology and Contemporary Sports (TCS)*, 2022, pp. 10–13. https://doi.org/10.1109/TCS56119.2022.9918744
17. Cho K., Merriënboer B., Gulcehre C., Bahdanau D., Bougares F., Schwenk H., Bengio Y. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *Proc. of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014, pp. 1724–1734. https://doi.org/10.3115/v1/D14-1179
18. Dey R., Salem F.M. Gate-variants of Gated Recurrent Unit (GRU) neural networks. *Proc. of the IEEE 60th International Midwest Symposium on Circuits and Systems (MWSCAS)*, 2017, pp. 1597–1600. https://doi.org/10.1109/MWSCAS.2017.8053243
19. Kothadiya D., Bhatt C., Sapariya K., Patel K., Gil-González A.-B., Corchado J.M. Deepsign: Sign language detection and recognition using deep learning. *Electronics*, 2022, vol. 11, no. 11, pp. 1780. https://doi.org/10.3390/electronics11111780
20. Verma U., Tyagi P., Kaur M. Single input single head CNN-GRU-LSTM architecture for recognition of human activities. *Indonesian Journal of Electrical Engineering and Informatics (IJEEI)*, 2022, vol. 10, no. 2, pp. 410–420. https://doi.org/10.52549/ijeei.v10i2.3475

3. Camgoz N.C., Hadfield S., Koller O., Ney H., Bowden R. Neural sign language translation // Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2018. P. 7784–7793. https://doi.org/10.1109/CVPR.2018.00812
4. Sridhar A., Ganesan R.G., Kumar P., Khapra M. INCLUDE: A large scale dataset for indian sign language recognition // Proc. of the 28th ACM International Conference on Multimedia. 2020. P. 1366–1375. https://doi.org/10.1145/3394171.3413528
5. Ying X. An overview of overfitting and its solutions // Journal of Physics: Conference Series. 2019. V. 1168. N 2. P. 022022. https://doi.org/10.1088/1742-6596/1168/2/022022
6. Creswell A., White T., Dumoulin V., Arulkumaran K., Sengupta B., Bharath A. Generative adversarial networks: An overview // IEEE Signal Processing Magazine. 2018. V. 35. N 1. P. 53–65. https://doi.org/10.1109/MSP.2017.2765202
7. Gupta K., Singh S., Shrivastava A. PatchVAE: Learning local latent codes for recognition // Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). 2020. P. 4745–4754. https://doi.org/10.1109/CVPR42600.2020.00480
8. Karras T., Aila T., Laine S., Lehtinen J. Progressive growing of GANs for improved quality, stability, and variation // Proc. of the ICLR 2018 Conference Blind Submission. 2018.
9. Ma L., Jia X., Sun Q., Schiele B., Tuytelaars T., Van Gool L. Pose guided person image generation // Proc. of the 31st Conference on Neural Information Processing Systems (NIPS 2017). 2017.
10. Sushko V., Gall J., Khoreva A. One-shot GAN: Learning to generate samples from single images and videos // Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW). 2021. P. 2596–2600. https://doi.org/10.1109/CVPRW53098.2021.00293
11. Li J., Jing M., Lu K., Ding Z., Zhu L., Huang Z. Leveraging the invariant side of generative zero-shot learning // Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). 2019. P. 7394–7403. https://doi.org/10.1109/CVPR.2019.00758
12. Madrid G.K.R., Villanueva R.G.R., Caya M.V.C. Recognition of dynamic Filipino Sign language using MediaPipe and long short-term memory // Proc. of the 13th International Conference on Computing Communication and Networking Technologies (ICCCNT). 2022. https://doi.org/10.1109/ICCCNT54827.2022.9984599
13. Adhikary S., Talukdar A.K., Sarma K.K. A vision-based system for recognition of words used in Indian Sign Language using MediaPipe // Proc. of the 2021 Sixth International Conference on Image Information Processing (ICIIP). 2021. P. 390–394. https://doi.org/10.1109/ICIIP53038.2021.9702551
14. Zhang S., Chen W., Chen C., Liu Y. Human deep squat detection method based on MediaPipe combined with Yolov5 network // Proc. of the 41st Chinese Control Conference (CCC). 2022. P. 6404–6409. https://doi.org/10.23919/CCC55666.2022.9902631
15. Quiñonez Y., Lizarraga C., Aguayo R. Machine learning solutions with MediaPipe // Proc. of the 11th International Conference on Software Process Improvement (CIMPS). 2022. P. 212–215. https://doi.org/10.1109/CIMPS57786.2022.10035706
16. Ma J., Ma L., Ruan W., Chen H., Feng J. A Wushu posture recognition system based on MediaPipe // Proc. of the 2nd International Conference on Information Technology and Contemporary Sports (TCS). 2022. P. 10–13. https://doi.org/10.1109/TCS56119.2022.9918744
17. Cho K., Merriënboer B., Gulcehre C., Bahdanau D., Bougares F., Schwenk H., Bengio Y. Learning phrase representations using RNN encoder-decoder for statistical machine translation // Proc. of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). 2014. P. 1724–1734. https://doi.org/10.3115/v1/D14-1179
18. Dey R., Salem F.M. Gate-variants of Gated Recurrent Unit (GRU) neural networks // Proc. of the IEEE 60th International Midwest Symposium on Circuits and Systems (MWSCAS). 2017. P. 1597–1600. https://doi.org/10.1109/MWSCAS.2017.8053243
19. Kothadiya D., Bhatt C., Sapariya K., Patel K., Gil-González A.-B., Corchado J.M. Deepsign: Sign language detection and recognition using deep learning // Electronics. 2022. V. 11. N 11. P. 1780. https://doi.org/10.3390/electronics11111780
20. Verma U., Tyagi P., Kaur M. Single input single head CNN-GRU-LSTM architecture for recognition of human activities // Indonesian Journal of Electrical Engineering and Informatics (IJEEI). 2022. V. 10. N 2. P. 410–420. https://doi.org/10.52549/ijeei.v10i2.3475

Authors

Khanh Dang — PhD Student, ITMO University, Saint Petersburg, 197101, Russian Federation, https://orcid.org/0009-0009-5882-7653, dangkhanhmta.2020@gmail.com
Igor A. Bessmertny — D.Sc., Full Professor, ITMO University, Saint Petersburg, 197101, Russian Federation, sc 36661767800, https://orcid.org/0000-0001-6711-6399, bessmertny@itmo.ru

Авторы

Данг Хань — аспирант, Университет ИТМО, Санкт-Петербург, 197101, Российская Федерация, https://orcid.org/0009-0009-5882-7653, dangkhanhmta.2020@gmail.com
Бессмертный Игорь Александрович — доктор технических наук, профессор, профессор, Университет ИТМО, Санкт-Петербург, 197101, Российская Федерация, sc 36661767800, https://orcid.org/0000-0001-6711-6399, bessmertny@itmo.ru

Received 08.12.2023
Approved after reviewing 06.02.2024
Accepted 14.03.2024

Статья поступила в редакцию 08.12.2023
Одобрена после рецензирования 06.02.2024
Принята к печати 14.03.2024