Application of vision transformers and 3D convolutional neural networks for sign language cluster recognition

Serhii Smirnov[a] and Nataliia Kuznietsova[a]

[a] *National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", ave. Peremohy 37, Kyiv, 03056, Ukraine*

Abstract

In this study the overview of the methods for sign language recognition was done and the existing datasets in this area were analyzed. It was shown how based on the real data to develop and test different approaches. The models based on the Vision Transformer (ViViT) and 3D Convolutions CNN (3dCNN) using different batch sizes were built and compared. It was also shown how to learn models on different data sizes and to search the compromise between accuracy, speed and overfitting of the models. Our research provides valuable insights into the strengths and limitations of different models of the task, not solved tasks and offer a direction and possible improvements of existed methods in this area by using vision transformers.

Keywords 1

Gesture recognition, computer vision, neural networks, deep learning, hand shape recognition, sign language interpreter, vision transformers, 3D convolutions.

1. Introduction

Sign language is a visual language used by people who are deaf or hard of hearing to communicate with each other and with hearing individuals. It involves using a combination of hand gestures, facial expressions, and body language to convey meaning. While sign language is an effective mean of communication, it can be challenging for non-signers to understand and communicate with sign language users. This has led to the development of sign language recognition technology, which uses computer algorithms to interpret and translate sign language into spoken or written language. Sign language recognition has the potential to improve communication and inclusion for people who are deaf or hard of hearing. It also poses unique challenges, such as the necessity for accurate hand shape and movement detection, real-time recognition, and dealing with the complexity and variability of different sign languages. Advantages and new achievements in machine learning, computer vision, and sensor technology give now the possibility to overcome these challenges and make sign language recognition more accurate, efficient, and accessible. Machine learning techniques, such as deep learning, can then be used to learn a mapping between these visual features and the corresponding sign language gestures. While CNNs have limitations in capturing long-term dependencies and global context, which are crucial for complex image understanding tasks such as object detection and segmentation, special transformers have gained significant popularity in the field of computer vision in recent years due to their ability to process sequential data such as images and videos. In this article we will compare two approaches for sign language recognition and define for the real practical task which of them is more effective and perspective for improvements for next studies.

2. Sign language recognition problem statement

Sign language is an essential mode of communication for deaf or hard-of-hearing individuals. Sign language recognition (SLR) is a challenging task, as sign languages are highly complex, with a wide

The Sixth International Workshop on Computer Modeling and Intelligent Systems (CMIS-2023), May 3, 2023, Zaporizhzhia, Ukraine
EMAIL: sergej.smirnov.sss@gmail.com (Smirnov S.); natalia-kpi@ukr.net (Kuznietsova N.)
ORCID: 0000-0002-5495-0680 (Smirnov S.); 0000-0002-1662-1974(Kuznietsova N.)

range of variations and nuances. SLR involves the hand gestures identification, facial expressions, and body movements to interpret the meaning of a sign [1, 2]. The goal of SLR is to develop systems that can recognize and translate sign language into written or spoken language which enable communication between hearing and non-hearing individuals. In this research we will discuss the problems, limitations, not solved issues, and existing solutions in SLR.

One of the primary challenges in SLR is the complexity of sign languages. There are over 300 sign languages used worldwide, each with its own grammar, vocabulary, and dialects. Furthermore, sign languages are highly context-dependent, with the meaning of signs often varying depending on the speaker's location, age, gender, and culture. Thus, developing an SLR system that can accurately recognize and interpret the nuances of different sign languages is a significant challenge.

Another challenge in SLR is the variability in signing styles. Signers may use different hand shapes, positions, and movements to convey the same message. Moreover, the speed and duration of signs can vary, adding further complexity to the task. Thus, SLR systems must be robust to variations in signing styles, as well as to variations in lighting conditions and camera angles. To address these challenges, researchers have proposed various techniques, such as data augmentation, transfer learning, and multi-modal fusion, which combine visual and other modalities such as audio or depth information.

One of the main challenges in sign language recognition is collecting and annotating large datasets of sign language gestures, which are required to train and evaluate machine learning models. This can be particularly difficult for sign languages that are not widely spoken or documented. A limitation of SLR is the lack of large-scale annotated datasets. While there are several datasets available for SLR, they are relatively small, limiting the performance of machine learning algorithms. A brief comparison of existed datasets for sign recognition task is made in Table 1. Moreover annotating sign language data is a time-consuming and challenging task, as it requires the expertise of sign language experts.

Table 1
Sign language recognition datasets comparison

Id	Name	Country	Classes	Samples	Language level	Availability
1	DGS Kinect 40	Germany	40	3000	Word	Contact author
2	RWTH-PHOENIX-Weather	Germany	1200	45760	Sentence	Publicly available
3	SIGNUM	Germany	450	33210	Sentence	Contact author
4	GSL 20	Greek	20	~840	Word	Contact author
5	Boston ASL LVD	USA	3300+	9800	Word	Publicly available
6	PSL Kinect 30	Poland	30	300	Word	Publicly available
7	PSL ToF 84	Poland	84	1680	Word	Publicly available
8	LSA64	Argentina	64	3200	Word	Publicly available
9	MSR Gesture 3D	USA	12	336	Word	Publicly available
10	DEVISIGN-G	China	36	432	Word	Contact author
11	DEVISIGN-D	China	500	6000	Word	Contact author
12	DEVISIGN-L	China	2000	24000	Word	Contact author
13	IIITA-ROBITA	India	23	unknown	Word	Contact author
14	Purdue ASL	USA	unknown	unknown	Word/ Sentence	Request DVDs/HD
15	CUNY ASL	USA	unknown	~33000	Sentence	Unknown
16	SignsWorld Atlas	Arabia	multiple types	unknown	Handshape, Words, Sentences	Unknown
17	LSA-T	Argentina	translation	14880	Sentence	Publicly available
18	LSFB-CONT	Belgium	6883	85000+	Word, Sentence	Publicly available

19	LSFB-ISOL	Belgium	400	50000+	Word	Publicly available
20	WLASL	EEUU	2000	21083	Word	Publicly available

Another issue in SLR is the lack of real-time performance. SLR systems often require high computational resources, making it challenging to achieve real-time performance on mobile devices or in low-resource settings.

Researchers have proposed various solutions to deal with these challenges and limitations [3–12]. There is an approach that proposes to use different deep learning algorithms, such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs), to recognize the signs from video sources. These algorithms have shown promising results in SLR, achieving state-of-the-art performance on several benchmark datasets.

Another solution is to use depth sensors, such as Microsoft Kinect, to capture 3D motion data, which can be used to recognize signs accurately [8]. Depth sensors are advantageous as they can capture the 3D shape and position of the signer's hands, providing more robust and accurate sign recognition.

Furthermore, researchers have proposed the use of transfer learning, where pre-trained models on large datasets such as ImageNet are fine-tuned on sign language datasets. This approach has shown to improve the performance of SLR models, particularly for low-resource sign language datasets.

So, the SLR is a challenging task that requires robust and accurate recognition of complex hand gestures, facial expressions, and body movements. While several solutions have been proposed to address the limitations and challenges of SLR, there are still several unsolved issues, such as real-time performance, variability in signing styles, and lack of large-scale annotated datasets. With the development of more advanced algorithms and the availability of larger annotated datasets, it is hopeful that these challenges can be addressed, enabling better communication between hearing and non-hearing individuals. Overall, sign language recognition is still an important problem with potential applications in fields such as assistive technologies, education, and communication for deaf and hard-of-hearing individuals.

3. Brief overview of sign language recognition methods

There are various methods for sign language recognition (SLR) that have been proposed and tested over the years. Here are some of the most widespread methods for SLR:

1. ***Template matching*** is a simple and intuitive approach where the hand movements of the signer are matched against a predefined set of templates to recognize the sign [3, 4]. The template matching method involves capturing a series of hand poses and storing them as templates. During recognition, the input sequence is compared to each template, and the sign is identified based on the closest match. While this method is easy to implement, it is limited by the need for manually defining the templates and the inability to handle variations in signing styles.
2. ***Hidden Markov Models (HMMs)*** are probabilistic models used in speech recognition as a tool that can model the temporal dependencies in sign language by capturing the transitions between hand shapes and movements. The method involves training an HMM on a dataset of sign language gestures and using it to recognize signs in new sequences [5, 6]. However, HMMs can be limited in capturing the complex and context-dependent variations in sign language.
3. ***Support Vector Machines (SVMs)*** as a type of machine learning algorithm can classify sign language sequences by finding the hyperplane that separates the data points into their respective classes [7]. SVMs have been shown to achieve good accuracy in SLR, but they require large amounts of training data and may not be robust to variations in signing styles.
4. ***3D Depth Sensors*** using 3D depth sensors, such as Microsoft Kinect give the possibility to capture the 3D shape and position of the signer’s hands. This approach has the advantage of being more robust to variations in lighting and camera angles, and it can capture the depth

information of the hand movements[8, 9]. The depth information can be used to recognize signs more accurately.

5. ***Deep Learning*** methods, such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs), have been used in SLR and have shown significant improvements in performance. CNNs can learn the spatial features of sign language sequences, while RNNs can capture the temporal dependencies between hand movements [10, 11]. Additionally, attention mechanisms can be used to focus on the most relevant parts of the sequence, improving the accuracy of recognition.

Several approaches to deep learning could be defined for the relevant tasks and could be quite efficient in the real application.

PoseTCN is a deep learning model that uses temporal convolutional networks (TCNs) to capture the temporal dependencies in sign language gestures. PoseTCN takes as input a sequence of 3D hand pose data and outputs the recognized sign. The model uses dilated convolutions to increase the receptive field of the network and improve the model's ability to capture long-term dependencies [12, 13].

PoseTGCN is a deep learning model that uses a graph convolutional network (GCN) to capture the spatial dependencies between the joints in sign language gestures, and a temporal convolutional network (TCN) to capture the temporal dependencies. The model takes as input a sequence of 3D joint positions and outputs the recognized sign. The GCN operates on a graph structure where the joints are nodes and the edges represent the spatial relationships between them [14, 15]. The TCN operates on the resulting feature maps and uses dilated convolutions to capture long-term dependencies.

Inflated 3D ConvNet (I3D): I3D is a deep learning model that uses a 3D convolutional neural network (CNN) to extract spatio-temporal features from sign language gestures. The model takes as input a sequence of RGB or depth frames and as the outputs gives the recognized sign. The 3D CNN is pre-trained on large-scale video datasets, such as Kinetics or Sports-1M, and fine-tuned on the sign language recognition task [16, 17]. The pre-training allows the model to learn generalizable features that can be applied to sign language gestures.

Sign Language Transformers (SLT) is a transformer-based model that uses self-attention mechanisms to learn the spatial and temporal features of sign language gestures. SLT takes as input a sequence of RGB or depth frames and as the outputs the recognized sign. The model uses a pre-trained backbone network, such as ResNet or EfficientNet, to extract visual features from the frames, which are then fed into a transformer encoder-decoder architecture [18, 19]. The attention mechanisms in the model allow it to focus on the most relevant parts of the sequence and improve recognition accuracy. Now this approach is widely used in continuous sign language translation.

Transformers were originally designed for natural language processing (NLP) tasks where they exceed in capturing long-range dependencies and global context. They achieved this by incorporating self-attention mechanisms that allow the model to weigh the importance of different parts of the input sequence when making predictions. The same mechanism can be applied to images by treating each pixel or patch as a token, allowing the model to attend to different parts of the image when making predictions. Another advantage of transformers in computer vision is their ability to handle variable input sizes without requiring resizing or cropping. This is important for tasks such as object detection and segmentation where the size and aspect ratio of the objects can vary significantly. Additionally, transformers can leverage pre-training on large data amounts, allowing them to learn useful representations that can be fine-tuned on smaller datasets for specific tasks. Overall, the use of transformers in computer vision has shown promising results, outperforming traditional CNN-based architectures on various benchmarks and achieving state-of-the-art results on challenging tasks such as image captioning and visual question answering.

In summary, there are various deep learning models that can be used for sign language recognition, such as PoseTGCN, I3D, PoseTCN, and Sign Language Transformers (SLT). These models differ in their architecture and their ability to capture spatial and temporal dependencies in sign language gestures.

In Ukraine the task of sign recognition is really actual and important in context of the war and necessity to develop the special governments to support and provide assist and inclusion in society the

people who were suffered from the war and have problems with hearing. There are several works of national scientists who have investigated the problem of sign recognition and proposed special techniques and systems for communication and translating into the sign language [20–21]. Nevertheless, there are still unsolved issues and necessity of new approaches and adapting the existed methods is quite high.

4. Practical task of video interpretation for sign language

The ***goal*** in this article was to test mentioned above approaches and to build the simple transformer-based model for sign language recognition and compare its efficiency with the standard approach of using 3D convolutions. The idea was to clarify and define on a very first level (without any neural networks tricks or additional approaches) which approach could be more accurate for our task.

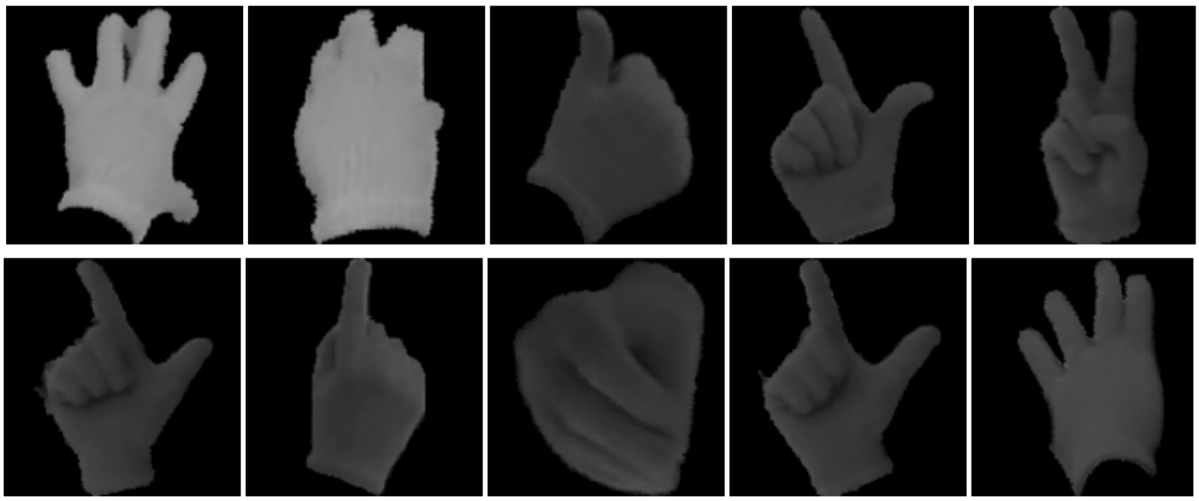
4.1. Dataset

For our experiments the LSA64 dataset [22] was used. LSA64 is a dataset for Argentinian Sign Language (LSA) and it represents a collection of video sequences designed for the task of sign language recognition in the Argentinian Sign Language. The dataset was created by researchers at the National University of Córdoba in Argentina, and contains 64 different LSA signs performed by 20 signers (10 male and 10 female).

The videos were recorded in a controlled environment using a high-definition camera and have a resolution of 1280x720 pixels at 25 frames per second. Each sign was performed five times by each signer and results were presented in a total of 6400 video sequences.

The LSA64 dataset also includes ground-truth annotations for each video, indicating the start and end frames of each sign. These annotations were performed manually by experts in LSA sign language.

The LSA64 dataset is quite challenging dataset due to variations in signing speed, camera viewpoint, and lighting conditions, making it a valuable resource for researchers working on developing robust and accurate sign language recognition algorithms (see some examples on Figure 1 and Figure 2).



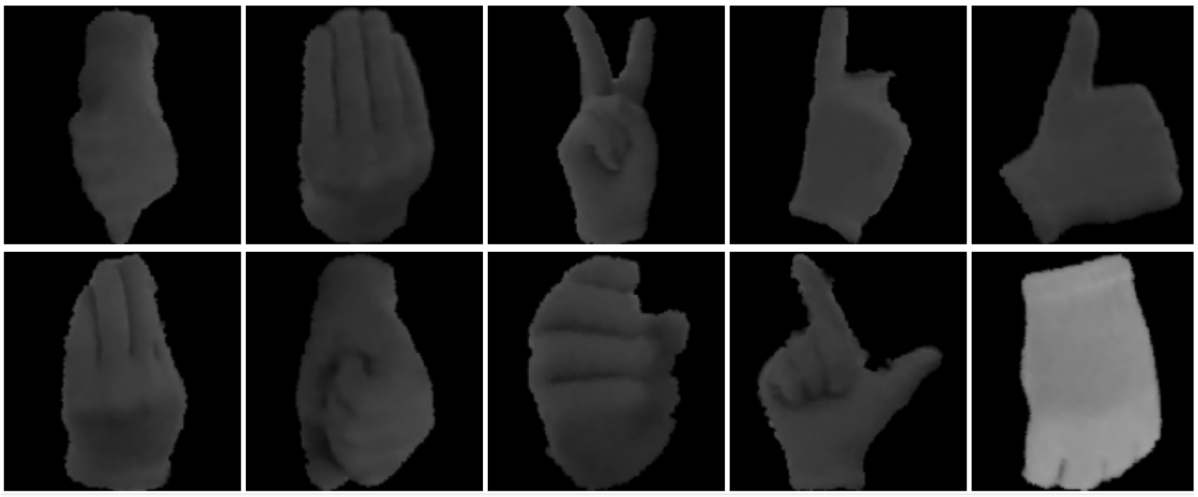


Figure 1: Screenshots of some examples of LSA64 dataset for sign language recognition

Figure 2: Example of one video storyboard

For our experiments firstly we made some labels for the dataset and then clustered them into 3 logical groups. That was done to have more labels in each of the ground-truth classes. The classes are: “colors” signs (consists of the next initial classes: “red”, “green”, “yellow”, “light-blue”), “food” signs (consists of the next initial classes: “sweet milk”, “water”, “food”) and “verbs” signs (consists of the next initial classes: “help”, “thanks”). The labels were randomly splitted into the train-test sets: 385 labels went to train and 165 labels - to the test set. Note that for experiments the proportion of each class was saved in both train and test sets. The general proportion of classes in data are: 36,3636% of the first class, 36,3636% of the second class, 27,2727% of the third class.

4.2. Used approaches

In this paragraph the transformer-based and 3D convolution-based models that were used in the experiments are described more detailed.

4.2.1. 3D convolutions

A neural network that uses 3D convolutions for video analysis typically consists of multiple layers of 3D convolutional, pooling, and fully connected layers [23].

3D convolutions are a type of convolutional layer that considers the spatial and temporal dimensions of the input data. In the case of video analysis, the input is a sequence of frames, and the 3D convolutional layer applies a kernel to each frame and its neighboring frames in the temporal dimension to extract features that capture both spatial and temporal information. This allows the model to learn patterns and movements over time, which is crucial for tasks such as action recognition and gesture recognition (see Figure 3) [24].

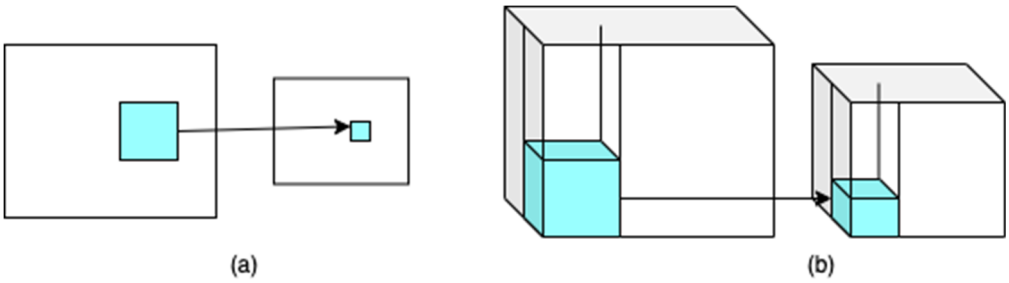


Figure 3: Comparison of 2D (a) and 3D (b) convolutions

After the 3D convolutional layers, pooling layers are often used to downsample the feature maps and reduce the spatial dimensionality of the data. This helps to reduce the parameters number in the model and prevent overfitting.

Finally, fully connected layers are used to classify the input video sequence into one or more classes. These layers take the flattened feature maps from the convolutional layers and apply a set of weights to produce a probability distribution over the possible classes.

4.2.2. Vision transformer

The approach of Video Vision Transformer (ViViT) [25, 26] involves dividing the video into small spatiotemporal regions of interest, called tubelets, and processing them using self-attention mechanisms similar to those used in the NLP models.

Here is a brief overview of how the ViViT model with tubelet embeddings works:

1. Tubelet Extraction: The first step is to extract tubelets from the input video. This can be done using a variety of techniques such as object detection, tracking, or motion analysis. Each tubelet is represented as a sequence of T frames, where each frame is a H x W x C tensor representing the pixel values of the video frame (see Figure 4).

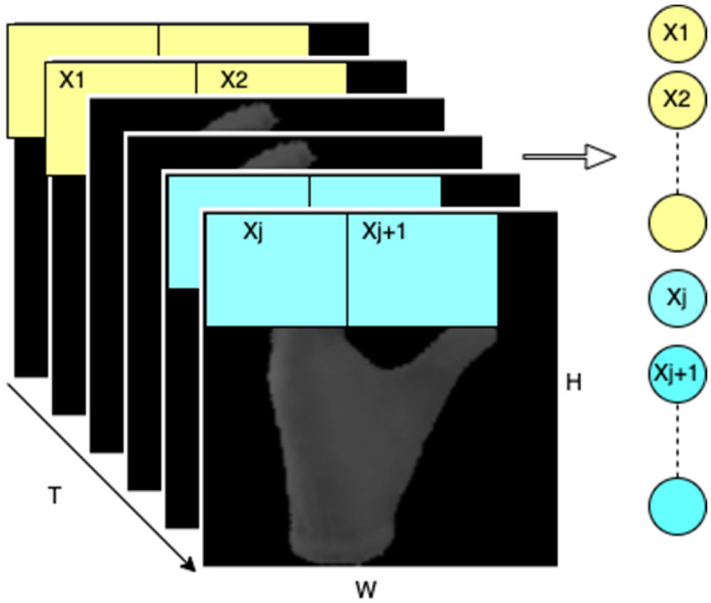


Figure 4: Tubelet embedding

2. Flattening and Linear Projection: Each frame in the tubelet is flattened into a sequence of patches, and these patches are then linearly projected into a higher-dimensional embedding

space of size D using a trainable linear layer. This results in a sequence of patch embeddings for each frame in the tubelet.

3. Multi-Head Self-Attention: The projected sequences are then passed through multi-head self-attention layers. Each layer computes attention weights between all pairs of patch embeddings in the sequence, and uses these weights to compute a weighted sum of the patch embeddings. This allows the model to attend to different regions of the tubelet depending on the task at hand.
4. Feedforward Network: After each self-attention layer, the output is passed through a feedforward network with a ReLU activation function. This network applies a linear transformation to the input, followed by a non-linear activation function. This helps the model capture more complex relationships between the patch embeddings in the sequence.
5. Aggregation: Finally, the output of the last self-attention layer is aggregated across all frames in the tubelet to obtain a single vector representation for the tubelet. These vectors are then passed through a linear layer to predict the class label for the entire tubelet.

By processing tubelets using self-attention mechanisms, ViViT with tubelet embeddings is able to better capture the spatiotemporal relationships between different regions of the video, resulting in improved performance on video recognition tasks such as action recognition and video classification.

5. Modelling & Results

In our experiments we used the vision transformer (ViViT) and 3D convolutions CNN (3dCNN) for comparison on our dataset, which was trained on the different batch sizes (3, 32, 128, 385 (the length of train dataset)). Each of the selected models was trained on 30 epochs, with learning rate equal to 1e-5, with input size of the videos equal to (25, 64, 64, 3). During the training the history of the accuracies is stored, so as a result only those model weights are saved and loaded for that approaches that had the best test accuracy. That was done to prevent the possible model overfitting. On Table 2 the models comparison table is presented, where top-1 (also known as accuracy) and top-2 are the top-K accuracy metrics calculated by the formula 1 below:

$$ttop-K\ accuracy = \frac{1}{n_{samples}} \sum_{i=0}^{n_{samples}-1} \sum_{j=1}^{k} 1\left(\widehat{f_{i,j}} = y_i\right) \quad (1)$$

where $\widehat{f_{i,j}}$ is the predicted class for the i-th sample corresponding to the j-th largest predicted score, y_i is the corresponding true value, k is the number of guesses allowed and 1(x) is the indicator function. Top-K accuracy is often used for sign language recognition problems because it is a useful metric for evaluating the models performance dealing with a large number of possible signs and variations, as well as accounting for the flexibility required in recognizing signs.

Table 2
Comparison of vision transformer (ViViT) and 3D convolutions CNN (3dCNN) approaches

Approach name	test top-1	test top-2	train top-1	train top-2
ViViT/3	**69.7%**	89.7%	**87.01%**	89.7%
ViViT/32	64.85%	**90.3%**	78.96%	**90.3%**
ViViT/128	63.03%	89.09%	63.9%	89.09%
ViViT/385	60.0%	83.03%	65.97%	83.03%
3dCNN/32	63.64%	75.76%	58.18%	73.77%
3dCNN/128	52.12%	72.73%	51.17%	72.73%

In table 2 it is seen that the ViViT models with small batch sizes have higher quality. But having in mind the accuracy on training dataset as well we can notice that such models are more easily

overfitted. That means that ViViT model is faster in training flow to get the high quality (see Figure 5). On the other hand, for future investigations more approaches for fixing the model overfitting issue should be applied (e.g., augmentation), especially by working with small data amounts. Such logic could be traced on the plots below (Figures 6-11). Note, that in Figure 6-11 the loss and accuracy are normalized to be presented on the same scale, which gives the opportunity to analyze overfitting issues there.

We see that from the 20th epoch the accuracy for 3D convolution network (3dCNN/128) are extremely increasing. From the other hand, the accuracy on both training and test datasets for ViViT/3 and ViViT/32 models (Figure 6-7) as well as test losses are growing up and at the same time the losses on training dataset are decreasing. It also confirmed that the model was good learned on training dataset but on the test dataset it faced with some troubles. As for 3D convolution networks (Figures 10-11) the losses on both training and test datasets are decreasing with similar speed as well as accuracy are increasing.

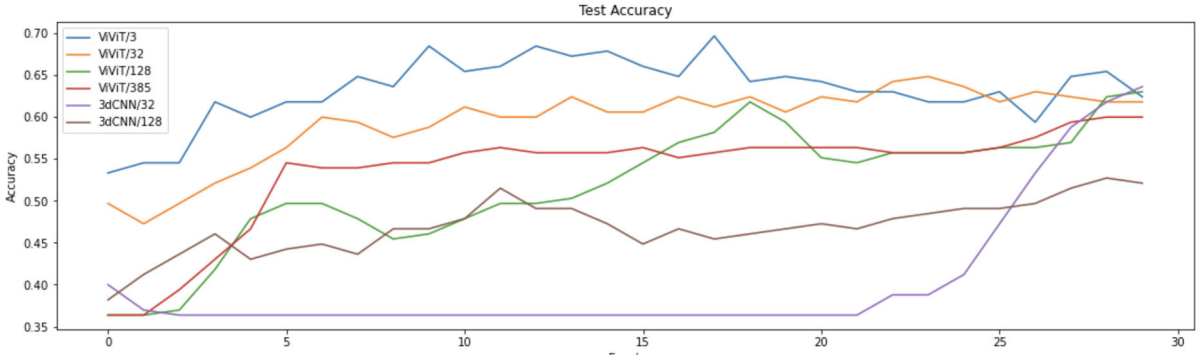


Figure 5: Accuracy on a test set for the different models on epochs scale

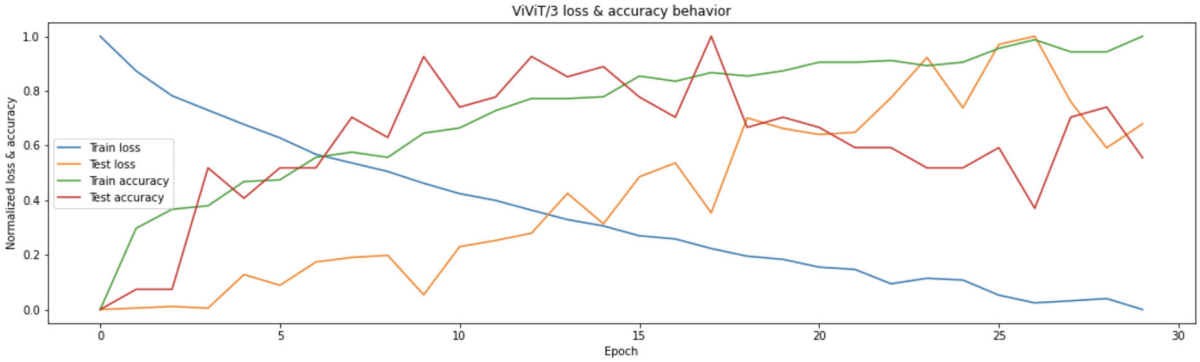


Figure 6: Train and test loss and accuracy comparison of ViViT/3 model on epochs scale

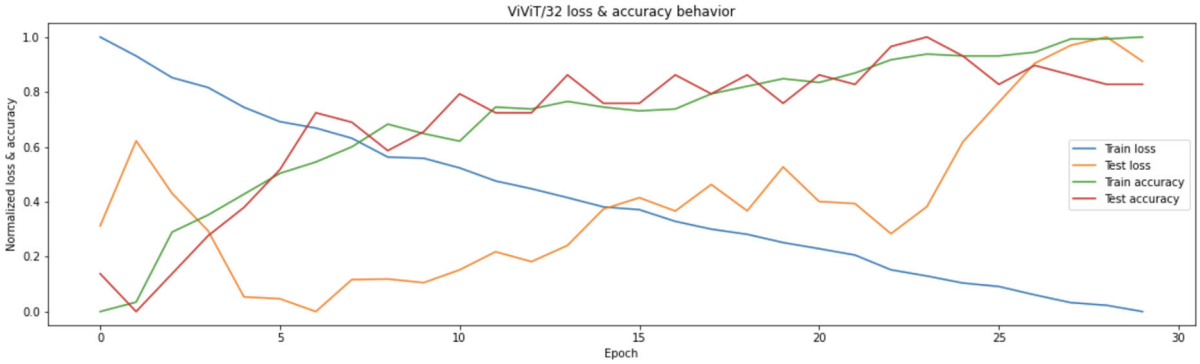


Figure 7: Train and test loss and accuracy comparison of ViViT/32 model on epochs scale

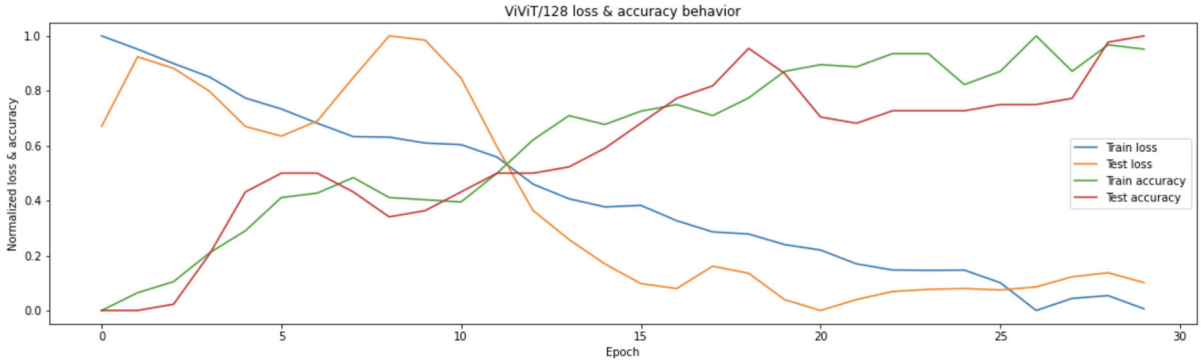


Figure 8: Train and test loss and accuracy comparison of ViViT/128 model on epochs scale

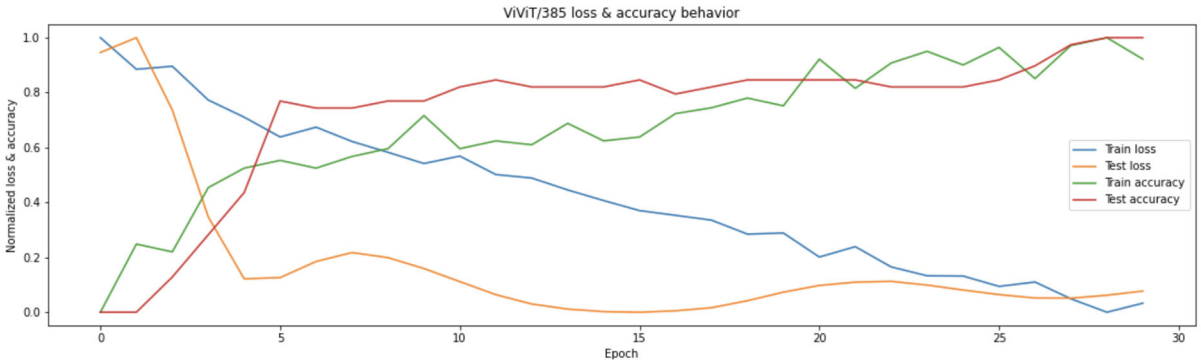


Figure 9: Train and test loss and accuracy comparison of ViViT/385 model on epochs scale

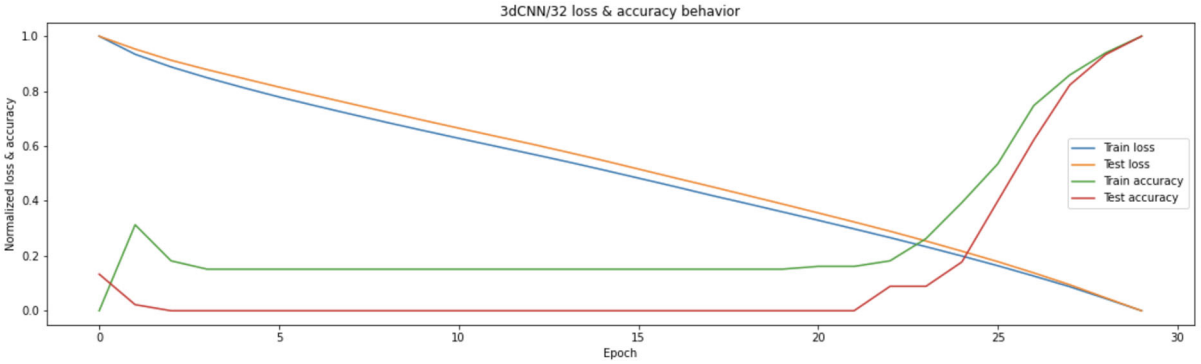


Figure 10: Train and test loss and accuracy comparison of 3dCNN/32 model on epochs scale

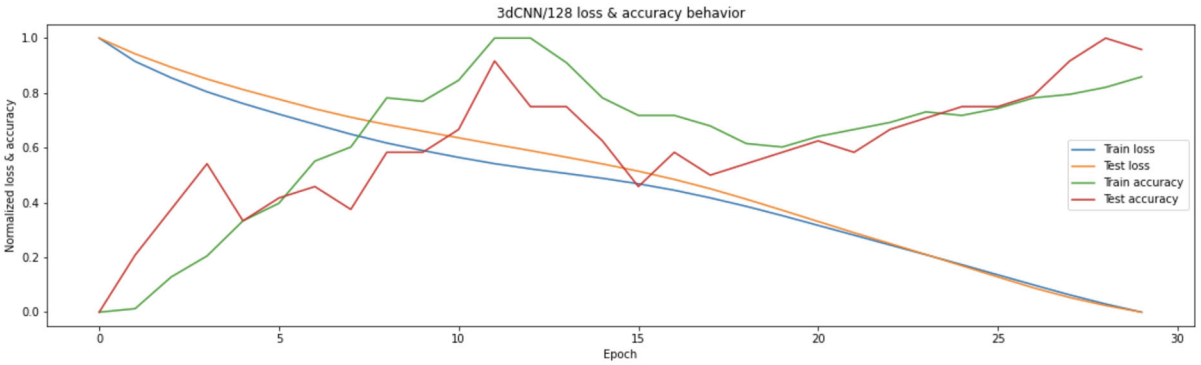


Figure 11: Train and test loss and accuracy comparison of 3dCNN/128 model on epochs scale

6. Conclusion

In this paper we discussed the most relevant practices and approaches for sign language recognition. While significant progress has been made in sign language recognition using modern methods, there are still some important issues that remain unsolved:

1. Large variability in sign language. Sign language can vary widely across different regions, cultures, and even individuals. This variability poses a significant challenge for sign language recognition systems, which must be robust to these variations.
2. The availability of large, diverse datasets is crucial for training and evaluating machine learning models for sign language recognition. However, there is still a limited availability of such datasets, particularly for less widely spoken sign languages.
3. Real-time sign language recognition is important for many applications, such as assistive technology and communication. However, real-time recognition remains a challenge, as it requires processing sign language videos in real-time, which can be computationally intensive.
4. Sign language gestures can be occluded or noisy due to factors such as clothing, lighting, and background clutter. Handling these occlusions and noise is still a challenge for sign language recognition systems.
5. Sign language recognition systems are typically trained on a limited set of sign language gestures, which can impact their ability to recognize new or rare signs.

The task of sign recognition in this paper was solved for real dataset. It was shown how to implement the existed approaches on different sizes of the existed data and what to do to receive the higher accuracy. Our experiments aimed to compare the performance of the Vision Transformer (ViViT) and 3D Convolutions CNN (3dCNN) models on sign language recognition. We trained both models on different batch sizes and evaluated their accuracy using the Top-K metric. Our analysis shows that ViViT models with small batch sizes achieved higher quality, but were more prone to overfitting. The best obtained numerical results were 69,7% for ViViT/3 on test-top1 and 89,7% on test-top2 and for ViViT/32 were achieved the accuracy 89,7% on test-top1 and 90,3% for test-top 2. To address the issue of overfitting, future investigations should explore approaches such as data augmentation to improve the generalization of ViViT models, especially when working with limited amounts of data.

The practical value of this paper is that it was also shown on real dataset how and why it is needed to search the compromise between speed, accuracy and overfitting issues as well as between length of the dataset and how existed methods needed to improve.

Overall, our results provide valuable insights into the strengths and limitations of different models for sign language recognition, as well as their practical implementation. It was offered in the paper a starting point for further research for sign language recognition by using vision transformers and additional approaches in conjunction with, for example, pose estimation, hands recognition, etc., with vision transformers before classification itself.

7. References

[1] W. Hameed and A.A. Al-Jumaily, "Sign language recognition: Dataset and challenges", in Proceedings of the 2019 International Conference on Innovations in Intelligent Systems and Applications, pp. 1-7, 2019.

[2] Sehili, M. E. A., & Melkemi, M. (2021). Challenges and Opportunities in Sign Language Recognition. IEEE Access, 9, 52470-52487. doi:10.1109/ACCESS.2021.3062127.

[3] V. Murino, C. S. Regazzoni, Template Matching Techniques in Computer Vision: Theory and Practice, IEEE Transactions on Pattern Analysis and Machine Intelligence 22 (2000) 743-760. doi:10.1109/34.85661.

[4] Dong, L., Jiang, S., Huang, Q., & Li, W. (2010). Template matching-based human action recognition in videos. Pattern Recognition, 43(3), 1199-1206. doi:10.1016/j.patcog.2009.07.022.

[5] Makris, D., & Ellis, T. (2006). Sign language recognition using hidden Markov models. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), 36(3), 514-524. doi:10.1109/TSMCB.2005.856082.

[6] Garg, G., Sharma, S., & Saraswat, M. (2016). Sign Language Recognition Using Hidden Markov Models and HOG Features. In 2016 International Conference on Signal Processing and Communication (ICSC) (pp. 717-722). IEEE. doi:10.1109/ICSC.2016.7953781.
[7] Kumari, N., Gupta, N., & Sharma, S. K. (2016). A Review on Hidden Markov Models and Support Vector Machines in Sign Language Recognition. International Journal of Computer Applications, 138(7), 6-12. doi:10.5120/ijca2016908784.
[8] El-Fishawy, Z., Rizk, M., & Abdel-Wahab, M. A. (2018). Sign Language Recognition Using Kinect Sensor: A Review. In 2018 11th International Conference on Developments in eSystems Engineering (DeSE) (pp. 191-196). IEEE. doi:10.1109/DeSE.2018.00042.
[9] Guo, X.-L., & Yang, T.-T. (2016). Gesture recognition based on HMM-FNN model using a Kinect. Journal on Multimodal User Interfaces, 11. doi:10.1007/s12193-016-0215-x.
[10] Xia, W., Zhai, X., & Liu, Y. (2019). Sign language recognition using deep learning models: A review. ACM Transactions on Accessible Computing, 12(3), 1-30. doi: 10.1145/3301417.
[11] Zhang, C., Yang, J., & Kim, G-M. (2017). Sign Language Recognition with Convolutional Neural Networks Trained on Synthetic Data. In 2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS) (pp. 1-6). doi: 10.1109/AVSS.2017.807856.
[12] Keze Wang, Xiaoguang Zhao, and Jing Liu. Sign Language Recognition Using Temporal Convolutional Networks and Skeleton Data. IEEE Access, vol. 7, pp. 158074-158083, 2019. doi: 10.1109/ACCESS.2019.2951043.
[13] R. Girdhar, G. Gkioxari, L. Torresani, and M. Paluri, "PoseTCN: Efficient Convolutional Neural Networks for Human Pose Estimation and Action Recognition," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 2019, pp. 3332-3341. doi: 10.1109/CVPR.2019.00344.
[14] Deng, Z., Wan, J., & Xie, X. (2020). PoseTGCN: A Temporal Graph Convolutional Network for 3D Human Pose Forecasting. IEEE Transactions on Neural Networks and Learning Systems, 32(1), 67-77. doi:10.1109/TNNLS.2020.3010193.
[15] Andrea Vanzo, Carlo Ciliberto, and Alberto Montagner, "Sign Language Recognition Based on Pose Estimation with Temporal Graph Convolutional Networks," Sensors 20(13), 3759, July 2020. doi:10.3390/s20133759.
[16] Kardoostsiami, A., Shafiee, M. J., & Plataniotis, K. N. (2021). Sign Language Recognition with Inflated 3D Convolutional Networks. IEEE Transactions on Multimedia, 23, 313-326. doi:10.1109/TMM.2020.3043619.
[17] Tran, D., Wang, H., Torresani, L., Ray, J., LeCun, Y., & Paluri, M. (2015). Learning Spatiotemporal Features with 3D Convolutional Networks for Gesture Recognition. In Proceedings of the IEEE International Conference on Computer Vision (pp. 4489-4497). doi: 10.1109/ICCV.2015.510.
[18] Efstratios Gavves, Thomas van de Weijer, and Jan C. van Gemert. Transferring Knowledge from Text to Sign Language Video Recognition with Transformers. 2021. doi: 10.1109/CVPRW50498.2021.00443.
[19] Chung, J.S., Kim, J., & Kim, J. (2021). Sign Language Transformers: Joint End-to-end Sign Language Recognition and Translation. arXiv preprint arXiv:2103.01197.
[20] S. Kondratiuk, I. Krak, V.A. Kuznetsov, A. Kulias, Using the Temporal Data and Three-dimensional Convolutions for Sign Language Alphabet Recognition, CEUR Workshop Proceedingsthis link is disabled, 2022, 3137, pp. 78–87.
[21] S. Kondratiuk, I. Krak, A. Kylias, V. Kasianiuk. Fingerspelling Alphabet Recognition using Cnns with 3d Convolutions for Cross Platform Applications. Advances in Intelligent Systems and Computing. Vol. 1246 AISC. 2021, pp.585-596. doi:10.1007/978-3-030-54215-3_37.
[22] Ronchetti, F., Quiroga, F., Estrebou, C., Lanzarini, L., and Rosete, A. LSA64: A Dataset of Argentinian Sign Language. In XXII Congreso Argentino de Ciencias de la Computación (CACIC), 2016.
[23] Heng Wang, Yang Wang, and Gang Zeng, “Sign Language Recognition Using 3D Convolutional Neural Networks,” in Proceedings of the 24th ACM international conference on Multimedia (MM ’16), Amsterdam, The Netherlands, October 15-19, 2016, pp. 1033–1042. doi:10.1145/2964284.2964318.

[24] Lu, C., Chen, Y., & Lu, H. (2019). Sign Language Recognition Using 3D Convolutional Neural Networks with Softmax Probability Map. IEEE Access, 7, 116256-116267. doi: 10.1109/ACCESS.2019.2937045.
[25] Yuan, L., Chen, Y., Wang, T., Gan, W., Liu, Z., & Kornilov, S. (2021). ViViT: A Video Vision Transformer for Efficient Video Recognition. arXiv preprint arXiv:2103.15691.
[26] Elnaggar, A., Mahmoud, A., Abdou, A., Elgammal, A., & Abdel-Razek, M. (2021). ViViT-Sign: A Video Vision Transformer for Sign Language Recognition. arXiv preprint arXiv:2104.07441.