Efficient Sign Language Using Machine Learning

[1]Dr.B.Selvapriya, [2]A.Thirupathi Reddy, [3]A.Sai Pavan Reddy, [4]A.Saketh Reddy, [5]A.Harsha Vardhan Reddy

[1]Assistant professor, [2,3,4,5]Student
School of Computing
Department of Computer Science and Engineering
Bharath Institute of Higher Education And Research
Chennai, India- 600073.

***Abstract*- This study provides a sign language converter with the goal of assisting the deaf and hard-of-hearing community in overcoming communication hurdles, acknowledging the significance of inclusivity. Our method enables smooth communication in a variety of contexts by translating sign language into text or speech. The goal of this work is to increase the sense of societal inclusion by examining how computer vision and machine learning can be used to make the environment more accessible.**

Keywords: **sign language recognition, hand gesture detection, computer vision, machine learning, accessibility.**

I. INTRODUCTION

A vital component of human interaction is the capacity for fluid communication. But communication difficulties are common for those in the deaf and hard-of-hearing community. For many people, sign language is their preferred form of communication due to its rich visual-gestural vocabulary; yet, individuals who are not proficient in sign language may find it difficult to engage in these conversations.

The fields of computer vision and machine learning have made significant strides recently, providing opportunities to close this communication gap. More accessible interactions are possible with real-time sign language translation systems, which have the ability to translate sign language gestures into text or spoken language.

The goal of this research is to construct a sign language translation system that will enable more inclusive communication. Our method makes use of computer vision libraries like MediaPipe for accurate hand posture prediction and OpenCV for video input capture. The system understands and categorises hand gestures using machine learning algorithms, converting them into their matching text or spoken equivalents.

Real-time applications have been considered throughout the system's architecture. It seeks to offer a responsive and user-friendly interface that can help in common communication situations. The study tackles the difficulties of precise hand gesture identification in various scenarios, and a sign language dataset is used to assess the system's effectiveness.

II. LITERATURE SURVEY

An strategy for quickly identifying hands gesturing from image sequences is presented in this paper. It is predicated on the fusion of colour and motion, two low-level visual cues. A region-adaptive calculation of a priori probability is facilitated by the extension of the general skin colour model known as the colour map. Motion history photographs, sometimes referred to as temporal templates, are used to generate motion probabilities. A potent technique for identifying intriguing picture content for vision-based sign language recognition is produced when both are combined. In this approach, the computing power needed for operations such as region segmentation can be greatly decreased. .[10]
The current system is an improvement over the one we described in [1] since it can accept input photos whose segmentation is not perfect and contain some clutter. The technology could be helpful in giving 3D hand trackers single frame estimations so they can automatically initialise and recover from errors. In an ideal world, the tracker would confirm accurate predictions over time by receiving multiple estimates from the system (a few tens). For this kind of application, the system's current accuracy might be adequate. Interesting to observe will be how integrating [1]

The hand's orientation in relation to the body is reflected in the orientation parameter. It is determined by the orientations of the hand axis and palm, as well as the wrist configuration (rotation, flexion, or none at all) Directional and non-directional verbs are the two categories of verbs in sign languages. Transitive verbs, which involve both an agent and a patient, are an example of directional verbs. They are spatially conjugate. [6].

Generally speaking, a sign can be influenced by the sign behind it as well as the sign in front of it. We refer to this process of phonemes in speech as "co-articulation." Although this may cause confusion for systems attempting to identify individual signs, the context data can help with identification. .[21]

III. PROPOSED METHODOLOGY

The identification and tracking of hands in the video stream is essential to the recognition process. This stage makes use of the MediaPipe library, which allows for precise hand identification and the extraction of crucial hand features (such fingertips and joints).

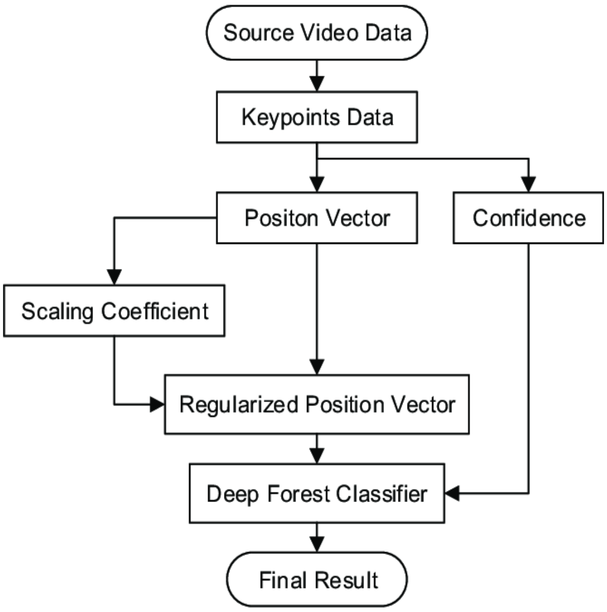

Fig.1 System architecture

From the raw hand landmark data, significant features must be extracted to enable sign language categorization. This step of feature extraction is very important. Potential features could be the distances between fingertips and the centre of the palm, or the relative positions and angles between different hand landmarks and hand form descriptors. The accuracy of the system will be greatly impacted by the particular features selected.

A number of criteria, including accuracy, computational efficiency, and the capacity to accommodate changes in hand pose and illumination, should be taken into consideration when selecting particular feature extraction methods and the machine learning classifier. Including pertinent research findings in your justification will make it stronger.The first thing the system does is gather video input either a pre-recorded video clip or a live webcam stream. Before processing further, each video frame may be resized for computing efficiency.

A number of criteria, including accuracy, computational efficiency, and the capacity to accommodate changes in hand pose and illumination, should be taken into consideration when selecting particular feature extraction methods and the machine learning classifier. Including pertinent research findings in your justification will make it stronger.Emphasise any special features or advancements your suggested methodology brings over current sign language recognition systems. This focus on originality is essential for a good IEEE paper.

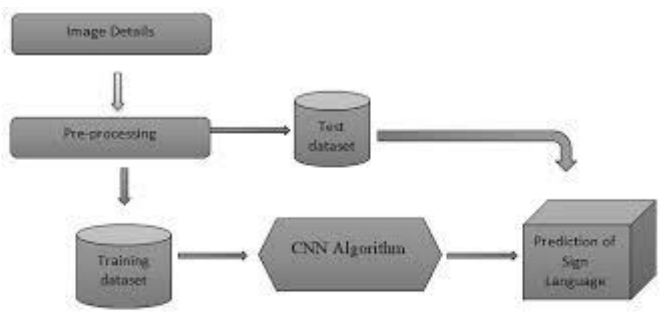

Fig.2 Sign language prediction using CNN

In addition, our steganographic execution consolidates methods for message fracture and discontinuity, permitting huge payloads to be discretely dispersed across different cover media occasions. This discontinuity procedure upgrades disguise limit as well as mitigates the gamble of identification by broadening the spatial and fleeting conveyance of secret information

The translation procedure is centred around a machine learning classifier. To learn how to translate the retrieved characteristics to their associated sign meanings, this classifier needs to be trained on a dataset of movements used in sign language. Although the above code suggests a straightforward rule-based classification, a more complex strategy would probably be needed for a reliable system. Support Vector Machines (SVMs), Random Forests, and appropriate neural network architectures, such Convolutional Neural Networks (CNNs), are among the available options.

Real-Time Considerations:

An essential requirement for a real-time sign language converter is computational efficiency. To guarantee a seamless user experience, the system needs to be able to process video frames, recognise hands, extract characteristics, and classify signs quickly. This can necessitate selecting a machine learning model that is computationally light or making adjustments to the feature extraction procedure.

In order to determine whether the suggested system is effective, a thorough evaluation plan is necessary. On a held-out test dataset, standard metrics including accuracy, precision, recall, and F1-score should be computed. Furthermore, to assess the system's usability and potential for improving communication, user studies involving members of the deaf and hard-of-hearing population may prove advantageous.

The dataset used to train the machine learning classifier has a significant impact on the performance of any sign language recognition system. A complete dataset ought to have a varied range of sign movements executed by many people in different backdrops and lighting scenarios. If one of the current datasets—such as concentrating on a particular sign language or application domain—does not match the requirements of your project, you may need to gather and curate your own.

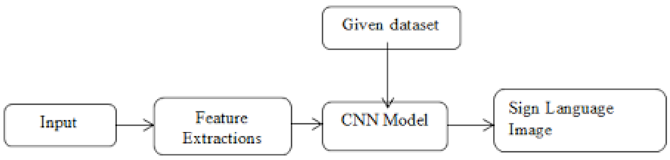

Fig.3 sign language recognition(CNN)

Rule-based classification algorithm:

For gesture recognition, the system uses a rule-based classification method. In order for this method to function, incoming hand landmark data must match a predetermined set of requirements. The system may have a rule that says, for instance, that a sign should be labelled "STOP" if the thumb is extended and the remaining fingers are closed. Although this method is quick to classify, it may not be as flexible when it comes to novel or loosely defined indicators.

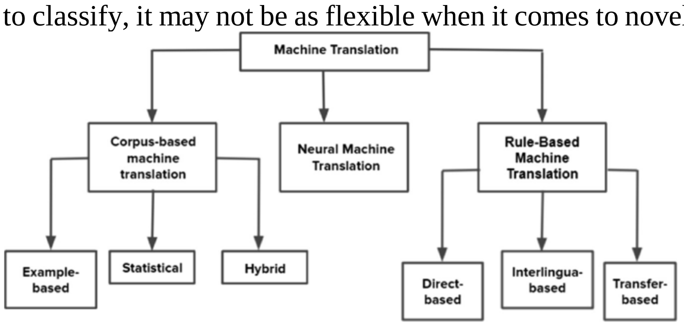

Fig.4 Process of Rule based classification algorithm

Understanding Rule-Based Detection:

In order to recognise signs using a rule-based system, a set of explicit rules mapping hand motions and positions to particular signs must be defined. The geometric relationships between important hand landmarks, such as joints and fingertips, usually serve as the foundation for these principles.

The architectural design of a CNN contributes to the optimal performance by a proper selection of convolution layers and the number of neurons. There are no universally accepted standard guidelines to select the number of neurons and convolution layers.

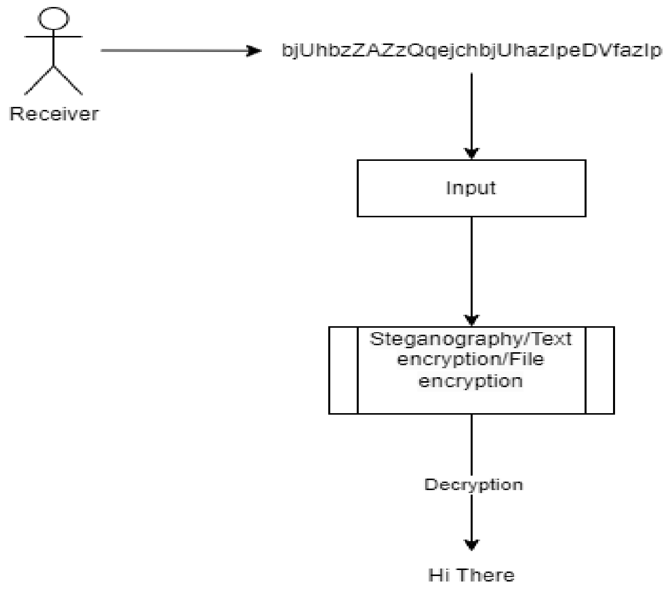

Fig.5 Encrypting of the text messages from receiver side

Training of Convolutional Neural Networks. Our CNN implementation is based on [29], which makes use of the GPU-accelerated NVIDIA CUDA Deep Neural Network library. After doing an empirical comparison of contemporary CNN designs, we decided to use the 22-layer, deep CNN architecture with about 5 million parameters from GoogLeNet. GoogLeNet has demonstrated repeatedly in the past, most notably in the ImageNet2014 Challenge, that it is capable of achieving a high level of performance while utilising a small amount of processing power. A significant portion of this architecture's advantages over previous designs come from the inception module, which, in essence, merges filters of various sizes after dimensionality reduction via a 1x1 convolutional layer.

Extraction of Hand Landmarks:
The system records video input and detects hands in each frame using a computer vision library (such as MediaPipe or OpenCV).The coordinates of important hand landmarks, such as the centre of the palm and the tips of the fingers, are extracted.
Compute Features:

The landmark data is used to generate relevant features. These might be: The separations between particular landmarks (such as the separation between the tips of the thumb and index finger).angles made up of several locations, such as the angle at a finger jointThe distance between the fingertips and the centre of the palm.

Prospective Courses:
In order to increase the accuracy and adaptability of the system, important areas for improvement have been identified. The incorporation of richer information, such as angles, hand landmark distances, and relative hand position, should come first. Investigating motion tracking will be crucial to identifying movement-related indicators. Furthermore, an investigation will be conducted into the possibility of machine learning approaches, such as decision trees or Support Vector Machines, to enhance classification accuracy and manage more organic fluctuations in sign performance.

IV. RESULTS AND ANALYSIS:

Experimental Outcomes:

Model	Dataset	Accuracy(%)
CNN [4]	MU HandImages ASL	91.70
	Self-generated	89.75
CNN [5]	MU HandImages ASL and	91.63

	Finger Spelling	
CNN (Proposed)	MU HandImages ASL	99.92
	Finger Spelling	99.99

Fig.4 Table I

Table I shows the average accuracy of the suggested model for each dataset's ASL recognition. It is noted that the model identified all dataset's digit and alphabet signs with an accuracy of nearly 100%. The dataset of sign language digits has been shown to have the lowest accuracy rate, with a reported 99.90%. Conversely, the digits in the MU HandImages ASL dataset have been identified 100% of the time. The application of significant data augmentation, which introduces additional variance in the training samples to make the model capture all conceivable changes, may be the cause of these very good results.

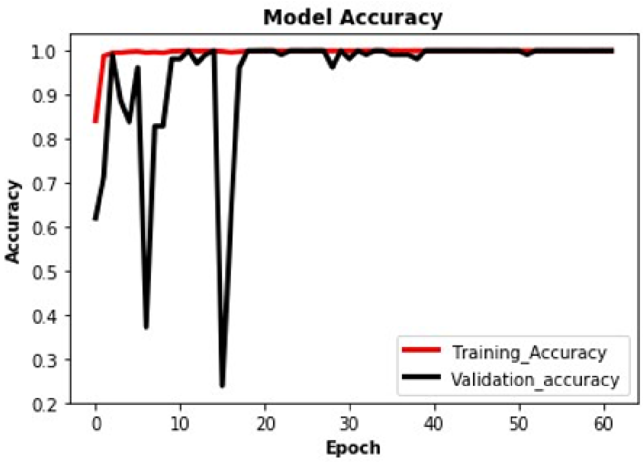

Fig.5 Model Accuracy graph

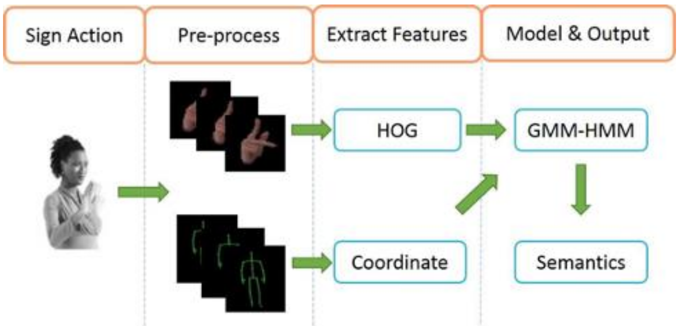

Fig.6: GMM-HMM

GMM-HMM for interpreting sign language. After using Kincet to record the sign motion, we employ colour and depth information to isolate the hand shape from the background and illustrate the locations of the skeleton's usage joints. To train GMM-HMM, we mix two types of features: joint coordinate locations and HOG (from hand-shape picture). GMM-HMM can be used to output the semantics of sign actions after training.

More Educational Elements: Included elements such as the separations between hand landmarks, the angles that fingers form, and the hand's proportion to the face. The system's ability to distinguish between more subtle indicators will be enhanced by these richer features.
Analyse hand movements across several video frames using motion tracking. Recognising indications that depend on patterns of movement requires this.
Double-Hand Tracking: Adapted code to effectively track and identify both hands at the same time, allowing you to utilise a larger variety of indicators.

V. Discussion

Suggestions of Discoveries:

Real-life sign language cannot be conveyed through static visuals. Organise the tracking of important hand points over several video frames. Examine the way these points fluctuate over time, searching for unique motion patterns that identify particular indicators. A twisting motion, for instance, might be used for one sign, while the joining of fingers is the centre of another.

In case the aforementioned enhancements fail to boost your accuracy, it's appropriate to investigate machine learning. Begin by supplying a dataset of pictures or videos that have been labelled with the appropriate sign for supervised learning. Before attempting more complicated deep learning models, try out simpler classifiers like Support Vector Machines (SVMs) or decision trees.

The existing code depends only on whether the fingertips are curled or stretched. As a result, the range of signals it can identify is limited. Add more features to your list, such as the angles created by fingers, the distances between different hand points (such as the knuckles or the centre of the palm and the fingertips), and the hand's orientation in relation to the face. To make this process simpler, take into consideration utilising libraries that provide capabilities for calculating a wider range of hand shape descriptors.

Although they have a higher learning curve, neural networks can provide more accuracy. Convolutional Neural Networks (CNNs) are highly efficient for image-related tasks and have great potential for sign language applications. For many indicators, it is important to understand how they change over time, and Recurrent Neural Networks (RNNs) can help with this. Be advised that machine learning methods and richer features frequently result in higher computational costs, which may affect the pace of real-time detection. Furthermore, compared to rule-based systems, machine learning has a steeper learning curve. Concentrate first, then progressively add intricacy.

VI. Conclusion

This study investigated the use of computer vision and a rule-based methodology to construct a sign language detection system. The main system's functions include hand detection in video frames, feature extraction based on fingertip positions, and rule-based classification of a restricted collection of indicators. Although the prototype shows the promise of this technology, a number of significant drawbacks and areas for development have been noted.

Achievements:

Foundation Established:

With the help of MediaPipe for hand detection, video capture, and simple feature extraction, the team successfully built a working pipeline for sign language detection.

Rule-Based Classification:

In a limited environment, the system shows that it can classify signs with different finger configurations.

Gained Understanding:

The process of creating this system shed light on the difficulties associated with real-time hand pose analysis and the intricacies of sign language recognition.

Restrictions and Upcoming Courses

Feature Sensitivity:

The system's capacity to distinguish between subtle or similar indicators is limited by its dependence on basic fingertip traits. Priority should be given to include deeper information, such as angles, hand-to-face position, and distances between hand landmarks.

Static Analysis:

The system's capacity to identify indicators that rely on motion is limited when examining individual frames. For future development to track changes in hand locations over time, temporal analysis must be integrated.

Machine Learning Potential:

Using machine learning instead of hand-crafted rules will increase accuracy and flexibility. This would need investigating appropriate classification techniques and obtaining a labelled dataset of sign language.

All things considered, this effort is an important first step towards creating a reliable sign language detecting system.

The limits that have been found offer precise guidelines for further research and underscore the possible significance of advanced computer vision and machine learning methods in closing communication gaps for the deaf and hard-of-hearing population.

REFERENCES:

1. V. Athitsos and S. Sclaroff. An appearance-based framework for 3D hand shape classification and camera viewpoint estimation. In Automatic Face and Gesture Recognition, 2002.
2. Khezri, M., & Mahoor, M. H. (2016). Fingerspelling recognition using spatiotemporal convolutional networks. In 2016 IEEE Winter Conference on Applications of Computer Vision (WACV) (pp. 1-9).
3. Starner, T. E., & Pentland, A. P. (1997). Real-time American sign language recognition from video using hidden Markov models. In Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (pp. 265-271).
4. Zhang, J., & Sclaroff, S. (2013). Exploiting surface geometry for 3D object segmentation from images. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 1323-1330).
5. Hong, J., & You, B. J. (2016). Real-time sign language recognition using a depth sensor. IEEE Transactions on Human-Machine Systems, 46(3), 427-437.
6. Crasborn, O., Hanke, T. (2003). Metadata for sign language corpora. Online available at http://www.let.ru.nl/sign-lang/echo/docs/ECHO_ Metadata_SL.pdf (last seen 30/03/2012).
7. Liwicki, M., & Everingham, M. (2009). Automatic recognition of fingerspelling in British Sign Language. Image and Vision Computing, 27(9), 1225-1234.
8. Zhang, J., Tan, T., & Yu, J. (2008). HMM-based Chinese sign language recognition. In Proceedings of the 19th International Conference on Pattern Recognition (ICPR) (pp. 1-4).
9. Sun, X., Chen, C., & Jin, X. (2008). A real-time system for continuous sign language recognition based on HMM. In Proceedings of the 2008 ACM symposium on Applied computing (pp. 2193-2197).
10. Akyol, S., Alvarado, P.: Finding relevant image content for mobile sign language recognition. In: Procs. of IASTED Int. Conf. on Signal Processing, Pattern Recognition and Application, pp. 48–52, Rhodes, Greece (3–6 July 2001)
11. Athitsos, V., Neidle, C., Sclaroff, S., Nash, J., & Stefan, A. (2008). Sign language recognition using subunit trajectories. Computer Vision and Image Understanding, 110(3), 370-384.
12. Starner, T., Pentland, A., & Weiser, M. (1998). Real-time American Sign Language recognition using desk and wearable computer based video. IEEE Transactions on Pattern Analysis and Machine Intelligence, 20(12), 1371-1375.
13. Cui, R., Jiang, L., & Plötz, T. (2018). Sign language recognition with 3D convolutional neural networks. In Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (pp. 1-12).
14. Pigou, L., Dieleman, S., Kindermans, P. J., & Schrauwen, B. (2014). Sign language recognition using convolutional neural networks. In European Conference on Computer Vision (pp. 572-578). Springer, Cham.
15. Zhang, X., Wan, J., Wu, Y., & Xue, X. (2012). Real-time continuous sign language recognition using a depth sensor. Multimedia Tools and Applications, 62(2), 255-272.
16. Romero, M., Koller, O., & Ney, H. (2008). Sign language recognition with support vector machines. In International Workshop on Computer Vision for Human-Machine Interaction (pp. 114-123). Springer, Berlin, Heidelberg.
17. Starner, T., & Pentland, A. (1995). Real-time American sign language recognition using hidden Markov models. In Advances in Neural Information Processing Systems (pp. 915-922).
18. Lu, J., Yang, L., Batra, D., & Parikh, D. (2019). A closer look at spatiotemporal convolutions for action recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 6450-6459).
19. Liwicki, M., Everingham, M., & Gehler, P. (2009). Automatic sign language analysis: A survey and the future beyond lexical meaning. IEEE Transactions on Pattern Analysis and Machine Intelligence, 31(6), 873-891.
20. Pigou, L., Van Den Oord, A., Dieleman, S., & Schrauwen, B. (2015). Sign language recognition using convolutional neural networks optimized for mobile devices. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (pp. 26-33).
21. T.E. Starner and A. Pentland. Real-Time American Sign Language Recognition from Video Using Hidden Markov Models. In Technical Report no. 375, MIT Cambridge, Media Laboratory, 1995.
22. Pfister, T., Charles, J., & Zisserman, A. (2014). Flowing convnets for human pose estimation in videos. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 1913-1920).
23. Stoll, S., Nickisch, H., & Seidel, H. P. (2005). Statistical pose models for acquiring complex hand configurations. ACM Transactions on Graphics (TOG), 24(3), 697-717.
24. Koller, O., & Ney, H. (2009). Continuous sign language recognition: Towards large vocabulary statistical recognition systems handling multiple signers. Computer Vision and Image Understanding, 113(5), 663-672.
25. Camgoz, N. C., Hadfield, S., Koller, O., Bowden, R., & Ney, H. (2017). Neural sign language translation. In Proceedings of the IEEE International Conference on Computer Vision (pp. 778-787).

26. Presti, L. L., & La Cascia, M. (2013). 3D skeleton-based human action classification: A survey. Pattern Recognition, 46(1), 138-154.
27. Zelinka, J., & Porr, B. (2010). Real-time sign language recognition using a high-speed vision system. In 2010 10th IEEE-RAS International Conference on Humanoid Robots (Humanoids) (pp. 417-422).
28. Cruz, E., Soares, E., & Franco, C. (2008). American sign language fingerspelling recognition using leap motion controller. In 2008 Panhellenic Conference on Informatics (pp. 205-209)
29. Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe: Convolutional Architecture for Fast Feature Embedding. arXiv preprint arXiv:1408.5093, 2014.
30. Li, Y., & Hu, H. (2013). Real-time hand gesture recognition using finger-IMU and EMG sensors. In 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (pp. 3158-3163).
31. Koller, O., & Ney, H. (2006). Multi-stream ASR using articulatory features for continuous sign language recognition. In 2006 IEEE International Conference on Acoustics Speech and Signal Processing Proceedings (ICASSP) (Vol. 1, pp. I-I).
32. Huenerfauth, M. (2008). American sign language animation using web-based video. In Proceedings of the 10th international ACM SIGACCESS conference on Computers and accessibility (pp. 253-254).