Real-time Italian Sign Language Recognition with Deep Learning

Veronica J. Schmalz[1,2]

[1]*ITEC, imec research group at KU Leuven, Etienne Sabbelaan 51, 8500, Kortrijk, Belgium*
[2]*Freie Universität Bozen-Bolzano, Universitätsplatz, 1, 39100, Bozen, Italy*

Abstract

Image recognition systems have evolved so much that they can actually be exploited to solve significant challenges today, such as facilitating communication for people with hearing impairments relying on sign languages. This project aims to apply deep learning and fine-tuning techniques to build an automatic recognition system for the Italian Sign Language (LIS). More specifically, our goal is a real-time image recognition system capable of accurately identifying the letters of the LIS alphabet provided by a user in a Human Computer Interaction (HCI) framework by means of Python's Open Source Computer Vision (OpenCV) library and two models based on convolutional neural networks, namely CNN and VGG19, applied for large-scale image and video recognition. In addition to testing the performance of different architectures, our work constitutes a novel step towards the application of automatic image recognition techniques with the recently acknowledged LIS and a lately released open-source dataset, also representing the only source available for this type of research on single-handed isolated signs. This project may not only play a role in the interpretation and learning of the Italian Sign Language, encouraging its spread and study, but also in the inclusion of hearing-impaired individuals in the language research domain.

Keywords

sign languages, image recognition, deep learning, Italian Sign Language

1. Introduction

Sign languages (SLs) represent the most well-structured and organised means of communication apart from the oral languages spoken around the world. They are primarily used among individuals suffering from hearing loss and acoustic impairments via signs and gestures in the visual space. Similarly to spoken languages, SLs do not constitute a universal language but differ according to the areas and community groups from which they originate.

To date, there is no official data confirming the current number of sign languages used in the world, yet at least 161 SLs have been documented [1]. The main distinctive features of SLs are the multimodality, simultaneity and iconicity of the communicative act. Indeed, the linguistic information is conveyed by means of visual and manual interactions to which the interlocutor needs to simultaneously pay attention. These are hand shapes and movements, together with oriented gestures, facial expressions and mouthing[2, 3]. Given the complex set of elements

CEUR Workshop Proceedings (CEUR-WS.org)

that must be taken into account when analysing SL use, in this project we will mainly focus on one-handed isolated static signs, generally performed with the signer's dominant hand. More specifically, in our case we will be considering the letters of the LIS alphabet, one of the key elements to acquire when learning a sign language. The rest of the paper is organised as follows. Section 2 provides relevant details concerning the Italian Sign Language. A brief overview about the use of automatic recognition strategies in sign languages is presented in Section 3. Next, Section 4 describes the datasets taken into account for this project. In Section 5 we outline the tools and methodologies used for our experiments, which are analysed in detail in Section 6. Moreover, in Section 7 we report the obtained models' results and in the subsequent section, namely 8, we briefly describe an evaluation made in HCI using our best architecture and OpenCV. Finally, in Section 9 we draw our conclusions and consider possible future work.

2. The Italian Sign Language (LIS)

In Italy, approximately 10,2% of the population, namely 6,198,000 people, suffers from hearing problems, including absolute deafness or hearing loss [4]. Particularly, 1/1000 of the country's population is attested to already have hearing impairments from birth. Nevertheless, only children born from hearing impaired parents, 5% circa of the infantile deafness cases, spontaneously acquire the LIS given their exposure to it by the parents. On the contrary, 95% of children with hearing loss from non-deaf parents rarely have access to SL in early age [5]. Although highly variable factors such as the degree of hearing loss, the age of the loss and the adoption of implants or rehabilitation therapies differentiate deaf subjects, there is scientific evidence on the effectiveness and support of SL to facilitate communication, socialisation and integration [6]. Nevertheless, the interest in SL in Italy has not always been keen. Indeed, prejudices and misconceptions have taken time to be dismantled in families, the medical community and society in general [7, 4]. Recently, the Italian Parliament has recognised the LIS [8] as an official language used in the country. Despite being the last state of the European Union acknowledging the effective status of this linguistic minority, this act offers grounds for greater inclusion of hearing-impaired people in society and the diffusion of their language. In fact, unlike other widespread sign languages, such as the American Sign Language (ASL), LIS counts a reduced number of users and is rarely studied by both language learners and scholars. It is no coincidence that there are few datasets available for research in the field of language technologies and the like [9, 10]. More specifically, for the creation of an automatic recognition system like ours, based on the static, isolated and one-handed letters of the LIS alphabet, there is currently only one corpus available online [11]. One of the aims of this work is, therefore, also to promote studies in this field and encourage the creation of material available for further research in this domain.

3. Related Work

The goal of this project was the creation of a system for Italian Sign Language recognition based on static alphabet signs. During its conception, however, we were confronted with two main issues: the lack of isolated static LIS data, together with limited online accessibility

and dissemination material. In fact, although current literature reviews [12, 13, 14] on sign language recognition count more than 400 articles and 90 deep sign language recognition models covering 25 different sign languages, mainly ASL, Indian Sign Language (ISL), Arabic Sign Language (ArSL) and Chinese Sign Language (CSL), there are few traces related to research on LIS. Nevertheless, recently Italian scholars in the domain of Applied Linguistics and the like have begun to pay more attention to this topic and new research trends have emerged, especially dealing with dynamic sign language recognition and extraction [10, 15].

In the global sphere of SL recognition research, the first recognition experiments date back to the 1990s using colored gloves, neural networks [16, 17, 18] and Hidden Markov Models [19] for the classification isolated signs. This type of research focuses on single letters or words at a time, as opposed to continuous and dynamic sign recognition, where transition movements and speed play a crucial role for the recognition task [20, 21]. Additionally, for the latter, powerful computer vision and image processing techniques are required in order to be able to capture spatial and temporal features. Among these, skin color detection, background subtraction, hand motion detection and trajectory estimation are some of the most widely adopted [22, 23, 24, 25, 26]. With the advent of Microsoft Kinect, new data formats were introduced using depth and RGB streams [27, 28]. The emergence of deep learning models, such as CNN, contributed to the automatic extraction of features [29, 30, 31], while sequence models like RNN, GRU and LSTM revolutionised the encoding of temporal information [32, 33].

As far as resources for SL recognition are concerned, there are several international benchmark corpora available. Among them the most widely adopted are those dedicated to ASL, which in a recent literature review [12] counts over 24 different systems applied on data of diverse nature. These include systems that achieve significantly high accuracy, such as 99.8% on single-handed, static isolated alphabet signs with KNN [34] and NN [35], and 82.3% on real time signing with SVM [36]. Second in terms of quantity of research is the ISL, whereby one system based on leap motion acquired signs achieves up to 100% accuracy in the recognition of single handed, dynamic, isolated signs using ANN [37]. Next are ArSL and CSL, reaching 99.97% accuracy with HMM based systems on camera, single handed, static isolated signs [38] and 100% on Kinect, single and double handed, dynamic isolated words [39]. In the Italian scenario, instead, the majority of existing corpora concerns video contents, such as the one created by the PRIN project, on spontaneous conversations, narratives and picture naming [40], the Italian Sign Language Bank, a large parallel corpus between Italian and LIS related to lexicon and meaning [9], and the one concerning the Italian Sign Language Alphabet generated through the Myo Gesture Control Armband [10]. To our knowledge, only one open source corpus containing static images of the LIS alphabet signs exists [11]. More information about it is provided in the following section.

4. Dataset

Similar to oral languages, the LIS is supported by an alphabet consisting of 26 letters to orthographically represent words. The first signed Italian alphabet, established by the theologian Assarotti in Genoa, dates back to the 19th century and involved combining single and double handed signs by placing them in respective body areas [4]. In the 1970s, however, due to the

influence of the other more widely used manual alphabets, Italian signers began to adopt a different system which quickly became widespread. The latter involved using only the dominant hand and producing the signs in a neutral space.
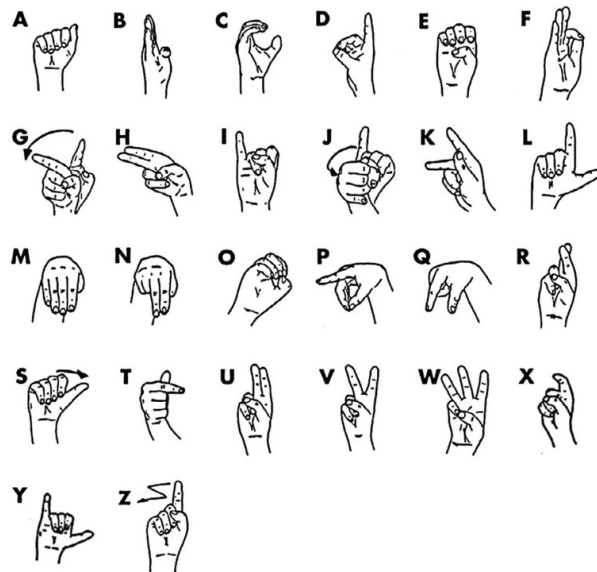

Figure 1: The Italian Sign Language (LIS) alphabet represented with static frontal signs [10]. Arrows are used to indicate the expected movements in the case of G, J, S and Z signs.

In Figure 1 a representation of the LIS alphabet used to this day is displayed. As can be seen from the signs in Figure 1, most letters can be statically signed, whereas letters such as G, J, S and Z, require slight movements as indicated by the arrows. Fingerspelling is rarely used in sign languages, especially in LIS, as it is generally applied to express concepts that do not have a corresponding sign in the language. This phenomenon is also known as *local lexicalisation*. However, the alphabet constitutes an important element in SL and is in fact one of the first components to be learned. Our project for static sign language recognition is based on it, and in particular on the images from the dataset created by Donnici and Monica [11] with the collaboration of the Gualandi Foundation.
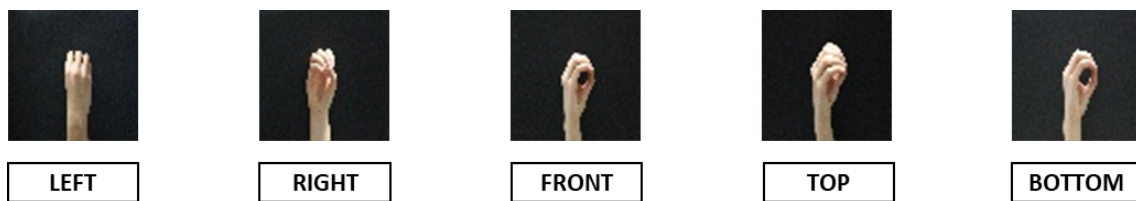

Figure 2: Examples from the dataset [11] for the *O* sign in five different perspectives, namely left (1), right (2), front (3), top (4) and bottom (5) from the same signer.

It represents the only available corpus of the LIS isolated and static alphabet signs and consists

of 22 signed letters' categories, excluding G, S, J and Z, photographed from five different angles, namely front, top, bottom, left and right. The total number of elements is 11,008 images from eleven different signers, which have been divided into three sub-datasets. The first sub-portion of the dataset contains images on a black background from all angles (11,008) (see Figure 2 for an example), the second contains only top, front and bottom images (5,954), and the third only frontal images (1,869). For our experiments we resorted to the first and third one in particular. The adopted methods are described in detail in the following section.

5. Methodology

Automatic recognition of sign languages represents a complex matter with which the research community has been dealing for some time. Since the advent of deep learning techniques, especially with regard to the use of convolutional neural networks (CNN), researchers have been experimenting with approaches of automatic feature engineering and classification via image recognition [41, 11, 42], achieving significantly accurate results (usually well above 90% of accuracy). Given the ability of CNN components for feature extraction from the provided data with a high degree of generalization, they recognise variable elements in images, ranging from people's faces, to animals, characters, etc. while guaranteeing robustness and efficiency. On account of their effectiveness, we use a CNN model trained from scratch and VGG19, a pre-trained one, for our experiments with the LIS alphabet. The first of these is inspired by the architecture used by Bheda and Radpour [41] for ASL recognition. It consists of three groups of convolutional layers, a Max-Pooling layer and a Dropout one, followed by two groups of fully connected layers, a Dropout layer and a final Output layer (see Figure 3 left).
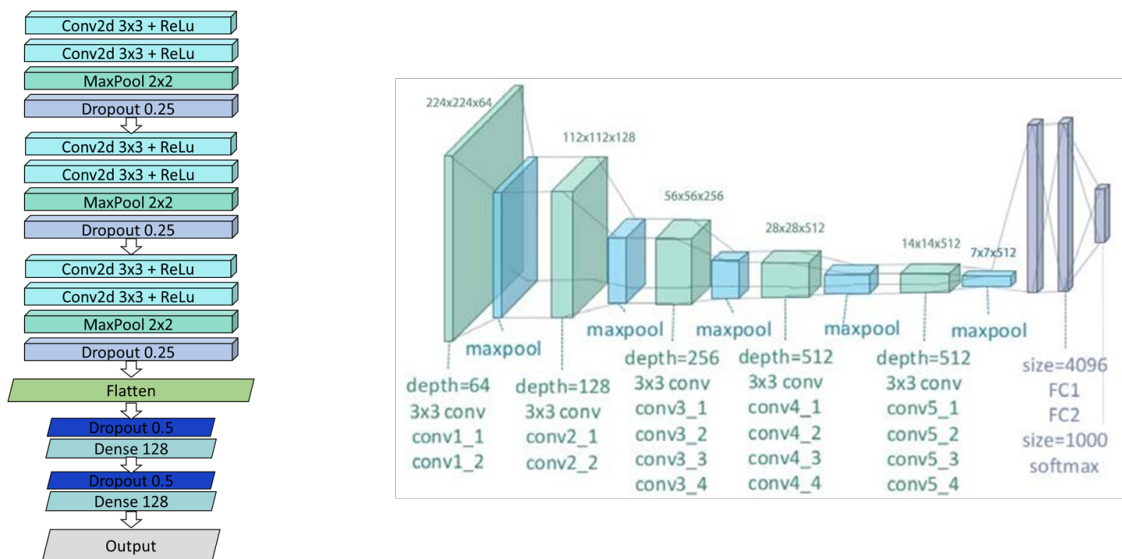

Figure 3: CNN architecture (left) and VGG19 architecture (right).

Alongside with this model, which we trained from scratch using isolated static LIS sign

images, we also experimented with a pre-trained deep neural model, VGG19 (see Figure 3 right), constituted by a convolutional encoder of 16 layers, two fully connected Dense layers, a Softmax layer and five Max-Pooling layers [43]. The reason for this was that the latter had been successfully trained on a larger amount of data, namely 14,197,122 images from ImageNet [44]. However, since the original training set did not include SL signs, we had to fine-tune the model with our LIS images. In this process, we froze the first six layers to avoid re-training and added four final Dense layers, following [45]. Therefore, the above-described models, implemented in Keras, received as input during the training, validation and test phases, the images of the LIS alphabet categorised in 22 classes. For data augmentation we adopted Keras' *ImageDataGenerator*, which generates batches of tensor images to be looped. Following the training and evaluation of the models, we finally opted to test the best one in a Human Computer Interaction (HCI) perspective. To do this, we employed OpenCV [46], a Python library for computer vision and image recognition adopting a cascade approach. After accessing the user's camera, we proceeded to capture images of the hand signs currently produced in a predefined area and processed them through our pre-trained model. The latter classified the LIS signs at the moment, providing and displaying three different hypotheses varying in terms of accuracy. The specific details of our conducted experiments are illustrated in the following section.

6. Experiments

To perform our automatic LIS alphabet sign recognition task, we conducted two types of experiments. First, we used the CNN architecture (see Figure 3) trained from scratch with two sub-datasets containing fingerspelling images from five to a single different perspectives [11]. In the second case, attempting to improve the results, we employed the pre-trained VGG19 model [43], fine-tuning 10 of the 16 layers contained in it, inspired by Cabana et al. [45]. Details about the training parameters of each experiment are provided in Table 1.

Table 1
Parameters and details about the experiments with CNN and VGG19 models using either frontal or 5-distinct-angles images.

Parameters	CNN model	VGG19
Input size	64x64	28x28
Batch size	32/64	32/64
Epochs	50/100	50/100
Optimizer	Adam	SGD
Loss	Categ. cross-entropy	Categ. cross-entropy

In both cases we performed two types of experiments, first using the dataset with only frontal sign images (1,105 images in the training set and 519 images in the validation set) and then the larger dataset from different angles (6,028 images in the training set and 2,952 images in the validation set), for 50 and 100 epochs respectively, with a learning rate of 0.01. The two models were fed images of different sizes in pixels, 64x64 and 28x28 respectively, which were processed in batches of 32 or 64 in each one of the experiments. When using the CNN model, we adopted

Adam (Adaptive Moment Estimation) as optimizer and applied the categorical cross-entropy loss. In contrast, for the experiments carried out with the pre-trained VGG-19, which we fine-tuned, we selected SGD (Stochastic Gradient Descent) and the same loss as before.

7. Results

Once we had trained our eight different models, we proceeded to evaluate them. Since we had employed a data generator during the training, we adopted Keras' *evaluate_generator*, to which we passed the test sets of the two respective sub-corpora. Subsequently, we mainly considered the models' accuracy, or the number of correct predictions, together with precision, recall and F1-scores extracted with the *classification_report* function from Scikit-learn. A summary of the accuracy obtained values is available in Table 2 for CNN and in Table 3 for VGG19. These two provide information about the model, the number of training epochs and the images contained in the training set, reflecting the less numerous dataset with only front signs, and the one with five distinct angles' images.

Table 2
Results from the different experiments with CNN using frontal images and images from five different angles.

Model	Epochs	Test set	Accuracy
CNN1	50	238	92%
CNN2	50	1220	94%
CNN3	100	238	**97%**
CNN4	100	1220	93%

Table 3
Results from the different experiments with VGG19 using frontal images and images from five different angles.

Model	Epochs	Test set	Accuracy
VGG19.1	50	238	95%
VGG19.2	50	1220	98%
VGG19.3	100	238	97%
VGG19.4	100	1220	**99%**

The results in Table 3 prove that the pre-trained VGG19 model performed overall better than the simple CNN (see Table 2), achieving 99% accuracy on 22 characters after 100 epochs with the larger sub-dataset. This is presumably due both to the number of images used to pre-train and fine-tune it, 11,008 in total, and to the possibility of extracting different features considering

the numerous angles from which the images of each letter of the LIS alphabet were taken. From the classification report in Figure 4 we can confirm that the total accuracy of the model, 99%, is also reflected in the F1-scores, the harmonic mean between precision and recall, of all 22 classes. In fact, the values range from a minimum of 0.96 to a maximum of 1.00 on 10 letters of the LIS alphabet.

	precision	recall	f1-score	support
A	1.00	1.00	1.00	46
B	0.98	1.00	0.99	48
C	0.98	1.00	0.99	53
D	0.96	1.00	0.98	52
E	1.00	1.00	1.00	56
F	1.00	0.98	0.99	58
H	0.98	1.00	0.99	59
I	1.00	1.00	1.00	58
K	1.00	0.98	0.99	59
L	1.00	1.00	1.00	59
M	1.00	0.96	0.98	57
N	0.97	1.00	0.98	56
O	1.00	1.00	1.00	56
P	1.00	1.00	1.00	55
Q	1.00	1.00	1.00	55
R	0.98	0.95	0.96	56
T	1.00	1.00	1.00	54
U	0.97	0.98	0.97	57
V	0.98	0.96	0.97	52
W	1.00	1.00	1.00	59
X	1.00	1.00	1.00	57
Y	1.00	0.98	0.99	58
accuracy			0.99	1220
macro avg	0.99	0.99	0.99	1220
weighted avg	0.99	0.99	0.99	1220

Figure 4: Classification report of the results obtained with the VGG19.4 model for each sign image class.

In contrast, among the CNN models trained from scratch, the best model is the one trained for 100 epochs using the dataset of frontal sign images only. The latter displays an overall accuracy of 97%, and remarkable precision on several signs (see Figure 5 for more details).
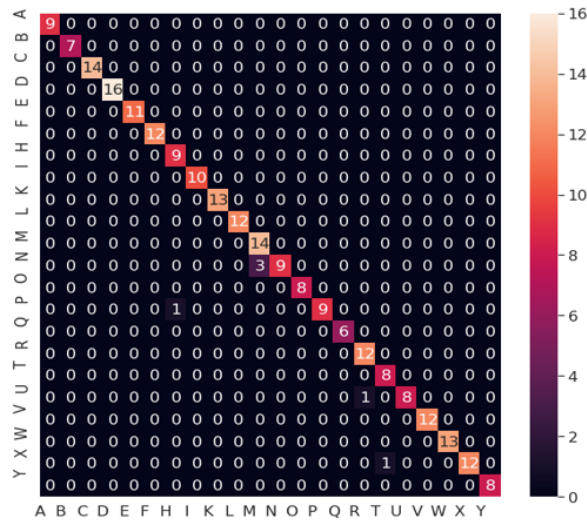

Figure 5: Confusion matrix representing the classification results with the CNN3 model (97% accurate).

8. Testing our best model with OpenCV

In order to verify the actual functioning of our best model, VGG19.4, in the recognition of the signs of the LIS alphabet, we decided to evaluate it in HCI using OpenCV [46]. Therefore, we employed this real-time image recognition tool by means of a script through which we accessed the camera of the user's PC. Then, we detected in real-time the sign performed within a pre-defined 28x28 square displayed in the user's window. During the image detection process, to distinguish the sign from the background, i.e. the foreground area, we calculated the accumulated weighted average to subtract from the latter. Then, with a threshold value of 25, we determined the sign contours to precisely define the bounding box to be processed, captured the frame, resized it and flipped it to avoid mirror view. The image, in RGB colour format, was then passed to our pre-trained model. The predictions of the latter appeared on the user's screen in the camera feed window. The most accurate prediction was represented with a green character, corresponding to the orthographic sign of the finger-spelled LIS sign, while the next two predictions, less accurate, appeared in red at the bottom of the square where the user performed the signing. For this experiment a user signed each of the 22 letters of the dataset 50 times in three different settings with diverse backgrounds and lighting conditions. Note that the user is not a highly proficient signer but a learner. Figure 6 below displays some examples [1].
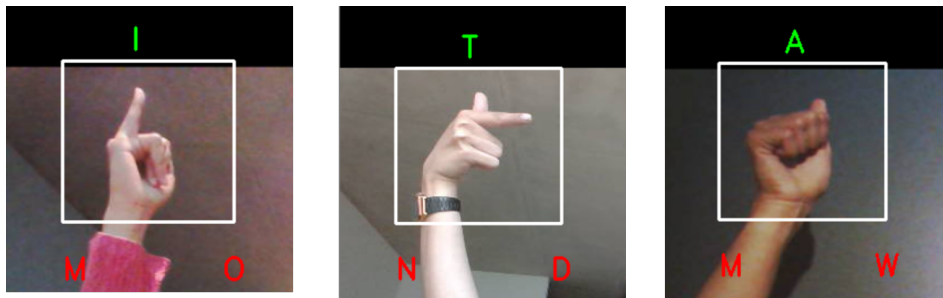

Figure 6: Examples of HCI in LIS alphabet signs recognition with OpenCV and VGG19.4

After considering the predictions for each letter, an overall accuracy of 77.2% was determined with an average of 3.46s for the correct recognition of each sign. Among them, the signs for which the most incorrect predictions were found were E, Q, X and M, very often misclassified as A, P, C and W. Considering the relative similarity between these signs and the fact that the user was not an highly experienced signer, these errors were somewhat predictable. Since we used the best model trained with images from several angles, i.e. front, right, left, above and below, through the attempts with the OpenCV library tools we have additionally demonstrated the model's capacity to effectively detect most of the signs, including some that were made in unconventional or imprecise angles by the user, i.e. differently from what was represented in Figure 1.

In addition, if users prefer, by pressing the space key on the keyboard they can first display on a pop-up window of their screen a random image from the dataset to be signed, which they

---

[1]A demo and more details about this experiment and project can be found under https://github.com/VeroJulianaSchmalz/LIS-Recognition-with-Deep-Learning-/

will use as a reference, and then try to reproduce it. In this case, to assess the proper execution of the sign, models such as CNN3 (see Figure 5), trained with front-only images reflecting the references in Figure 1, could be used.

9. Conclusion and Future Work

In this paper we described the application of deep learning and fine-tuning techniques for the recognition of LIS alphabet signs. We described the different experiments we conducted using the only available isolated static LIS dataset for image recognition and classification, and two models, a CNN designed with Keras that we trained from scratch, and a deeper pre-trained model, VGG19, which we had to fine-tune. Based on two different sections of the corpus, one with numerous multi-angle images, and one with frontal images only, we obtained an accuracy of 99% with VGG19.4 and of 97% with the CNN3. Finally, by means of OpenCV we confirmed the accuracy of our best model in HCI through real-time user's inputs. The obtained results provide comparability with other related studies conducted in state-of-the-art sign recognition experiments, despite the fact that no similar studies on LIS one-handed isolated alphabet signs have been published so far. Additionally, we managed to demonstrate that it is possible to achieve significant results from relatively simple image recognition technologies, as long as there is a consistent dataset with which to train and test the neural models. It might be possible to build an even larger corpus with the collaboration of national hearing-impaired communities and organizations, as there are currently no other similar static hand-signed datasets concerning the basic LIS alphabet. In the future, we might consider extending the recogniser to interpret single-handed isolated words rather than simple alphabet signs. However, given the dynamism and multi-channelled nature of sign language, one possibility would be to design a system that takes into account double-handed signs, again in real-time HCI or continuous image data. Finally, with a broader number of video data from different signers, a real-time LIS interpreting system could be envisaged. Projects of this kind constitute important resources for valuing the diversity and inclusion of people with hearing impairments in society. Additionally, our system could also help promoting the learning of Italian Sign Language, allowing learners to independently practice fingerspelling and become fluent SL users.

References

[1] Ethnologue, Sign language, 2021. URL: https://www.ethnologue.com/subgroups/sign-language.

[2] A. Kusters, M. Spotti, R. Swanwick, E. Tapio, Beyond languages, beyond modalities: Transforming the study of semiotic repertoires, International Journal of Multilingualism 14 (2017) 219–232.

[3] J. C. Lu, S. Goldin-Meadow, Creating images with the stroke of a hand: Depiction of size and shape in sign language, Frontiers in psychology 9 (2018) 1276.

[4] C. Branchini, L. Mantovan, A grammar of italian sign language (lis), 2020.

[5] E. Tomasuolo, T. Gulli, V. Volterra, S. Fontana, The italian deaf community at the time of coronavirus, Frontiers in Sociology 5 (2021) 125.

[6] P. Rinaldi, E. Tomasuolo, A. Resca, La sordità infantile: nuove prospettive di intervento, Erickson, 2018.

[7] V. Volterra, Chi ha paura della lingua dei segni?, Psicologia clinica dello sviluppo 18 (2014) 425–427.

[8] Decreto legge 22 marzo 2021, n. 41: Misure urgenti in materia di sostegno alle imprese e agli operatori economici, di lavoro, salute e servizi territoriali, connesse all'emergenza da covid-19. art. 34 ter- misure per il riconoscimento della lingua dei segni italiana e l'inclusione delle persone con disabilità uditiva (2021).

[9] P. Prinetto, U. Shoaib, G. Tiotto, The italian sign language sign bank: Using wordnet for sign language corpus creation, in: 2011 International Conference on Communications and Information Technology (ICCIT), IEEE, 2011, pp. 134–137.

[10] I. Pacifici, P. Sernani, N. Falcionelli, S. Tomassini, A. F. Dragoni, A surface electromyography and inertial measurement unit dataset for the italian sign language alphabet, Data in Brief 33 (2020).

[11] M. Donnici, G. Monica, Italian sign language fingerspelling recognition, 2018. URL: https://github.com/maghid/italian_fingerspelling_recognition.

[12] A. Wadhawan, P. Kumar, Sign language recognition systems: A decade systematic literature review, Archives of Computational Methods in Engineering 28 (2021) 785–813.

[13] R. Rastgoo, K. Kiani, S. Escalera, Sign language recognition: A deep survey, Expert Systems with Applications 164 (2021) 113794.

[14] R. Elakkiya, Machine learning based sign language recognition: A review and its research frontier, Journal of Ambient Intelligence and Humanized Computing 12 (2021) 7205–7224.

[15] G. Saggio, P. Cavallo, M. Ricci, V. Errico, J. Zea, M. E. Benalcázar, Sign language recognition using wearable electronics: implementing k-nearest neighbors with dynamic time warping and convolutional neural network algorithms, Sensors 20 (2020) 3879.

[16] K. Murakami, H. Taguchi, Gesture recognition using recurrent neural networks, in: Proceedings of the SIGCHI conference on Human factors in computing systems, 1991, pp. 237–242.

[17] S. S. Fels, G. E. Hinton, Glove-talk: A neural network interface between a data-glove and a speech synthesizer, IEEE transactions on Neural Networks 4 (1993) 2–8.

[18] S. A. Mehdi, Y. N. Khan, Sign language recognition using sensor gloves, in: Proceedings of the 9th International Conference on Neural Information Processing, 2002. ICONIP'02., volume 5, IEEE, 2002, pp. 2204–2206.

[19] K. Grobel, M. Assan, Isolated sign language recognition using hidden markov models, in: 1997 IEEE International Conference on Systems, Man, and Cybernetics. Computational Cybernetics and Simulation, volume 1, IEEE, 1997, pp. 162–167.

[20] R.-H. Liang, M. Ouhyoung, A sign language recognition system using hidden markov model and context sensitive search, in: Proceedings of the ACM symposium on virtual reality software and technology, 1996, pp. 59–66.

[21] S. Marcel, O. Bernier, J.-E. Viallet, D. Collobert, Hand gesture recognition using input-output hidden markov models, in: Proceedings Fourth IEEE International Conference on Automatic Face and Gesture Recognition (Cat. No. PR00580), IEEE, 2000, pp. 456–461.

[22] N. B. Ibrahim, M. M. Selim, H. H. Zayed, An automatic arabic sign language recognition system (arslrs), Journal of King Saud University-Computer and Information Sciences 30

(2018) 470–477.

[23] J. Han, G. Awad, A. Sutherland, Modelling and segmenting subunits for sign language recognition based on hand motion analysis, Pattern Recognition Letters 30 (2009) 623–633.

[24] X. Yang, X. Chen, X. Cao, S. Wei, X. Zhang, Chinese sign language recognition based on an optimized tree-structure framework, IEEE journal of biomedical and health informatics 21 (2016) 994–1004.

[25] F.-S. Chen, C.-M. Fu, C.-L. Huang, Hand gesture recognition using a real-time tracking method and hidden markov models, Image and vision computing 21 (2003) 745–758.

[26] R. Elakkiya, K. Selvamani, S. Kanimozhi, R. Velumadhava, A. Kannan, Intelligent system for human computer interface using hand gesture recognition, Procedia engineering 38 (2012) 3180–3191.

[27] Z. Zhang, Microsoft kinect sensor and its effect, IEEE multimedia 19 (2012) 4–10.

[28] Z. Zafrulla, H. Brashear, T. Starner, H. Hamilton, P. Presti, American sign language recognition with the kinect, in: Proceedings of the 13th international conference on multimodal interfaces, 2011, pp. 279–286.

[29] O. Koller, H. Ney, R. Bowden, Deep hand: How to train a cnn on 1 million hand images when your data is continuous and weakly labelled, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 3793–3802.

[30] D. Tran, L. Bourdev, R. Fergus, L. Torresani, M. Paluri, Learning spatiotemporal features with 3d convolutional networks, in: Proceedings of the IEEE international conference on computer vision, 2015, pp. 4489–4497.

[31] J. Huang, W. Zhou, H. Li, W. Li, Attention-based 3d-cnns for large-vocabulary sign language recognition, IEEE Transactions on Circuits and Systems for Video Technology 29 (2018) 2822–2832.

[32] O. Koller, N. C. Camgoz, H. Ney, R. Bowden, Weakly supervised learning with multi-stream cnn-lstm-hmms to discover sequential parallelism in sign language videos, IEEE transactions on pattern analysis and machine intelligence 42 (2019) 2306–2320.

[33] O. M. Sincan, A. O. Tur, H. Y. Keles, Isolated sign language recognition with multi-scale features using lstm, in: 2019 27th Signal Processing and Communications Applications Conference (SIU), IEEE, 2019, pp. 1–4.

[34] D. Aryanie, Y. Heryadi, American sign language-based finger-spelling recognition using k-nearest neighbors classifier, in: 2015 3rd International Conference on Information and Communication Technology (ICoICT), IEEE, 2015, pp. 533–536.

[35] M. Zamani, H. R. Kanan, Saliency based alphabet and numbers of american sign language recognition using linear feature extraction, in: 2014 4th International conference on computer and knowledge engineering (ICCKE), IEEE, 2014, pp. 398–403.

[36] C. Savur, F. Sahin, Real-time american sign language recognition system using surface emg signal, in: 2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA), IEEE, 2015, pp. 497–502.

[37] D. Naglot, M. Kulkarni, Ann based indian sign language numerals recognition using the leap motion controller, in: 2016 International Conference on Inventive Computation Technologies (ICICT), volume 2, IEEE, 2016, pp. 1–6.

[38] A. A. Ahmed, S. Aly, Appearance-based arabic sign language recognition using hidden markov models, in: 2014 international conference on engineering and technology (ICET),

IEEE, 2014, pp. 1–6.

[39] J. Zhang, W. Zhou, C. Xie, J. Pu, H. Li, Chinese sign language recognition with adaptive hmm, in: 2016 IEEE international conference on multimedia and expo (ICME), IEEE, 2016, pp. 1–6.

[40] C. Geraci, K. Battaglia, A. Cardinaletti, C. Cecchetto, C. Donati, S. Giudice, E. Mereghetti, The lis corpus project: A discussion of sociolinguistic variation in the lexicon, Sign Language Studies 11 (2011) 528–574.

[41] V. Bheda, D. Radpour, Using deep convolutional networks for gesture recognition in american sign language, arXiv preprint arXiv:1710.06836 (2017).

[42] A. Wadhawan, P. Kumar, Deep learning-based sign language recognition system for static signs, Neural Computing and Applications 32 (2020) 7957–7968.

[43] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, arXiv preprint arXiv:1409.1556 (2014).

[44] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al., Imagenet large scale visual recognition challenge, International journal of computer vision 115 (2015) 211–252.

[45] G. M. Cabana, Elisa, J. Viader, Sign language translator opencv, 2019. URL: https://github.com/ecabestadistica/sign-language-translator-python-opencv.

[46] G. Bradski, A. Kaehler, The opencv library, Dr Dobb's Journal of Software Tools 25 (2000) 120.