Introducing Deep Learning with Data Augmentation and Corpus Construction for LIS

Manuela Marchisio[1,*,†], Alessandro Mazzei[1,†] and Dario Sammaruga[2,†]

[1]Università degli Studi di Torino - Corso Svizzera 185, 10149, Torino, Italy
[2]Orbyta Tech S.r.l. - Piazza Castello 113, 10121 Torino, Italy.

Abstract

The development of home video recording has had a big impact in the development of video documents containing Italian Sign Language (LIS) sentences. LIS2SPEECH is an ongoing project by Orbyta Tech s.r.l. to build a complete translation chain from LIS to speech. The idea is to build a free software framework to transform video containing LIS sentence into Italian vocal sentences. In this way, LIS signers can indirectly produce Italian vocal sentences. In this paper we describe two milestones for LIS2SPEECH, that are: i. the development of some deep neural models trained by using data augmentation technique, and ii. the construction of a new dataset for LIS to Italian. Referring to the first point, a number of deep learning models were developed and tested. Then data augmentation was performed by using some geometric transformations to the videos belonging to the original training set. With reference to the second point, we constructed the TGLIS-227 dataset by using video and audio segmentation techniques, starting from a corpus of RAI newscasts. This dataset is a novelty in the current research panorama as there are no public datasets for LIS with sentence-level granularity.

Keywords

Sign Language Recognition, LIS language, Deep Learning, RNNs, CNN, LIS dataset, Data Augmentation,

1. Introduction

In this paper, we present the architecture of a real-time translation system from Italian Sign Language (henceforth LIS) to Italian speech. An ideal system platform for this task should be composed of three main modules:

- a first module which, starting from an input video, returns the glosses[1] contained in the video. So, this module performs a *Sign Language Recognition* Task.
- a second module that translates the glosses into the Italian language. So, this module performs a *Sign Language Translation* task.
- a third module for text-to-speech system, that is for pronouncing the sentences in Italian. So, this module performs a *Text to Speech* task.

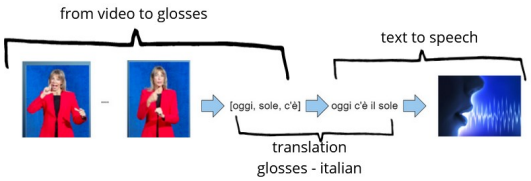


Figure 1: High level architecture for LIS2SPEECH

The task of Sign Language Recognition (SLR) is a classification task that allows to automatically obtain the glosses corresponding to the signs performed by a signer in a video. In general, the SLR is approached as a multiclass classification problem, i.e. a type of supervised learning which, on the basis of a (statistical) model, can associate the correct gloss among the k, where k is the cardinality of the LIS dictionary considered. In the specific settings of the LIS2SPEECH[2] project, the input is a sequence of signs encoded in a video while the labels are the corresponding glosses. So, this is a case of supervised learning, where the models are trained on a dataset containing numerous examples (signs) labeled with the relative class (gloss).

Considering the module for translation from LIS to Italian, this implement a task where is true the rule "more data is better data". In this paper we follow this prescription in two distinct directions. In Section 2, we describe a number of experiments with deep neural models trained

CLiC-it 2023: 9th Italian Conference on Computational Linguistics, Nov 30 — Dec 02, 2023, Venice, Italy
*Corresponding author.
†These authors contributed equally.
manuela.marchisio@edu.unito.it (M. Marchisio);
alessandro.mazzei@unito.it (A. Mazzei);
dario.sammaruga@orbyta.it (D. Sammaruga)
https://github.com/march2345 (M. Marchisio);
https://github.com/alexmazzei (A. Mazzei);
https://github.com/BeanRepo (D. Sammaruga)
0009-0002-6658-9340 (M. Marchisio); 0000-0003-3072-0108 (A. Mazzei); 0009-0005-4276-9269 (D. Sammaruga)

[1]We write "gloss" to denote a naming system for signs together with the encoding of the relevant morpho-syntactic features.

[2]Copyright for LIS2SPEECH project by Orbyta Tech s.r.l.

by using a data augmentation technique. We enlarge the possibility given by a relatively small initial dataset, by using a number of geometrical transformations to the original videos. We describe these transformations and experiment their impact on the performances of the Isolated Sign Language Recognition (ISLR) task, i.e. when each video contains a single sign. Moreover, in Section 3, we describe the initial steps toward the release of a new dataset for LIS in the news domain. We provide a description of the algorithmic process used to provide a sentence level segmentation of the original videos. In Section 4, we summarizing the contributions of this paper and provide a brief description of the ongoing work.

2. Deep Learning models for Italian SLR

In all experiments concerning neural learning there are two crucial ingredients, these are *the dataset* and the *neural architecture*. In this Section, we describe a number of work related to our project (Section 2.1), we describe the dataset employed in our experiments (Section 2.2), we describe the details of the preprocessing and training applied (Section 2.3) and, finally we report results of our experiments (Section 2.4).

2.1. Related Works on SLR

The major related works in sign language recognition consider 2 important features: the input and neural models used. First, the input could be static (an image for each sign) or dynamic (a video for each sign). The different granularity of the input allows the SLR to be divided into two different tasks: isolated sign language recognition (ISLR) and continuous sign language recognition (CSLR). The former takes a single sign as input and outputs the corresponding gloss. The latter instead takes as input a sentence or a sequence of signs and returns the correct sequence of glosses. [1] is a quantitative analysis of the state of the art on SLR based on more than 400 results from 1983 until today. In this analysis you note that the number of publications on isolated sign recognition is always greater than on a continuous one. Moreover, there are some works that use a *computer vision* approach to detect information from the input and others that use an *electronic* approach by using some gloves with electronic sensor. By limiting to neural networks models, [2] reports an analysis of the main models used in automatic sign language recognition (SLR) up to now, and includes too traditional machine learning classifiers such as SVM (support vector machine), HMM (hidden markov model), K-NN (k-nearest neighbours), ensemble learning and systems based on fuzzy logics. Recently, some other research studies have also been based on Transformers architectures [3, 4] and attention-based models [5]. However, most studies in this SLR task uses two specific neural architectures, that are convolutional [6, 7, 8] and recurrent neural networks [9, 10, 11]. On this basis, we have chosen to develop and train five different NN models: LSTM, GRU, BILSTM, BIGRU and CONVNET[3] (architecture's details in appendix A).

2.2. The LIS Dataset employed in the experiments

For the LIS there are few datasets (see Section 3), and as a consequence we used the only one suitable for SLR, that is the A3LIS-147 dataset [12]. A3LIS-147 was built by the A3LAB research group of the Università politecnica delle Marche, in collaboration with the ENS (Ente Nazionale Sordi, the Italian National Deaf institution) of Ancona. The dataset is composed of 1480 video. The corpus contains 147 standard (*natural*) signs, plus one special (*artificial*) sign for representing the "silence" (*sil sign*). The latter is not a sign belonging to an LIS natural dictionary, but it encodes the common resting position in corpus conversations.

Crucially, and in contrast to most electronic LIS dictionaries, all the signs of A3LIS-147 have been performed by 10 different signers. This peculiar property of the corpus allows us to use it as a training set for the isolated sign language recognition task (ISLR henceforth). For this specific task, each video represents a single sign preceded and followed by the sil sign (or rest).

2.3. Preprocessing and Training

In this Section, we describe the development of a deep neural system for ISLR trained on the A3LIS-147.

First of all, we have a preprocessing phase for converting videos into numerical data suitable for learning. In preprocessing, we extracted a total of 543 keypoints for each frame of the A3LIS-147 videos using the google model Mediapipe Holistics [13]. We decided to reduce this number to 535 since we eliminated 8 keypoints representing lower limbs. Indeed, very often the LIS signers in the videos are framed from the hip up.

Second, we used these keypoints as input for neural networks trainings. We splitted this set of data into two parts: in the initial phase we[4] use 70% in the training set and 30% in the test set. The split is stratified, i.e. it

[3] All these models have been developed using the Keras API, using a GPU NVIDIA-GeForce GTX 1650 with 4GB of RAM and a CPU Intel i7-9750H with 6 core and 16GB of RAM.

[4] In the final part of this work we use k-fold cross-validation to determine what is the best split between test and training. We obtain the best results with k=5, so best split is 80% in training set and 20% for test set

maintains the proportions of the classes in the training and in the test. After a number of initial experiments in training, we observed two emergent issues:

1. The size of the input was too high when the number of signs increased, provoking an out of memory error.
2. The results had a very bad recognition accuracy when considering the entire 148 signs dataset.

For these two reasons, we decided to perform two other preprocessing steps on the data for solving these issues.

These two steps work, in some senses, in two opposite directions. On the one hand, we performed data reduction (Section 2.3.1) for optimizing the number of features given in input to the neural ISLR classification model. On the other hand, we performed data augmentation (Section 2.3.2) on the number of videos for each sign. Indeed, we realized that 10 videos for each sign are too small number to allow the network to correctly classify. We discuss the impact of these two steps in Section 2.4 where we report the results of the experiments with various neural models.

2.3.1. Data Reduction

We reduced the number of keypoints extracted for each frame with the Google model by considering:

- the number of keypoints of the face is higher than other parts of the body and this could negatively affect the training of the model giving too much importance to this part of the body. For this reason we developed a function which allows us to go from 468 to 128 representative keypoints on the contours of the face, eyes, eyebrows and mouth.
- The Mediapipe documentation recommends discarding the z dimension because the Google system still has low performances in predicting the depth. For this reason, in some tests we discarded the z of each keypoint. In other words, we converted the original 3D data produced by mediapipe into 2D by just removing the z value[5].
- Finally, we applied the principal component analysis (PCA) to reduce the total number of keypoints to four-six components, which represents the 95% of explained variance.

2.3.2. Data Augmentation

We applied a Data Augmentation technique by increasing the number of videos for each sign by making some geometric transformations to the originals. In particular we performed: translation, rotation, flip and smoothing.

[5]In a different trial, we have tried to set the z coordinate to zero.

Translation The following transformation was applied to each original keypoint (x,y,z):

$$x_{trasl}, y_{trasl}, z_{trasl} = (x + \Delta x, y + \Delta y, z)$$
$$\Delta x = np.random.uniform(-max_{sx}, max_{dx})$$
$$\Delta y = np.random.uniform(-max_{giu}, max_{su})$$

where Δx represents the displacement on the x axis, while Δy represents the displacement on the y axis. Both these delta were randomly extracted from an uniform distribution by using, as the range, the values representing the maximum translation downwards, upwards, rightwards, leftwards. Using these limits, we guarantee that all the keypoint coordinates are still values between 0 and 1. The randomly extracted values are the same for all frames of a video. In Figure 2c, we report an example of applying this transformation.

Rotation The following transformation was applied to each original keypoint (x,y,z):

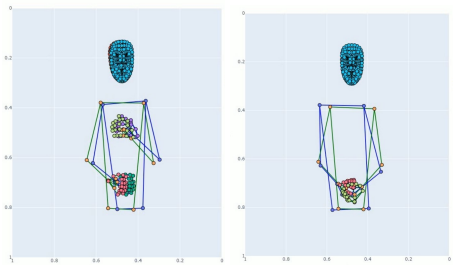
$$x_{rot} = (x - x_{centro})cos\theta - (y - y_{centro})sin\theta + x_{centro}$$
$$y_{rot} = (y - y_{centro})cos\theta + (x - x_{centro})sin\theta + y_{centro}$$
$$z_{rot} = z$$

The rotation was performed with respect to the center of all keypoints $centro = (x_{centro}, y_{centro})$. θ is the angle of rotation: the value is randomly extracted from the uniform distribution between $(-20, 20)$ and is the same for all the keypoints of the frames of one video. In Figure 2a we can see an example of applying this transformation to the keypoints of a video frame.

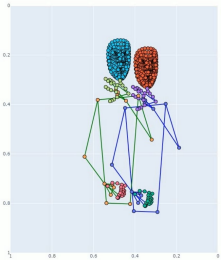
Flip The following transformation was applied to each original keypoint (x,y,z):

$$x_{flip}, y_{flip}, z_{flip} = (2k + x, y, z)$$

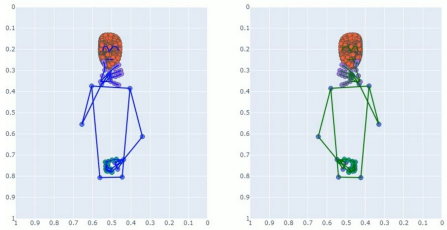
It is an axial symmetry with respect to the straight line $x = k$ parallel to the y axis. In our case k corresponds to the x coordinate of the center of all keypoints. This type of transformation is important because the same sign can also be performed symmetrically, because there are right-handed and left-handed signers. Without this transformation, symmetrical signs cannot managed



(a) original and rotated (b) original and smooth



(c) original and translated



(d) original and flipped

Figure 2: Frame's plot

properly by the neural models, since it recognizes them as different stimulus. In Figure 2d, we can see an example of applying this transformation to the keypoints of a video frame.

Smooth This type of transformation was implemented by considering the study presented in [3]. It consists in applying a different random rotation for each single keypoint up to a maximum of 13 degrees. Note that this transformation was not applied to every part of the body, but only to the keypoints related to the pose. Indeed, keypoints of pose have greater variations when the body moves independently by the head, which in most cases remains in a static position. Applying a different rotation to each point allows to capture variations in the execution of a sign due to a different signer: a slightly more bent elbow, one shoulder lower than the other, different proportions between body parts, etc. This is a crucial transformation because it really produces a kind of totally *new* keypoints, that is really different from those in the original dataset. In Figure 2b, we can see an example of applying this transformation to the keypoints of a video frame.

2.4. Experiments and results

We performed 800 tests divided into two different groups. The first group, called the general group, contains all the possible combinations (480 tests) of the parameters shown in Table 2 (Appendix B). In this group all the values of the parameters were tested. The second group, called specific data augmentation group, is designed to test the impact of each data augmentation transformation on results. It contains some combinations of all the possible combinations in Table 3 (Appendix B)

The results of each test is reported with the results of accuracy, precision and recall curve, confusion matrix, F1-score.

2.4.1. Test Evaluation

We upload the results of the test on the online platform QlikSense[6] in order to build a dashboard that allows us to visualize them. In this Section, we use the these graphical representations to comment results.

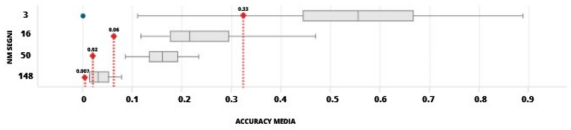


Figure 3: Box plot avg accuracy VS number of signs - without data augmentation

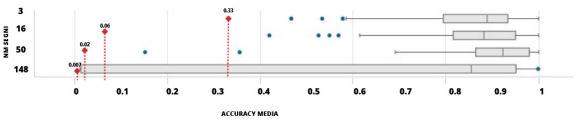


Figure 4: Box plot avg accuracy VS number of signs - with data augmentation

In Figure 3 we observed that if we consider the first group of tests, without data augmentation, the average accuracy decreases if the number of signs increases. The accuracy of a naive model that selects a class at random among the N possible ones using a uniform probability distribution are indicated in red. We use this baseline for accuracy for the models that we have trained. Moreover,

[6] https://www.qlik.com/it-it/products/qlik-sense

to correctly evaluate the generalization power, we consider for accuracy, the precision, the recall, the F1-score and the confusion matrix. In the Figure 3 and 4, we have that the baselines are 33% for 3 signs, 6% for 16 signs, 2% for 50 signs, 0.6% for 148 signs. In general, considering N signs, the baseline is equal to $\frac{1}{N}$ since the classes are balanced in the training dataset.

In Figure 5, we consider average accuracy for different neural models, and we have a very wide range of values for accuracy (from 0 to 0.8). In contrast, in the data

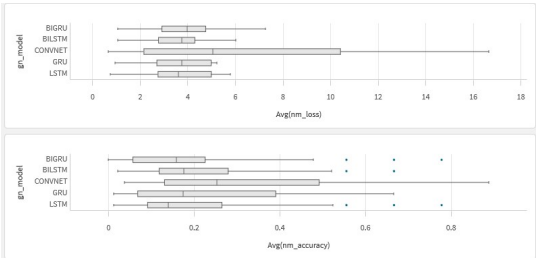


Figure 5: Box plot avg accuracy VS models - without data augmentation

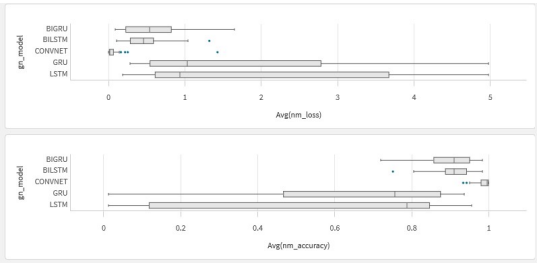


Figure 6: Box plot avg accuracy VS models - with data augmentation

augmentation experimentation, there is no decrease in accuracy when the number of signs increases (Figure 4) and there is a significant increase in the accuracy with respect to the models (Figure 6). Furthermore from the second group of tests we understood that the smooth is the best transformation. This demonstrates that increasing the number of videos for each sign, that is simulating the generation of new videos, had an important impact on the results. Moreover, the results show that in Figure 7 it seems that there is no real difference in using all the coordinates for each keypoint (that are x,y,z) or only two (that are x,y). From experimentations, the best neural model seems to be CONVNET (configured with the specific parameters in Table 4, Appendix B). Indeed, we achieved 100% accuracy (Figure 8) and 100% precision, recall, F1-score on each class. However, we are awzre that these impressive results can be due to overfitting,

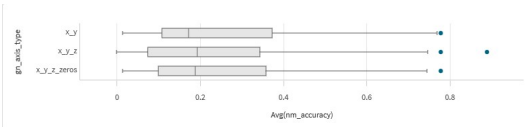


Figure 7: Box plot avg accuracy VS keypoint's coordinates

since the classification task is evaluated on a relatively small dataset of distinct signs (148 signs).

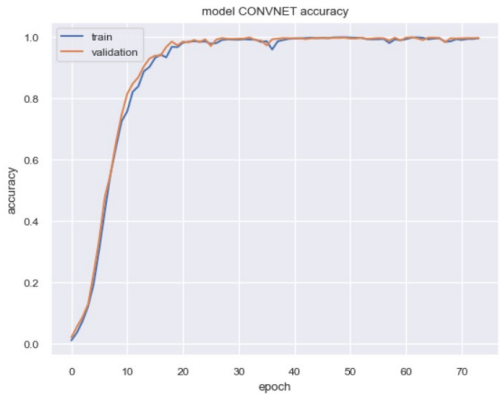


Figure 8: Best model accuracy for CONVNET

2.4.2. A Specific Test for Real Time computation

A small prototype system was developed to test the performances of the neural models implemented in real-time. A number of issues arose for this specific context. As we discussed, all the neural models were trained by applying PCA to the data, but this specific preprocessing step creates problems for a real time application. Indeed, for each prediction of the input, it is necessary to apply PCA to the data collected slowing the real-time performance. To solve this specific issue, we used a specific parallel thread, and by running the PCA on another parallel thread: in this way we have been able to reduce the impact of this problem. By using the OpenCV library we have developed a function that allowed us to read the video from the webcam frame by frame. So, for each frame, the detection of the mediapipe keypoints was performed and saved them in an array. The first prediction took place when we have collected this information for at least N frames, where N is the number of frames of the input shape of the model being tested. After that, each extraction would corresponded to a prediction which is still based on the last N frames present in the array of extracted keypoints. Predictions that exceeded a certain threshold of probability fixed α (in tests performed α = 0.7), are shown in a bar at the top of the window.

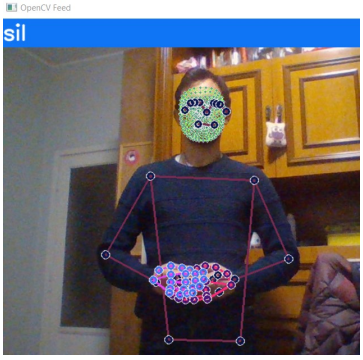


Figure 9: Example of real time detection

3. TGLIS-227: A new dataset for LIS

According to [1] the number of datasets for sign languages is proportional to the number of available datasets of the corresponding national vocal language. As noted in Section 2, the only dataset that can be used in a SLR task for LIS is A3LIS-147 [12], that is composed of 147 signs/videos, performed by 10 different signers. There are other linguistic resources for LIS, that are SpreadTheSign [14], Segni in movimento and SIGNHUB [15]. However, these datasets cannot be effectively used in SLR neural training because they contain only one video for each sign. Moreover, there are no public datasets for LIS with sentence-level granularity.

3.1. Towards a New Dataset

In order to build a new dataset, we considered the available sources of LIS videos. We decided to use the RAI newscasts[7] for three reasons: 1. the quality of LIS production, 2. the availability of many videos, 3. the continuous production of new videos (at least three daily editions). All newscasts have the same video format:

Figure 10: RAI LIS newscast example

[7] https://www.rai.it/dl/easyweb/LIS-e2a267d2-e9a0-4af7-b2ff-baad1f5d060e.html

1. On the left box there is the signer, on the right there is the speaker or, alternatively, the images related to the news.
2. The duration is comprised of between 2 and 5 minutes.
3. Each video contains 5 or 6 different news items.
4. Each news item is preceded and concluded by the sign of silence.
5. The speaker waits for the signer to finish before moving to the next news item.
6. Each news item is accompanied by a subtitle representing the topic.

So, exploiting these features, we developed a system to automatically segment each newscasts video. By identifying the parts of the newscast video containing the silence-LIS in the images and the silence in the audio, we produced a number of videos containing a single news item for each one. The working hypothesis is that the the silences correspond to the transition from one news to an another.

To perform silence-LIS detection we use YOLOv7 (an object detector). To train a YOLO model we needed many images that represented the object to detect, that is the silence-LIS. Since no silence datasets exist, we built it by extracting frames from RAI newscasts. So, we obtained a silence dataset that contained 8000 silence LIS images with resolution 640x640 and with two annotations: only hands or hands+elbows.

3.2. Pipeline construction

We downloaded 20 newscasts from Rai Play and we did a pre-processing step by removing the theme song and by cropping the video to focus on the LIS signer. Then we detected silence-LIS using the YOLO model trained on the silence dataset. Thereafter we detected silences audio for each video: first we extracted audio from video (using moviepy library functions) then we detected only silence that were at least 2 seconds long (using PyDub library function). All detections were recorded into tabular CSV format. Moreover, we built a filter algorithm that took in input detections and returned the ranges that corresponds to the transition from one news item to an other. By using this information, we splitted the newscast in the corresponding news by obtaining a new video file and an audio file for each news. Finally, we annotated each news item (in CSV) with two extra fields:

1. the topic: we applied an optical character recognition (pytesseract) to crop the title of the news items shown in the video
2. the transcript: we applied speech recognition (the Azure SDK speech to text)

These annotations are important in a translation context because the topic represents the context of translation and the transcript represents the target of translation.

3.3. Dataset Final Structure

The final dataset is called TGLIS-227 since it is composed of 227 distinct news items extracted from LIS newscasts editions. For each of them we have:

1. a video (mp4): containing the LIS news item.
2. an audio (wav): that is the audio of the news item in the Italian (vocal) language.
3. topic (in csv): containing the topic of the news item (Appendix E),
4. transcript (in csv): that is the automatic transcription of the news item(Appendix F).

Note that we built an automatic procedure that could be applied several times in order to increase the size of the dataset.

A crucial weakness of the actual dataset is the lack of a standard for LIS transcription in some written form. This linguistics issue requires the collaboration with Deaf organizations and could be performed by using annotation tools for videos such as ELAN [16].

Finally note that for copyright issues we cannot distribute the audio/video content of news items directly, but only the annotations (Appendixes C, D, E, F)[8]. However, by using the timestamps of each news item (Appendix C and Appendix D), and requesting access to the "Teche Rai", it is possible to extract the video and the audio from original newscasts[9].

3.4. Testing deep learning for ISLR on TGLIS-227 videos

We tried to test the best model described in section 2 on the TGLIS-227 dataset (section 3). Since we do not have an LIS annotation (e.g in glosses) for each video, we did a very raw evaluation of the correctness of the ISLR predictions by using the lemma corresponding to the Italian news transcript. In particular, we counted the number of matches between predictions and lemmas, obtaining around 33% of correct matches. This low value is consequence of the the different size of the training dataset, containing only 147 signs, with respect to the size of the TGLIS-227 dataset, containing around 5000 lemma.

[8] data are available at this GitHub: https://github.com/BeanRepo/TGLIS-227

[9] Note that the timestamps are calculated on the videos without start-end theme songs

4. Conclusion and future works

In this work we have presented two main results obtained in the LIS2SPEECH project. First, we have described the application of a number of data augmentation techniques to some deep neural models in the task of ISLR. We proved with experiments that some of these transformations have a strong impact on the final performance of the classification task. Second, we built the TGLIS-227, a new sentence-level dataset for LIS, applying a new procedure for the automatic segmentation of the newscasts.

In future work we intend to develop the following two ideas:

1. to annotate TGLIS-227 video with the glosses that they contain;
2. to develop a system like Common Voice [17] to collect more data to build an open source dataset for LIS;

Moreover, a more challenging development could be to encode additional two video features that are: the lips and facial expressions. Finally, we noted that very often in the news the signers "read" the gloss by using their lips and, moreover, express an emotion related with the gloss by using their facial expression.

References

[1] O. Koller, Quantitative survey of the state of the art in sign language recognition, CoRR abs/2008.09918 (2020). URL: https://arxiv.org/abs/2008.09918. arXiv:2008.09918.

[2] I. Adeyanju, O. Bello, M. Adegboye, Machine learning methods for sign language recognition: A critical review and analysis, Intelligent Systems with Applications 12 (2021) 200056. URL: https://www.sciencedirect.com/science/article/pii/S2667305321000454. doi:https://doi.org/10.1016/j.iswa.2021.200056.

[3] M. Bohacek, M. Hruz, Sign pose-based transformer for word-level sign language recognition, 2022 IEEE/CVF Winter Conference on Applications of Computer Vision Workshops (WACVW) (2022) 182–191.

[4] R. Rastgoo, K. Kiani, S. Escalera, Zs-slr: Zero-shot sign language recognition from rgb-d videos, ArXiv abs/2108.10059 (2021).

[5] J. Huang, W. gang 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 (2019) 2822–2832.

[6] G. A. Rao, K. Syamala, P. V. V. Kishore, A. S. C. Sastry, Deep convolutional neural networks for

sign language recognition, 2018 Conference on Signal Processing And Communication Engineering Systems (SPACES) (2018) 194–197.

[7] J. Huang, W. gang Zhou, H. Li, W. Li, Sign language recognition using 3d convolutional neural networks, 2015 IEEE International Conference on Multimedia and Expo (ICME) (2015) 1–6.

[8] R. Kumar, A. Bajpai, A. Sinha, Mediapipe and cnns for real-time asl gesture recognition, 2023. arXiv:2305.05296.

[9] M. Borg, K. P. Camilleri, Sign language detection "in the wild" with recurrent neural networks, in: ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2019, pp. 1637–1641. doi:10.1109/ICASSP.2019.8683257.

[10] G. Samaan, A. Wadie, A. Attia, A. Asaad, A. Kamel, S. Slim, M. Abdallah, Y.-I. Cho, Mediapipe's landmarks with rnn for dynamic sign language recognition, Electronics 11 (2022) 3228. doi:10.3390/electronics11193228.

[11] D. Kothadiya, C. Bhatt, K. Sapariya, K. Patel, A.-B. Gil-González, J. M. Corchado, Deepsign: Sign language detection and recognition using deep learning, Electronics 11 (2022). URL: https://www.mdpi.com/2079-9292/11/11/1780. doi:10.3390/electronics11111780.

[12] M. Fagiani, S. Squartini, E. Principi, F. Piazza, A new italian sign language database, 2012. doi:10.1007/978-3-642-31561-9_18.

[13] V. Grishchenko, V. Bazarevsky, R. Engineers, G. Research, Mediapipe holistic — simultaneous face, hand and pose prediction, on device, 2020. URL: https://ai.googleblog.com/2020/12/mediapipe-holistic-simultaneous-face.html.

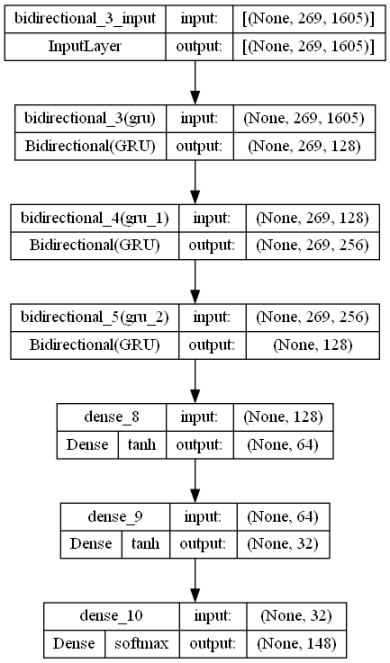
[14] A. Cardinaletti, Il progetto spread the sign, BLITYRI (2016). doi:https://hdl.handle.net/10278/3691616.

[15] Sign-Hub, Sign-hub: Wp 2.4, 2020. URL: https://hdl.handle.net/11403/sign-hub-wp-24/v1, ORTOLANG (Open Resources and TOols for LANGuage) –www.ortolang.fr.

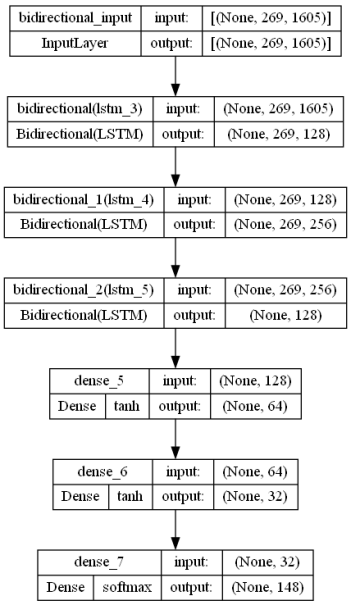
[16] T. L. A. R. f. h. Nijmegen: Max Planck Institute for Psycholinguistics, Elan[computer software], 2023. URL: https://archive.mpi.nl/tla/elan.

[17] R. Ardila, M. Branson, K. Davis, M. Henretty, M. Kohler, J. Meyer, R. Morais, L. Saunders, F. M. Tyers, G. Weber, Common voice: A massively-multilingual speech corpus, in: Proceedings of the 12th Conference on Language Resources and Evaluation (LREC 2020), 2020, pp. 4211–4215.

Appendix A. Neural Networks Architectures

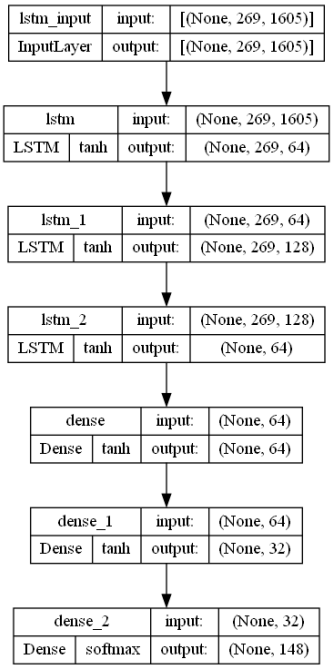


(a) BIGRU

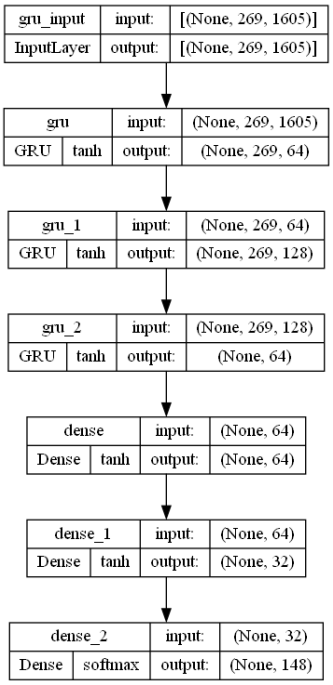


(b) BILSTM

Figure 11: Biderectional Recurrent Architectures



(a) LSTM



(b) GRU

Figure 12: Recurrent architectures

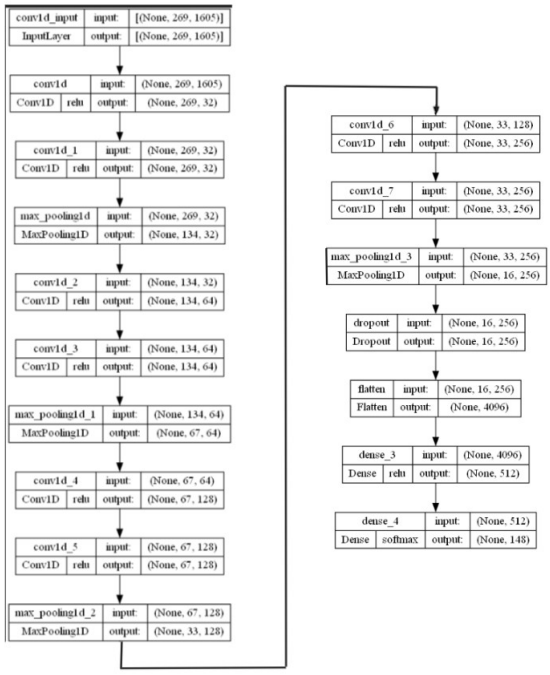


Figure 13: CONVNET Architecture

Appendix B. Test documentation

Table 1
Test documentation: all parameters

info_salvate	type
cd_test_code	integer
cd_train_log_name	string
cd_test_log_name	string
dt_testing_timestamp	datetime
nm_signs	integer
nm_avg_sign_videos	integer
nm_kps	integer
nm_kps_face	integer
nm_kps_pose	integer
nm_kps_lh	integer
nm_kps_rh	integer
gn_axis_type	string
fl_is_normalized	boolean
fl_is_face_red	boolean
gn_face_red_type	string
fl_is_data_aug	boolean
fl_rotated_data	boolean
fl_flipped_data	boolean
fl_smoothed_data	boolean
fl_translated_data	boolean
fl_is_pca	boolean
gn_model	string
gn_model_layers	string
nm_model_parameters	integer
nm_epochs	integer
nm_batch_size	integer
fl_is_early_stop	boolean
nm_patience	integer
gn_test_path	string
gn_optimizer_type	string
gn_accuracy_type	string
gn_loss_type	string
nm_accuracy	float
nm_loss	float

Table 2
General group of tests: parameters

parameters	values
nm_signs	[3,16,50,148]
nm_avg_sign_videos	[10]
gn_axis_type	['x_y','x_y_z', 'x_y_z_zeros']
fl_is_face_red	[True, False]
fl_is_normalized	[True, False]
fl_is_data_aug	[True, False]
fl_is_pca	[True]
gn_model	['GRU','LSTM', 'BIGRU','BILSTM', 'CONVNET']

Table 3
Specific data augmentation group of tests: parameters

parameters	values
nm_signs	[3,16,50,148]
nm_avg_sign_videos	[10]
gn_axis_type	['x_y']
fl_is_face_red	[True]
fl_is_data_aug	[True, False]
fl_is_pca	[True]
fl_is_normalized	[True]
fl_rotated_data	[True, False]
fl_flipped_data	[True, False]
fl_smoothed_data	[True, False]
fl_translated_data	[True, False]
gn_model	['GRU','LSTM', 'BIGRU','BILSTM', 'CONVNET']

Table 4
Best model parameters for the best neural model, i.e. CONVNET

parametro	valore
gn_axis_type	['x_y']
fl_is_face_red	[True]
fl_is_normalized	[True]
fl_is_data_aug	[True]
fl_is_pca	[True]

Appendix C. Video Timestamps[10] This appendix is a sample of video timestamps file. The full version is available on github: https://github.com/BeanRepo/TGLIS-227

Table 5
video range for each newscast

video	range_notizia
01_25_2022.mp4	"['00:00', '00:41'], ['00:44', '01:13'], ['01:15', '01:40'], ['01:44', '02:06'], ['02:09', '02:26'], ['02:28', '02:37']"
01_26_2022.mp4	"['00:00', '00:34'], ['00:39', '01:00'], ['01:06', '01:36'], ['01:40', '02:09'], ['02:14', '02:36'], ['02:38', '02:47']"
01_27_2022.mp4	"['00:00', '00:35'], ['00:39', '01:16'], ['01:20', '01:55'], ['01:59', '02:31'], ['02:36', '03:14'], ['03:16', '03:24']"
01_28_2022.mp4	"['00:00', '00:45'], ['00:48', '01:09'], ['01:13', '01:40'], ['01:44', '02:03'], ['02:05', '02:22'], ['02:24', '02:32']"
02_16_2022.mp4	"['00:00', '00:45'], ['00:50', '01:22'], ['01:27', '01:59'], ['02:04', '02:39'], ['02:42', '03:04'], ['03:05', '03:15']"
02_17_2022.mp4	"['00:00', '00:41'], ['00:45', '01:36'], ['01:40', '02:15'], ['02:21', '02:48'], ['02:52', '03:09'], ['03:12', '03:21']"
...	...

Appendix D. Audio Timestamps[11] This appendix is a sample of audio timestamps file. The full version is available on github: https://github.com/BeanRepo/TGLIS-227

Table 6
audio range for each newscast

audio	range_notizia
01_25_2022.wav	"['00:00', '00:38'], ['00:41', '01:09'], ['01:13', '01:35'], ['01:40', '02:03'], ['02:07', '02:22'], ['02:27', '02:37']"
01_26_2022.wav	"['00:00', '00:29'], ['00:36', '00:55'], ['01:03', '01:29'], ['01:38', '02:06'], ['02:13', '02:32'], ['02:37', '02:47']"
01_27_2022.wav	"['00:00', '00:30'], ['00:37', '01:14'], ['01:18', '01:51'], ['01:57', '02:28'], ['02:34', '03:10'], ['03:14', '03:24']"
01_28_2022.wav	"['00:00', '00:41'], ['00:47', '01:04'], ['01:11', '01:36'], ['01:42', '01:58'], ['02:04', '02:18'], ['02:22', '02:32']"
02_16_2022.wav	"['00:00', '00:43'], ['00:48', '01:18'], ['01:25', '01:55'], ['02:02', '02:36'], ['02:41', '03:00'], ['03:03', '03:15']"
02_17_2022.wav	"['00:00', '00:37'], ['00:43', '01:32'], ['01:39', '02:13'], ['02:18', '02:45'], ['02:51', '03:07'], ['03:11', '03:21']"
...	...

[10] All data in this appendix is protected by Creative Commons Licence CC BY-NC-SA 4.0.

[11] All data in this appendix is protected by Creative Commons Licence CC BY-NC-SA 4.0.

Appendix E. Topic News[12]

This appendix is a sample of topic file. The full version is available on github: https://github.com/BeanRepo/TGLIS-227

Table 7
topic extracted for each news

01_25_2022_chunk_1	I RISULTATI DELLE ELEZIONI
01_25_2022_chunk_2	USA: ITALIA PARTNER IMPORTANTE
01_25_2022_chunk_3	FUGA DALLA RUSSIA PER NON ARRUOLARSI
01_25_2022_chunk_4	IL PRIMO ESPERIMENTO DI DIFESA PLANETARIA
01_25_2022_chunk_5	L'ITALIA BATTE L'UNGHERIA ED È NELLE FINAL FOUR
01_25_2022_chunk_6	
01_26_2022_chunk_1	ELEZIONE PRESIDENTE, IERI FUMATA NERA
01_26_2022_chunk_2	NUOVO IMPULSO AL CONFRONTO TRA I PARTITI
01_26_2022_chunk_3	CRISI UCRAINA, ALTA TENSIONE
01_26_2022_chunk_4	RALLENTA LA CURVA DELL'EPIDEMIA
01_26_2022_chunk_5	OGGI BERRETTINI GIOCA I QUARTI DI FINALE
01_26_2022_chunk_6	
01_27_2022_chunk_1	L'ELEZIONE DEL PRESIDENTE, CONTATTI TRA I PARTITI
01_27_2022_chunk_2	COVID, ALLO STUDIO ESTENSIONE GREEN PASS
01_27_2022_chunk_3	CRISI UCRAINA, DIPLOMAZIA AL LAVORO
01_27_2022_chunk_4	STRAGE DI LICATA, UN PAESE IN LUTTO
01_27_2022_chunk_5	GIORNO DELLA MEMORIA, PAPA : MAI PIÙ QUESTI ORRORI
01_27_2022_chunk_6	
01_28_2022_chunk_1	QUIRINALE, ALLE 11 COMINCIA LA QUINTA VOTAZIONE
01_28_2022_chunk_2	UCRAINA, TELEFONATA BIDEN-ZELENSKY
01_28_2022_chunk_3	OK DELL'EMA ALLA PILLOLA ANTI-COVID
01_28_2022_chunk_4	
01_28_2022_chunk_5	TENNIS, SEMIFINALE BERRETTINI-NADAL
01_28_2022_chunk_6	
...	...

[12]All data in this appendix is protected by Creative Commons Licence CC BY-NC-SA 4.0.

Appendix F. Transcript News[13] This appendix is a sample of transcript file. The full version is available on github: https://github.com/BeanRepo/TGLIS-227

Table 8
transctipt extracted for each news

transcript	
01_27_2022_chunk_6.wav	Ed è tutto grazie per averci seguito. Il tg uno torna alle 8, buona giornata.
01_28_2022_chunk_1.wav	Un giorno dal tg uno la corsa al Quirinale comincerà alle 11, il quinto giorno di votazioni. Il centrodestra sarebbe orientato a votare uno dei nomi proposti nei giorni scorsi. Contrario a questa scelta il centrosinistra, che per protesta potrebbe uscire dall'Aula al momento del voto.
01_28_2022_chunk_1.wav	Intanto il presidente della Camera Roberto Fico.
01_28_2022_chunk_1.wav	Ha convocato alle 10:15 la Conferenza congiunta dei capigruppo di Camera e Senato per decidere se procedere a una doppia votazione giornaliera.
01_28_2022_chunk_2.wav	Cresce la tensione tra Stati Uniti e Russia sulla questione Ucraina telefonata tra Zelensky e Biden.
01_28_2022_chunk_2.wav	Per il Presidente americano c'è la possibilità concreta che i russi invadano l'Ucraina nel mese di Febbraio.
01_28_2022_chunk_3.wav	La situazione Covid in Italia rallenta la curva dei contagi, calano recovery e terapie intensive e si discute della possibilità di cambiare il sistema delle fasce a colori delle regioni e anche le regole che riguardano la scuola.
01_28_2022_chunk_3.wav	Intanto è arrivato l'OK dell'EMA alla pillola anti COVID di freezer.
01_28_2022_chunk_4.wav	Tamponi sospetti e Green pass fasulli chiuso un centro analisi in provincia di Trento e Stop a un secondo punto prelievi nel capoluogo Trentino.
01_28_2022_chunk_4.wav	5 le persone indagate.
01_28_2022_chunk_5.wav	Il tennis nella semifinale degli Australian Open, in campo Matteo Berrettini e Rafa Nadal. Il punteggio al momento è di due set a uno per lo spagnolo.
01_28_2022_chunk_6.wav	Ed è tutto grazie per averci seguito. Il tg uno torna alle 8, buona giornata.
...	...

[13] All data in this appendix is protected by Creative Commons Licence CC BY-NC-SA 4.0.