Local-global Data Augmentation for Contrastive Learning in Static Sign Language Recognition

Basso Madjoukeng, Ariel; Kenmogne, Edith Belise; Poitier, Pierre; Frénay, Benoît; Fink, Jerome

Published in:
Advances in Intelligent Data Analysis XXIII - 23rd International Symposium on Intelligent Data Analysis, IDA 2025, Proceedings

DOI:
10.1007/978-3-031-91398-3_5

Publication date:
2025

Local-global Data Augmentation for Contrastive Learning in Static Sign Language Recognition

Ariel Basso Madjoukeng[1], Edith Bélise Kenmogne[2], Pierre Poitier[1⋆], Benoît Frénay[1⋆⋆], and Jérôme Fink[1]

[1] University of Namur - NADI - PReCISE - HuMaLearn, Namur - Belgium
[2] University of Dschang - FS, Dschang - Cameroon

Abstract. Sign language (SL) is a visual language used by the Deaf community. Static sign language recognition (SLR) consists of classifying static hand configurations, i.e., signs, present in isolated images. Due to the expertise required for manual annotation, SLR suffers from a data scarcity issue. Recent studies show that contrastive learning is an effective method for addressing this issue by proposing an efficient unsupervised pre-training. Contrastive learning leverages data augmentation techniques applied to entire images (global-global augmentation). However, fine-tuned, contrastive models often rely on irrelevant aspects of those images, like the background, without focusing solely on the regions of interest. Such models are prone to bias that could lead to unreliable predictions. In response, this paper proposes a new local-global data augmentation technique that helps contrastive models focus during the fine-tuning step on regions of interest, i.e., the signer's hands. This approach (i) improves the accuracy of contrastive learning by up to 15% on some SLR datasets, and (ii) help the fine-tuned contrastive models to better focus on relevant regions of images for SLR.

Keywords: Contrastive Learning · Data Augmentation · Sign Language Recognition · Local-Global Augmentation.

1 Introduction

Sign language recognition (SLR) studies the translation of sign language into written or oral languages. The lack of people able to understand and translate sign languages (there is no universal sign language, but many of them) slows down data collection. To overcome this shortage in annotation, a solution is to leverage unsupervised methods, enabling models to learn useful features for SLR.

Contrastive learning is a popular and versatile, unsupervised approach that relies on augmentations of images to learn useful representations that are invariant to those augmentations. The resulting model can be fine-tuned on a small proportion of available annotated data of the same nature. The fine-tuned model

⋆ Ph.D. grant from FRIA (F.R.S.-FNRS)
⋆⋆ Supported by SPWR under grant n°2010235 - ARIAC by DIGITALWALLONIA4.AI.

performances are closely tied to the representations learned during the unsupervised training. In many application scenarios, particularly when the size of the unsupervised training dataset is not substantial, instead of learning representations useful for classification, the model sometimes learns representations based on all the features of the image. During the classification phase, predictions could be based on the background or other irrelevant parts of the image.

The above problem is particularly acute in static SLR where the region of interest (RoI), the signer's hands, contains most of the information needed to classify the sign. The background should play no role in the prediction. For SLR, a model whose predictions often rely on aspects of the image other than the RoI may be both unreliable and irrelevant from a linguistic point of view. To overcome those limitations, this paper introduces a novel augmentation method for contrastive pre-training called *local-global* augmentation. It encourages the model to focus on the RoI in the images. This method is applied to static SLR for datasets of single-hand configurations without dynamic components.

Experiments show that, on small datasets, our method helps the model to primarily focus on the RoI and improves accuracy by up to 15%. For large datasets, it concentrates saliency on the RoI without significantly loss of performance.

The rest of this paper is organized as follows. Section 2 introduces SimCLR and MoCo v2, two popular contrastive learning algorithms used here to assess the proposed augmentation strategy. A review of static SLR is also presented, and the datasets used in our experiments are presented. Section 3 shows issues that occur when using a *global-global* augmentation strategy. Our *local-global* augmentation strategy developed to address those issues is introduced in Section 4. Section 5 describes the architecture for the experiments and the meta-parameters for the training. Section 6 shows quantitative and qualitative results to assess our new method. Finally, those results are discussed in Section 7.

2 Preliminaries and Related work

This section introduces SimCLR and MoCo v2, two popular contrastive learning algorithms used in our experiments. It also provides an overview of related work in SLR and discusses what augmentation methods can be used for SLR. Finally, the diverse SLR datasets used in our experiments are presented.

2.1 Contrastive Learning

Contrastive learning enables deep learning architectures to learn self-supervised representations through various data augmentations, without the need for prior annotations. There exist several contrastive algorithms, among which the most popular are SimCLR [1] and MoCo v2 [2]. This section briefly presents them.

SimCLR Algorithm : SimCLR is one of the first successful contrastive algorithms and is still widely used in the literature [3–5]. This algorithm consists of four components. First, an *augmentation module* transforms an input image into

a positive pairs of augmented images, i.e., augmentations. The other instances in the same batch act as negative examples. Second, the *encoder network* projects the input data in a latent space. Finally, the *projection head*, commonly a multilayer perceptron, transforms the latent space into lower-dimensional vectors called the projections. The contrastive loss function is applied to the projections. The loss function is used to maximize the similarity between positive pairs and minimize the similarity between negative pairs. Once trained, the projection head is removed and only the latent space is used for inference. The obtained embeddings can be grouped with any clustering algorithm [6].

MoCo v2 Algorithm : SimCLR requires large batches to have enough positive and negative examples to converge. It is not always possible to increase the batch size when training a model due to memory limitations. To tackle this issue, MoCo v2 [2] introduces the concept of *dictionary*: a queue containing embeddings of the data from previous batches. Those embeddings serve as negative examples for the current batch. At each step, MoCo v2 receives two inputs: a query and keys. A query is an augmented image from the dataset, while keys are a set of samples stored in a dictionary. MoCo v2 uses two encoders: a query encoder and a key encoder (commonly called *momentum encoder*). During the training, the query encoder parameters are updated during the backpropagation, while the momentum encoder parameters are updated using a momentum update rule.

2.2 Sign Language Recognition

Sign language recognition (SLR) studies the creation of systems to recognize, classify, or translate sign language. This work focuses on static SLR that aims to recognize static signs in images. Since the advent of deep learning, significant progress has been made in this field [7]. Given the visual nature of the data, convolutional neural networks (CNNs) are often used. Murali et al. [8] introduce a three-layer CNN architecture, trained on the American Sign Language (ASL) dataset [9] with 87k images of the ASL alphabet. The results show that their model, trained from scratch, achieves a remarkable accuracy of 90%. Mohsin et al. [10] benchmark several CNN architecture for the American Sign language recognition and achieved 95% of accuracy with a VGG16 model.

However, for many medium datasets such as the *Indian Sign Language* [11] or the *Arabic Sign Language* [12] datasets, training from scratch does not lead to satisfying results. To address this issue, researchers have proposed to rely on knowledge transfer. Al-Barham et al. [13] show that transfer learning with a pre-trained ResNet-15 achieves 96.77% of accuracy on the ArASL dataset with 7.8k images. Despite these advances, the scarcity of annotated data and the existence of underrepresented sign languages (i.e., small datasets) hinders the usage of transfer learning. For underrepresented datasets like *Bahasa Isyarat Indonesia* [14], transfer learning alone often obtains poor accuracy.

Zhao et al. [15] conducted a study on self-supervised learning methods and concluded that contrastive learning is more efficient than transfer learning in

more general tasks such as hands motion estimation. Accordingly, Kothadiya et al. [16] propose a contrastive architecture for SLR, applied to the *American Sign language* (ASL) and *Indian Sign Language* (ISL) datasets. They apply augmentations such as color distortion, Gaussian blur, translation, and flipping globally (i.e., applied to the whole image) to form positive contrastive pairs. These transformed images are then used to train contrastive algorithms such as SimSiam [17], SimCLR, MoCo, and others. With SimCLR, they achieve 74.08% of accuracy on ISL and 74.66% on ASL; SimSiam reaches 74.59% on ISL and 77.41% on ASL, while MoCo achieves 70.21% on ISL and 68.03% on ASL.

To the best of our knowledge, in contrastive learning algorithms, data augmentation is only applied globally to the entire image [1, 2, 17, 11], i.e., all of its regions are affected. However, during training, this poses several issues that are further exacerbated when the unsupervised training set is not sufficiently large. Section 3 provides a detailed discussion on the issues associated with the global application of augmentations in contrastive learning for SLR.

2.3 Image Augmentation for SLR

Data augmentation is a critical step in contrastive learning, it should be designed with care as it plays a critical role in the performance of the final model. The SimCLR paper [1] categorizes image augmentations in two main groups, *geometric and appearance transformations*. Geometric transformations are modifications that affect the geometric properties of an image, such as its shape, size, orientation, etc. Such transformations include rotations, translations, scaling, vertical/horizontal flips, and deformations. Appearance transformations focus on the texture without affecting the geometric nature of the image. Color distortion and Gaussian blur are two common appearance augmentations.

In SLR, rotations and flips could be damageable, as the same hand configuration with a different orientation could have a different meaning. These transformations should thus be avoided for contrastive learning in SLR. Moreover, for our method that only transform parts of the image (see Section 4.1 for details), only the appearance transformations will be used, as the geometric transformations are hard to apply to the local parts or objects in the image.

2.4 SLR Datasets

Several SLR datasets are used to illustrate issues of current augmentation techniques and to assess our approach: (i) *Bahasa Isyarat Indonesia* (BISINDO) [14] with 312 images across 26 classes representing the alphabet, (ii) *Arabic Alphabets Sign Language* (ArASL) [12] with 7,8k images from 31 unique letter signs, (iii) *the Indian Sign Language* (ISL) [11] with 42k images from 35 static signs from the Indian SL and (iv) the *American Sign Language* (ASL) [9] dataset with 87k images from 26 gestures signs from the American manual alphabet. These datasets are quite diverse and representative of the challenges in static SLR.

3 Issues of Global-global Augmentation For Contrastive learning

Contrastive approaches rely on data augmentation to generate positive pairs for their training process. The positive pairs for an image are obtained by applying two distinct augmentations to all the pixels of the image. Since both augmentations are applied to the entire image, we refer to this approach as *global-global* augmentation. As mentioned in Section 2, after generating the positive pairs, learning aims to capture invariance between the augmented versions of the same image. However, it often happens that instead of learning invariance related to the RoI (in our case, the hands of the signers), the model captures invariance in regions of the image that are irrelevant for the downstream task (e.g., clothes). There is no reason for the model to focus only on the RoI during the unsupervised pre-training. This issue is even more acute on small datasets.

To assess the existence and seriousness of this issue, a model is trained using a contrastive loss on all the static SLR datasets introduced in Section 2.4. The details of the training and the architecture are available in Section 5.

To visualize what regions of the image are used by the model to classify a sign, saliency maps [18] are computed. The saliency map of an image highlights the pixels that are the most prominent for the model to take its decision. Therefore, by looking at those maps, it is possible to observe what parts of the image are the most important for the model. Figure 1 shows, for each SLR dataset, two randomly selected images along with their label and a saliency map.

In Figure 1, the following cases can be observed. Sometimes, the RoI aligns with the saliency map. This happens when the contrastive model captures relevant similarities in the input. It is common for larger datasets. In other cases, the saliency spreads over the whole image. This occurs when the model fails to capture the relevant invariance between the different augmented views of the same image. Typically, it happens with smaller datasets such as BISINDO and ArASL in our experiments, and sometimes with medium-sized datasets.

In response to observations from our preliminary experiment in Figure 1, this work proposes a new method to mitigate issues for contrasting pre-training in SLR. This new *local-global* augmentation is presented in the next section.

4 Local-global Data Augmentation

This section introduces the proposed *local-global* augmentation for contrastive training. The intuition and advantages of the method are presented, and we empirically motivate why only applying the augmentation to the RoI for all positive images (i.e., relying on a local-local augmentation) is not desirable.

4.1 Main Idea and Method

The proposed *local-global* augmentation module for contrastive learning is graphically described in Figure 2. An augmentation is selected and applied to the input image to create a positive pair of images. The key difference with a classical,

(a) Letter A (b) Saliency map (c) Letter H (d) Saliency map

(e) Letter A (f) Saliency map (g) Letter H. (h) Saliency map

(i) Teh Marbuta (j) Saliency map (k) Meem (l) Saliency map

(m) Letter B (n) Saliency map (o) Space symbol (p) Saliency map

Fig. 1: Images and saliency maps from BISINDO [14] (first line), ISL [11] (second line), ArASL [19] (third line) and ASL [9] (fourth line) datasets. Each image is processed by a model pre-trained using a contrastive loss (see text for details).

global-global augmentation module is that both images in the positive pairs are augmented differently in our local-global module. For the first image of the pair, the augmentation is only applied on the RoI. For the second one, the augmentation is applied on the whole image. Figure 3 shows a comparison of a *global-global* and a *local-global* augmentation applied to sign language data.

As already stated in Section 3, contrastive algorithms learn invariance between different augmented versions of the same instance. The *local-global* augmentation makes it easier for the model to find similar patterns in the relevant parts of images. On the one hand, maximizing the similarity of the projections for the positive pairs encourages the model to only use the RoI. Indeed, for both

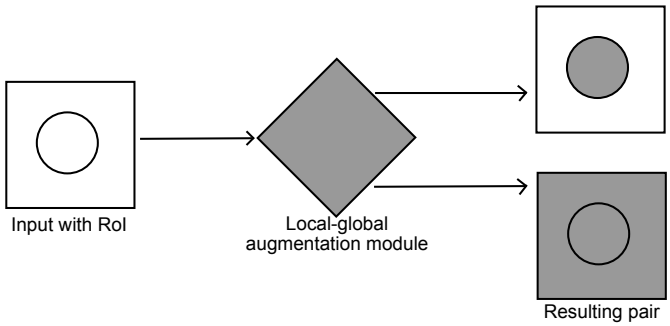


Fig. 2: Overview of the local-global augmentation process. The input with an identified RoI is passed to the augmentation module. It randomly selects a transformation and applies it once to the whole image and once to the RoI. The resulting pair is sent to the contrastive algorithm, here SimCLR or MoCo v2.

Fig. 3: *global-global* (left) and *local-global* augmentation (right), illustrated on an image from the ArASL dataset [19]. Local augmentation is applied to the RoI.

images in the positive pairs, *pixels inside the RoI are similar in both positive images as they are altered by the same augmentation*, whereas *pixels outside the RoI are altered in only one of the positive images*. If projections are based only on the RoI, it will be easier to maximize similarity. On the other hand, the negative pairs play the same role as in classical contrastive learning: forcing the model to learn relevant projections to distinguish images. Local-global augmentation makes it easier for the algorithm to find invariance and leads to better results on smaller dataset, as experimentally illustrated in Section 6.

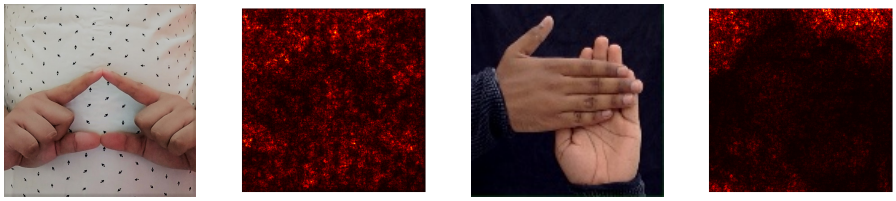
A limitation of the *local-global* is the need of RoI annotation to apply the augmentation. For our experiments on static SLR, the RoI is automatically extracted using a network pre-trained for image segmentation called Mask R-CNN [20]. This proved to be effective as we only need to detect hands on a background, and no manual annotation is needed. The *local-global* method also prevents the ap-

plication of some augmentations impossible to apply locally (e.g., rotation). This explains why our experiments only consider appearance-based augmentation.

4.2 About Local-local Augmentation

One might wonder why the augmentation is not applied solely on the RoI, leading to a *local-local* augmentation module for contrastive learning. However, this produces the opposite of the desired result: instead of focusing on the RoI, the model seems to divert its focus towards all the other regions of the image, excluding the RoI. This phenomenon suggests that the model, instead of capturing the relevant features of the RoI, ends up capturing the other features of the image. Intuitively, maximizing the similarity of images in the positive pairs no longer encourages contrastive learning to focus on the RoI, *since the background is now what is identical between both images*. Figure 4 presents examples of saliency maps obtained with a contrastive model trained with local-local augmentations. It is clear that the model saliency focuses on everything but the RoI.

(a) Letter B (b) Saliency map (c) Dad (d) Saliency map



(e) Letter B (f) Saliency map (g) Letter H (h) Saliency map

Fig. 4: Saliency map for images from the ASL [9] (first), ArASL [19] (second), BISINDO (third) and ISL (fourth) datasets with *local-local* augmentation.

5 Architecture and Training Parameters

Let us now introduce the architecture and the meta-parameter used in our experiments. Experiments in Section 6 are implemented in PyTorch version 2.4. The model is composed of an encoder and a projection head selected as follows:

– **Encoder**: the encoder is a ResNet-50 pre-trained on ImageNet. This choice is supported by its widespread use [21, 22] in transfer learning tasks.

– **Projection head**: the projection head consists of a multilayer perceptron (MLP) with a single hidden layer with an input size of 1000 as it is the size of the ResNet50 embedding and a projection size of 200 followed by a ReLU.

Models are trained using an SGD optimizer with a learning rate of 10^{-3} during 70 epochs and with a batch size of 128 images. The temperature of the contrastive loss is set to 0.1 and, in the case of MoCo v2, the momentum is set to 0.9. Those values match the recommended values reported in the SimCLR and MoCo v2 papers. The contrastive training is unsupervised and does not use the labels provided by the datasets. After the contrastive pre-training, the encoder is fine-tuned with a cross-entropy loss during 70 epochs using the same optimizer, batch size and number of epochs as the contrastive pre-training.

To pre-train, fine-tune and evaluate the contrastive model, three data splits are needed. For the evaluation, the test set is made of 30% of the data. The remaining 70% are used to pre-train and fine-tune the model: 80% of the training set are for the unsupervised training and 20% are for the supervised fine-tuning.

A pre-trained Mask R-CNN [20] with a confidence score value of 0.5 to extracts the RoI: all regions detected by the R-CNN are included in the RoI for the *local-global* augmentation module, whose implementation is available[3].

6 Experimental Results

Tables 1 and 2 show the average accuracies for 10 models trained using SimCLR and MoCo v2 for each dataset. One can observe that the accuracy of the model pre-trained with *local-global* augmentation is larger or equal in all the experiments. It yields a 15% accuracy improvement in performance for the small BISINDO dataset. This suggests that our approach is particularly effective compared to the traditional *global-global* augmentation when data are scarce. This was confirmed for the ASL dataset in an additional experiment (not shown here due to space limitations), where local-global augmentation outperformed global-global augmentation when only 400 instances are used for the supervised part.

A qualitative analysis for the *local-global* method is shown in Figure 5. The saliency is more concentrated on the RoI, regardless of the dataset size. This proves that the model learns to better focus on the relevant object in the image to learn good representations, whether on a small, medium or large dataset. To complement this qualitative analysis, we computed the average proportion of saliency within the RoI with local-global and global-global augmentation, and the results were respectively: 70% vs. 30% for BISINDO, 63.2% vs. 51.46% for ISL, 83.21% vs. 77.69% for ArASL and 73.91% vs. 72.29% for ASL. This confirms the relocation of saliency.

The limitations of this approach are that it is harder and, sometimes, impossible to apply some augmentation locally. In the case of images, only the appearance-based transformation can be applied locally. This limits the amount of available transformations. It also requires the identification of the RoI in the images, which might not be possible for every use cases.

[3] https://github.com/arielbassodev/local-global-data-Augmentation

Table 1: Comparison of the accuracies obtained using a *global-global* or *local-global* augmentation module with SimCLR for several static SLR datasets. Accuracy is averaged on 10 repetitions and 95% confidence intervals are given.

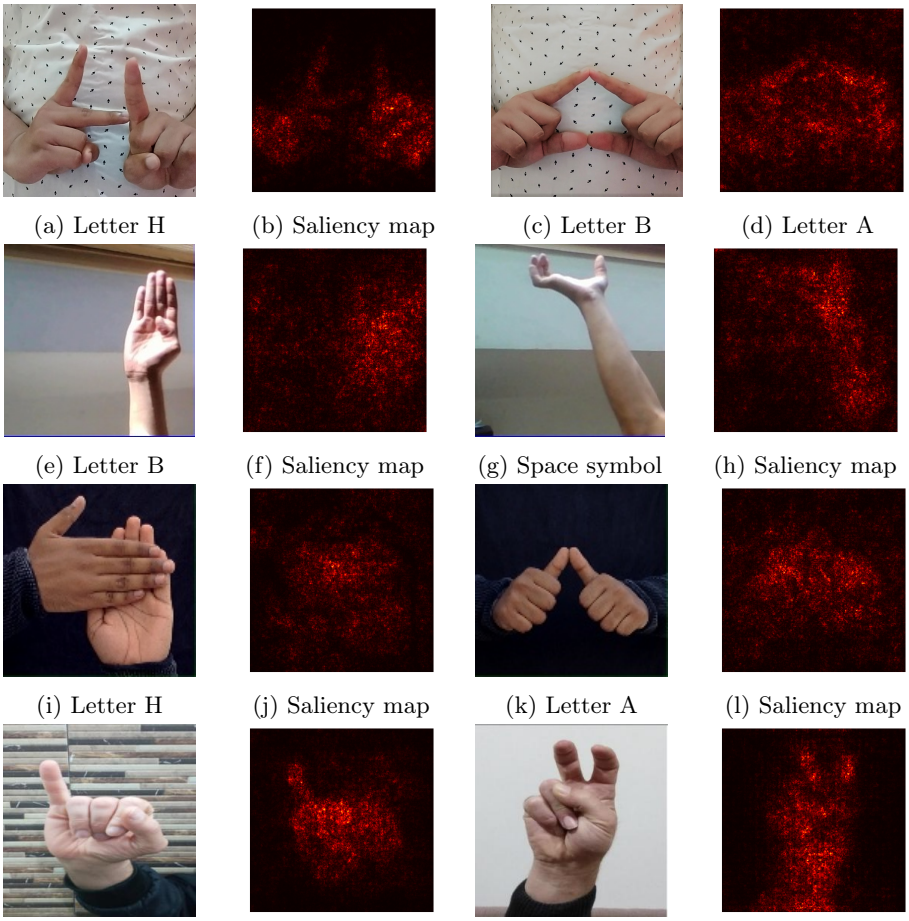
Dataset	Augmentation	global-global	local-global
BISINDO	Color Distortion (C_d)	9.72%± 1.27	22.05%±1.70
	Gaussian Blur (G_b)	7.95%±1.43	18.65%±2.45
	G_b and C_d	9.77%±2.06	27.92%±2.60
ISL	Color Distortion (C_d)	54.92%±1.89	62.01%±1.51
	Gaussian Blur (G_b)	52.95%±2.60	54.20%±1.27
	G_b and C_d	57.13%±1.89	65.51%±1.41
ArASL	Color Distortion (C_d)	40.29%±1.04	43.07%±0.75
	Gaussian Bur (G_b)	34.97%±0.69	37.02%±1.03
	G_b and C_d	44.29%±0.86	49.01%±1.17
ASL	Color Distortion (C_d)	67.32%±0.75	67.36%±0.95
	Gaussian Blur (G_b)	64.72%±0.56	64.72%±0.65
	G_b and C_d	69.45%±0.80	70.08%±0.96

Table 2: Comparison of the accuracies obtained using a *global-global* or *local-global* augmentation module with MoCo v2 for several static SLR datasets. Accuracy is averaged on 10 repetitions and 95% confidence intervals are given.

Dataset	Augmentation	global-global	local-global
BISINDO	Color Distortion (C_d)	8.08%±1.36	14.87%±1.51
	Gaussian Blur (G_b)	7.65%±1.09	13.07%±2.15
	G_b and C_d	11.52%±1.51	18.60%±2.30
ISL	Color Distortion (C_d)	57.02%±1.51	59.25%±0.58
	Gaussian Blur (G_b)	54.35%±0.88	55.09%±0.73
	G_b and C_d	59.07%±0.58	62.59%±0.66
ArASL	Color Distortion (C_d)	39.53%±3.02	39.01%±2.89
	Gaussian Blur (G_b)	38.07%±0.27	39.49%±0.59
	G_b and C_d	38.11%±0.43	39.6%±1.04
ASL	Color Distortion (C_d)	64.72%±0.63	64.75%±0.94
	Gaussian Blur (G_b)	61.63%±0.51	62.09%±0.48
	G_b and C_d	65.78%±0.70	66.50%±0.80

7 Conclusion

This work presents a new data augmentation module for contrastive pre-training that forces the model to focus more on regions of interest than on the other parts of the images. The approach is evaluated on the use case of static sign language recognition with four datasets of different sizes. First, a qualitative analysis of a model trained with a classical *global-global* augmentation module shows the pitfalls of this approach. Second, experiments conducted with the proposed local-global augmentation strategy show qualitative improvement. The observation of saliency maps clearly shows that models trained with our *local-global* augmentation module mainly focus on the regions of interest. Those results



(a) Letter H (b) Saliency map (c) Letter B (d) Letter A

(e) Letter B (f) Saliency map (g) Space symbol (h) Saliency map

(i) Letter H (j) Saliency map (k) Letter A (l) Saliency map

(m) Meem (n) Saliency map (o) Teh Marbuta (p) Saliency map

Fig. 5: Saliency map for the BISINDO (1st row), ASL (2nd row) and ISL (3rd row), and ArASL (4th row) dataset images with local-global augmentation.

are quantitatively confirmed by the results obtained by the contrastive model. An improvement of 15% accuracy can be observed on the BISINDO sign language dataset, one of the smallest considered in our SLR experiments.

Although our experiments focus on sign language recognition, they can be transferred to other use cases where patterns to be identified are located in an easily definable region of interest, if masks can be obtained at reasonable cost.

For future work, aiming to further enhance the performance and reliability of data augmentation techniques for sign language recognition, dynamic sign language recognition will be explored by assessing the effectiveness of our method on sign language videos. Local-global augmentation methods for videos will be developed and tested to leverage unannotated sign language data in the wild.

References

1. Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *Proc. ICML*, pages 1597–1607, 2020.
2. Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proc. CVPR*, pages 9729–9738, 2020.
3. Hadi Hojjati, Thi Kieu Khanh Ho, and Naregs Armanfard. Self-supervised anomaly detection in computer vision and beyond: A survey and outlook. *Neural Networks*, page 106106, 2024.
4. Jianhong Ma, Hao Duan, Cong Yang, Xiyao Wang, Yongpeng Niu, and Ying Han. Simclr-unet: An ecg feature wave segmentation algorithm based on a self-supervised learning strategy. In *Proc. of the 2022 4th RICAI*, pages 1354–1359, 2022.
5. Adrian Willi, Pascal Baumann, Sophie Erb, Fabian Gröger, Yanick Zeder, and Simone Lionetti. Self-supervised and few-shot learning for robust bioaerosol monitoring. *arXiv preprint arXiv:2406.09984*, 2024.
6. Ariel Basso Madjoukeng, Edith Belise Kenmogne, and Benoît Frenay. Fast k-means with stable instance sets. In *2024 Proc. IJCNN*, pages 1–8. IEEE, 2024.
7. Muhammad Al-Qurishi, Thariq Khalid, and Riad Souissi. Deep learning for sign language recognition: Current techniques, benchmarks, and open issues. *IEEE Access*, 9:126917–126951, 2021.
8. Romala Sri Lakshmi Murali, LD Ramayya, and V Anil Santosh. Sign language recognition system using convolutional neural network and computer vision. *Int. J. Eng. Innov. Adv. Technol*, 4:138–141, 2022.
9. Akash Nagaraj. ASL alphabet, 2018.
10. S Mohsin, BW Salim, AK Mohamedsaeed, BF Ibrahim, and SR Zeebaree. American sign language recognition based on transfer learning algorithms. *IJISAE*, pages 390–399, 2024.
11. Prathum Arikeri. Indian sign language character level dataset, 2021. Retrieved 2021-06-05.
12. Muhammad Al-Barham. Rgb arabic alphabets sign language dataset, 2023.
13. Muhammad Al-Barham, Osama Ahmad Alomari, and Ashraf Elnagar. Arabic sign language alphabet classification via transfer learning. In *ICETAI*, pages 226–237. Springer, 2023.
14. Achmad Noer. Bahasa isyarat indonesia (bisindo) alphabets. Kaggle Dataset.
15. Zehui Zhao, Laith Alzubaidi, Jinglan Zhang, Ye Duan, and Yuantong Gu. A comparison review of transfer learning and self-supervised learning: Definitions, applications, advantages and limitations. *Expert Syst. Appl.*, page 122807, 2023.
16. Deep R Kothadiya, Chintan M Bhatt, and Imad Rida. Simsiam network based self-supervised model for sign language recognition. In *International Conference on Intelligent Systems and Pattern Recognition*, pages 3–13. Springer, 2023.
17. Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *Proc. CVPR*, pages 15750–15758, 2021.
18. Karen Simonyan. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013.
19. Ammar Sayed. Arabic sign language dataset 2022, 2022.
20. Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proc. CVPR*, pages 2961–2969, 2017.

21. Bader Alsharif, Ali Salem Altaher, Ahmed Altaher, Mohammad Ilyas, and Easa Alalwany. Deep learning technology to recognize american sign language alphabet. *Sensors*, page 7970, 2023.
22. Baraa Wasfi Salim and SR Zeebaree. Kurdish sign language recognition based on transfer learning. *IJISAE*, pages 232–245, 2023.