Proceedings of the 9th Workshop on the Representation and Processing of Sign Languages, pages 165–170
Language Resources and Evaluation Conference (LREC 2020), Marseille, 11–16 May 2020

Automatic Classification of Handshapes in Russian Sign Language

Medet Mukushev*, Alfarabi Imashev*, Vadim Kimmelman†, Anara Sandygulova*
*Department of Robotics and Mechatronics, School of Engineering and Digital Sciences, Nazarbayev University
Kabanbay Batyr Avenue, 53, Nur-Sultan, Kazakhstan
†Department of Linguistic, Literary and Aesthetic Studies, University of Bergen
Postboks 7805, 5020, Bergen, Norway
mmukushev@nu.edu.kz, alfarabi.imashev@nu.edu.kz, vadim.kimmelman@uib.no, anara.sandygulova@nu.edu.kz

Abstract

Handshapes are one of the basic parameters of signs, and any phonological or phonetic analysis of a sign language must account for handshapes. Many sign languages have been carefully analysed by sign language linguists to create handshape inventories. This has theoretical implications, but also applied use, as an inventory is necessary for generating corpora for sign languages that can be searched, filtered, sorted by different sign components (such as handshapes, orientation, location, movement, etc.). However, creating an inventory is a very time-consuming process, thus only a handful of sign languages have them. Therefore, in this work we firstly test an unsupervised approach with the aim to automatically generate a handshape inventory. The process includes hand detection, cropping, and clustering techniques, which we apply to a commonly used resource: the Spreadthesign online dictionary (www.spreadthesign.com), in particular to Russian Sign Language (RSL). We then manually verify the data to be able to apply supervised learning to classify new data.

Keywords: Sign Language Recognition, Machine Learning Methods, Information Extraction

1. Introduction

Signs in sign languages are composed of phonological components put together under certain rules (Sandler and Lillo-Martin, 2006). In the early days of sign language linguistics, three main components were identified: handshape, location on the body, and movement, while later orientation and non-manual component were added. A recent paper stresses the need to combine interdisciplinary approaches in order to build successful sign language processing systems that account for their complex linguistic nature (Bragg et al., 2019).

By deploying a number of computer vision approaches, this paper aims to automate one of the most time-consuming tasks for linguists i.e. creation of a handshape inventory. Many researchers worked on establishing phonetic handshape and phonemic handshape inventories (see e.g. (Van der Kooij, 2002; Nyst, 2007; Tsay and Myers, 2009; Kubuş, 2008; Klezovich, 2019). In all of these works, handshapes were extracted and annotated manually (Klezovich, 2019).

Klezovich (2019) proposed the first handshape inventory for Russian Sign Language (RSL) by applying semi-automatic approach of extracting hold-stills in a sign video based on images overlay approach. The reason for extracting hold-stills from the rest of the video frames is due to the fact that handshapes are the most clear and visible in hold positions, and transitional movements never contain distinct handshapes. Klezovich proposed to extract hold-stills and then manually label only these frames, which can significantly speed up the process of creating handshape inventories (Klezovich, 2019).

In this paper, we test an automatic approach to generating handshape inventory for Russian Sign Language. First, we try an unsupervised learning and demonstrate that the results are unsatisfactory, because this method cannot distinguish handshapes separately from orientation and location in their classification. Second, we manually label a training dataset according to HamNoSys handshapes (Hanke, 2004), and demonstrate the utility of the supervised learning on new data.

2. Handshape as a phonological component

Ever since the seminal book by Stokoe (1960) on American Sign Language (ASL), signs in sign languages are analyzed as consisting of several parameters, one of the major ones being handshape (Sandler and Lillo-Martin, 2006). Handshape itself is not considered an atomic parameter of a sign, usually being further subdivided into selected fingers and finger flexion (Brentari, 1998).

Much research has been devoted to theoretical approaches to handshapes (see (Sandler and Lillo-Martin, 2006) for an overview), as well as to descriptions of handshape inventories in different sign languages (see e.g. (Caselli et al., 2017; Sutton-Spence and Woll, 1999; Fenlon et al., 2015; Kubuş, 2008; Klezovich, 2019; Kubuş, 2008; Van der Kooij, 2002; Prillwitz, 2005; Tsay and Myers, 2009)).

Several issues have been identified in studying handshapes that can be currently addressed using novel methods. First, many researchers identify the existence of the so-called unmarked handshapes (Sandler and Lillo-Martin, 2006, 161-162). These handshapes are maximally distinct in terms of their overall shape, they are the easiest to articulate, the most frequently occurring in signs, the first to be acquired by children, etc. For instance, in ASL, the following handshapes are generally treated as unmarked: A (fist), 5 (all fingers outstretched), 1 (index finger straight, all the other closed), E (all fingers bent and touching).

Since unmarkedness of handshapes derives from their visual and articulatory properties, it is expected that the same handshapes should be unmarked across different sign languages. This appears to be the case, although slight variation can also be observed. For instance, in Turkish Sign Language (TID), 7 handshapes can be identified as being the most frequent, including two handshapes based on the fist with or without outstretched thumb (Kubuş, 2008).

Figure 1: 135 top activated clusters for HOG descriptors.

In addition to the observation that (approximately) the same handshapes are the most frequent, a surprising finding is that the frequency of the most frequent handshapes is extremely similar across different sign languages. For instance, in British Sign Language (BSL), 50% of signs have one of the four unmarked handshapes (Sutton-Spence and Woll, 1999); in Turkish Sign Language, if we only consider the four most frequent handshapes, this would account for 57% of the signs (Kubuş, 2008), and, in ASL, the four most frequent handshapes in the ASL-LEX dataset (Caselli et al., 2017) account for 49% of all signs.

Secondly, some researchers argue that sign languages differ in their phonemic inventories, including the inventories of handshapes. For instance, Sign Language of the Netherlands has 70 phonetic and 31 phonemic handshapes (Van der Kooij, 2002), and many other sign languages are reported to have inventories of similar sizes (Kubuş, 2008; Caselli et al., 2017; Prillwitz, 2005). At the same time, Adamorobe Sign Language has been reported to have only 29 phonetic and 7 phonemic handshapes (Nyst, 2007). On the opposite end, a recent study of Russian Sign Language (RSL) based on semi-automatic large scale analysis has claimed that RSL has 117 phonetic but only 23 phonemic handshapes (Klezovich, 2019). Note however, that it is very difficult to directly compare results from different sign languages because different methods of assessing phonemic status of handshapes are used.

So we can observe both similarities in handshapes across different sign languages, as well as considerable variation. At the same time, it is difficult to make direct comparison because different datasets and annotation and classification methods are applied in different studies.

In the current study, we propose and test a method that can be applied to classifying handshapes across many sign languages using a common data set: the Spreadthesign online dictionary (www.spreadthesign.com). As a proof of concept, we analyze data from Russian Sign Language.

3. Dataset pre-processing

3.1. Dataset

The dataset was created by downloading videos from the Spreadthesign online dictionary (www.spreadthesign.com). We have downloaded a total of 14875 RSL videos from the website. The videos contain either a single sign or a phrase consisting of several signs.

Klezovich (2019) used the Spreadthesign online dictionary too, and after removing compounds, dactyl-based and number-based signs, she ended up working with 3727 signs or 5189 hold-stills.

In our case, blur images are removed using variation of Laplacian with a threshold of 350. If the variance is lower than the threshold then image is considered blurry, otherwise image is not blurry. Normally, we select threshold by trial and error depending on a dataset, there is no universal value. This reduced the number of total images from 141135 images to 18226 cropped images of hands.

3.2. Hand extraction

Hand detection can be considered as a sub-task of object detection and segmentation in images and videos. Hands can appear in various shapes, orientations and configurations, which creates additional challenges. Object detection frameworks such as MaskRCNN (He et al., 2017) and CenterNet (Duan et al., 2019) can be applied for this task.

However, occlusions and motion blur might decrease accuracy of the trained models. For these reasons, in this work, we used a novel CNN architecture namely Hand-CNN (Narasimhaswamy et al., 2019). Its architecture is based on the MaskRCNN (He et al., 2017) with an additional attention module that includes contextual cues during the detection process. In order to avoid issues with the occlusions and motion blur, Hand-CNN's proposed attention module is intended for two types of non-local contextual pooling, feature similarity and spatial relationship between semantically related entities. The Hand-CNN model provides segmentation, bounding boxes and orientations of detected hands. We utilize the predicted bounding boxes to crop hands with two padding parameters: a 0-pixel padding and a 20-pixel padding. As a result, the first group contains cropped images of detected hands only, while the other group contains cropped images of hands and their positions relative to the body.

3.3. Image pre-processing

To images with a 0-pixel-padding on detected hands, we apply Histogram of Oriented Gradients (HOG) descriptors (Dalal and Triggs, 2005). HOG feature descriptors are commonly used in computer vision for object detection (e.g. people detection in static images). This technique is based on distribution of intensity gradients or edge directions. Firstly, an image is divided into small regions and then each region has its histogram of gradient directions calculated. Concatenations of these histograms are used as features for clustering algorithm. In this work we use "feature" module of the scikit-image library (van der Walt et al., 2014) with the following parameters: orientations = 9, pixels per cell = (10,10), cells per block = (2,2) and L1 used as a block normalization method. Prior to this pre-processing, all images are transformed to grayscale and resized to 128 by 128 pixel images.

To images with a 20-pixel-padding on detected hands we utilize AlexNet (Krizhevsky et al., 2012). It is a Convolutional Neural Network (CNN) commonly used for various image processing tasks as a baseline architecture. We use only the first five convolutional layers with 96, 256, 384, 384 and 256 filters, as we only need to extract features for clustering purposes without the need for classification of images. Prior to feature extraction all images are resized to 224 by 224 pixels. CNN features are PCA-reduced to 256 dimensions before clustering.

4. Unsupervised Methodology

4.1. Clustering

We utilize a classical clustering algorithm, namely k-means. Thus, k-means implementation by (Johnson et al., 2019) is applied to ConvNet features, while scikit-learn (Pedregosa et al., 2011) implementation is applied to HOG features. Each training is performed for 20 iterations with random initialization.

We experimentally determined the number of clusters to be specified for clustering. It seemed like handshape orientation was also accounted for by the clustering algorithm, the idea was to increase the number of clusters to force the algorithm to differentiate between orientations. By trying

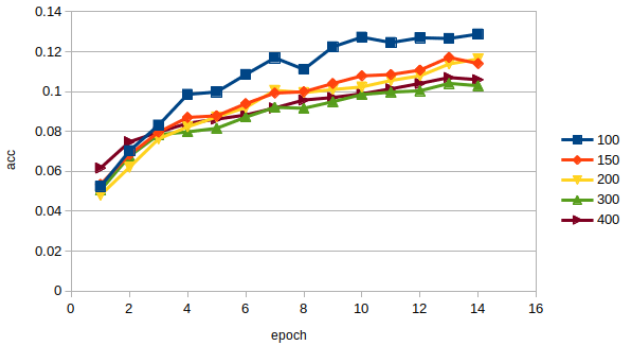

Figure 2: Average Silhouette Coefficient scores for the model trained on AlexNet features

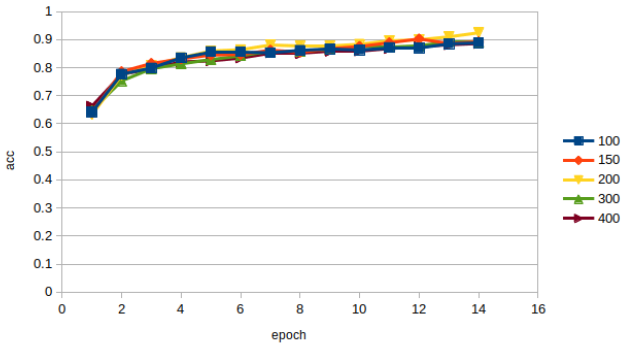

Figure 3: Normalized mutual information

varying step size, we ended up with setting for the following sizes: 100, 150, 200, 300 and 400 clusters.

4.2. Analysis and evaluation

We use two metrics to evaluate the performance of the clustering models: the Silhouette Coefficient and Normalized Mutual Information (NMI).

When the ground truth labels are not known for predicted clusters, Silhouette Coefficient score is applied. Silhouette method is used for interpretation and validation of clustering analysis (Rousseeuw, 1987). Its value gives understanding of how similar an item is to its own cluster compared to other clusters. Silhouette is bounded between -1 and +1, where a higher value means that a clustered item is well matched to its cluster and less matched to other clusters. As can be seen from Figure 2, the maximum value of Silhouette Coefficient score is observed for the model trained on AlexNet features for 100 clusters after 15 epochs. However, the score itself is just slightly over 0.12 which indicates that our clusters are overlapping.

In addition, we use predicted labels to measure NMI. It is a function that measures the agreement between predicted and actual labels. Perfect labeling gives score of +1 and bad labeling gives negative scores. As we can see from Figure 3, all models with different number of clusters result in the scores reaching 0.9 after 15 epochs.

The reason for such results might be that image descriptors for hands are too close to each other, which makes it difficult for the algorithm to differentiate. At the same time, NMI score indicates that predicted labels are almost the

same after each training epoch. In order to increase density of predicted clusters additional pre-processing of images is required.

4.3. Results

Figure 1 gives us insights about the results of applying unsupervised clustering to handshapes. First, it is clear that the algorithm does not distinguish classes only based on handshapes, but also based on orientation (for images with 0-pixel-padding), and also based on localization (for images with 20-pixel-padding). If the linguistic task of creating an inventory of phonemic handshapes is at stake, this is a clear disadvantage of this approach.

Second, despite its shortcomings, the method does provide some linguistically interesting results. Specifically, one can see that the handshapes which are expected to be unmarked (A, 5, 1) appear frequently and as labels for multiple classes. Thus, even though the classification is not linguistically relevant, the effect of markedness is still visible in the results of this unsupervised approach.

5. Supervised Methodology

5.1. Dataset

Given that the unsupervised approaches did not result in a clustering reflective of relevant handshape classes, we turned to a supervised approach. The results of HOG clustering was used as the initial dataset that contained 140 clusters of 18226 images. It was decided to manually clean the automatically generated clusters for inaccuracies. This task was performed by four undergraduate students, who divided the folders first between each other and then one person merged all of them.

First, each cluster (folder) was visually scanned for the most frequently classified handshape in order to remove handshapes that did not belong there from that folder. These steps were performed for all 140 folders. Since there were many folders of the same handshape with the only difference in orientation, they were merged, which resulted in 35 classes and a large unsorted (junk) folder. Thus, the final version of the dataset contains 35 classes of 7346 cropped images with 0-pixel-padding.

The classes were created using intuitive visual similarity as a guide, and by linguistically naive annotators. However, a post factum analysis shows that the manual classification is linguistically reasonable as an approximation of a phonological inventory. Specifically, the classes that were created are distinguished by selected fingers, spreading (spread or not), and finger position (straight, bent, curved). Thumb position is only used as a distinguishing feature for opposed thumb vs. all other possibilities. Non-selected finger position is not taken into account. This reasonably approximates features relevant for proposed phonological inventories in other sign languages, and, as such, can be used for RSL as well.

If phonetic classes were the target, then classes would also need to be distinguished by exact thumb position and also by the differences in non-selected fingers. In such a case the full inventory of possible handshapes described in HamNoSys (Hanke, 2004) could be used as the basis for manual

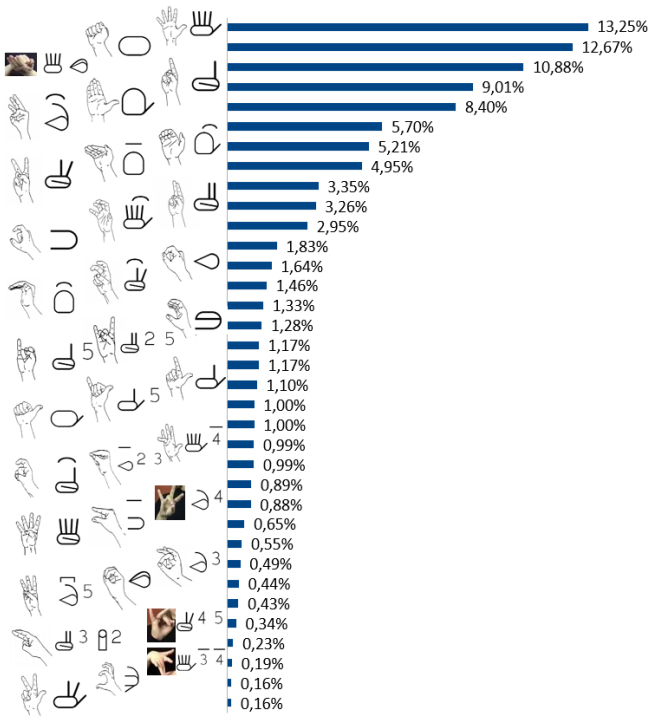

Figure 4: Handshape classes count

classification. However, the dataset we used appears to be too small to attempt a phonetic classification.

The manually labeled subset was later divided into a training set with 6430 images and a validation set with 916 images. Figure 4 shows the number of tokens for each class in a training and validation sets combined. Figure 4 also shows a linguistically relevant result: our manual classification of handshapes also demonstrates the expected frequency properties of marked and unmarked handshapes. In particular, the most frequent handshapes are the ones expected to be unmarked: A (fist), 5 (hand with all fingers spread), 1 (index finger), and B (a flat palm)).[1] These forms together constitute 48% of all handshapes (if the two-handed signs are disregarded).

5.2. ConvNet and transfer learning

Training an entire ConvNet from scratch for a specific task requires big computational resources and large datasets, which are not always available. For this reason, a more common approach is to use ConvNet that was pretrained on large datasets, such as ResNet-18 or ImageNet (which contains 1.2 million images divided into 1000 categories) as a feature extractor for a new task. There are two common transfer learning techniques based on how we use pretrained ConvNet: finetuning the ConvNet and ConvNet as a fixed feature extractor. In the first technique, we use weights of a pretrained model to initialize our network instead of random initialization. All the layers of the ConvNet are trained. In the second approach, we freeze the weights for all of the network layers and only the last final fully connected layer is changed with random weights and

[1]The *10.88%* class in Figure 4 includes all two-handed signs, which we do not attempt to classify according to handshape at the moment.

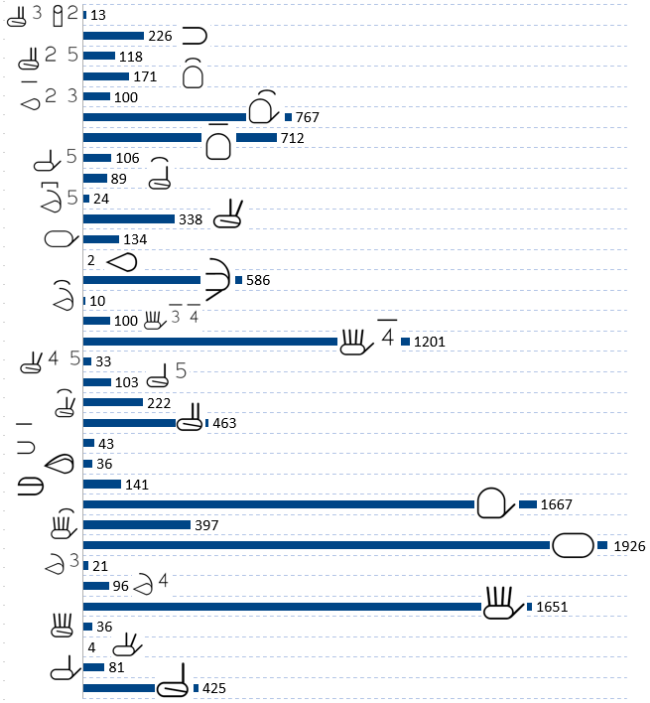

Figure 5: Handshape classes count using classifier

only this layer is trained. We implemented our networks using PyTorch (release: 1.4.0) that is an open source machine learning library (Paszke et al., 2019). Our code is based on Chilamkurthy's Transfer Learning for Computer Vision Tutorial (Chilamkurthy, 2017). ResNet-18 (He et al., 2016) model was used as a pretrained model.

5.3. Results

We trained two networks using both approaches. Each model was trained for 200 epochs. Using the second approach (i.e. ConvNet as a fixed feature extractor with only the last layer trained), the best accuracy of 43.2% was achieved. On the other hand, the first approach (i.e. finetuning the ConvNet and training all layers) demonstrated a better accuracy of 67%. Therefore, the finetuned model was used for further accuracy improvements. First, we added data augmentation to increase the number of samples. Samples were randomly rotated and visual parameters (brightness, contrast, and saturation) were randomly changed with a probability of 0.25. This helped to increase the accuracy of the best model up to 74.5% after 200 epochs. Later, we used this trained model to predict labels for all 18226 handshapes. In order to remove cases that were misclassified, a threshold for prediction probability was set to 0.7. And as a result, 12042 samples were classified. Figure 5 demonstrates the number of predicted samples for each class.

6. Discussion

6.1. Insights from unsupervised and supervised approaches

The current study shows that the unsupervised approach does not seem promising in the task of automating handshape recognition. The main problem is that the category of handshape is linguistically relevant, but not visually separable from orientation and location by this very basic data-driven approach.

We have demonstrated that an alternative approach involving a manual classification step can be quite effective. However, manual classification is problematic for obvious reasons, as it involves human judgment.

Both approaches, however, offer some linguistically relevant insights, specifically concerning unmarked handshapes. In the unsupervised approach, it is clear that many clusters are assigned unmarked handshapes as labels, which can be explained by both their frequency and visual salience. In the supervised approach, our manual classification of 7346 handshapes demonstrated that the unmarked handshapes (A, 1, 5, B) are indeed the most frequent ones. Finally, applying the ConvNet model to the whole dataset of 18226 handshapes has shown that top 3 classes are A, B, 5. Interestingly, the 1 handshape is not in the top most frequent ones. The most likely explanation is that this handshape is frequently misclassified as the handshape with middle finger bent and the other fingers outstretched (the 'jesus' handshape in the figures), which is a rare marked handshape in the manually classified dataset, but frequent in the results using the classifier.

Thus, both successful and less successful applications of machine learning methods show the importance of unmarked handshapes in RSL. It would be interesting to extend these approaches to other sign languages for comparative purposes.

6.2. Comparison with Klezovich 2019

As discussed above, Klezovich (2019) proposed the first handshape inventory for RSL by applying semi-automatic approach of extracting hold-stills in a sign video using the same dataset used here (Spreadthesign). This gives us the opportunity to compare the results of a more traditional linguistic analysis of handshape classes in RSL with the approach used in the current study.

A direct comparison is possible between Klezovich's results and the results of our unsupervised learning approaches. Both result in a classification of handshapes. However, we have demonstrated that the results of unsupervised clustering are unsatisfactory, so it cannot be used for any linguistically meaningful applications.

As for the supervised approach, both our approach and Klezovich's analysis include manual annotation, but in different ways. Klezovich manually classified handshapes into potential phonemic classes using linguistic criteria, which resulted in a large linguistically informed inventory. We manually classified handshapes based on visual similarity into a smaller number of classes, and then used this as a dataset for machine learning.

The comparison between Klezovich's and our manual classifications is not very informative, as only the former was based on linguistic criteria. Given that Klezovich's classification was not used as a training set for automatic recognition, no comparison is possible for this aspect either. This issue is left for future research.

7. Conclusion

We have shown that by deploying a number of classical machine learning algorithms, it is possible to partially automate one of the most time-consuming tasks for linguists i.e. creation of a handshape inventory, and, in addition, to investigate frequencies of various handshapes in a large data set. At the moment, it seems that unsupervised approaches cannot be used to create handshape inventories because orientation and location differences also influence clustering, and to an even greater extent than handshape itself. A supervised approach is clearly more effective, however, it requires a manual annotation component where a substantial number of handshapes is manually classified. This introduces additional problems of determining the number of classes for manual classification. Upon achieving the satisfying unsupervised clustering results, future work will focus on comparing and applying this framework to other sign languages.

8. Bibliographical References

Bragg, D., Koller, O., Bellard, M., Berke, L., Boudreault, P., Braffort, A., Caselli, N., Huenerfauth, M., Kacorri, H., Verhoef, T., Vogler, C., and Ringel Morris, M. (2019). Sign language recognition, generation, and translation: An interdisciplinary perspective. In *The 21st International ACM SIGACCESS Conference on Computers and Accessibility*, ASSETS '19, pages 16–31, New York, NY, USA. ACM.

Brentari, D. (1998). *A prosodic model of sign language phonology*. MIT Press.

Caselli, N. K., Sehyr, Z. S., Cohen-Goldberg, A. M., and Emmorey, K. (2017). ASL-LEX: A lexical database of American Sign Language. *Behavior Research Methods*, 49(2):784–801, April.

Chilamkurthy, S. (2017). Transfer learning for computer vision tutorial. *https://chsasank.github.io/*.

Dalal, N. and Triggs, B. (2005). Histograms of oriented gradients for human detection.

Duan, K., Bai, S., Xie, L., Qi, H., Huang, Q., and Tian, Q. (2019). Centernet: Keypoint triplets for object detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 6569–6578.

Fenlon, J., Cormier, K., and Schembri, A. (2015). Building BSL SignBank: The Lemma Dilemma Revisited. *International Journal of Lexicography*, 28(2):169–206, June.

Hanke, T. (2004). Hamnosys: representing sign language data in language resources and language processing contexts. In Oliver Streiter et al., editors, *LREC 2004, Workshop proceedings: Representation and processing of sign languages.*, pages 1–6, Paris.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

He, K., Gkioxari, G., Dollár, P., and Girshick, R. (2017). Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969.

Johnson, J., Douze, M., and Jégou, H. (2019). Billion-scale similarity search with gpus. *IEEE Transactions on Big Data*.

Klezovich, A. (2019). *Automatic Extraction of Phonemic Inventory in Russian Sign Language*. BA thesis, HSE, Moscow.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105.

Kubuş, O. (2008). *An Analysis of Turkish Sign Language Phonology and Morphology*. Diploma thesis, Middle East Technical University, Ankara.

Narasimhaswamy, S., Wei, Z., Wang, Y., Zhang, J., and Hoai, M. (2019). Contextual attention for hand detection in the wild. *arXiv preprint arXiv:1904.04882*.

Nyst, V. (2007). *A Descriptive Analysis of Adamorobe Sign Language (Ghana)*. LOT, Utrecht.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al. (2019). Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, pages 8024–8035.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Prillwitz, S. (2005). Das Sprachinstrument von Gebärdenspachen und die phonologische Umsetzung für die Handformkomponente der DGS. In Helen Leuninger et al., editors, *Gebärdesprachen: Struktur, Erwerb, Verwendung*, pages 29–58. Helmut Bukse Verlag, Hamburg.

Rousseeuw, P. J. (1987). Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20:53 – 65.

Sandler, W. and Lillo-Martin, D. (2006). *Sign language and linguistic universals*. Cambridge University Press.

Stokoe, W. (1960). *Sign Language Structure: An Outline of the Visual Communication Systems of the American Deaf*. Number 8 in Studies in Linguistics: Occasional Papers. Department of Anthropology and Linguistics, University of Buffalo, Buffalo.

Sutton-Spence, R. and Woll, B. (1999). *The Linguistics of British Sign Language*. Cambridge University Press, Cambridge.

Tsay, J. and Myers, J. (2009). The morphology and phonology of Taiwan Sign Language. In James Tai et al., editors, *Taiwan Sign Language and Beyond*, pages 83–130. The Taiwan Institute for the Humanities, Chia-Yi.

Van der Kooij, E. (2002). *Phonological Categories in Sign Language of the Netherlands. The Role of Phonetic Implementation and Iconicity.* LOT, Utrecht.

van der Walt, S., Schönberger, J. L., Nunez-Iglesias, J., Boulogne, F., Warner, J. D., Yager, N., Gouillart, E., Yu, T., and the scikit-image contributors. (2014). scikit-image: image processing in Python. *PeerJ*, 2:e453, 6.