D1.1 First Version of Visual Tracking and Feature Extraction Components

Partner Responsible	UTH-ECE
Other Contributors	AthenaRC
Document Reference	**D1.1**
Dissemination Level	Public
Version	1.0 (Final)
Due Date	July 2020 (**M06**)
Date of Preparation	July 2020

Contract No.: HFRI-FM17-2456

Editor

Gerasimos Potamianos (**UTH-ECE**)

Contributors

UTH-ECE: Katerina Papadimitriou, Gerasimos Potamianos
AthenaRC: Eleni Efthimiou, Stavroula-Evita Fotinea, Petros Maragos

SL-ReDu Principal Investigator:

Assoc. Prof. Gerasimos Potamianos
University of Thessaly, Electrical and Computer Engineering Department (**UTH-ECE**)
Volos, Greece 38221
email: gpotamianos@uth.gr (gpotam@ieee.org)

Executive Summary

SL-ReDu is an innovative project that aims to considerably advance the current state-of-the-art in the automatic recognition of Greek Sign Language (GSL) from videos, while focusing on the standardized teaching of GSL as a second language as a use-case. In this deliverable (D1.1), we present our initial work on the detection and tracking of the visual articulators of sign language (both manual and non-manual), as well as the extraction of corresponding visual features, which constitutes the first component for the recognition pipeline of WP1 and WP2. Specifically, we examine the detection and tracking of the various sign language (SL) articulators of interest, namely the hands and mouth employing traditional detection and tracking techniques, as well as deep learning-based approaches (Task T1.1). Regarding the visual feature extraction of the detected articulators, traditional shape, appearance, and motion-based feature representations, as well as more advanced deep-learning representations are explored (Task T1.2). Further, performance of the detection and tracking algorithms as well as of the extracted features is reported on three datasets of GSL. In particular, performance of the various feature sets is investigated in terms of computational efficiency and recognition accuracy on the task of isolated GSL recognition. This deliverable will be updated as D1.2 (M16), while also providing baseline components for D2.1 (M12).

Table of Contents

1 Introduction

Automatic sign language (SL) recognition constitutes an important human-computer interaction technology, allowing natural language communication for the speech and hearing impaired. At the same time, European and national policies on inclusion and accessibility, as well as the official recognition of national sign languages, have led to a dramatic increase in the need for communication and education in sign language (SL), both as mother language (L1) and as second language (L2), well beyond the approximately 1‰ of the deaf population [1]. Yet, nonnative SL education remains a cumbersome process, demanding extensive and iterative tutor-to-learner feedback on a one-to-one basis, while also suffering from a high degree of teacher subjectivity in the evaluation of student proficiency [2].

Due to its non-vocal nature, SL forms a means of expression that comprises both manual (i.e. hand shape, motion pattern, hand relative position, and orientation) and non-manual (i.e. body posture, facial expressions, and body motion) articulation that integrates simultaneously on multiple streams contributing to the formation of basic SL signs. Thus, it is clear that a successful SL recognition system should be able to accurately track both manual and non-manual articulators in space and time, recognize patterns in the respective articulatory streams, and fuse them at the appropriate temporal level to yield basic signs and their temporal sequence [3]. It should also be able to accomplish the above in a signer-independent fashion, thus accounting for natural variability in individual-signing style. Crucial components in addressing the problem are the visual detection, tracking, and visual feature representation of both manual and non-manual SL articulation. Achieving such goals requires an interdisciplinary effort, employing state-of-the-art techniques in computer vision, machine learning, and sign linguistics, while exploiting large amounts of SL data, and constitutes the focus of this deliverable (D1.1).

Over the last three decades, there has been significant research activity on the problem of hand detection [4-7] for human-computer interaction (HCI), while at the same time, substantial attention has been dedicated to the field of face detection and recognition, achieving tremendous success [8, 9]. There exists a large variety of such algorithms based on traditional RGB video input, exploiting for example skin color consistency to apply skin-based color models for manual and non-manual SL articulation detection [10], while others take advantage of the dynamic nature of gesturing incorporating motion-based algorithms for tracking [11]. Moreover, there exist various systems ensued from the combination of articulation position and/or movement estimation [12-15, 63, 64] and appearance-based descriptors [16-19, 65]. In recent years, significant attention has been paid to depth (RGB-D) cameras [20], such as the Kinect [21] and Intel RealSense [22], due to the inherent advantages of the depth modality information and availability of the human skeleton data stream [21]. Specifically, systems utilizing such cameras have been introduced [23-25], relying on hand-crafted feature descriptors that are extracted from the depth and/or skeleton streams. Lately, deep learning approaches have provided a breakthrough in the field of visual detection and feature learning, primarily in the form of convolutional neural networks (CNNs) [26], demonstrating superior performance on several tasks. Among the proposed deep learning architectures, region-based CNNs (RCNNs) [27] and their variants [28-30] have been extremely popular for object detection, and therefore for face and hand detection. Specifically, in [31] detection and tracking are accomplished through the faster-RCNN [29] and a two-stream 3D CNN for spatiotemporal SL feature extraction. Most of the aforementioned deep learning advances have been disseminated to the research community via open source toolkits. Such examples constitute the OpenFace framework regarding facial landmark tracking [32], as well as OpenPose [33] that results in accurate 2D human skeleton estimates including detailed hand information.

This deliverable aims to analyze and compare the methods implemented in previous researches concerning the first component of SL-ReDu recognition system (WP1), providing the necessary input to the machine learning algorithms for the GSL recognition task (WP2). Moreover, it aims to suggest the best method to explore for future research. Specifically, WP1 comprises two tasks, namely the detection and tracking of the visual articulators in SL video, as well as the extraction of visual features of tracked

articulators. Specifically, the first task (Task T1.1) focuses on the detection and tracking of the various SL articulators of interest, namely the hands and mouth. For this purpose, a "light-weight" scheme is first explored, where facial information accomplished by an efficient off-the shelf detector drives skin-tone based hand segmentation complemented with motion tracking to drive localization. Subsequently, the OpenPose framework [33], which is a more computationally demanding deep-learning approach based on skeletal data generation [34-36] is incorporated. Regarding the extraction of appropriate representations of the tracked manual and non-manual articulators (Task T1.2), traditional shape/geometric feature representations, appearance schemes such as PCA representations of the regions-of-interest and more advanced feature sets such as Gabor filterbank energies, SIFT, LBPs, HOGs, and optical flow representations are considered. Further, deep learning-based representations using CNNs as well as deep auto-encoders are also pursued, thus providing a multitude of visual streams to be utilized in the SL recognizer of WP2. Our approaches are evaluated on three isolated-sign GSL datasets: (a) The *Polytropon GSL corpus* [12]; (b) the *ITI GSL dataset* [37]; and the *Dicta-Sign dataset* [38], employing a CRNN (CNN and RNN) classifier for the sign prediction task. The approach adopted in this deliverable is schematically depicted in Figure 1.

The remainder of D1.1 is structured as follows: In Section 2, we overview methods for detection and tracking of the visual articulators (both manual and non-manual). In Section 3, we focus on the extraction of various visual feature sets. In Section 4, we overview the GSL datasets used in our evaluation, followed by experimental results in Section 5. Finally, in Section 6, we conclude this deliverable and discuss our future plans.

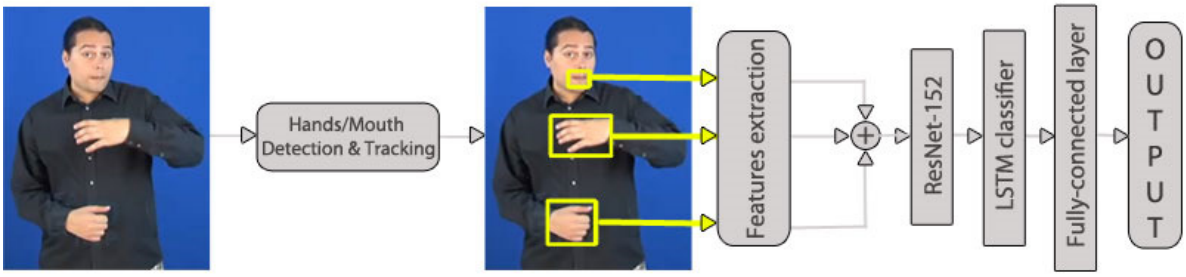


Figure 1: General overview of the deliverable methodology. **Left:** Detection and tracking methods (Task T1.1) are employed to extract regions-of-interest of the hands (manual articulators) and mouth (non-manual articulator). **Middle:** Various visual features are extracted from the aforementioned regions-of-interest (Task T1.2). **Right:** For evaluation purposes, the extracted feature sets of each region-of-interest are concatenated and evaluated for isolated sign classification from GSL video data.

2 Task T1.1: Detection and Tracking of Visual Articulators in SL Video

This section describes in detail the evaluated systems concerning the detection and tracking of manual and non-manual SL articulators from RGB video. As already mentioned, traditional detection and tracking techniques are first developed towards a low-resource SL recognition system baseline, refining recent work by the PI's team [11]. Further, the OpenPose deep learning-based toolkit [33] is also explored to yield facial landmarks and 2D body skeletons, including hand detail.

2.1 Light-Weight Hand Detection and Type Classification (HDTC)

The hand detection and type classification scheme (HDTC) investigated here refines our earlier work [11] for the visible signer hands extraction, as well as their classification into left and right types (as viewed by the camera). The approach is two-phase, relying on both traditional techniques for efficient articulation localization and tracking, as well as deep learning for hand type detection and classification, under the assumption that the signer frontal head pose data is visible, as in the case of typical SL videos. Specifically, the system relies on two distinct pillars: the image pre-processing pipeline and the classification phase, as also depicted in Figure 2. As it may be observed, the pre-processing component involves a series of individual steps, namely: (a) skin-tone estimation; (b) skin-tone based segmentation; (c) skin region motion tracking, and (d) hand segmentation. The essence of the first phase is the generation of a limited number of proposal windows to be subsequently fed to the CNN-based hand-type classification component.

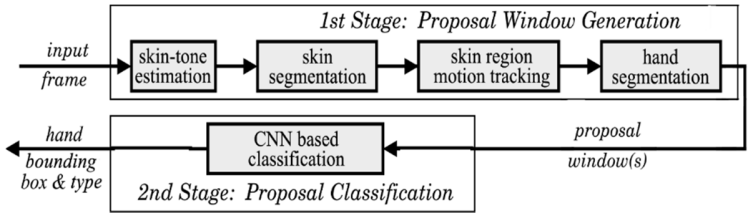


Figure 2: Block diagram of the two-stage HDTC system for hand detection and type classification.

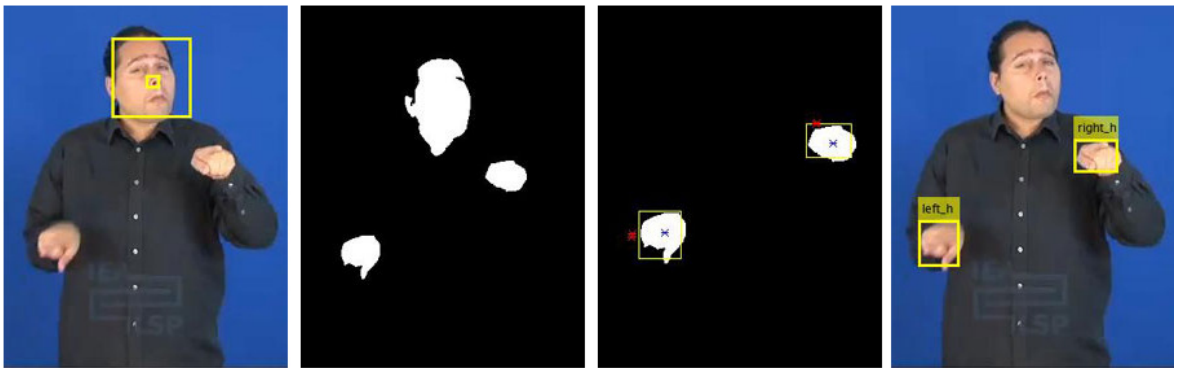
The pipeline commences with skin color range estimation via the Viola-Jones face detection algorithm [8], which is a widely used real-time face detector due to its computational efficiency and high efficacy. The ultimate aim of the face detection process is the central square of the facial region extraction, which does not involve background information and, thus, best captures the skin tone. Assuming a successful face detection (see Figure 3(a)), the nose area (covering 13% of the facial bounding box) is extracted and subsequently converted to the YCbCr color space [39], driving skin-tone based segmentation. Specifically, after image frame transformation into the YCbCr color space, skin pixels are classified regarding the range of the corresponding YCbCr values of the extracted nose region (see Figure 3(b)). In case of missing face detection, the frame is imposed on skin segmentation in the YCbCr color space via particular threshold values as defined in [39].

Following skin segmentation, hands are subjected to skin region motion tracking through the motion-based Kalman filtering [40] in order to address hand and facial regions overlapping. Treating hands as the only skin-tone moving objects in the frame, Kalman filter algorithm that has proven an optimal state estimator, assigns detections to related tracks regarding their previous location (see Figure 3(c)). Thus, in our task, Kalman filtering provides hand localization and tracking discarding skin-like objects that do not correspond to moving tracks.

The image processing pipeline concludes with the background subtraction of the previously produced rectangular bounding boxes using the Otsu's thresholding method [41]. Generating bounding boxes, the so-called proposal windows that involve only the target objects is critical for the classification phase.

During classification, the returned proposal windows pass through a CNN classifier for final hand detection and type classification, considering three classes of interest, namely left, right, and no hand (see Figure 3(d)). The CNN adopts the AlexNet architecture [42], due to its wide endorsement by the computer vision community and its high-performance capability. In more detail, each ROI is resized to the AlexNet input layer fixed size (227 x 227 pixels) and fed to an AlexNet CNN classifier for the label prediction task. More precisely, the AlexNet CNN follows a five convolutional and three fully-connected layers and is pretrained on the ImageNet corpus [43], which includes a large set of 1000 labeled images for each of 1000 categories. In order to adapt the network to our task, we modify the final fully-connected layer to have the same size as the number of classes of interest in this work (three). Finally, the network is fine-tuned employing the Visual Geometry Group hand dataset of the Oxford University [44], which consists of almost 6k labeled images of hand types. The dataset is randomly divided into training and test sets (70% and 30%). During training, Stochastic Gradient Descent with Momentum (SGDM) is employed with an initial learning rate of 0.004 decayed by a factor of 0.5. The maximum number of complete passes (epochs) is set to 60, and a mini-batch with 128 images is used.

In addition to the manual articulators (hands) extracted as above, the mouth regions (non-manual articulators) are extracted via the Viola-Jones mouth detector.



(a) **(b)** **(c)** **(d)**

Figure 3: An example of the hand detection and type classification approach of Task 1.1, applied to Polytropon GSL corpus. Depicted, left to right: (a) video frame marked with a rectangular box enclosing the detected facial region, as well as the central square of the detected face region; (b) segmented skin region; (c) tracked hands by Kalman filtering (yellow rectangles depict detected objects, red stars the predicted object positions, and blue stars their corrected positions); (d) frame marked with rectangular boxes illustrating the signer's left and right hands.

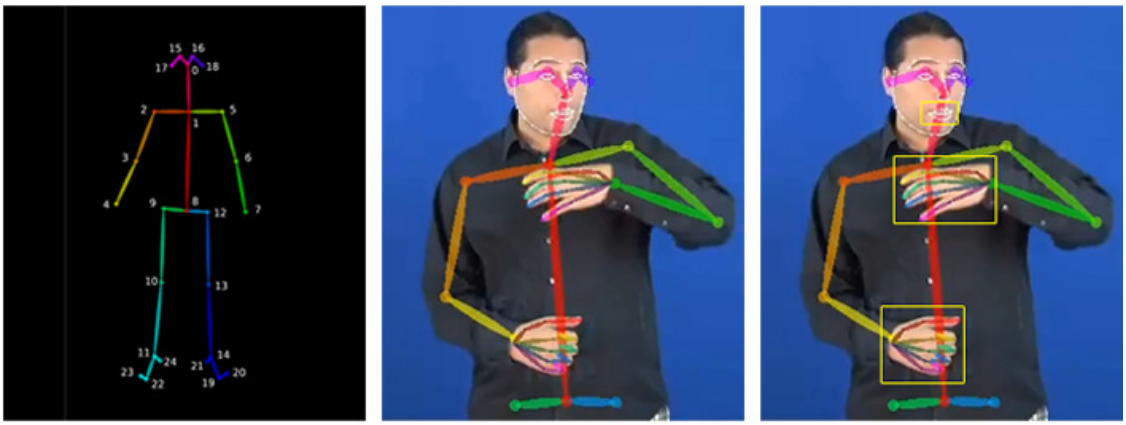
2.2 OpenPose-Based Articulator Detection

This approach relies on the extraction of 2D human skeletal data through the OpenPose human joint detector [33], which provides a descriptive motion and structural representation of the human body (body pose, hands, and face) relying on deep convolutional pose models. OpenPose has great potential for several real-life applications, enabling motion detection without dedicated hardware, like the Kinect or motion capture equipment. The OpenPose network initially extracts image features employing the first 10 layers of VGG-19 [45], which are then fed into two parallel convolutional layers branches. The first branch generates a set of confidence maps, each representing a specific human pose skeleton part, while

the second produces a set of part affinity fields (PAFs) [33] that represents the degree of confidence of the association for each pair of body part detections.

OpenPose provides a detailed spatio-temporal representation of the human skeleton, extracting in total 137 human skeleton joint descriptors in the form of image coordinates (see Figure 4(a)). Specifically, OpenPose renders 25 body pose keypoints, 21 joints for each hand, as well as 70 facial keypoints, as also depicted in Figure 4(b). Since in the majority of SL videos only the signer upper-body parts are involved in the signing process, here we exploit only 57 estimated coordinates, removing 10 human body joints associated with the invisible lower body parts of the signer, as well as the face joints.

The most dominant spatial information concerning SL is focused on hands involving handshape deformation and orientation, as well as lip shape. To this end, we segment the mouth and hand regions (see Figure 4(c)) using the skeletal coordinates of the regions-of-interest, which constitute descriptive spatial representations that can significantly enhance SLR performance.

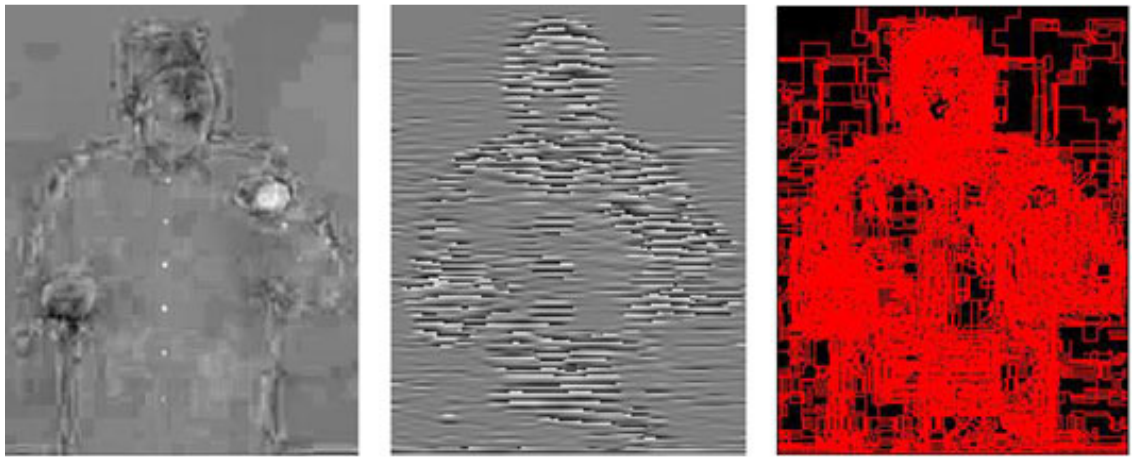


(a) **(b)** **(c)**

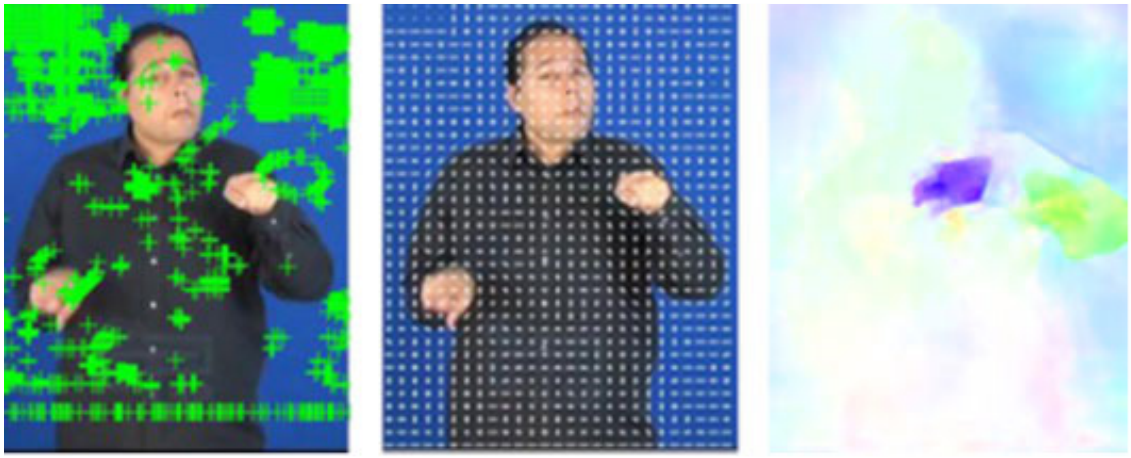
Figure 4: (a) An example of the skeleton representation obtained by the OpenPose library [33]; (b) input image frame from the Polytropon GSL corpus [12] with super-imposed keypoints generated by OpenPose; and (c) input image marked with rectangular boxes enclosing the handshapes and the mouth region derived based on the human skeleton.

3 Task T1.2: Extraction of Visual Features of Tracked Articulators

In this section, we consider various feature learners of the tracked manual and non-manual articulators, namely the hands and mouth, at a local or global level, emphasizing their shape and/or appearance, and concentrating on static and/or motion patterns, thus providing a multitude of visual streams to the SL recognizer of Section 5.1. Specifically, traditional, hand-crafted shape/geometric feature representations, appearance schemes such as principal component analysis (PCA) representations of the regions-of-interest and more advanced feature sets, such as Gabor filterbank energies, the scale-invariant feature transform (SIFT), local binary patterns (LBPs), histogram of oriented gradients (HOG), and optical-flow representations are investigated. Further, deep learning-based representations using CNNs as well as deep auto-encoders are also considered. Examples of the hand-crafted feature sets considered (Sections 3.1-3.6) are depicted in Figure 5 (extracted over an entire video frame, for better visualization).



(a) (b) (c)



(d) (e) (f)

Figure 5: Visualization examples of various feature representations of Section 3 on a video frame of the Polytropon GSL corpus: (a) PCA; (b) Gabor filters magnitude; (c) LBP features; (d) SIFT approach; (e) HOG features; and (f) SpyNet based optical flow.

3.1 Principal Component Analysis (PCA)

PCA [46] is an unsupervised feature learner used in data science for reducing the dimensionality of multi-variate data, while preserving as much relevant information as possible. Specifically, PCA utilizes the eigenvectors of the data covariance matrix that correspond to the matrix largest eigenvalues, employing them to project the data to a new subspace of typically significantly smaller dimension. In other words, PCA reduces feature dimensionality, retaining though a significant portion of the original information. In more detail, PCA converts data into a lower-dimensional space using an orthogonal transformation whilst maximizing the data variation. Here, we discard feature dimensions that correspond to small eigenvalues containing little information, in order to increase data sample density in the feature space and remove noise, resulting in 256-dimensional feature vectors. It should be noted that the eigenvalue decomposition of the data covariance matrix is used to perform PCA.

3.2 Gabor Filter-Banks

Gabor filters [47] were originally proposed for signal representation in both time and frequency. A Gabor filter is a linear filter that can be defined as a sinusoidal function multiplied by an elliptical Gaussian. A Gabor filter analyzes whether there exists specific frequency content in the image along specific directions at a localized region around the point or region of analysis. Frequency and orientation representations of Gabor filters are appropriate for representation and discrimination. Specifically, in the spatial domain, a 2D Gabor filter is a Gaussian kernel function modulated by a sinusoidal plane wave. Here, we generate a custom Gabor filter-bank, whose elements are 39x39 matrices with each matrix being a 2D Gabor filter. Subsequently, these filters are applied to the image, resulting to a 360-dimensional Gabor feature vector. The factor of down-sampling along rows and columns is 110.

3.3 Local Binary Patterns (LBP)

The LBP feature vector [48], in its typical form, divides the region-of-interest into cells. It then employs each pixel in the cell in a threshold searching process, calculating the optimal threshold that maximizes the between-class variance. Specifically, each pixel in the cell is compared with its 8 neighbours following the pixels in a clockwise manner. If the pixel value is greater than a neighbour value, it is replaced with 0, otherwise with 1, generating an 8-digit binary number. Subsequently, the cell histogram of the frequency of each previously produced number is computed. Finally, all histograms are concatenated giving the feature vector of the entire region. In this study, a set of 8 neighbours is used to encode LBP features for each pixel in the input image selected from a circularly symmetric pattern around each pixel, generating a 256-dimensional feature vector.

3.4 Scale-Invariant Feature Transform (SIFT)

The SIFT approach [49] transforms an input image to a large collection of local feature vectors. Each of these is scale- and rotation-invariant. In order to extract these features, the model applies scale-space extrema detection, for identification of those locations and scales, keypoint localization, for eliminating points with low contrast and/or poorly localisation on an edge, orientation assignment, for obtaining consistent orientations to the keypoints based on local gradient data; and finally extracts the keypoint descriptor, relying on the local gradient data. Thus, after extracting the keypoints location and orientation in the input image, we exploit them to generate a 256-dim SIFT feature vector. It should be noted that since there is a variability in the number of keypoints per frame and, by extension, to the resulted feature vector dimensionality, we employ t-distributed stochastic neighbour embedding (t-SNE) [50] that

constitutes a machine learning-based dimensionality reduction technique for generating feature vectors with fixed, predefined dimensionality.

3.5 Histogram of Oriented Gradients (HOG)

HOG [51] is a hand-crafted feature descriptor that considers the distribution of oriented gradients in the image as features, where the magnitude of gradients is large around regions of abrupt intensity changes capturing a lot more information about object shape. Here, the HOG feature vector is calculated to have dimensionality of 324, encoding local shape information from regions within the image.

3.6 Optical Flow

Since a critical aspect concerning SL is the motion estimation in time, we also evaluated the use of optical flow spatio-temporal feature representations. To acquire it, the well-known SpyNet [52] model is employed, which combines classical optical flow algorithms with deep learning techniques. Once the optical flow is estimated, optical flow vector with 256 dimensionality is generated according to the magnitude and orientation between two adjacent frames.

3.7 Convolutional Neural Network (CNN)

CNNs are composed of a series of convolutional layers complemented with non-linearity and pooling, followed by fully connected layers and an output layer. Here, we apply a pre-trained ResNet-18 network [53] (trained on the ImageNet database [43]) to each region-of-interest in order to extract feature maps by taking the output of the fully-connected layer. The network uses 3 x 3 convolutional kernels, down-sampling with stride 2, and is trained using the mean squared error loss function. Note that the input is resized to the fixed size of the ResNet-18 network input layer (224 x 224 pixels). The network outputs feature maps of 512 dimensions by taking the output of the global average pooling layer.

3.8 Vanilla Auto-Encoder (AE)

The vanilla AE [54] is a neural network that performs a mapping of the input image to a latent space representation through encoding, and then reconstructs the output by utilizing a decoder. In particular, it employs a multi-layer perceptron (MLP) encoder and decoder using the mean squared error loss function. Here, a vanilla stacked AE is used, consisting of two hidden layers with fixed dimensionality 100 on the encoder and the decoder. For weight initialization, we perform the Xavier process [55]. Auto-encoder network training is conducted using Scaled Conjugate Gradient Descent (SCGD) with an initial learning rate of 0.004 decreased by a factor of 0.8 and a mini-batch size of 64 images generating a 256-dimensional feature vector.

4 Datasets and Experimental Framework

The performance of the aforementioned algorithms is assessed on three publicly available isolated-sign GSL datasets: the *Polytropon GSL corpus* [12], the *ITI GSL dataset* [37], and the *Dicta-Sign database* [38]. These corpora exhibit significant differences among them, concerning the acted task and the recorded subjects, thus offering a desirable variation in the vocabulary content. More details follow.

Polytropon GSL corpus [12]: This contains 3 repetitions of 3,600 sentences performed by a single signer, recorded by two frontal-view cameras, a Kinect and an RGB one. Here, the RGB video data are used, which are provided at a frame-rate of 25 Hz and 848 × 480-pixel resolution. Corpus annotations based on ELAN [1, 7] are available at both the signed sentence and signed word level. The corpus signed vocabulary includes 2,664 unique words corresponding to proper nouns, adverbs, and verbs characterized by variability in sign duration. In this study, we explore an isolated sign small-vocabulary that involves words with a sufficient number of occurrences appearing between 30 to 110 times (52.6 on average). These yield 5,414 video snippets, which are obtained by "cutting" the longer video database files based on the ELAN annotation time-stamps of the words of interest.

ITI GSL dataset [37]: This includes 5x3 different dialogues organized in sets of 5 individual tasks in 3 public services, performed by 7 different signers. The dialogues, which appertain to a communication between a deaf person and a single service employee, are pre-defined and are performed by each signer 5 consecutive times (5x7x5x3). Signing is captured by an Intel RealSense D435 RGB+D camera at a rate of 30 Hz, providing simultaneously RGB and depth streams (bit depth:24) at the same spatial resolution of 648×480 pixels per frame. During recording, camera poses adjustments are made, offering a desirable variation in the videos. Corpus annotations accomplished through GSL linguistic experts are provided at both the signed sentence and signed word levels. The corpus signed vocabulary consists of 310 unique glosses (40,785 gloss instances) and 331 unique sentences (10,290 sentences), with 4.23 glosses per sentence on average. Here, an isolated sign recognition task is built for 305 unique words that appear between 4 and 10 times by each signer in the dataset, yielding 12,897 video snippets in total.

Dicta-Sign dataset [38]: This is a multilingual corpus on the domain "Travel across Europe" in four sign languages (including GSL), concerning communication for transport by different means and contexts as well as related personal experiences. The corpus comprises 10 different tasks with a session duration of approximately 2 hours on the same elicitation material, covering various interaction formats from monologues to sequences of very short turns, also with different levels of predictability. The data are recorded by seven cameras, two of them stereo cameras [56], capturing signing from different view-points (front, side, footage and bird's eye view). GSL data are expressed by 8 pairs of different signers (16 signers in total) consisting of 8 to 10 hours of signing. Corpus annotations are based on iLex export format [57] as well as ELAN [57, 58] and are provided at two different levels (signed sentence and signed word), containing labels and time-stamps. The corpus signed vocabulary consists of 1704 unique words. Here, we employ an isolated small-vocabulary subset of 152 unique words with a sufficient number of occurrences among the 7 signers between 4 and 20 times. These yield 5,959 video snippets of words obtained by "cutting" the longer video database files based on the ELAN annotation time-stamps of the words of interest.

The signing duration statistics of all three isolated sign GSL datasets vary significantly, as also depicted in Figure 6. Since the multi-signer datasets (i.e., excluding the single-signer Polytropon dataset) contain more than 4 recordings for each sign and every signer, experiments are performed in a multi-signer framework in an effort to retain a balance between the sets. In addition, all experiments on the three corpora are conducted using ten-fold cross-validation, where 80% of each fold is allocated to training, 10% to validation, and 10% to testing.

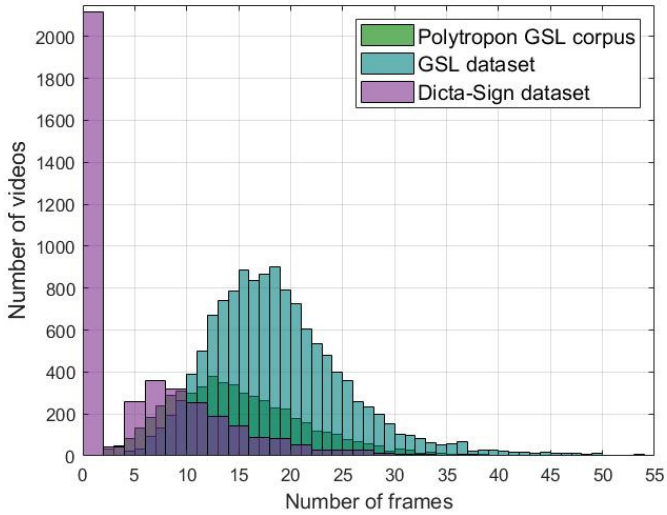


Figure 6: Duration histograms (in video frames) of signed words in all three isolated-sign GSL datasets considered in this deliverable.

5 Experimental Results

5.1 Implementation Details and SL Classifier

The extracted articulators regions (hands and mouth) from the first component of our system before being fed to the feature learner are resized to a fixed size of 227 x 227 pixels for system consistency. Additionally, in order to provide a fair comparison between the hand-crafted features and deep learning-based ones, all feature vectors had dimensionalities of the same order, ranging between 128 and 512.

Since there is both spatial and temporal content to be considered, the task of isolated sign recognition from SL videos is addressed by a CRNN model based on a long short-term memory (LSTM) network [60]. In the typical form, such model is a pair of a CNN encoder and an LSTM decoder complemented with a final fully-connected neural network for the prediction task. Specifically, the encoder receives latent-representation sequential data and outputs a sequence of hidden states, while the decoder maps the latter to the desired output (sign IDs) through a final fully-connected network. In this work, the CNN encoder is a pre-trained ResNet-152 model [53] using the ImageNet dataset [43]. For decoding, a one-layer LSTM decoder is used with hidden dimensionality equal to 128. Training is conducted employing the Adam optimizer with an initial learning rate of 0.001 decreased by a factor of 0.3. We also use dropout at a rate of 0.3.

Evaluation experiments were carried out on an Nvidia GTX 1050 Ti GPU. The hand extraction and type classification scheme as well as all feature learners except for optical flow estimation and CNN model implementations were deployed in the Matlab environment, while all others were implemented in PyTorch [61]. It should be noted that OpenPose was run on a system with Nvidia Tesla K80 and Cuda 10.1 toolkit.

5.2 Task T1.1: Evaluation of Articulator Detectors

The first set of experiments concerns the detection and tracking accuracy and computational efficiency of the various SL articulators of interest, namely the hands and lips, employing both approaches described in Section 2. For detection and tracking performance evaluation, we used 7 video clips from the Polytropon GSL corpus including 106 frames in total, 7 videos from the ITI GSL dataset comprising 96 image frames (13.71 frames per signer), and 16 video snippets of the Dicta-Sign dataset involving in total 156 image frames (9.75 frames per signer). Body parts like handshapes and lips for all 358 image frames were manually labeled for this evaluation.

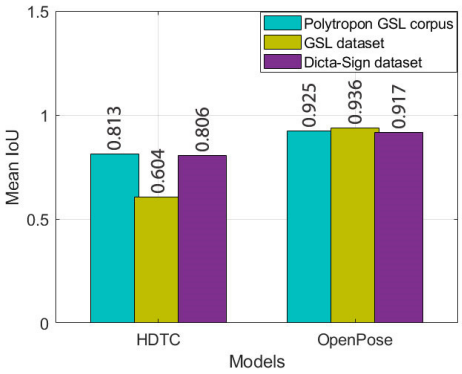
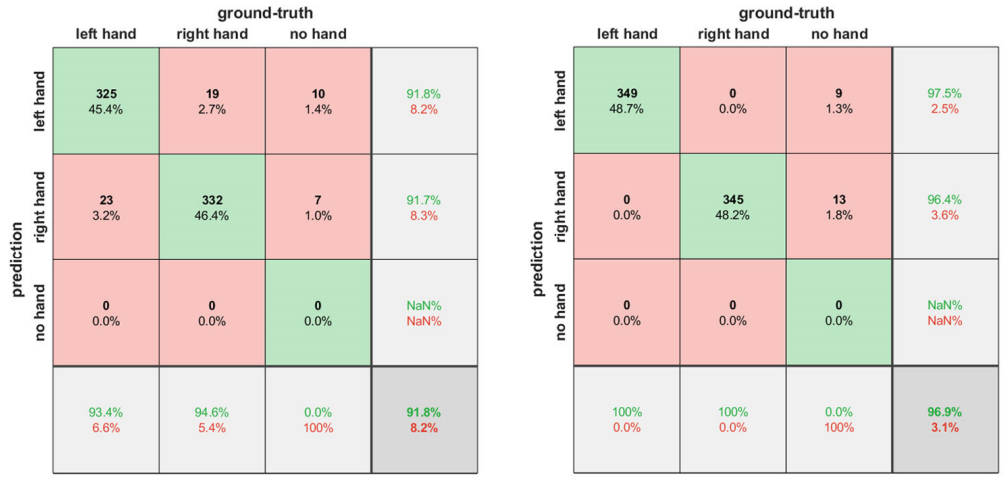


Figure 7: Accuracy comparison of the hand detection and type classification (HDTC) scheme against the OpenPose framework on all three evaluation sets in terms of mean intersection over union (IoU).

Accuracy results are provided in Figure 7 in terms of the mean intersection-over-union (mean IoU) [62], a standard metric to measure the overlap ratio between the ground-truth and predicted bounding boxes. The OpenPose detector reaches the highest IoUs in all three evaluation datasets, ranging between 0.917 and 0.936. The lowest performance is achieved on the GSL dataset (0.604) through the hand extraction and type classification (HDTC) scheme, probably due to the video quality that is rather low, the variability of standing posture of the signer, as well as the signing variability among subjects. The OpenPose framework seems to outperform the considered alternative on all three datasets, demonstrating the robustness of this deep-learning approach on signer skeleton detection and tracking, providing low risk of false detections and missing hand candidate regions. The implementation of the hand extraction and type classification (HDTC) method takes 0.48 sec per frame on average using Matlab, while OpenPose requires less time (0.22 sec per frame).



(a) **(b)**

Figure 8: Confusion matrices of the (a) hand detection and type classification scheme against (b) the OpenPose framework on all three evaluation sets (358 frames in total).

In Figure 8, the performance of the evaluated systems is assessed by providing the confusion matrices over all evaluation frames. As already mentioned, there are 358 frames in total where left and/or right hands participate in the signing process. In its computation, only detections with IoUs higher than 0.5 are considered correct. Note also that the pretrained CNN of Section 2.1 is used for the hand extraction and type classification (HDTC) scheme, while for the OpenPose framework the hand type is automatically provided through skeletal data. Clearly, the OpenPose model performs better than the HDTC scheme, with 96.9% of its classifications being correct and only 3.1% wrong. Among the 716 detections, there are 9 cases where no hand participating is wrongly detected and classified as left, and 13 cases for the right hand one. On the other hand, the HDTC model demonstrates lower accuracy performance with 91.8% correct classifications and 8.2% wrong. Here, there are 42 misclassifications between hand types, with 23 left hands being classified as right hands and 19 the other way. There are also 10 cases where no left hand participating is wrongly detected, and 7 cases for the right hand. It is apparent that OpenPose constitutes the best hand detector and tracker between the two evaluation systems, due to its computational efficiency and detection accuracy. For that reason, we provide the handshape and mouth regions generated by OpenPose as input to the visual feature extractors evaluated next.

5.3 Task T1.2: Evaluation of Visual Features for GSL

The performance of our GSL recognition system concerning all investigated feature learners of Section 3 is reported in Table 1 on all three isolated sign GSL datasets with respect to sign classification accuracy (%). Specifically, assuming that F denotes the latent representation generated by the feature learner corresponding to each articulator participating in the signing process, a 3 x F dimensional feature vector (for the two hands and mouth) is fed to the LSTM classifier for the sign classification task. As already mentioned, for articulator region-of-interest extraction, the OpenPose framework is used due to its efficacy in skeleton localization and tracking. As demonstrated in Table 1, the best results are achieved by the CNN (ResNet-18 network [53]), while the worst by the PCA and LBPs regarding the three datasets. Further, there is a wide range in classification accuracy between the feature learners, with CNN outperforming PCA and LBPs by almost 35%. It can be observed that the higher accuracy is achieved in the case of the ITI GSL dataset, which may be due to its larger size compared to the other two. It is also interesting to note that performance is relatively consistent across the three datasets.

Visual features	Polytropon	ITI GSL	Dicta-Sign
PCA	51.46	50.23	50.57
Gabor filter-bank	53.27	52.86	51.20
LBPs	51.62	50.01	49.97
SIFT	86.64	86.99	85.28
HOG	85.81	86.22	84.73
Optical flow	87.15	87.84	85.05
CNN	**88.24**	**89.05**	**86.15**
Vanilla AE	87.03	87.49	85.87

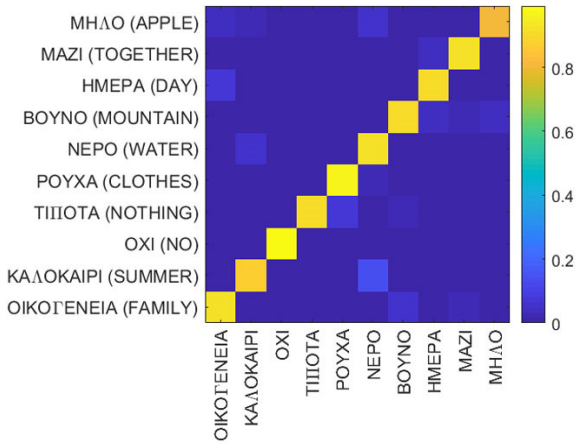
Table 1: Sign classification accuracy (%) of the isolated GSL recognition system on the three datasets of Section 4, using the various visual features of Section 3 in conjunction with an LSTM classifier.

Next, in Table 2, we investigate the running speed (sec per frame) of the various visual features. It can be observed that the lowest speed is obtained by the Gabor filters (0.65 sec per frame), which is primarily due to the filter-bank generation component, while the highest is achieved by the LBPs requiring less time than the others (0.0060 sec per frame).

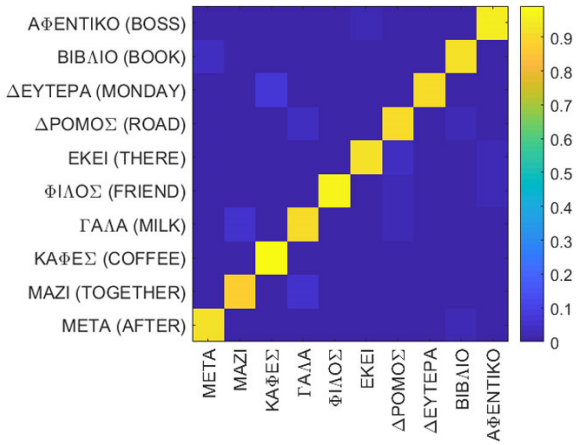
Feature	PCA	Gabor	LBPs	SIFT	HOG	Optical fl.	CNN	AE
Running time	0.0373	0.6469	0.0060	0.0099	0.0224	0.4612	0.0114	0.1304

Table 2: The required running time per frame of the various visual features of Section 3.

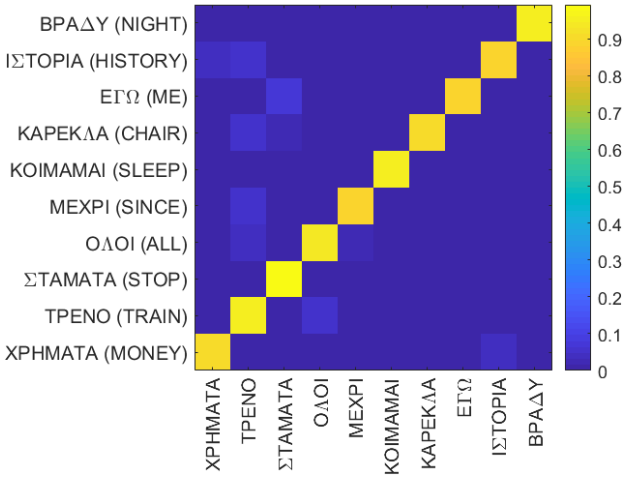
Finally, in Figure 9, we visualize the confusion matrix for a subset of ten words selected at random from all three datasets. The bright yellow diagonal demonstrates the successful classification achieved. As it can be observed, the alignment between prediction and ground truth is generally monotonic, with the ITI GSL dataset achieving the best results. Among the confusable pairs of these matrices, most of the misclassifications involve videos where the signing handshapes look very similar and their positioning (and track) do not coincide. This is expected, as the system investigated in this deliverable only encodes handshapes (and lips) ignoring the articulators tracks. It should also be noted that including only handshape features in the input feature vector (i.e., disregarding the non-manual lip information) achieves 0.673% less accuracy on average on all three GSL datasets.



(a)



(b)



(c)

Figure 9: Confusion matrices of a subset of ten words selected at random from all three datasets, namely (a) the Polytropon corpus, (b) the ITI GSL dataset, and (c) the Dicta-Sign database.

6 Conclusions

In this deliverable, we presented the SL-ReDu project initial work concerning the detection and tracking of the visual manual and non-manual articulators, as well as the extraction of the corresponding visual features, which constitutes the first component for SL recognition (WP1). Specifically, we investigated the detection and tracking of various SL articulators of interest using traditional techniques, as well as deep learning-based approaches. Additionally, regarding visual feature extraction, traditional shape, appearance, and motion-based features, as well as more advanced ones based on deep learning representations were explored. Our results on three isolated sign GSL datasets demonstrated that OpenPose is the most accurate framework for SL articulator localization and tracking, whilst CNN-based feature representations of manual and non-manual articulators turn out superior to the considered alternatives. This deliverable paves the way for D2.1 (M12) on and will be updated in D1.2 (M16).

References

[1] M. De Meulder, *The Power of Language Policy: The Legal Recognition of Sign Languages and the Aspirations of Deaf Communities*, Ph.D. Thesis, Faculty of Humanities, Univ. of Jyväskylä, 2016.

[2] G. Potamianos, K. Papadimitriou, E. Efthimiou, S. Fotinea, G. Sapountzaki, and P. Maragos, "SL-ReDu: Greek sign language recognition for educational applications. Project description and early results," in *Proc. of the Pervasive Technologies Related to Assistive Environments*, 2020.

[3] M. Brennan, "Word order: Introducing the issues", In: M. Brennan and G. Turner (Eds.), *Word-Order Issues in Sign Language: Working Papers*, pp. 9–46, Int. Sign Linguistics Assoc., 1994.

[4] S. S. Rautaray and A. Agrawal, "Vision based hand gesture recognition for human computer interaction: a survey," *Artificial Intelligence Review*, vol. 43, no. 1, pp. 1–54, 2015.

[5] S. Bambach, S. Lee, D. J. Crandall, and C. Yu, "Lending a hand: Detecting hands and recognizing activities in complex egocentric interactions," in *Proc. ICCV*, pp. 1949–1957, 2015.

[6] J. R. Pansare, S. H. Gawande, and M. Ingle, "Real-time static hand gesture recognition for American Sign Language (ASL) in complex background," *Journal of Signal and Information Processing*, vol. 3, no. 3, pp. 364–367, 2012.

[7] P. Heracleous, D. Beautemps, and N. Aboutabit, "Cued speech automatic recognition in normal-hearing and deaf subjects," *Speech Communication*, vol. 52, no. 6, pp. 504–512, 2010.

[8] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proc. CVPR*, vol. 1, pp. 511–518, 2001.

[9] "OpenCV: Open Source Computer Vision Library" [Online]: https://opencv.org/

[10] G. Awad, J. Han, and A. Sutherland, "A unified system for segmentation and tracking of face and hands in sign language recognition," in *Proc. ICPR*, vol. 1, pp. 239–242, 2006.

[11] K. Papadimitriou and G. Potamianos, "A hybrid approach for hand detection and classification in upper-body videos," in *Proc. EUVIP*, 2018.

[12] R. Bowden, D. Windridge, T. Kadir, A. Zisserman, and M. Brady, "A linguistic feature vector for the visual interpretation of sign language," in *Proc. ECCV*, 2004.

[13] R. Yang and S. Sarkar, "Detecting coarticulation in sign language using conditional random fields," in *Proc. ICPR*, 2006.

[14] M. Zahedi, P. Dreuw, D. Rybach, T. Deselaers, and H. Ney, "Geometric features for improving continuous appearance-based sign language recognition," in *Proc. BMVC*, 2006.

[15] M. M. Zaki, S. I. Shaheen, "Sign language recognition using a combination of new vision-based features," *Pattern Recognition Letters*, 32(4):572–577, 2011.

[16] A. Farhadi, D. Forsyth, and R. White, "Transfer learning in sign language," in *Proc. CVPR*, 2007.

[17] P. Dreuw, D. Rybach, T. Deselaers, M. Zahedi, and H. Ney, "Speech recognition techniques for a sign language recognition system," in *Proc. Interspeech*, 2007.

[18] S. Nayak, S. Sarkar, and B. Loeding, "Automated extraction of signs from continuous sign language sentences using iterated conditional modes," in *Proc. CVPR*, 2009.

[19] P. Buehler, M. Everingham, D. P. Huttenlocher, and A. Zisserman, "Upper body detection and tracking in extended signing sequences," *International Journal of Computer Vision*, 95(2): 180–197, 2011.

[20] A. Fossati, J. Gall, H. Grabner, X. Ren, and K. Konolige (Eds.), *Consumer Depth Cameras for Computer Vision, Research Topics and Applications*, Springer-Verlag, 2013

[21] I. Tashev, "Kinect development kit: A toolkit for gesture- and speech-based human-machine interaction", *IEEE Signal Process. Mag.*, 30(5): 129–131, 2013.

[22] "Intel RealSense Cameras" [Online]: https://realsense.intel.com/

[23] Z. Ren, J. Yuan, and Z. Zhang, "Robust hand gesture recognition based on finger-earth movers distance with a commodity depth camera," in *Proc. Multimedia*, pp. 1093–1096, 2011.

[24] A. Kurakin, Z. Zhang, and Z. Liu, "A real time system for dynamic hand gesture recognition with a depth sensor," in *Proc. EUSIPCO*, pp. 1975–1979, 2012.

[25] S. Escalera, J. Gonzalez, X. Baro, M. Reyes, I. Guyon, V. Athitsos, H. Escalante, L. Sigal, A. Argyros, C. Sminchisescu, R. Bowden, and S. Sclaroff, "ChaLearn multi-modal gesture recognition 2013: Grand challenge and workshop summary," in *Proc. ICMI*, pp. 365–368, 2013.

[26] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, 521: 436–444, 2015.

[27] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. CVPR*, pp. 580–587, 2014.

[28] R. Girshick, "Fast R-CNN," in *Proc. ICCV*, pp. 1440–1448, 2015.

[29] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards realtime object detection with region proposal networks," in *Proc. NIPS*, pp. 91–99, 2015.

[30] K. He, G. Gkioxari, P. Dollar, and R. Girshick, "Mask R-CNN," *CoRR*, abs/1703.06870, 2017.

[31] J. Huang, W. Zhou, Q. Zhang, H. Li, and W. Li, "Video based sign language recognition without temporal segmentation," *CoRR*, abs/1801.10111, 2018.

[32] "OpenFace" [Online]: https://cmusatyalab.github.io/openface/

[33] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh, "Realtime multi-person 2D pose estimation using part affinity fields," in *Proc. CVPR*, pp. 1302-1310, 2017.

[34] D. Konstantinidis, K. Dimitropoulos, and P. Daras, "A deep learning approach for analyzing video and skeletal features in sign language recognition," in *Proc. IEEE International Conference on Imaging Systems and Techniques*, pp. 1–6, 2018.

[35] F. Nugraha and E. C. Djamal, "Video recognition of American sign language using two-stream convolution neural networks," in *Proc. ICEEI*, pp. 400–405, 2019.

[36] S. Ko, J. Son, and H. Jung, "Sign language recognition with recurrent neural network using human keypoint detection," in *Proc. Conference on Research in Adaptive and Convergent Systems*, pp. 326–328, 2018.

[37] N. Adaloglou, T. Chatzis, I. Papastratis, A. Stergioulas, G. Papadopoulos, V. Zacharopoulou, G. Xydopoulos, K. Atzakas, D. Papazachariou, and P. Daras, "A Comprehensive Study on Sign Language Recognition Methods," *IEEE Transactions on Multimedia*, 2019.

[38] S. Matthes, T. Hanke, A. Regen, J. Storz, S. Worseck, E. Efthimiou, A.-L. Dimou, A. Braffort, J. Glauert, and E. Safar, "Dicta-Sign – Building a multilingual sign language corpus," in *Proc. Wksp. Repres. Proces. Sign Lang.: Inter. Between Corpus and Lexicon (Satellite to LREC),* 2012.

[39] K. B. Shaik, P. Ganesan, V. Kalist, B. Sathish, and J. M. M. Jenitha, "Comparative study of skin color detection and segmentation in HSV and YCbCr color space," *Procedia Computer Science*, vol. 57, pp. 41 – 48, 2015.

[40] J. Jeong, T. Yoon, and J. Park, "Kalman filter based multiple objects detection-tracking algorithm robust to occlusion," in *Proc. SICE Annual Conference*, pp. 941–946, 2014.

[41] N. Otsu. A threshold selection method from gray-level histogram, *IEEE Transactions on Systems, Man and Cybernetics*, 9(1):62–66, 1979.

[42] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems* (NIPS) 25, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds. Curran Associates, Inc., pp. 1097–1105, 2012.

[43] J. Deng, W. Dong, R. Socher, L. J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. CVPR*, 2009.

[44] A. Mittal, A. Zisserman, and P. H. S. Torr, "Hand detection using multiple proposals," in *Proc. BMVC*, 2011.

[45] K. Simonyan and A. Zisserman, "Very deep convolutional net-works for large-scale image recognition," in *Proc. ICLR*, 2015.

[46] I. T. Jolliffe, "Principal Component Analysis," *International Encyclopedia of Statistical Science,* (2011).

[47] H. Li, T. Celik, N. Longbotham, and W. Emery, "Gabor Feature Based Unsupervised Change Detection of Multitemporal SAR Images Based on Two-Level Clustering", *IEEE Geoscience and Remote Sensing Letters*, vol. 12, pp 1-5, 2015.

[48] C. Silva, T. Bouwmans, and C. Frelicot, "An eXtended Center-Symmetric Local Binary Pattern for Background Modeling and Subtraction in Videos," in *Proc. VISAPP*, 2015.

[49] D. G. Lowe, "Object recognition from local scale-invariant features," In *Proc. ICCV,* pp. 1150-1157, 1999.

[50] L.J.P. van der Maaten, "**Accelerating t-SNE using Tree-Based Algorithms**," *Journal of Machine Learning Research, vol.* 15, pp. 3221-3245, 2014.

[51] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," In *Proc. CVPR*, pp. 886–893, 2005.

[52] A. Ranjan and M. J. Black, "Optical flow estimation using a spatial pyramid network," in *Proc. CVPR*, pp. 2720–2729, 2017.

[53] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. CVPR*, pp. 770–778, 2016.

[54] P. Baldi, "Autoencoders, unsupervised learning and deep architectures," in *Proc. International Conference on Unsupervised and Transfer Learning Workshop*, pp. 37–50, 2011.

[55] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proc. International Conference on Artificial Intelligence and Statistics*, pp. 249–256, 2010.

[56] T. Hanke, L. König, S. Wagner, and S. Matthes, *DGS Corpus & Dicta-Sign: The Hamburg Studio Setup,* In: P. Dreuw et al. (Eds.): LREC 2010. 7th International Conference on Language Resources and Evaluation. Workshop Proceedings. W13. 4th Workshop on Representation and Processing of Sign Languages: Corpora and Sign Language Technologies, ELRA, pp. 106-109, 2010.

[57] T. Hanke, iLex - A tool for sign language lexicography and corpus analysis, in *Proc. International Conference on Language Resources and Evaluation*, 2002.

[58] O. Crasborn and H. Sloetjes, "Enhanced ELAN functionality for sign language corpora," in *Proc. Workshop on the Representation and Processing of Sign Languages: Construction and Exploitation of Sign Language Corpora*, pp. 39–43, 2008.

[59] 2019. ELAN (Version 5.8) [Computer software]. Nijmegen: Max Planck Institute for Psycholinguistics, The Language Archive. https://archive.mpi.nl/tla/elan.

[60] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computing*, vol. 9, no. 8, pp. 1735–1780, 1997.

[61] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in PyTorch," in *Proc. NIPS-W*, 2017.

[62] M. A. Rahman and Y. Wang, "Optimizing Intersection-Over-Union in deep neural networks for image segmentation," in *Proc. ISVC*, pp.234–244, 2016.

[63] S. Theodorakis, V. Pitsikalis, and P. Maragos, "Dynamic–static unsupervised sequentiality, statistical subunits and lexicon for sign language recognition," *Image and Vision Computing*, vol. 32, no. 8, pp. 533-549, 2014.

[64] V. Pitsikalis, S. Theodorakis, C. Vogler, and P. Maragos, "Advances in phonetics-based sub-unit modeling for transcription alignment and sign language recognition," in *Proc. CVPR-W*, pp. 1-6, 2011.

[65] A. Roussos, S. Theodorakis, V. Pitsikalis, and P. Maragos, "Dynamic Affine-Invariant Shape-Appearance Handshape Features and Classification in Sign Language Videos," *Journal of Machine Learning Research*, vol. 14, no. 15, pp. 1627-1663, 2013.