Sinhala Fingerspelling Sign Language Recognition with Computer Vision

Amal A. Weerasooriya
Department of Computer Science and Engineering
University of Moratuwa
Moratuwa, Sri Lanka
amal.20@cse.mrt.ac.lk

Thanuja D. Ambegoda
Department of Computer Science and Engineering
University of Moratuwa
Moratuwa, Sri Lanka
thanujaa@uom.lk

2022 Moratuwa Engineering Research Conference (MERCon) | 978-1-6654-8786-3/22/$31.00 ©2022 IEEE | DOI: 10.1109/MERCon55799.2022.9906281

***Abstract*—Computer vision based sign language translation is usually based on using thousands of images or video sequences for model training. This is not an issue in the case of widely used languages such as American Sign Language. However, in case of languages with low resources such as Sinhala Sign Language, it's challenging to use similar methods for developing translators since there are no known data sets available for such studies.**

In this study we have contributed a new dataset and developed a sign language translation method for the Sinhala Fingerspelling Alphabet. Our approach for recognizing fingerspelling signs involve decoupling pose classification from pose estimation and using postural synergies to reduce the dimensionality of features. As shown by our experiments, our method can achieve an average accuracy of over 87%. The size of the data set used is less than 12% of the size of data sets used in methods which have comparable accuracy. We have made the source code and the dataset publicly available.

Keywords—sign language recognition, Sinhala, fingerspelling, finger pose estimation

I. INTRODUCTION

Sign language is a system of communication that uses visual gestures and signs. Sign languages consist of three main parts which are manual features consisting of gestures made with the hands, non-manual features such as facial expressions, and fingerspelling [1].

Fingerspelling is a method of spelling words using hand movements [2]. Fingerspelling signs can be divided into two types, one being dynamic signs which are defined by a sequence of poses and the other being static signs which are defined by a single pose which doesn't vary with the time. The Fingerspelling Alphabet of Sinhala Sign Language (FASSL) has signs for vowels and consonants of Sinhala Language. Some of these signs are static, while others are dynamic. Some signs used in the FASSL are shown in Fig. 1.

Sign Language is used for communication with and among deaf and mute people in Sri Lanka as the preferred language [3] [4]. There is a Sinhala Sign Language (SSL) as well as a Tamil Sign Language. Further, there are several dialects adopted by different teaching institutes in different areas of the country [5].

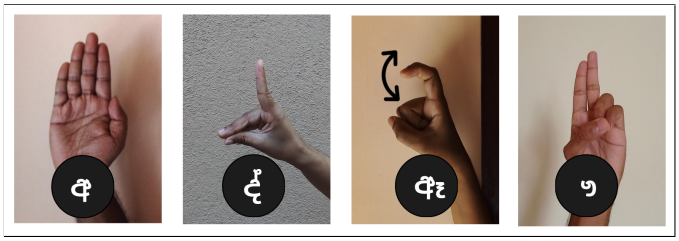

Fig. 1: Some signs from Fingerspelling Alphabet of Sinhala Sign Language

The majority of the Sri Lankan population don't understand SSL [4]. There have been some attempts in developing systems to capture the hand and body movements of SSL and translate them to Sinhala Language. Fernando et al. [4] have developed a system which can translate 15 words. Although it has a good accuracy for the selected words, fingerspelling is not included. Further, there has been no known study carried out to develop a translator for the FASSL.

Many studies have been done for classification of widely used sign languages such as American Sign Language (ASL). Hence there are datasets containing thousands of data points for such languages. However, in the case of FASSL there is no such known dataset.

In recent years, there have been significant improvements in the field of human body pose estimation using images as inputs. Such pose estimations can be used in various applications including sign language translation.

This paper presents a computer vision based translator for static signs of FASSL utilizing hand pose estimation techniques. Our contributions are as follows:

- A labelled dataset[1] which can be used for developing a classifier for FASSL.
- A hand pose based classification method for fingerspelling (static signs) which can be trained using a substantially smaller dataset compared to the existing sign language translators.
- An end-to-end real time translator using the above classifier which runs on general purpose computers such as laptops.

[1]The dataset is available at https://github.com/aawgit/signs.

II. PREVIOUS WORK

A widely used approach in the domain of vision based sign language translation is extracting features from images and classifying using Artificial Neural Networks (ANN) such as Convolutional Neural Networks (CNN). An alternative approach is deriving the underlying hand/ body pose as a skeleton model and recognizing signs based on features extracted from the said model. We are adopting a model based approach for our study and the rest of the literature review is focused on main steps involved in such an approach, which are hand detection & tracking, pose estimation, and pose classification.

As the first step of the classification pipeline, the position of the hand i.e. the Region of Interest (RoI) has to be determined. This is done based on skin color in [6], [7] and [8]. Alternatively in [9] and [10] a CNN based approach is used. In case of video data, the RoI has to be determined for each video frame. Instead of running a hand detection process for each frame, in some studies a tracking method is employed once a hand is detected [6] [9] [10].

Pose estimation step takes an image or video frame as the input and returns an estimation of the skeleton in 2D or 3D space. Both hand pose and body pose literature were reviewed since findings of the latter can be applied to the former. In [11] there are multiple stages, at each of which a classifier predicts confidences for locations of each anatomical landmark. A classifier in a certain stage takes features of the image data and the result of the previous classifier as inputs. At each stage a Random Forest is used. In [12] a 2.5D pose is generated from a RGB image using a CNN, a 3D pose is derived from it using camera parameters. In [13] a 2D pose is estimated using a CNN and then a 3D pose is generated as an inverse kinematics problem.

Pose classification is identifying the pose represented by coordinates of the skeleton joints. Zhang et al [10] have employed a simple algorithm where the state of each finger, e.g. bent or straight, is mapped to a set of predefined gestures. When there are dynamic gestures, temporal features also need to be taken into account.

Temporal data is captured using CNNs in [14], and [15] by converting a series of poses into a 2D image where one axis represent the time while the other represents the location information of the joins in the skeleton model. Liu et al in [16] use a Spatio-Temporal LSTM (ST-LSTM) model which simultaneously models the spatial and temporal information. Each unit of the ST-LSTM corresponds to one of the joints in the skeleton model. Each of these units receives information on its neighbouring unit and previous state of itself. In [17] poses of each body part such as an arm or leg are indexed. A body pose is represented with five indices corresponding to 5 body parts. A histogram of the poses is used as features for the classification using SVMs.

III. METHODOLOGY

The proposed system first generates a 3D hand pose from a video/ image input and then classifies the pose to recognize the signs. The input video/ image is generated from a single view. Depth information or parameters of the camera are not used. The system consists of the following 3 main steps.

1) Hand detection, tracking and pose estimation
2) Pre-processing
3) Pose classification

The high level architecture of the system is presented in Fig 2.

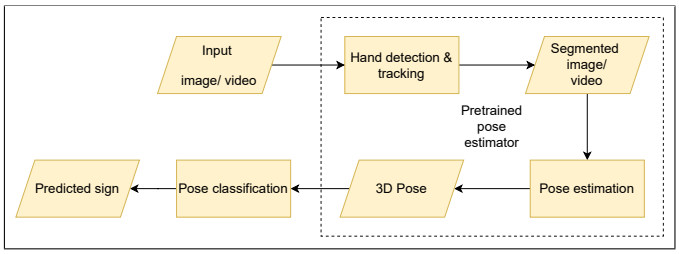

Fig. 2: High level architecture of the proposed system

This modular approach reduces the complexity of the pose classification in comparison to training a pose classifier directly on images, thus making it possible to train the system on a smaller dataset in comparison to the latter approach. Further, this approach utilizes the existing methods and pre-trained models for up to and including the pose estimation step.

A. Hand detection, tracking and pose estimation

There are multiple pre-trained hand pose estimators available. Considering the accuracy, speed, availability, and effort to set up & integrate, we selected the estimator in [10]. It returns the pose as a set of 3D coordinates of 21 points which is termed the hand landmark. The estimator uses a CNN model for pose estimation, and there is an integrated hand detecting & tracking component which also uses a CNN model. There is a pretrained model of the estimator available as a Python module. Details of the selected estimators implementation are shown in the Table I [18].

TABLE I. DETAILS OF THE SELECTED POSE ESTIMATOR

Criteria	**Value**	**Description**
Mean 3D error	1.4 cm	Mean Absolute Error in Euclidean 3D metric space
Speed	48 - 120 FPS	On laptop/ desktop computers

B. Features

The output of the pose estimator is the locations of the skeletal joints of the hand in the form of 3D cartesian coordinates. Useful features for classification were derived using the said coordinates. Identifying important features and reduction of dimensions are normally done using methods such as PCA. However, as the aim of this study is to develop a classifier using a small dataset, a method such as PCA is not applicable since the success of such depends heavily on the availability of a large dataset. Therefore an alternative approach was adopted.

The human hand has a degree of freedom (DoF) of 27 [18]. However, studies have shown that a hand pose can be represented with a model of lower DoF because movements of some limbs are correlated with each other [18], [19], [20]. This is termed as postural synergies. In the referred studies this has been analyzed using PCA and most important features describing a pose have been identified.

In [19] Cobos et al show that one extension/flexion angle of all fingers and adduction/ abduction angle of thumb and index finger are important features for gesture reconstruction. Based on findings of the said studies we developed our baseline classifier using the Metacarpophalangeal (MCP) joint angles of all fingers, Proximal interphalangeal (PIP) joint angles of fingers except for thumb, Trapeziometacarpal (TMC) joint angle of thumb, and adduction/ abduction angle of thumb and index finger as features. However, we observed that accuracy marginally improved when Interphalangeal (IP) angle of thumb, and Distal interphalangeal (DIP) angles of other fingers were added. Therefore, all extension/flexion angles of finger joints and adduction/ abduction angle of thumb & index finger were used.

Further, we concatenated the said feature vector with flattened coordinates of non stationary joints, which improved the accuracy. Furthermore, in order to distinguish the signs which only differ in the orientation of the palm, it was taken as a feature.

C. Pre-processing

With respect to the angle based features, no preprocessing was necessary. However, to use coordinates, they had to be normalized in order to remove variations resulting from camera angle, distance between camera and the signer, size of the hand, and the position of the hand in the image frame. Therefore following preprocessing steps ware done:

- Removing movement: The wrist joint of the skeleton model was taken as the origin of the coordinate system.
- Removing rotations around the 3 axes: 2 reference lines were identified for removing rotations. Reference line 1 was selected as the imaginary line which connects the wrist joint to the mean of the bases of index, middle, ring and small fingers, whereas the reference line 2 is the imaginary line which connects the small fingers base to the index fingers base. Rotations were removed in such a way that the resulting hand model's reference line 1 becomes coincident with the y axis and the projection of reference lines 2 on the x, z plane becomes coincident with the x axis.
- Scaling the dimensions to convert to a predefined size.

Although the rotations are removed, they are extracted and fed to the classifier at a later stage in the pipeline. There are limitations in the pose which could not be corrected by preprocessing which are differences in limb size ratios from person to person, personal variations in signs, and the estimators inaccuracies.

D. Pose classification

There are 58 signs in the FASSL and 27 of them are static. In this study, 26 static signs out of the 27 were classified. Symbol ◌a: was omitted because the thumb is hidden between other fingers in the sign, and therefore the estimator fails.

Classification happens at two levels. At the first level, the normalized hand pose is classified using the selected angles and coordinates. Since rotation of the hand is removed during normalization, another level of classification is used which takes the rotation and the prediction of the level 1 classification as features. Architecture of the classifier is shown in Fig. 3.

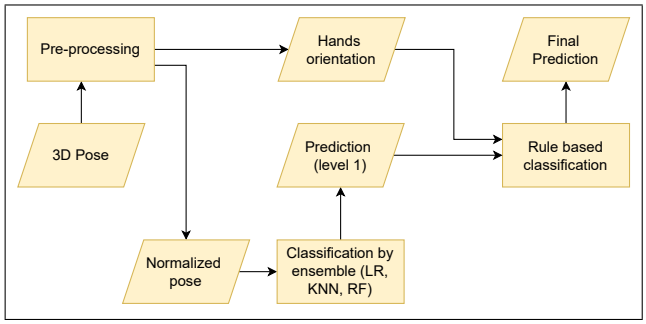

Fig. 3: Architecture of the classifier

Details of the classifiers are as follows.

1) Level 1: Ensemble consisting of following classifiers using majority vote.

- Random Forest (RF) classifier (100 trees, with Entropy criterion)
- K-Nearest Neighbour (KNN) classifier (3 neighbours, weighted by the inverse of distance to neighbours)
- Logistic Regression (LR) classifier (multinomial)

2) Level 2: A rule based classifier which distinguishes signs which only differ in palms orientation. It checks if the prediction is one of the predefined signs, and if so determines the correct prediction based on the palms orientation. There are three pairs of such signs, and they are distinguished based on rotation w.r.t vertical axis. The algorithm of the rule base classification is shown in Algorithm 1.

IV. RESULTS AND DISCUSSION

A. Dataset

The dataset was created using 2 educational videos sourced from YouTube, 3 videos created by sign language teachers, and 4 photo sets created by novices. The videos contain all the signs (58), while photo sets contain only the static signs (27). There is a variety in frames rates, resolutions and backgrounds among sources. In each video a signer performs letters in the alphabet sequentially. Each photo set contains up to 6 images per sign. The dataset was validated before annotating, and incorrect signs were removed. Each letter in the alphabet was given a unique index for annotation and internal functions of the system. The Images have been annotated using the image name against its index whereas videos have been annotated using the start frame number and the end frame number

Algorithm 1: Level 2 (rule based) classifier

input : Output of level 1 classifier, $Sign_{in}$,
Rotation angle w.r.t. Y axis *Angle*
output: Predicted sign, $Sign_{out}$

$Sign_{out} \leftarrow Sign_{in}$
if $Sign_{in}$ = ට or $Sign_{in}$ = ල් **then**
  if *Angle*>45° **then**
    $Sign_{out} \leftarrow$ ල්
  else
    $Sign_{out} \leftarrow$ ට
  end
else
  if $Sign_{in}$ = ද් or $Sign_{in}$ = ප් **then**
    if *Angle*>45° **then**
      $Sign_{out} \leftarrow$ ද්
    else
      $Sign_{out} \leftarrow$ ප්
    end
  else
    if $Sign_{in}$ = හ් or $Sign_{in}$ = ඹ **then**
      if *Angle*>45° **then**
        $Sign_{out} \leftarrow$ ඹ
      else
        $Sign_{out} \leftarrow$ හ්
      end
    else
      end
  end
end
return $Sign_{out}$

against the relevant index. All annotation files are in Comma Separated Value (CSV) format and available with the dataset. 2 image sets and 2 videos were used for training & validation and the rest were used as the test set. The size of the training set is 122 and its distribution is shown in Fig. 4.

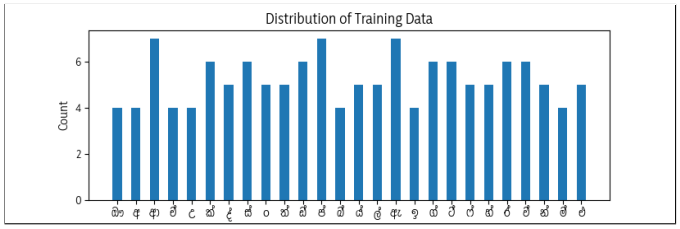

Fig. 4: Distribution of training data by sign

B. Evaluation

The test set consists of 2 photo sets and 3 videos. The 2 image sets had a total of 110 images. All the frames in each of the 3 videos were used for the evaluation. However, since most adjacent frames are almost identical, the results of each video were reduced to 2 per sign. This was done by dividing the duration of a sign into 2 equal parts and taking the mode. Due to slight variations of the pose, movements and noise, pose estimation could vary even in the same video. Hence 2 samples were considered instead of 1. Accuracy and precision were used as the evaluation criteria.

The classifier was evaluated for 2 cases. First with all the available test data and then after removing the samples which gave incorrect pose estimations. In the second case, estimations with distorted limb sizes and wrong angles have still been used as long as it's possible for a human to classify them correctly. Fig. 5 shows an example for a wrong pose estimate.

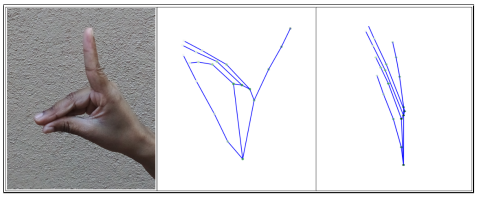

Fig. 5: The sign ද් (left), its correct pose estimate (middle), and a wrong pose estimate (right)

Details of results for different datasets are shown in Table II. Precision and recall values for each class is shown in Fig. 6.

TABLE II. PERFORMANCE OF THE CLASSIFIER ON THE TEST SET

		All data		Without wrong estimates	
Dataset	**Type**	**Accuracy**	**Precision**	**Accuracy**	**Precision**
Subject 5	Video	0.7308	0.6308	0.8636	0.8106
Subject 6	Video	0.7447	0.6875	0.8974	0.8571
Subject 7	Video	0.8043	0.7246	0.8810	0.8413
Subject 8	Photo	0.9111	0.8800	0.9111	0.8800
Subject 9	Photo	0.8519	0.8427	0.8519	0.8427
All		0.8074	0.8479	0.8795	0.9035

Removing estimation errors shows about a 7% increase in the accuracy. When analyzing the classifier independent of the estimation errors it was observed that 22 out of 26 signs are being recognized with a recall value over 75%. Signs එ, උ, and මි have the lowest scores which are 62.50%, 66.67%, and 66.67% respectively. The signs එ, and න් differ from each other mostly only by the position of the thumb, thus making their pose estimations close to each other. Therefore about 20% of the time එ is being misclassified as න් and න් is being misclassified as එ. Likewise, මි is being misclassified as සි and න් due to similarities between them. Although උ and ව් are not very similar to the eye, in the normalized pose their difference are not very prominent in some cases. Therefore 30% of the time, උ is misclassified as ව්. The similarity between මි, සි and න් signs are shown in Fig 7.

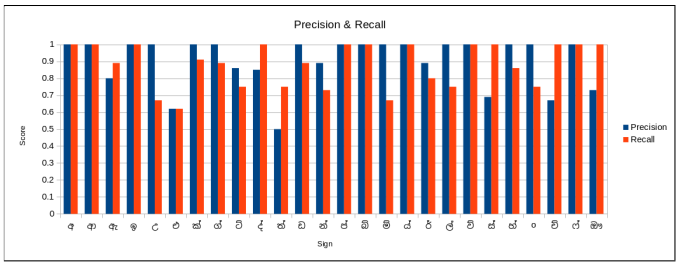

Fig. 6: Precision and Recall for each class

Fig. 7: Similarity between signs ම (left), ස (middle), and න (right)

TABLE III. ABLATION STUDY

Classifier	Features	Accuracy
RF	Angles + Coordinates	0.6964
KNN	Angles + Coordinates	0.6696
LR	Angles + Coordinates	0.7410
RF + Rule Based	Angles + Coordinates + Orientation	0.7902
KNN + Rule Based	Angles + Coordinates + Orientation	0.7723
LR + Rule Based	Angles + Coordinates + Orientation	0.8482
Ensemble + Rule Based	Angles + Orientation	0.8080
Ensemble + Rule Based	Angles + Coordinates + Orientation	0.8795

The results of the ablation study done on the test set is shown in Table III.

We compared the results with some other sign language classification methods. In order to do a fair comparison, methods matching the following criteria were selected:

- Developed for static signs
- Have at least 20 signs
- Use images/videos without depth information
- Accuracy and training dataset size are published

Comparison of accuracy to other method are shown in Table IV.

TABLE IV. COMPARISON TO OTHER FINGERSPELLING CLASSIFIERS

Authors/ year	Training set size	No. of signs	Accuracy	Context
Pugeault et al [21]/ 2011	24000	23	75%	ASL Fingerspelling
Sanalohit et al [22]/2022	2518	30	84.57%	Thai Sign Language Fingerspelling
Wang et al [23]/2020	1050	30	89.48%	Chinese Sign Language Fingerspelling
Rastgoo et al [24]/2018	2524 - 131000	24 - 36	90.1% - 99.31%	ASL Fingerspelling
Ours - all	122	26	80.7%	FASSL
Ours - correct estimates	122	26	87.9%	FASSL

While the compared methods have higher and lower accuracies than the proposed method, it should be noted that all the said methods have been developed using much larger training datasets. In the case of the smallest compared training set, it's still about 9 times larger, and in the case of the largest, it's about 197 times larger than the proposed methods training set. When compared with methods having close accuracies to the proposed method such as [22] 84.57%, [23] (89.48%), it can be seen that the proposed method achieves results in par with them using a training set of size of 4.8% and 11.6% of the compared training sets respectively.

The hardware used for the experimental setup was a computer having a 8 core 1.6 GHz processor, 8 GB of RAM and no GPUs. Average processing speed for pose estimation was 30.7 frames per second (FPS). Average speed for preprocessing and classification was 89.20 FPS. The system was implemented in such a way that pose estimation and classification run parallelly utilizing 2 CPU cores. Therefore the speed of the pose estimation step becomes the speed of the end to end system.

V. CONCLUSION

We present a method for developing a classifier for static signs of FASSL using a small training data comprised of only 122 images. Our key idea is to develop a simpler classification model such that it doesn't need a large data set for training, and yet performs well enough to use in real world applications. The results show that our method achieves an accuracy which is on par with that of the methods which use much larger datasets. Further, it's speed is enough to be used with common video formats for real time applications. In the future we plan to improve this by including dynamic signs of FASSL.

ACKNOWLEDGMENT

We thank Mr. Buddika Gunathilaka, Ms.Samantha Madurangi and Ms. Geshani Amila for their contributions to building the dataset. We also thank Ms. Sunitha Karunaratna of The Ceylon School for the Deaf & Blind, Mr. Brayan Susantha and Mr. Janaka Perera of Sri Lanka Central Federation of the Deaf for connecting us with resource people of their organizations.

REFERENCES

[1] H. Cooper, B. Holt, and R. Bowden, "Sign Language Recognition," in *Visual Analysis of Humans*, Journal Abbreviation: Visual Analysis of Humans, Jan. 2011, pp. 539–562, isbn: 978-0-85729-996-3. doi: 10.1007/978-0-85729-997-0_27.

[2] "Fingerspelling Alphabet - British Sign Language (BSL)." (2021), [Online]. Available: https://www.british-sign.co.uk/fingerspelling-alphabet-charts/ (visited on 01/11/2021).

[3] M. Punchimudiyanse and R. G. N. Meegama, "Computer Interpreter for Translating Written Sinhala to Sinhala Sign Language," *OUSL Journal*, vol. 12, no. 1, p. 70, Jul. 2017, issn: 2550-2816, 1800-3621. doi: 10.4038/ouslj.v12i1.7377.

[4] P. Fernando and P. Wimalaratne, "Sign Language Translation Approach to Sinhalese Language," *GSTF Journal on Computing (JoC)*, vol. 5, no. 1, p. 9, Sep. 2016, issn: 2010-2283. doi: 10.7603/s40601-016-0009-8.

[5] M. Punchimudiyanse and R. G. N. Meegama, "Animation of Fingerspelled Words and Number Signs of the Sinhala Sign Language," *ACM Transactions on Asian and Low-Resource Language Information Processing*, vol. 16, no. 4, pp. 1–26, Sep. 2017, issn: 2375-4699, 2375-4702. doi: 10.1145/3092743.

[6] J. Singha, A. Roy, and R. H. Laskar, "Dynamic hand gesture recognition using vision-based approach for human–computer interaction," *Neural Computing and Applications*, vol. 29, no. 4, pp. 1129–1141, Feb. 2018, issn: 0941-0643, 1433-3058. doi: 10.1007/s00521-016-2525-z.

[7] J. Singha and K. Das, "Indian Sign Language Recognition Using Eigen Value Weighted Euclidean Distance Based Classification Technique," *International Journal of Advanced Computer Science and Applications*, vol. 4, no. 2, 2013, issn: 2158107X, 21565570. doi: 10.14569/IJACSA.2013.040228.

[8] J. Rekha, J. Bhattacharya, and S. Majumder, "Shape, texture and local movement hand gesture features for Indian Sign Language recognition," in *3rd International Conference on Trendz in Information Sciences & Computing (TISC2011)*, Chennai, India: IEEE, Dec. 2011, pp. 30–35, isbn: 978-1-4673-0133-6 978-1-4673-0134-3 978-1-4673-0132-9. doi: 10.1109/TISC.2011.6169079.

[9] M. Zhang, X. Cheng, D. Copeland, *et al.*, "Using Computer Vision to Automate Hand Detection and Tracking of Surgeon Movements in Videos of Open Surgery," *arXiv:2012.06948 [cs]*, Dec. 2020, arXiv: 2012.06948.

[10] F. Zhang, V. Bazarevsky, A. Vakunov, *et al.*, "MediaPipe Hands: On-device Real-time Hand Tracking," *arXiv:2006.10214 [cs]*, Jun. 2020, arXiv: 2006.10214.

[11] V. Ramakrishna, D. Munoz, M. Hebert, J. Andrew Bagnell, and Y. Sheikh, "Pose Machines: Articulated Pose Estimation via Inference Machines," in *Computer Vision – ECCV 2014*, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds., vol. 8690, Series Title: Lecture Notes in Computer Science, Cham: Springer International Publishing, 2014, pp. 33–47, isbn: 978-3-319-10604-5 978-3-319-10605-2. doi: 10.1007/978-3-319-10605-2_3.

[12] U. Iqbal, P. Molchanov, T. Breuel, J. Gall, and J. Kautz, "Hand Pose Estimation via Latent 2.5D Heatmap Regression," in *Computer Vision – ECCV 2018*, V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, Eds., vol. 11215, Series Title: Lecture Notes in Computer Science, Cham: Springer International Publishing, 2018, pp. 125–143, isbn: 978-3-030-01251-9 978-3-030-01252-6. doi: 10.1007/978-3-030-01252-6_8.

[13] P. Panteleris, I. Oikonomidis, and A. Argyros, "Using a single RGB frame for real time 3D hand pose estimation in the wild," *arXiv:1712.03866 [cs]*, Dec. 2017, arXiv: 1712.03866.

[14] F. Baradel, C. Wolf, and J. Mille, "Pose-conditioned Spatio-Temporal Attention for Human Action Recognition," *arXiv:1703.10106 [cs]*, Aug. 2017, arXiv: 1703.10106.

[15] D. C. Luvizon, D. Picard, and H. Tabia, "2D/3D Pose Estimation and Action Recognition Using Multitask Deep Learning," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT, USA: IEEE, Jun. 2018, pp. 5137–5146, isbn: 978-1-5386-6420-9. doi: 10.1109/CVPR.2018.00539.

[16] J. Liu, A. Shahroudy, D. Xu, and G. Wang, "Spatio-Temporal LSTM with Trust Gates for 3D Human Action Recognition," *arXiv:1607.07043 [cs]*, Jul. 2016, arXiv: 1607.07043.

[17] C. Wang, Y. Wang, and A. L. Yuille, "An Approach to Pose-Based Action Recognition," in *2013 IEEE Conference on Computer Vision and Pattern Recognition*, Portland, OR, USA: IEEE, Jun. 2013, pp. 915–922, isbn: 978-0-7695-4989-7. doi: 10.1109/CVPR.2013.123.

[18] M. Santello, M. Flanders, and J. F. Soechting, "Postural Hand Synergies for Tool Use," *The Journal of Neuroscience*, vol. 18, no. 23, pp. 10 105–10 115, Dec. 1998, issn: 0270-6474, 1529-2401. doi: 10.1523/JNEUROSCI.18-23-10105.1998.

[19] S. Cobos, M. Ferre, M. Ángel Sánchez-Urán, J. Ortego, and R. Aracil, "Human hand descriptions and gesture recognition for object manipulation," *Computer Methods in Biomechanics and Biomedical Engineering*, vol. 13, no. 3, pp. 305–317, Jun. 2010, issn: 1025-5842, 1476-8259. doi: 10.1080/10255840903208171.

[20] N. Bhatt and V. Skm, "Posture similarity index: A method to compare hand postures in synergy space," *PeerJ*, vol. 6, e6078, Dec. 2018, issn: 2167-8359. doi: 10.7717/peerj.6078.

[21] N. Pugeault and R. Bowden, "Spelling it out: Real-time ASL fingerspelling recognition," in *2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*, Barcelona, Spain: IEEE, Nov. 2011, pp. 1114–1119, isbn: 978-1-4673-0063-6 978-1-4673-0062-9 978-1-4673-0061-2. doi: 10.1109/ICCVW.2011.6130290.

[22] J. Sanalohit and T. Katanyukul, "TFS Recognition: Investigating MPH]{Thai Finger Spelling Recognition: Investigating MediaPipe Hands Potentials," *arXiv:2201.03170 [cs]*, Jan. 2022, arXiv: 2201.03170.

[23] X. Jiang, B. Hu, S. Chandra Satapathy, S.-H. Wang, and Y.-D. Zhang, "Fingerspelling Identification for Chinese Sign Language via AlexNet-Based Transfer Learning and Adam Optimizer," *Scientific Programming*, vol. 2020, pp. 1–13, May 2020, issn: 1058-9244, 1875-919X. doi: 10.1155/2020/3291426.

[24] R. Rastgoo, K. Kiani, and S. Escalera, "Multi-Modal Deep Hand Sign Language Recognition in Still Images Using Restricted Boltzmann Machine," *Entropy*, vol. 20, no. 11, p. 809, Oct. 2018, issn: 1099-4300. doi: 10.3390/e20110809.