Real-Time Hand Detection using Convolutional Neural Networks for Costa Rican Sign Language Recognition

Juan Zamora-Mora
Escuela de Ingeniería del Software
Universidad Cenfotec
San José, Costa Rica
izamora@ucenfotec.ac.cr

Mario Chacón-Rivas
Escuela de Computación
Instituto Tecnológico de Costa Rica
Cartago, Costa Rica
machacon@itcr.ac.cr

ABSTRACT

Sign language is the natural language for the deaf, something that comes naturally as a form of non-verbal communication between signers, ruled by a set of grammar expressions in constant evolution since the universe of signs represents a small fraction of all words in Spanish. This limitation, combined with the lack of knowledge about sign language by verbal speakers, creates a separation where both parties (signers and non-signers) are unable to efficiently communicate. Such a problem increases under a specific context such as emergency situations, where first-response teams such as EMTs, firefighters or police officers might be unable to properly attend an emergency given that interactions between the involved parties become a barrier for decision making process when time is scarce. Developing a cognitive-capable tool that serves to recognize sign language in a ubiquitous way, is a must to reduce barriers between the deaf and emergency corps under this context.

Hand detection is the first step toward building a Costa Rican sign language (LESCO) recognition framework. Important advances in computing, particularly in the area of deep learning, open a new frontier for object recognition that can be leveraged to build a hand detection module. This study trains the MobileNet V1 convolutional neural network against the EgoHands dataset from Indiana University's UI Computer Vision Lab to determine if the dataset itself is sufficient to detect hands in LESCO videos, from five different signers that wear short-sleeve shirts under complex backgrounds. Those requirements are key to determine the usefulness of the solution. The consulted bibliography indicates that tests are performed with single-color backgrounds and long-sleeve shirts that ease the classification tasks under controlled environments only. The two-step experiment obtained 1) a mean average precision of (ninety-six dot one percent) 96.1% for the EgoHands dataset and 2) a (ninety one percent) 91% average accuracy for hand detection across the five LESCO videos. Despite the high accuracy reported by the tests in this paper, the hand detection module was unable to detect certain hand shapes such as closed fists and open hands pointing perpendicular to the camera lens. This fact suggests that the complex egocentric views as captured in the EgoHands dataset might be insufficient for proper hand detection for Costa Rican sign language.

CCS CONCEPTS

• Human-centered computing • Applied computing • Human computer interaction (HCI) • Computer vision.

KEYWORDS

Sign language recognition, LESCO, hand detection, convolutional neural network, real time, deep learning, machine learning.

1 Introduction

In Costa Rica, sign language is the natural language for the deaf as stated by law number 9049 since 2012 [1]. Costa Rican sign language has evolved since the beginning of the 20th century but without formal research focus since 1991 [3].

Since Woodward [2] and until 2016, there has been a void in academic research that addressed the use of computing to automatically recognize Costa Rican sign language. In 2016, Quesada et al [4], proposed a method for sign recognition using Intel RealSense technologies which require a special binocular camera able to use depth to construct a skeletal model of the hand and face for motion and gesture capture using vector-based primitives. These primitives are non-coplanar vectors used to describe object position in 2D and 3D environments.

This research proposes the use of common monocular vision hardware such as webcams or mobile phone cameras, to capture the signer's hands in real-time under complex backgrounds so that they can be used for feature extraction to build a Costa Rican sign language recognition framework.

Extensive research has been performed in the domain of hand tracking and recognition using multiple techniques; some of them require different assumptions about the data such as probability distributions [5], skin color [6], steady backgrounds or special clothing [7] to facilitate classification tasks.

Because of the nature of real-time object detection, we can safely assume that the signer could be anywhere wearing different clothing establishing a premise for the object recognition

978-1-7281-5436-7/19/$31.00 ©2019 IEEE
DOI 10.1109/CONTIE49246.2019.00042

architecture which cannot afford to use a special attire, gloves or any other type of complex devices to improve hand classification accuracy without affecting its generalization capabilities in multiple settings.

Convolutional neural networks (CNN) have been widely used for real-time object detection since they do not require assumptions about data distributions [5], and they use the convolution layers, layers that use specialized linear operations instead of the traditional matrix multiplications [25] to automatically extract features from the target objects; the hands in a given frame or picture [8] which facilitates the detection model training.

The initial requirements to run an object detection experiment with CNN are: 1) to have an extensive amount of images of the target object to detect, in this case: hand images in different rotations, forms, and shapes, 2) the information about the hand's location (x,y), size (height,width) and class-type (hand) from each picture in the dataset, and 3) the selection of a CNN architecture such as MobileNet, LeNet or ResNet [10] among others to test out the recognition capabilities. Fortunately, for requirements 1 and 2, there are open source tools for hand labeling and public databases with thousands of hands such as the EgoHands dataset [9] that can speed up the testing and classification accuracy improvement process. For requirement 3, there are also pre-trained models under different CNN architectures that can be used to get an initial hands recognition model.

The final results presented in this paper aim to validate the use of the selected dataset, and CNN architecture to be used for hand detection in real-time. Two different tests were executed to validate the effectiveness of the proposed method for hand detection. The first test will evaluate the classification accuracy of the selected architecture over the training and test sets prepared for this research. The second set of tests were executed on a set of videos recorded from members of the Costa Rican deaf community to assess the identification of hands over a video stream.

2 Related Work

There are many ways to perform hand detection, some of them require special hardware such as gloves [26], depth-capable sensors such as the Microsoft Kinect sensor [27] or Leap Motion cameras [28]. Proprietary APIs for such technologies provide useful information for feature extraction and sign classification; however, it increases the cost of production which negatively impacts its commoditization.

A vision-based approach that can use monocular cameras for hand detection, such as the one on mobile phones, tablets and laptops, is a must for a solution that looks for reducing the consumer cost by reusing the existing mobile architectures.

Several computer vision techniques could be used to fulfill this requirement, some of them subject to special conditions that could affect the detection capabilities in complex environments. Differences in such detection methods will be discussed in the following sections.

2.1 Color-Based Detection

Skin color has been used to identify hands on pictures and videos. Nayak et al. [17] used skin-color filtering to obtain color blobs to create relational distributions, where the signer uses long sleeves and the background is easily separable from the hand color. Du et al. [18] uses skin segmentation based on the color and motion likelihood. A different approach is used by Nayak et al. [19] which involves the use of multiple techniques such as background subtraction, skin color filtering and color-blob extraction. The examples shown in [17], [18] and [19] are long sleeved under differentiable background, something that helps the segmentation process but requires the signer to be aware of the environment constraints for proper detection.

2.2 Haar-Based Detection

Haar detection is based on a set of low-level features selected from several rectangular regions across the image which produces a high number of features filtered by Adaboost [21], selecting the best features for classification, making the training phase slow, since this requires a high number of positive and negative examples but with the ability of detecting objects in a real-time fashion. The Haar-Cascades machine learning algorithm was proposed by Paul Viola and Michael Jones in 2001 [20] and has been used for general recognition tasks such as face recognition.

Haar features are manually set, and because of the simplicity of the feature types, it might fail on recognizing certain objects because it is prone to get confused with background elements [22].

2.3 Convolutional-Based Detection

A CNN is a type of neural network architecture with many subsampling layers capable of performing object detection with high-level accuracy [23]. One advantage of CNNs is that it requires no feature engineering, as each layer in the network is able to automatically extract the target object features [25], by encoding low-level attributes in the initial hidden layers such as shapes and edges, to more complex objects such as body parts in the following layers [10].

CNN requires also a great number of samples for training that ranges from hundreds to thousands for each class which might take some time for a dataset to get prepared, due to the fact that labeling each image is usually a manual task performed with a tool able to select the target object on each image. Refer to Figure 1 for an example of a labeled image entry.

Rakowski et al. [10] reported that pre-trained models outperform state-of-the-art algorithms. This points to another key characteristic from this neural network topology; using pre-trained or transferred models can improve detection accuracy [25].

Hand detection is a complex problem that requires a classification algorithm with higher levels of freedom, capable of identifying multiple hands in an image or video frame with complex backgrounds and with no clothing requirement; meaning that the signer could wear sleeveless shirts. Success in the area of hand detection with convolutional networks can be consulted at [10], [23], [24] and [8].

3 Methods

To detect and track hands, a two-step experiment is suggested. The first step trains a convolutional neural network using the EgoHands [9] dataset. This first step will test its accuracy using the mean average precision metric [12], a commonly used numerical value in object detection that summarizes precision and recall. As the EgoHands dataset is just a collection of images, a second step is needed to test the ability of the trained model to recognize hands in video recordings of deaf people signing phrases in Costa Rican sign language (LESCO). The former process offers a visual feedback for each image in the video stream where the trained model was not able to detect hands. This fact shows important information about additional hand poses that might need to be included in the training set to improve the hand detection model capabilities.

The full experiment aims to demonstrate the relationship between the mean average precision obtained from training the model and its ability to detect hands in Costa Rican sign language videos. The results obtained are important toward the implementation of a Costa Rican sign language recognition framework to text. These results would show if the EgoHands dataset, under the chosen convolutional architecture, can be used for feature extraction for continuous sign language classification leading to the eventual translation of signemes [11], a portion of the sign that is similar across many sentences, to text.

3.1 Dataset

The EgoHands public dataset from Indiana University's UI Computer Vision Lab, is composed of four thousand eight hundred (4,800) labeled images (frames) taken from forty-eight (48) videos documenting the interaction between two people playing cards, chess and solving puzzles. Each image is 720x1280px JPEG file that shows a first-person view of the interaction between the players [9].

In addition to the 8.2 gigabytes of videos and images, Indiana University's UI Computer Vision Lab also offers labeled metadata for each image in the dataset where hands appear. For a particular image that contains multiple hands, a set of labels describes where each hand is located in the image in terms of x-y locations and height-width primitives that can serve to describe a bounding box for each image. The following is an example of a labeled entry which starts with the image name "Image1.png"

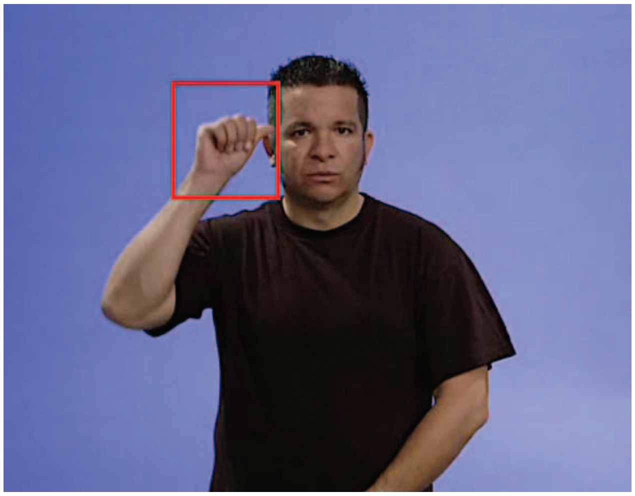

Figure 1. Training set entry example: [Image1.png, 1280, 720, hand, 416, 126, 647, 387]

where 1280 and 720 correspond to the height and width of the image in pixels, and 416, 126, 647, and 286 correspond to the red bounding box coordinates where a single hand appears. If there are many hands in a single image, then additional annotated entries such as the one above must be defined; one for each hand.

3.2 Hand-Detection using CNN

For hand detection, the EgoHands dataset is separated into a training-set with twenty-seven thousand two hundred sixty-two (27,262) images, and a test-set with two thousand five hundred five (2,505) images each as shown in [13]. This duality serves as an input for the depth wise, separable convolutional network MobileNet-V1, a deep neural network capable of executing efficiently in embedded applications [29]. The MobileNet architecture has been designed to run efficiently in mobile devices as well, producing a featherweight deep neural network [15]. This design sets an excellent premise for building a fast hand-detection application capable of using consumer-grade mobile technology.

For this experiment, the MobileNet-V1 model uses pre-initialized weights from the COCO dataset, a technique that has improved performance in object detection as reflected in [16]. The training routine is composed of two hundred thousand (200.000) iterations (epochs) with a learning rate of 0.004 and with constant measurement of the Loss at each step, Regularization/Loss and Mean Average Precision (mAP) to assess model fit and classification accuracy. A checkpoint of the model was also obtained at the 50 thousand iteration for comparison against the 200 thousand version.

Several models can be obtained during the training process and assessed using a small Python script able to draw boxes in the predicted coordinates where the images should appear. This will serve as a visual confirmation as to how the algorithm is able to detect hands as the mAP metric gets closer to one.

3.3 Hand-Detection over LESCO Videos

The second part of this experiment determines the capabilities of the trained model (at 50k and 200k iterations) to detect hands. For this purpose, a set of five (5) different videos from the National Resource Center for Inclusive Education [30] will be tested against the models where a bounding box is drawn at each frame where a hand should appear under a specific probability threshold. Given a particular frame from the video under evaluation, if the model finds a match, it will return to a list of objects with their respective probabilities; in other words, everything that looks like a hand based on the training set. The probability threshold for object detection is a parameter of the experiment that must be manually tuned to determine its value to reduce false positives. In this way, we will be able to label each frame from the videos to determine if there is one visible hand, two visible hands or none. With the hands labeled at each frame, we can offer a metric of hand detection for a given model, and additional feedback if the model fails to recognize a particular hand shape; something valuable for model tuning for LESCO recognition.

4 Results

The MobileNet-V1 algorithm was executed against the EgoHands dataset and two checkpoints of the model were obtained at 50 thousand (50k) and 200 thousand (200k) iterations. The 50k model obtained a value of (ninety-two dot three percent) 92.3% for the mAP, and the second model with 200k iterations a value of (ninety-six dot one percent) 96.1%. The mean average precision value measures the overlap between the two areas: in this case the area for each labeled image and the predicted area for each image that the model was able to detect.

The mAP measures the percentage of predictions where there is at least (fifty percent) 50% of match between the two areas. This does not mean a perfect match on locating hands in a particular frame, which might cause some predictions to create a bounding box around a hand with a different shape in terms of size and position. At this point, it is difficult to assess whether the mAP obtained from both models can generalize beyond the training set for real-time object detection. To test this out, each model was set to recognize hands from a set of LESCO videos. It also determined how many frames were correctly labeled, and how many were incorrectly detected (drawing a box around something that is not a hand). Table 1 shows the results obtained from this evaluation, for each LESCO video under the 200k model. If a frame contains two hands and only one was detected, the frame will be added into the "Label Missing" metric.

Table 1. Frame-based hand detection results

	Call	Die	Grandfather	Pain	Run
Label Missing	18	41	16	20	30
Wrongly Labeled	1	0	0	2	0
Total Frames	309	279	284	314	265
Accuracy	**94%**	**85%**	**94%**	**93%**	**89%**

The results from the 50k model are not reported numerically as the model failed to properly recognize hands across all the LESCO videos, which gives a hint about the minimum number of training iterations needed to obtain the results shown in Table 1. Figure 2 shows some frames incorrectly labeled by the 50k model.

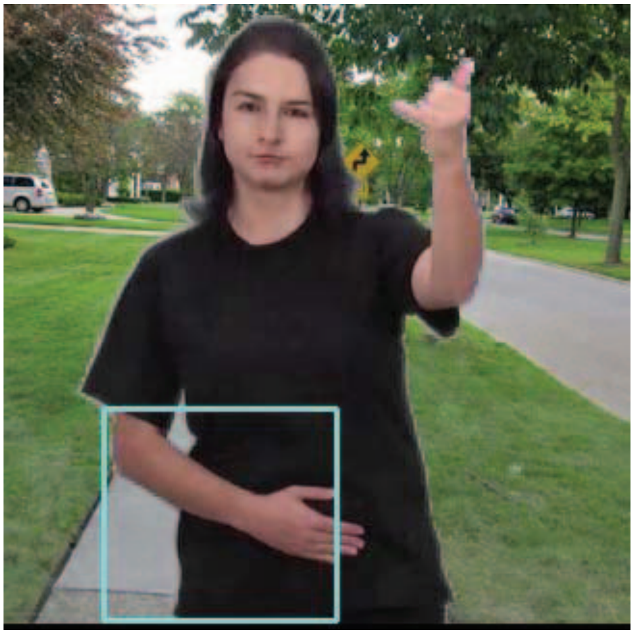

Figure 2-a: Sign for call (phone call).

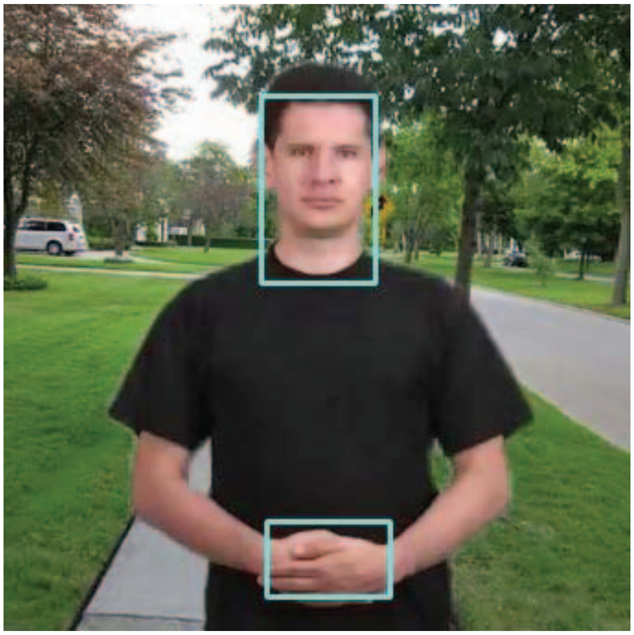

Figure 2-b: Sign to Die (before sign execution)

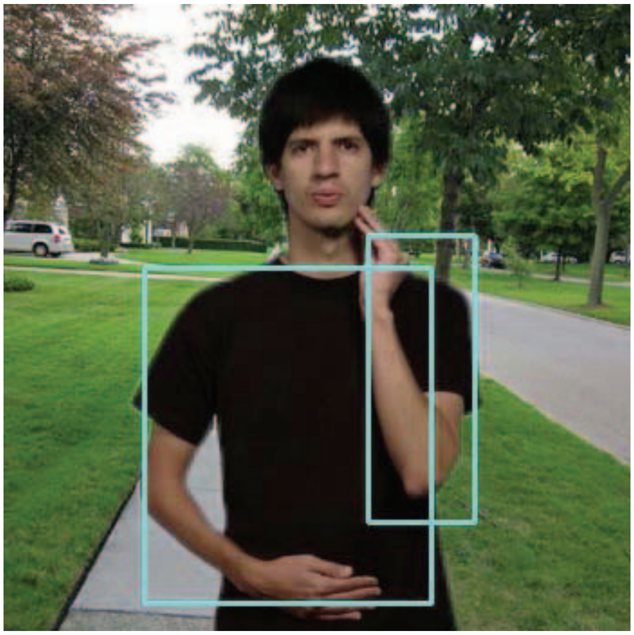

Figure 2-c: Sign for grandfather

Figure 2 shows three examples of how the 50k model failed to properly detect hands. In 2-a, the whole arm is labeled instead of the hand, and the left hand is not detected at all. 2-b shows a false positive where the face is labeled as a hand and 2-c shows both hands and arms labeled incorrectly.

The 200k model did a better job performing the hand detection on the same videos with high level of accuracy as shown in Table 1. Figure 3 shows some examples of frames properly labeled where only hands were detected and decorated with a bounding box.

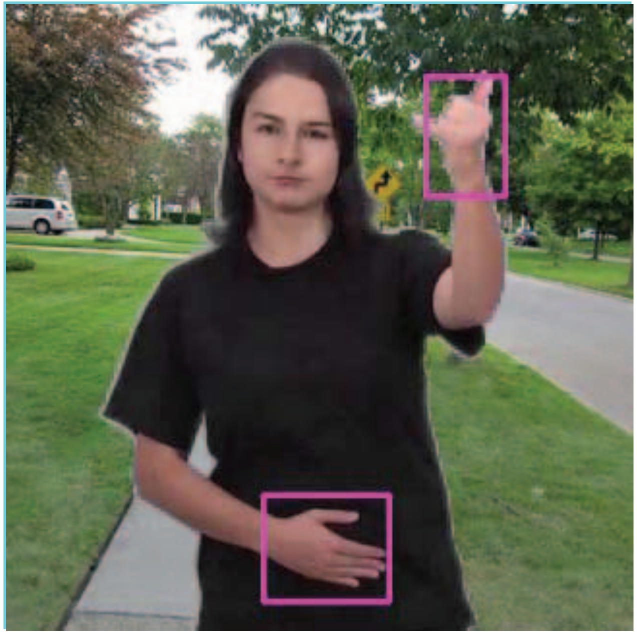

Figure 3-a: Sign for call (phone call)

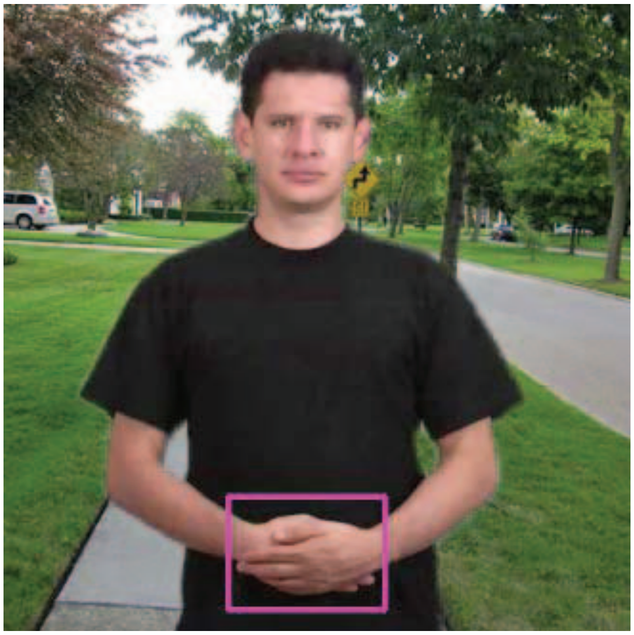

Figure 3-b: Sign to Die (before sign execution)

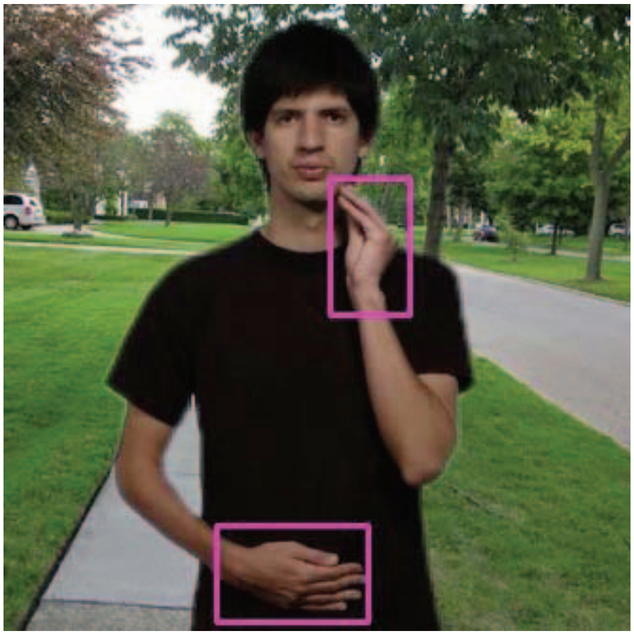

Figure 3-c: Sign for grandfather

5 Conclusions

The MobileNet V1 model was able to detect hands in LESCO videos successfully until the two hundred thousand (200.000) training iteration. The same algorithm under fifty thousand

(50.000) iterations was not able to properly label hands in the five (5) test videos used despite of the small difference of 3.8 in mAP.

The results show a small percentage of frames where the algorithm was not able to detect hands. In most of these cases, it can be attributed to the movement epenthesis [32], the transition phase between two signs where hands get blurred due to the fast movements when recorded with cameras with slow shutter speed. Other cases where the algorithm failed to detect hands happened with the closed fist and the open hand with the fingertips pointing to the camera.

To strengthen the recognition capabilities, it will be recommended to append another dataset with more hand images improving the recognition accuracy as the EgoHands dataset might be insufficient for all hand postures in sign language.

After the release of MobileNet V2 by Google, it will be recommended to run this same experiment with more data under this new neural network architecture; an inverted residual structure that introduces thin bottleneck layers and shortcuts between bottlenecks that outperformed the previous MobileNet V1 experiments tested on the Common Objects in Content (COCO) dataset for object detection [14].

Although MobileNet V1 was able to identify hands successfully, the recognition speed (about 7 frames per second for the test videos) is not optimal for real-time sign language recognition, but it could be used for near-real-time processing. The number of frames per second that the algorithm was able to process increased to 20 (in average) if no image was displayed in the screen. These videos used in the experiment rendered at thirty (30) frames per second, what caused a delay in the display when the bounding box was drawn. An alternative architecture, such as YOLO "You Only Look Once" for fast object recognition able to detect objects in speeds between forty-five (45) and one hundred fifty-five (155) frames per second [32] might also be candidate for real-time hand recognition for LESCO.

FUTURE WORK

This paper opens the door for different configurations of the current experiment that looks for improving the mean average precision results as well as the detection accuracy of hands for LESCO.

To improve the mAP, it will be recommended to try other fast-region convolutional neural network algorithms such as ResNet, ResNet50, Inception-ResNet, Inception-V2, MobileNet-V2 and YOLO as previously mentioned.

The results from this paper suggest that the MobileNet-V1 algorithm performed well until an extraneous hand gesture appeared, for example, closed fists in different rotations. To increase the hand detection accuracy for LESCO, it is important to either try a different dataset or to append more hand images to the EgoHands repository so that the new hands shapes are used to cover the gap between the dataset and the target signs to evaluate. This also suggests a deeper contextual analysis of the specific hand gestures not recognized by the object detection module.

In parallel to these recommendations, it will be good to use additional LESCO videos to validate more signs. Doing so not only serves to validate the detection capabilities, but to detect possible hand positions that the solution was not able to identify.

An additional future work will consist on building a LESCO-driven dataset of images from hands gathered specifically from the videos of CENAREC's online LESCO dictionary.

ACKNOWLEDGMENTS

We would like to thank the IncluTec team at the Instituto Tecnológico de Costa Rica (TEC) for their support and expertise in inclusive technology.

We show our gratitude to Luis Carlos Naranjo, Doctoral Candidate from the University of Alicante for his thoughtful reviews and numerous advices that greatly improved this paper.

We would like to thank Luis Quesada, Doctor from the University of Costa Rica (UCR) for providing us with research material as well as for receiving us numerous times at the UCR's computer science school.

REFERENCES

[1] 2012. Ley 9049 de Reconocimiento del Lenguaje de Señas Costarricense (LESCO) como Lengua Materna. La Gaceta, Alcance n.° 99.

[2] J. Woodward. (1991). Sign Language Varieties in Costa Rica. Sign Language Studies. 1073, 1, 329–345.

[3] P. Retana. (2013). Aproximación a la Lengua de Señas Costarricense (LESCO). Revista de Filología y Lingüística de la Universidad de Costa Rica. 37, 2, 137.

[4] L. Quesada et al. (2016) Sign Language Recognition Model Combining Non-Manual Markers and Handshapes. Ubiquitous Computing and Ambient Intelligence Lecture Notes in Computer Science. 400–405.

[5] D. Wu and L. Shao. (2014). Multimodal Dynamic Networks for Gesture Recognition. Proceedings of the ACM International Conference on Multimedia - MM 14.

[6] H. Cooper and R. Bowden. (2007) Large Lexicon Detection of Sign Language. Lecture Notes in Computer Science Human–Computer Interaction. 88–97.

[7] M. Hassan et al. (2019) Multiple Proposals for Continuous Arabic Sign Language Recognition. Sensing and Imaging. 20, 1.

[8] A. Tang et al. (2015) A Real-Time Hand Posture Recognition System Using Deep Neural Networks. ACM Transactions on Intelligent Systems and Technology. 6, 2, 1–23.

[9] S. Bambach et al. (2015) Lending A Hand: Detecting Hands and Recognizing Activities in Complex Egocentric Interactions. 2015 IEEE International Conference on Computer Vision (ICCV).

[10] A. Rakowski and L. Wandzik. (2018) Hand Shape Recognition Using Very Deep Convolutional Neural Networks. Proceedings of the 2018 International Conference on Control and Computer Vision - ICCCV 18.

[11] S. Nayak et al. (2005) Unsupervised Modeling of Signs Embedded in Continuous Sentences. 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR05) - Workshops.

[12] Szeliski, R. (2011) Computer vision: algorithms and applications. Springer, London.

[13] V. Dibia. (2017) HandTrack: A Library for Prototyping Real-time Hand Tracking Interfaces using Convolutional Neural Networks.

[14] M. Sandler et al. (2018) MobileNetV2: Inverted Residuals and Linear Bottlenecks. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition.

[15] N. Ma et al. (2018) ShuffleNet V2: Practical Guidelines for Efficient CNN Architecture Design. Computer Vision – ECCV 2018 Lecture Notes in Computer Science. 122–138.

[16] B. Zoph et al. (2018) Learning Transferable Architectures for Scalable Image Recognition. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition.

[17] S. Nayak et al. (2017) Finding Recurrent Patterns from Continuous Sign Language Sentences for Automated Extraction of Signs. Gesture Recognition The Springer Series on Challenges in Machine Learning. 203–230.

[18] W. Du and J. Piater. (2015) Hand Modeling and Tracking for Video-Based Sign Language Recognition by Robust Principal Component Analysis. Trends and Topics in Computer Vision Lecture Notes in Computer Science. 273–285.

[19] S. Nayak et al. (2009) Distribution-Based Dimensionality Reduction Applied to Articulated Motion Recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence. 31, 5, 795–810.

[20] P. Viola and M. Jones. (2001) Rapid object detection using a boosted cascade of simple features. Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001.

[21] H. Cooper and R. Bowden. (2009) Sign Language Recognition: Working with Limited Corpora. Lecture Notes in Computer Science Universal Access in Human-Computer Interaction. Applications and Services. 472–481.

[22] H. Thieu Le et al. (2013) A method for hand detection using internal features and active boosting-based learning. Proceedings of the Fourth Symposium on Information and Communication Technology - SoICT '13.

[23] K. Jacobs et al. (2016) Hand Gesture Recognition of Hand Shapes in Varied Orientations using Deep Learning. Proceedings of the Annual Conference of the South African Institute of Computer Scientists and Information Technologists on - SAICSIT 16.

[24] D. Chakraborty et al. (2018) Trigger Detection System for American Sign Language using Deep Convolutional Neural Networks. Proceedings of the 10th International Conference on Advances in Information Technology - IAIT 2018.

[25] I. Goodfellow et al. (2016) Deep learning. MIT Press, Cambridge, USA.

[26] M. Kadous. (1996) Machine recognition of Auslan Signs using powergloves: Towards large-lexicon recognition of sign language. Proceedings of the Workshop on the Integration of Gesture in Language and Speech, 165-174.

[27] C. Dong et al.: American Sign Language alphabet recognition using Microsoft Kinect. 2015 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW).

[28] P. Chophuk et al. (2018) Fist American sign language recognition using leap motion sensor. 2018 International Workshop on Advanced Image Technology (IWAIT).

[29] A. Howard et al. (2017) MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications.

[30] CENAREC. (2019) Diccionario LESCO, http://cenarec-lesco.org/DiccionarioLESCO.php.

[31] R. Yang et al. (2010) Handling Movement Epenthesis and Hand Segmentation Ambiguities in Continuous Sign Language Recognition Using Nested Dynamic Programming. IEEE Transactions on Pattern Analysis and Machine Intelligence. 32, 3, 462–477.

[32] J. Redmon et al. (2016) You Only Look Once: Unified, Real-Time Object Detection. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). doi:10.1109/cvpr.2016.91