ISBN: 978-981-18-1791-5
2021 the 11th International Workshop on Computer Science and Engineering (WCSE 2021)
doi: 10.18178/wcse.2021.06.038

Senyales: FSL Translator for Day-to-Day Emergencies

Alain Vincent Mindaña [1], Daian Villa [2], Marie Sheila Martin [3], Qaeda Vennett C. Cinco [4], Anthony D. Aquino [5+] and Shaneth C. Ambat [6]

[1] College of Computer Studies, Far Eastern University Institute of Technology, Manila, Philippines

Abstract. In the Philippines, there is still a barrier language between the deaf and hard of hearing and abled communities. This could be attributed to how time and resource-consuming it is to learn FSL. This study aims to develop an FSL emergency sign detection in hopes to diminish the prejudice against dhh and the barrier that prevent them from being entirely accepted by non-dhh. To test the hypothesis of how fast hand detection works in emergencies using mobile, YOLO v4 tiny was used as the main algorithm for FSL detection for selected phrases often used in emergencies backed by an online survey for its usefulness. The results yielded an average accuracy of 85% in detecting the hand signs in different background environments with ample lightning and that using the mobile application to detect emergency hand signs is fast and accurate enough when in different backgrounds.

Keywords: YOLO, dhh, Android, Mobile programming, object detection

1. Introduction

FSL is a sign language that originated from the Philippines. Unlike spoken languages such as Filipino and English, FSL has its grammar, syntax, and accent. FSL has its structure that does not resemble anything from the spoken languages due to its visual-spatial nature. Moreover, FSL is the primary binding force for dhh Filipinos residing in the country. Additionally, it is the core of the progressive view of deafness as a culture and of the deaf and hard of hearing as a linguistic and cultural minority. Despite this, some researchers believe that FSL with indigenous roots is at risk of being lost in time because of the growing influence of foreign sign languages like ASL. Sign language has been a practical tool and medium for the deaf and mute community. In the Philippines, there are two significant categories of sign languages used by many: FSL and ASL, in which both are "naturally emanating from the impaired" [1].

The emergence and fast development of technology aided many disabled people in dealing with their disorder and impairment. HCI significantly contributed to computer design and user experience and improved haptic technologies to specifically cater to people's needs. Based on the Development of English-to-Sign-Language Translation System on Android by Wairagya, Buana, and Sukarsa (2019), a mobile application was developed to translate sign from ASL to text [2]. The researcher used Android Application Program Interface (API) to create a server where the video is processed, segmented to frames, and translated. The result was a translated sentence structured form of ASL that the user can understand.

2. Problem Statement

During the 1500s, the first thriving community to have both abled and non-abled people coexist in harmony was in the island called Marth's Vineyard. However, despite the industrial revolution and technological breakthroughs, the lack of social awareness and stigma against Persons with Disabilities (PWD) enabled these people's oppression. An example is an article published by the Philippine Daily Inquirer last 2018 that featured a deaf couple being mocked and driven out in a prominent mall in Cubao by security staff [3]. This sparked outrage from the people in social media who learned about the couple's unpleasant encounter. Furthermore, the US State Department said that PWDs face more discrimination in job landing during the annual human rights reports last 2018 as only 10% of the PWDs in the Philippines can find work

+ Corresponding author. Tel.: + 63 2 8281 8888
E-mail address: registrar@feutech.edu.ph

[4]. In addition, the language barrier between the non-impaired and impaired people dramatically contributes to this ignorance and bigotry.

3. Objective

3.1. General Objective

The study's general objective is to develop an FSL detector that caters to day-to-day emergencies through a mobile application.

3.2. Specific Objective

The specific objectives of the study are the following: (1) To provide an FSL emergency hand sign detector in real-time. (2) To provide a user interface for navigable interaction and usage. (3) To create an FSL detector using a neural network and object detection. And (4) To make FSL emergency sign dataset.

- To provide an FSL emergency hand sign detector in real-time.
- To provide a user interface for navigable interaction and usage.
- To create an FSL detector using a neural network and object detection.
- To make FSL emergency sign dataset.

4. Scopes and Limitations

4.1. Scopes

The study is an Android-based mobile application that features the FSL emergency sign detection in real-Time and basic FSL phrases clips. Using the system allows the user to use the features and functions stated to understand the deaf and hard of hearing's emergency signs. Senyales' dataset contains Filipino standard emergency sign language for use in detecting the hand signs.

The application is applicable to Android mobile devices with Android mobile operating system versions 9.0 Pie environment and above. The mobile device must be equipped with 8 megapixels or higher for the front camera for sign detection. The system can only detect the sign with ample lighting and less detailed background (i.e., monochrome, minimal designs). In addition, it detects signs within 1-2 meters away from the camera. The non-dhh can choose a phrase from the list that will play a 5-10 second video.

4.2. Limitations

The application can only recognize certain skin hand pigments that have lighter tones. Birthmarks, skin tags, finger deformities, and arm-length sleeves may also significantly affect hand recognition. Furthermore, facial expressions are not part of sign detection, which can only detect dynamic and static hand movements. The application can see only two hands that are using sign languages. The study contains data set on emergency FSL to detect and translate signs. Moreover, the application developed can only be used to translate and detect emergency FSL and not teach the basic FSL signs to users.

5. Classes/Functions

5.1. Senyales' User Interface View Class

- Home View – shows the home or main activity of the application.
- Translate View – launches a camera activity for sign detection.
- Detect View – has the machine learning model that pre-processes the camera activity feed and outputs the detected and labelled hand sign.
- About Us View – contains information about the application and researchers.

5.2. Senyales' Controller Class

- onCreate() – is where the other functions are placed and coded.
- onClickListener() – prompts the corresponding views.
- initBox() – contains the detection part of the application.
- playVideo() – plays the corresponding videos.

- run() – contains the thread for the splash screen and the number of second delays before the home view appears.
- onPreviewSizeChosen() – contains the dimensions for the bounding boxes, tracker, and specifications of the text.
- getScreenRotation() – refers to the canvas that is relative to the camera orientation.
- processImage() - contains the timestamp for image detection.

6. Figure/Captions

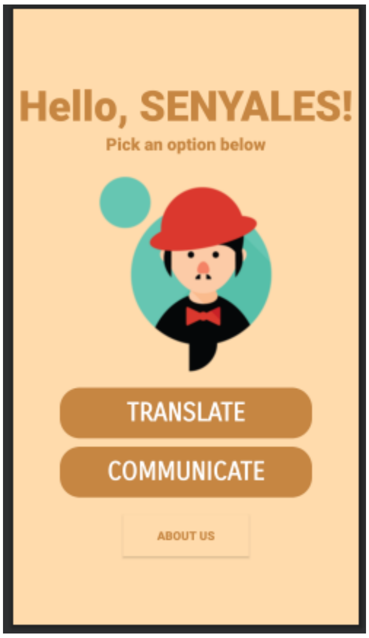


Fig. 1: Mobile Home Screen.

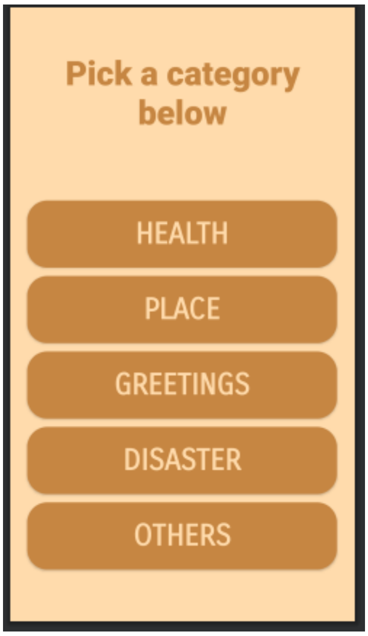


Fig 2: Communicate Option (List of Categories).

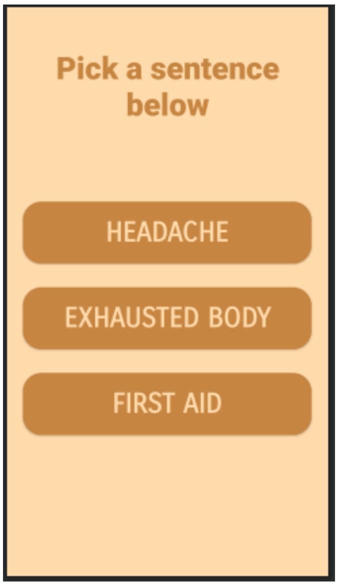


Fig. 3: “Health” Option (List of Categories).

Fig. 4: “Headache” Emergency FSL Clip.

Fig. 5: Translate Option Emergency FSL Hand Sign Detection.

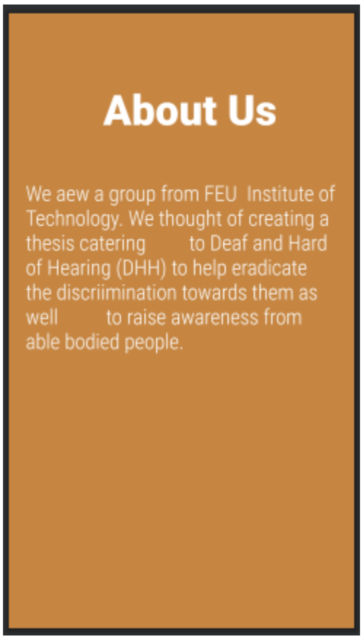


Fig. 6: About Us Option.

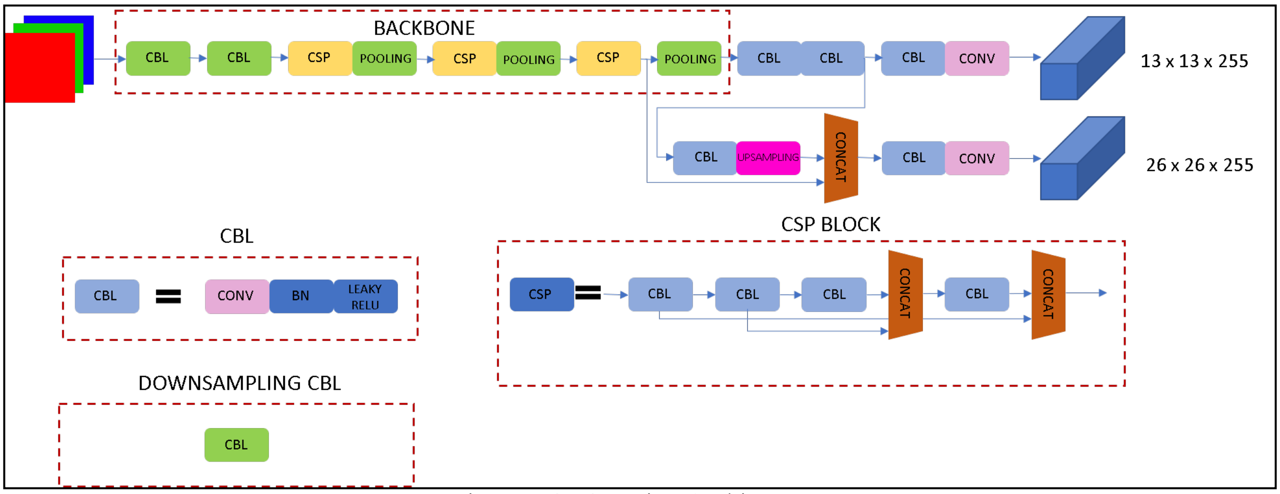


Fig. 7: YOLOv4-tiny Architecture.

7. Ablation Study/Findings

The ablation study conducted by the researchers determines the performance of the object detection model overall. This scientific examination of the model yielded various results to analyze the object detection capability and limits.

7.1. Dataset

Table 1: SENYALES Dataset Metadata

SIGNS (CLASS)	TOTAL IMAGES/CLASS (.png)	LABELLED IMAGES (.txt)
CALL_HELP	722	722
EARTHQUAKE	1,773	1,773
FIRE	831	831
THIEF	765	765
TORNADO	752	752
SAFE	3,351	3,351
STOP	3,177	3,177
FOLLOW ME	3,443	3,443
EMERGENCY	3,557	3,557

STORM	2,414	2,414
STAY	3,275	3,275
WAIT	2,258	2,258

7.2. Mobile System Algorithm (YOLOV4-Tiny)

The algorithm used in the mobile application is based on CSPDarknet53 (53 is the number of convolution layers in Darknet). YOLOv4-tiny consists of a backbone that is CSPDarknet53, Neck such as Spatial Pyramid Pooling and PANet path aggregation, and the head or the mainframe that YOLOv4-tiny is built on top of YOLOv3 [5] [7].

The backbone module in Figure 7. contains Convolution-Bn-mish Layer (CBL) with convolution-bn-leak_relu, which convolves the layer before pooling the image. In the application's case, it is the video feed from cameraActivity(). The output vector size of the backbone is 13x13x255. Upsampling and concatenating the output vector size of 13x13x255 result in 26x26x255.

7.2.1. Mobile System Algorithm (YOLOV4-Tiny)

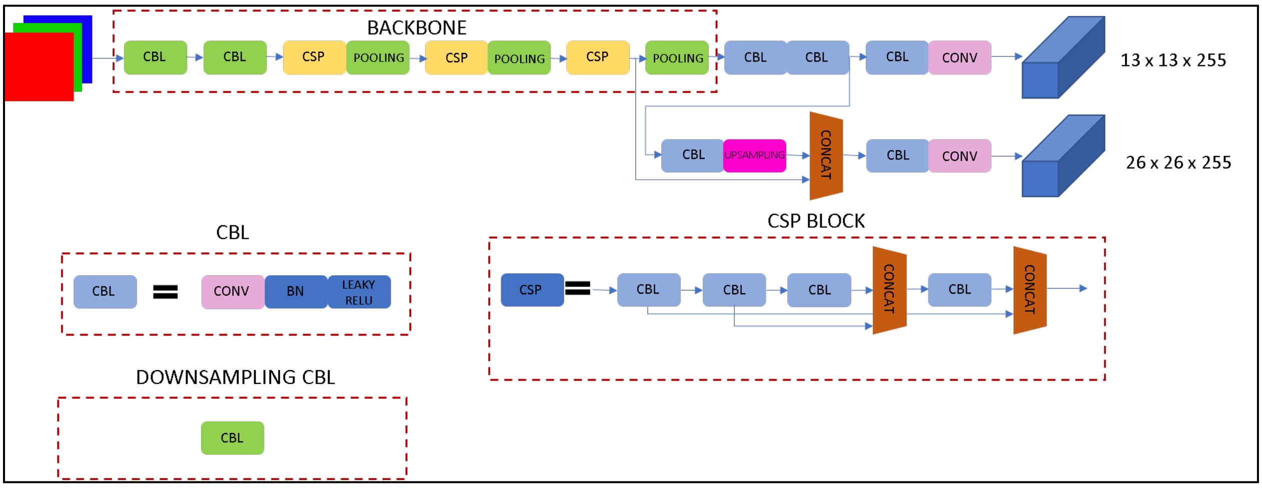


Fig. 7: YOLOv4-tiny Architecture

7.3. Tensorflow Lite FLOAT16 Quantization YOLOV4-Tiny Weights Conversion

The weights resulted from training the model are used to convert it to a. tflite format. Weights are transformed to a 16-bit floating value in TensorFlow Lite's flat buffer format. This results in a model size reduction up to twice its original network size [6].

7.4. Model Analysis for each Devices/Medium

The first device, a Dell Laptop or Laptop, yielded a mean Average Precision Intersection over Union @ 0.5 = 50% around 75%-85% on a Tiny-Darknet backbone optimized with TensorRT FP16 with an average FPS of 14.9. The mobile device Galaxy S8+ with only a processor and without any GPU yielded a mean Average Precision Intersection over Union @ 0.5 = 50% around 70%-80% with an average FPS of approximately 24.3, its mAP IOU fluctuated irregularly up to 90%-95% on the mobile device upon conducting test; the table reflects the average mAP that the researchers computed based on the output given by the model on different devices. Lastly, Google Colaboratory, a data analysis tool that uses cloud computing on powerful GPUs, resulted the highest among the two other mediums used. The Tesla P100-PCIE-16GB GPU, which is a highly robust GPU, with the same backbone as the other two, yielded a mean Average Precision Intersection over Union @ 0.5 = 50% around 85%-95% again fluctuating irregularly or dependent on some signs that can reach up to 98% confidence score and an average FPS of 175.8.

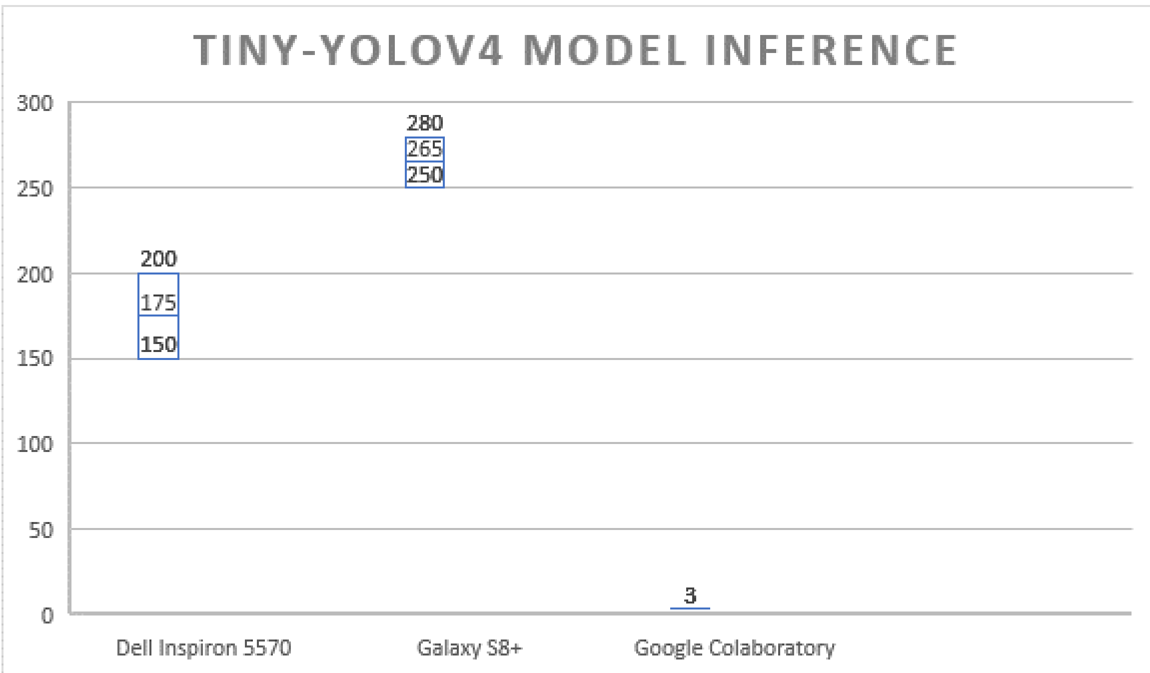
Table 2: FPS Measurement

Model	Device	CPU/Processor	GPU	Backbone	Input	mAP IOU @0.5 (Default)	Average FPS
Tiny-YOLOv4	Dell Inspiron 5570	Intel(R) Core (TM) i5 – 8250U CPU @ 1.60GHz [Cores 4] [Logical processor 8]	GPU 0: Intel(R) UHD Graphics 620	Tiny-Darknet + TensorRT_float16	416 x 416	75% - 85%	14.9
			GPU 1: AMD Radeon R7 M460				
	Galaxy S8+	Qualcomm Snapdragon 835 or Samsung Exynos 8895	-			70% - 80%	≈24.3
	Google Colaboratory	-	Tesla P100-PCIE-16GB			85% - 95%	175.8

7.5. Evaluating Model Using Inference Time

Inference time in machine learning is essential in determining the performance of the model. It is the process of using the trained model to make a prediction. The goal of all machine learning algorithms is to yield a low inference time or time of predicting to achieve a robust detection system. The researchers' model based on Tiny-YOLOv4 and optimized with TensorRT FP16 yielded an inference time for the three mediums used for the model created. Figure 8 displays the inference time for each medium. For Dell Inspiron 5570 Laptop, it yielded an inference time that ranges from 150ms – 200ms with a mean of 178.5ms and a median of 175. For Samsung Galaxy 8+, it yielded an inference time range 250ms – 280ms with a mean of 273.8ms and a median of 265. Lastly, Google Colaboratory using cloud GPU (Tesla P100-PCIE-16GB) yielded an inference time of 3ms with a mean of 3ms.

Table 3: Model Inference



7.6. Sign Detection Evaluation using Confusion Matrix

The total mAP for each class was computed to determine the predicted values for each sign using Galaxy S8+ as the medium for sign detection. The confusion matrix presented on Table 4 is the mAP for all classes. First class, call help, yielded a total of 0.99 or 99% total mAP while scoring a zero (0) across the other classes, meaning that it did not generate or classify the object incorrectly, a false positive. Earthquake yielded 0.53 or 53% total mAP while zero (0) on other classes. Fire yielded 0.94 or 94% total mAP while zero (0) on others. Thief yielded 0.85 or 85% total mAP while zero (0) on other classes. Tornado yielded 0.85 or 85% total mAP while zero (0) on other classes. Safe yielded 0.95 or 95% total mAP while zero (0) on other classes. Stop yielded 0.84 or 84% total mAP while zero (0) on other classes. Follow me yielded 0.91 or 91% total mAP while zero (0) on other classes. Emergency yielded 1.02 or 102% total mAP while zero (0) on other classes. Storm yielded 0.97 or 97% total mAP while zero (0) on other classes. Stay yielded 1.01 or 101% total mAP while zero (0) on other classes. Lastly, wait yielded 0.98 or 98% total mAP while zero (0) on other classes. These were based on the total mAP of both environments where the detection was tested (indoors and outdoors). Data was collated and computed to generate the total average per class.

Table 4: Hand Sign Detection Confusion Matrix

Predicted Values		CALL_HELP	EARTHQUAKE	FIRE	THIEF	TORNADO	SAFE	STOP	FOLLOW_ME	EMERGENCY	STORM	STAY	WAIT
	CALL_HELP	0.99	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	EARTHQUAKE	0.0	0.53	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	FIRE	0.0	0.0	0.94	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	THIEF	0.0	0.0	0.0	0.85	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	TORNADO	0.0	0.0	0.0	0.0	0.85	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	SAFE	0.0	0.0	0.0	0.0	0.0	0.95	0.0	0.0	0.0	0.0	0.0	0.0
	STOP	0.0	0.0	0.0	0.0	0.0	0.0	0.84	0.0	0.0	0.0	0.0	0.0
	FOLLOW_ME	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.91	0.0	0.0	0.0	0.0
	EMERGENCY	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.02	0.0	0.0	0.0
	STORM	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.97	0.0	0.0
	STAY	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.01	0.0
	WAIT	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.98

8. Survey Analysis

The researchers successfully developed and designed a mobile application that can detect 12 FSL emergency signs used frequently in disasters. When evaluated, the application was able to score 4.4 and 4.5 in terms of its functionality. The respondents were CS/IT students and professional programmers. Moreover, according to a dhh personnel who also evaluated the application's functionality said that it works according to the application's objectives. The application was able to apply the formula used to gather the average and total average value for each data. The users also found the application navigable when asked for Usability. In terms of its reliability, the application garnered 4.3, proving that the application is reliable and capable of producing desired results. In addition, the application's performance received a 4.0 from the respondents, proving that it is excellent in sign detection and communication function. The respondents, however, gave a poor rating on the application's supportability since high specifications are needed to run the application.

9. Conclusion

Based on the evaluation results gathered and the study's objectives, the conclusions derived by the researchers are as follows. SENYALES is developed for DHH and Non-DHH personnel to detect emergency FSL hand signs during times of crisis. Thus, the application is helpful for non-DHH personnel who are always on duty when disasters happen to understand and address the needs of the dhh (1). The application is user-friendly and navigable (2). The application detects 12 emergency FSL hand signs (3). The user can play video clips provided in the application (4). The customized dataset for emergency FSL hand signs can be used to further the research on emergency-related studies (5).

10.Recommendation

Future researchers interested in engaging in the same study may consider the following recommendations. Future researchers may implement robust algorithms to speed up the model's detection and accuracy (1). Implement more data augmentation on the dataset to produce better detection (3). Add more emergency FSL signs in the dataset taken from outside environments and places where disasters are prone to occur (4). Create stabilized version on Android that can cater to all versions (5). Expand the application's supportability by developing a stabilized IOS version (6).

11.Acknowledgements

We would wish to express our sincerest gratitude to Dr. Shaneth C. Ambat, our CS Director, for assisting us throughout our thesis journey.

We would also like to thank Mr. Anthony D. Aquino, thesis mentor, for aiding us in project development and paper revisions.

And finally, our dear instituiton, FEU Instutitute of Technology, for giving us to opportunity to present our research in various conferences.

12.References

[1] Maria Feona Imperial. 2015. Kinds of sign language in the Philippines. (June 2015). Retrieved January 22, 2021 from https://verafiles.org/articles/kinds-sign-language-philippines.

[2] IBN Wairagya, PW Buana, IM Sukarsa. "Development of English-to-Sign-Language Translation System on Android." International Journal of Computer Engineering and Information Technology 11, no. 7 (2019): 157-163.

[3] Cody Cepeda. 2018. Deaf couple mocked, driven away by security guard in Cubao mall; management takes quick action. (May 2018). Retrieved 2021 from https://lifestyle.inquirer.net/295789/p2fb-deaf-couple-mocked-driven-away-by-security-guard-in-cubao-mall-management-takes-quick-action/

[4] "Policies for PWDs 'not effectively enforced' in the Philippines: US report." ABS-CBN News. Mar. 14, 2019. https://news.abs-cbn.com/news/03/14/19/policies-for-pwds-not-effectively-enforced-in-the-philippines-us-report (accessed: Mar. 28, 2021).

[5] Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. 2020. YOLOv4: Optimal Speed and Accuracy of Object Detection. (April 2020). Retrieved 2020 from https://arxiv.org/abs/2004.10934.

[6] “Post-training float16 quantization | TensorFlow Lite.” tensorflow. Mar. 27, 2021. https://www.tensorflow.org/lite/performance/post_training_float16_quant (accessed: Mar. 27, 2021).

[7] Chia-Wei Chuang and Chih-Peng Fan, "Deep-Learning Based Joint Iris and Sclera Recognition with YOLO Network for Identity Identification," Journal of Advances in Information Technology, Vol. 12, No. 1, pp. 60-65, February 2021. doi: 10.12720/jait.12.1.60-65

[8] Junwen Sun, Waleed Abdulla, Weiming Wang, Qiong Wang, and Hai Zhang, "Band Selection for Palmprint Recognition," Vol. 7, No. 4, pp. 287-290, November, 2016. doi: 10.12720/jait.7.4.287-290

[9] Rabikumar Meitram and Prakash Choudhary, "Palm Vein Recognition Based on 2D Gabor Filter and Artificial Neural Network," Vol. 9, No. 3, pp. 68-72 August 2018. doi: 10.12720/jait.9.3.68-72

[10] Baorong Xie, Xinzhong Zhu, Chuanzhao Han, Yang Wang, Xian Li, and Yu Zhang, "Research of the Space-Borne Infrared Ship Target Recognition Technology Based on the Complex Background, "Journal of Advances in Information Technology, Vol. 10, No. 2, pp. 48-53, May 2019. doi: 10.12720/jait.10.2.48-53

[11] Ganesh G. Patil and Rohitash K. Banyal, "A Dynamic Unconstrained Feature Matching Algorithm for Face Recognition," Journal of Advances in Information Technology, Vol. 11, No. 2, pp. 103-108, May 2020. doi: 10.12720/jait.11.2.103-108

[12] José Augusto Cadena Moreano and Nora Bertha La Serna Palomino, "Global Facial Recognition Using Gabor Wavelet, Support Vector Machines and 3D Face Models," Journal of Advances in Information Technology, Vol. 11, No. 3, pp. 143-148, August 2020. doi: 10.12720/jait.11.3.143-148