A Real Time Malaysian Sign Language Detection Algorithm Based on YOLOv3

Mohamad Amar Mustaqim Mohamad Asri, Zaaba Ahmad, Itaza Afiani Mohtar, Shafaf Ibrahim

Abstract— Sign language is a language that involves a movement of hand gestures. It is a medium for the hearing impaired person (deaf or mute) to communicate with others. However, in order to communicate with the hearing impaired person, the communicator has to have knowledge in sign language. This is to ensure that the message delivered by the hearing impaired person is understood. This project proposes a real time Malaysian sign language detection based on the Convolutional Neural Network (CNN) technique utilizing the You Only Look Once version 3 (YOLOv3) algorithm. Sign language images from web sources and recorded sign language videos by frames were collected. The images were labelled either alphabets or movements. Once the preprocessing phase was completed, the system was trained and tested on the Darknet framework. The system achieved 63 percent accuracy with learning saturation (overfitting) at 7000 iterations. Once it is successfully conducted, this model will be integrated with other platform in the future such as mobile application.

Keywords: Convolutional Neural Network (CNN), Sign Language Translation, YOLO.

I. INTRODUCTION

Communication is the key in our daily life. Communicating is the process of exchanging information between sender and receiver through any medium available. The understanding between two parties is very important to ensure that the message delivered is well interpreted by the receiver. Deaf people use sign language as a medium to communicate with others. Sign language is the way of communication that is based on hand movement and visual orientation. Sign language experts stated that the visual used consists of handshape (the way the hand and fingers form a sign), a location of the hand, palm orientation and movement of the hand as its features [1]. This claim clarifies that each hand gesture or movement has a different meaning to be represented.

Different countries have different sign languages because the sign language itself was developed by the deaf communities based on their local culture and heavily influenced and translated from the spoken language [1]. Experts believes that sign language is unique within the communities and not simply extracted from spoken language [2]. The sign language that represents word "Doctor" in American Sign Language (ASL) is represented by forming "D" handshape but straighten the pointer finger and then pointing it to the left-hand pulse. In contrast, for Malaysian Sign Language (MSL), a signer acts like a doctor wearing a stethoscope to represents the word doctor [3]. Therefore, the choice of word and sign language was developed based on the culture of the region and the communities understanding on certain words.

In Malaysia, there are two types of sign language that are used among hearing impaired communities, which are Malaysian Sign Language (MSL) and Kod Tangan Bahasa Melayu (KTBM). MSL is an informal language that is created naturally by the deaf communities while KTBM is a formal language which involved handshape movement cued with a speech that is introduced by the government in the education system. KTBM is a teaching module that was released in 1985 by the Ministry of Education, which was adapted from American Sign Language (ASL) and converted into Malay language for education purpose in school [4]. Meanwhile, MSL is like a layman language that is usually used as communication medium among deaf communities [5].

In a real situation, when normal people meet with deaf people, communication breakdown happens due to different style of communication. In order to communicate with the hearing impaired person, the knowledge of sign language is necessary, but it becomes a barrier for those who do not learn the language. The common constraint faced by deaf people in communication is the absence of a signal interpreter [6]. This assertion is supported by Ting, a sign language teacher at Sarawak Society for The Deaf when she claimed that people are not interested to learn sign language due to the low demand of sign language class for normal people [7].

II. BACKGROUND OF STUDY

A. Object detection

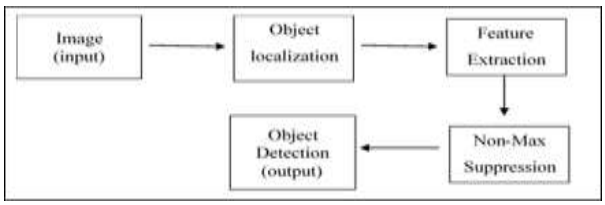

Fig. 1: Object detection framework

Revised Version Manuscript Received on September 16, 2019.

Mohamad Amar Mustaqim Mohamad Asri, Faculty of Computer and Mathematical Sciences, Universiti Teknologi MARA, Perak Branch Tapah Campus, 35400 Tapah Road, Perak, Malaysia.

Zaaba Ahmad, Faculty of Computer and Mathematical Sciences, Universiti Teknologi MARA, Perak Branch Tapah Campus, 35400 Tapah Road, Perak, Malaysia.

Itaza Afiani Mohtar, Faculty of Computer and Mathematical Sciences, Universiti Teknologi MARA, Perak Branch Tapah Campus, 35400 Tapah Road, Perak, Malaysia.

Shafaf Ibrahim, Faculty of Computer and Mathematical Sciences, Universiti Teknologi MARA, Melaka Branch Jasin Campus, 77300 Merlimau, Melaka, Malaysia.

Object detection is an approach that enables the computer to recognize the class that belongs to the object or locates the location of the object. The purpose of object detection is to determine what type of objects that are present and where the object is located in an image or video. The first step is object localization, which the algorithm predicts whether there is an interesting information on the image and where the presence of instances. This region will be bounded by a large set of bounding boxes that covers the entire image. For instance, in Region-based Convolutional Neural Network (RCNN), selective search algorithm was adopted in generating the region proposals.

Selective search approach tends to group pixels of an image and clustering the pixelated groups. This approach begins by extracting the pixels of an image, and then groups the nearest neighbour pixels in order to reduce the correlation of the two pixels. The full image as the largest segment is achieved from the iteration of this process [8]. The second step of object detection is the evaluation of the extracted visual features from the image or feature extraction. Feature extraction refers to the process of identifying the key points in an image (interest points) that can help to define the image's contents such as corners, shapes, edges, and blobs [9]. Next, the extracted features were evaluated through matrix computation and the output matrix is used to determine the pattern and class of the object.

The third step in object detection is the combination of multiple overlapping bounding boxes into a single box by using non-max suppression. During the object classification and localization on each grid cell of the image, it is possible that more than one grid cell will think that the centre of the object is in it, which may produce multiple bounding boxes of object detection. To solve this problem, non-max suppression will group those boxes into one, by choosing the highest probabilities as the most confident detection among each bounding boxes. Non-max suppression is an algorithm that will select the most scoring detection and replace the other lower scoring detection that bound the same object [10]. As a result, the highest probabilities of classification will be suppressed with other remaining bounding boxes and it will be taken as the output. Fig. 5 shows a general process that usually referred in experimenting the object detection project.

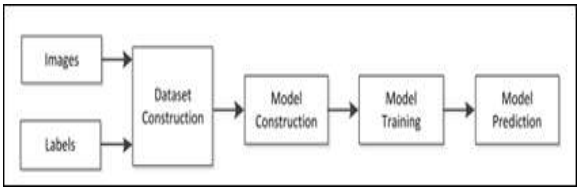

Fig. 2: Example of object detection procedure [11]

Based on the experiment process conducted by [11], the pre-processing process starts with a dataset construction where the images were collected and labelled to form a dataset which will be further used for training, validation and test sets. The model construction involves the task of creating a CNN structure and settings its parameters. Next, the training sets and validation sets are trained in the model training step. After the model training, the model prediction is executed by testing the test set first and utilizes the result for model evaluation.

B. Convolutional Neural Network (CNN)

Convolutional Neural Network (CNN) is a deep neural network that consist of more than two layers of neural network. CNN primarily used to perform image classification, object recognition and object detection in today's technology. CNN are comprised of learnable weights and biases of neurons that works by receiving inputs and perform dot product computation, which then determine the output of the network.

Differs from regular Neural Network, the neurons of CNN architecture were arranged in three dimensions that are known as width, height and depth. Depth of CNN do not resemble the number of layers in the network of CNN, but refers to the dimension of the activation volume instead. Fig. 2 represents comparison between CNN and regular Neural Network. Convolutional Neural Networks have a sequence of layers that perform different functions. There are three main types of layers in CNN architecture, which are Convolutional layer, Pooling layer and Fully Connected layer. Convolutional layer is layer that will operate dot product computation between the weights and a small region that connected to the input [12].

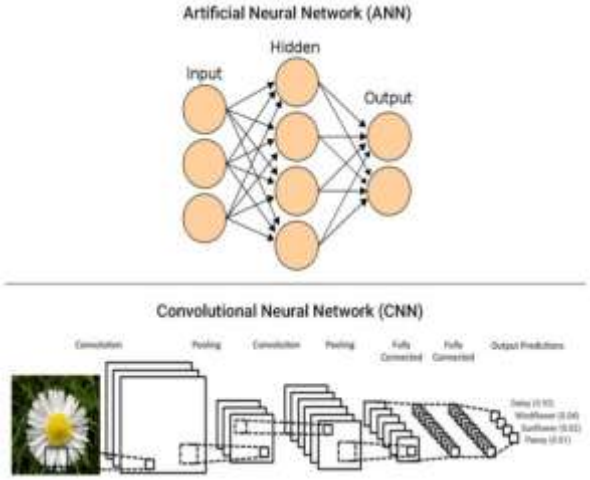

Fig. 3: Comparison between regular Neural Network (top) and Convolutional Neural Network (bottom) [12]

In the pooling layer, the dimensionality of the network is reducing continuously in order to decrease the number of parameters and computation, which also controls overfitting and shortens the training time [12]. The process of determining the class scores will be compute in the Fully Connected layer which then resulting the output of the network.

C. YOLO (You Only Look Once)

YOLO is an advanced deep learning object detection implementation that was introduced by [13]. Initially, object detection works by sliding a small window or known as classifier across the image to make a prediction. This process will consume more time since the classifier, which is the small window is running repeatedly through the entire image until the most certain prediction is determined. However, Ng also reported that YOLO approach is totally distinct from the

other object detector [14]. YOLO looks at the image just once, which is one of the main reasons for its name and detects the object quickly.

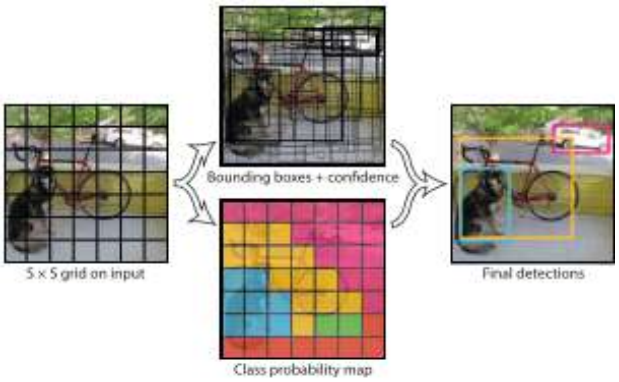

Fig. 4: YOLO operation [12]

YOLO sees the entire image during training and test time, then it encodes the information related to classes object based on their appearance. YOLO network also predict the bounding box by using feature extraction from the entire image. During the detection, YOLO algorithm divides the image into separated numerous grids, S x S grid. The center of the object that falls into a grid cell will be detected as the representation of the object and each grid cell predicts a number of bounding boxes and confidence score for those boxes. Every bounding box created have five predictions, x,y,w,h and confidence. (x,y) coordinates are the center of the box related to the bound of the grid cell, while width(w) and height(h) are predicted relative to the whole image. Confidence prediction resembles the intersection of union (IOU) between the ground truth of the box and the predicted box. Fig. 4 shows how YOLO algorithm approach works on detecting the object. Moreover, Redmond also expressed that YOLO architecture was inspired by GoogLeNet model, but being modified with 24 convolutional layers followed by two fully connected layers [13]. Fig. 5 describes the architecture of YOLO network.

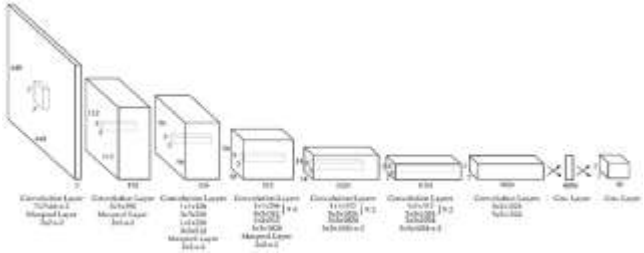

Fig. 5: YOLO architecture [13]

In [14] introduces the latest version of Yolo named Yolo9000 with real-time object detection system that can detect over 9000 object categories. It is an improved model of YOLOv2, does a standard detection tasks like PASCAL VOC and COCO. It is a novel, multi-scale training method operate similarly like YOLOv2 model that can run at varying sizes, with good speed and accuracy. Later, YOLOv3 is introduce with better performance and fast detector category when speed is important [15].

This project runs on YOLOv3 where it uses multi-label classification. For example, the output labels may be "alphabet A" and "sign gesture I" which are not non-exclusive. YOLOv3 changes the softmax function with self-determining logistic classifiers. This calculate the likeliness of the input belongs to a specific label. YOLOv3 then uses binary cross-entropy loss for each label instead of utilizing mean square error in calculating the classification loss. This significantly reduces the computation complexity by evading the softmax function altogether.

III. METHODOLOGY

The system design of this project is as shown in Fig. 6, it was separated into Training model and Detection model, where the Training model consist of labelled and trained dataset, and the Detection model as the testing process of this project. The system will begin to work by receiving input of sign language image in real time. YOLO algorithm will process the image by identifying the existence of trained images in the input image. If the trained image is existing in the input, a bounding box with label that covers the expected object will be produced.

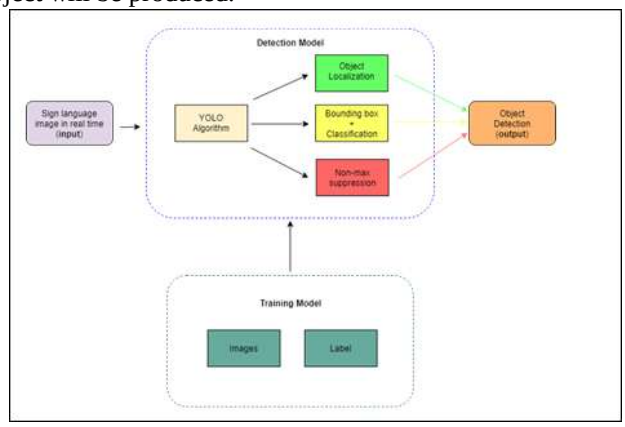

Fig. 6: System design

A. Darknet setup

Darknet is an open source neural network for YOLO object detection that was written in C and CUDA. In order to execute Darknet neural network in an independent machine, it requires to be installed with some dependencies. The installation and compilation of Darknet was referred from AlexeyAB's Github, as the instructions and steps mentioned was clear and easy to follow. Firstly, the files from the mentioned Github was cloned. Another dependency that were required to be compiled in Darknet also were downloaded such as CUDA 10, OpenCV and CUDA Deep Neural Network (cuDNN) library. Moreover, it was suggested to use machine that have GPU with cuda cores of 3.3 or above, as object detection requires high computation capability [16].

After all the minimum requirements for Darknet setup were prepared, the cloned files from AlexeyAB's Github were compiled with CUDA, OpenCV and cuDNN library using Microsoft Visual Studio 2015. Once the compilation completed, the darknet.exe program file was created automatically to remark that the Darknet compilation was a success.

B. Dataset preparation

During the data collection of this project, the sign language dataset from online source, Kaggle dataset that consist of 26 alphabets of American Sign Language images were collected. Each sign language consists of 3000 images (.jpg) of 400x400 resolution with variation of brightness. As the alphabet of Malaysian Sign Language was partially emulated from American Sign Language, therefore some of the letters can be directly used-in this project such as the letters A, B and C. From the dataset, the type of sign language images was further divided into two types, which were referred as static and dynamic. In this context, static sign language refers to a single hand movement and multiple hand movement was identified as dynamic sign language. Fig. 7 presents the example of image labelling that was conducted during the dataset preparation. This project focused on the development of the static sign language as it is easier to be labelled compared to the dynamic sign language. After the images were collected, the images were labelled using 'yolo marker', image annotation tool by AlexeyAB on Github [15]. The annotated images information that contains the coordinates of the image object was stored in the text file and will be processed in the training session.

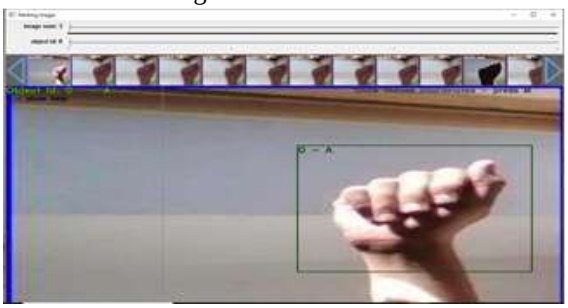

Fig. 7: Image labelling

C. Model training

After the preparation of the dataset completed, the labelled images were trained using Darknet framework. Before the training process was executed, the required parameters such as number of classes and filters were edited in the config file and the other parameters were copied from the default YOLOv3 config file. Next, the .names file, which contains list of the objects to be labelled was created. Furthermore, the training process was executed using the pre-trained convolutional weights, darknet53.conv.74 and being compiled in the Darknet framework. The training process was done until iterations reached 9000 as suggested by AlexeyAB. AlexeyAB suggested that ideal training iterations should be around 4000 – 9000. Fig. 8 shows the training process that was executed.

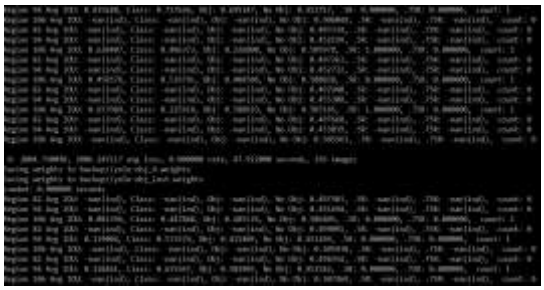

Fig. 8: Model training process

D. Model testing

After the training process of the model completed, the model testing process was executed to determine the accuracy of the trained weights. In Darknet framework, the trained weights will be saved in .weights file after each 1000 iterations. Therefore, if 9000 iterations were executed, there will be nine different trained weights. Fig. 9 shows the process weights validation executed in this project. After the most accurate weights was identified, the weights were tested in real-time, whereby the video was captured using a web-cam. The results of the other weights were tabulated and will be discussed in the next section.

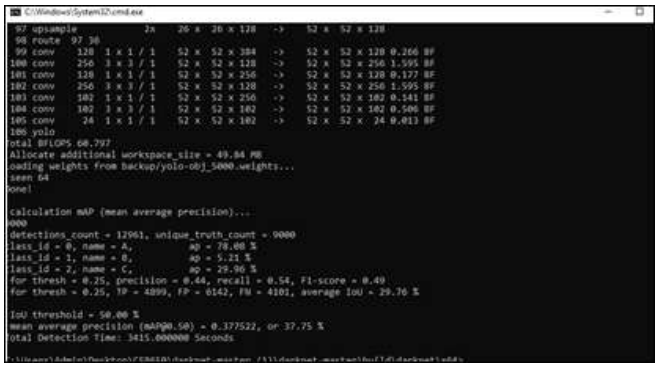

Fig. 9: Weight accuracy validation process

IV. RESULTS AND DISCUSSION

After the validation process was conducted, the results of the mean average precision (mAP) that depicts the accuracy of the detection was recorded. Table 1 summarizes the weights accuracy that was validated after training phase.

Table 1. Performance results by iterations

Iterations	Mean Average Precision (mAP)	
1000	Precision: 0.26	Recall: 0.13
	TP: 1208	FP: 3438
	FN: 7792	Average IoU: 24.06%
	mAP score: 14.7%	
	Total Detection Time(seconds): 316	
2000	Precision: 0.30	Recall: 0.36
	TP: 3283	FP: 7629
	FN: 5717	Average IoU: 18.98%
	mAP score: 37.12%	
	Total Detection Time(seconds): 317	

3000	**Precision: 0.33**	**Recall: 0.38**
	TP: 3445	**FP: 7041**
	FN: 5555	**Average IoU: 20.73%**
	mAP score: 37.34%	
	Total Detection Time(seconds): 318	
4000	**Precision: 0.38**	**Recall: 0.45**
	TP: 4091	**FP: 6690**
	FN: 4909	**Average IoU: 24.43%**
	mAP score: 55.14%	
	Total Detection Time(seconds): 317	
5000	**Precision: 0.44**	**Recall: 0.53**
	TP: 4768	**FP: 6162**
	FN: 4232	**Average IoU: 28.53%**
	mAP score: 56.02%	
	Total Detection Time(seconds): 320	
6000	**Precision: 0.47**	**Recall: 0.57**
	TP: 5109	**FP: 5674**
	FN: 3891	**Average IoU: 30.54%**
	mAP score: 58.8%	
	Total Detection Time(seconds): 320	
7000	**Precision: 0.49**	**Recall: 0.60**
	TP: 5417	**FP: 5537**
	FN: 3583	**Average IoU: 32.67%**
	mAP score: 63.06%	
	Total Detection Time(seconds): 318	
8000	**Precision: 0.46**	**Recall: 0.57**
	TP: 5118	**FP: 5936**
	FN: 3882	**Average IoU: 30.47%**
	mAP score: 57.32%	
	Total Detection Time(seconds): 318	
9000	**Precision:0.46**	**Recall: 0.57**
	TP:5118	**FP:5936**
	FN: 3882	**Average IoU:**
	mAP Score: 57.32%	
	Total Detection Time(seconds): 319	

Based on the observation in Table 1, TP means detecting Malaysian Sign Language correctly as in trained dataset. FP means detecting other objects in the images as Malaysian Sign Language mistakenly while FN represents the Malaysian Sign Language that are not detected in the detection. The weights that being trained on iterations 7000 have the highest accuracy among the others with 0.49 precision rate, 0.60 recall and 63.06% mAP score. The weights also have the most TP value and the lowest FN and FP values compared to the other weights. However, the weights still cannot be considered as good for the detection model because a good detection model should have an accuracy that close to 100%, usually 95% and above. Besides, from the accuracy of the trained weights, it was suspected that the training process have achieved overfitting, where objects on images from training-dataset can be detected, but cannot on other images. Fig. 10 visualizes the weights accuracy by iterations of this project.

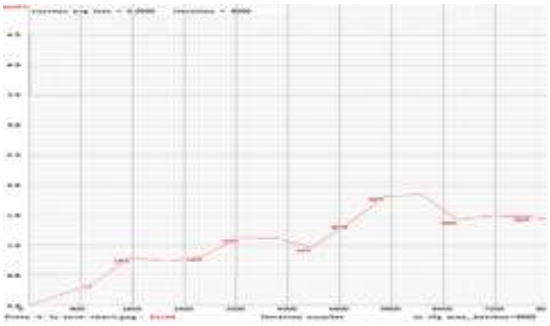

Fig. 10: Weights accuracy by iterations

From Fig. 10, it shows that the accuracy of the weights starts to drop drastically after 7000 iterations and decrease continuously. It is believed that during iterations 7000, it was the peak of the training, where the learning rate was on its maximum and the overfitting happened after 7000 iterations. Overfitting happens because of the pre-trained model neural network architecture, where the learning rate cannot be control and the neural network was fitted closely to the training set, which then leads to difficulty to generalize and predicts new data. Moreover, when this model is being tested in real world situation by executing it through webcam, sometimes it detects the object that was not trained to be detected such as it classifies white wall as letter 'A' and sometimes it does not give any response or detection when a certain sign was shown. Therefore, it can be confirmed that this project detection model fits too well. The results also show that this project has low robustness and accuracy for detecting the Malaysian Sign Language. This might happen due to few factors that influence the accuracy of the detection such as high image noise, lack of variety of distraction in the object images and insufficient number of trained images.

V. CONCLUSION

This paper proposed a study on automatic Malaysian Sign Language Detection using Convolutional Neural Network (YOLO approach). The project focuses on the translation of hand gesture movement that comprise of alphabet and fingerspelling of Malaysian Sign Language. An open source framework of YOLO algorithm, which is Darknet was implemented in conducting this project. The performance of the translation (weights file) were evaluated by iterations within the Darknet framework. It was found that the best weights were on 7000 iterations and it is believed that the model experiences overfitting after 7000 iterations. The performance and consistency of the detection and identification system are not very promising with an accuracy of 63.06 percent from training and 72 percent from actual system implementation results. Based on the development carried on, the system able to detect the sign language, but it is highly depended on the video captured image, background and angle of the hand. For future works, we plan to investigate on the deeper domain of the sign language or a bigger dataset and to explore on the other popular approaches such as recurrent and generative adversarial neural network which has been proven an outstanding achievement in other research fields.

VI. ACKNOWLEDGMENT

This research has been supported by Universiti Teknologi MARA Perak Branch, Tapah Campus.

REFERENCES

1. W. C. Stokoe, "Sign language structure: An outline of the visual communication systems of the American deaf," Journal of deaf studies and deaf education, 10(1), 2005, pp. 3-37.
2. A. F. B. M. Sahid, W. S. W. Ismail, and D. A. Ghani, "Malay Sign Language (MSL) for beginner using Android application," 1st International Conference on Information and Communication Technology, 2017, pp. 189–193.
3. N. F. Baharuddin, personal communication, October 26, 2018.
4. M. S. Shaari, S. Ahmad, T. K. Hoe, and W. Z. Abu, Bahasa isyarat Malaysia. Selangor: Persekutuan Orang Pekak Malaysia, 2000.
5. K. F. Khairuddin, S. Miles, and W. Mccracken, "Deaf learners' experiences in Malaysian schools: Access, equality and communication. Social Inclusion, 6(2), 2018, pp. 46–55.
6. L. J. Muir and Iain E. G. Richardson, "Perception of sign language and its application to visual communications for deaf people," Journal of Deaf Studies and Deaf Education, 10(4), 2005, pp. 390-401.
7. A. Lai, Understanding the silent world. 2015, Available: https://www.thestar.com.my/metro/focus/2015/05/05/understanding-the-silent-world-society-on-a-mission-to-spread-awareness-of-deaf-culture/.
8. A. Kanezaki and T. Harada, "3D selective search for obtaining object candidates," IEEE International Conference on Intelligent Robots and Systems, 2015, pp. 82–87.
9. E. Salahat and M. Qasaimeh, "Recent advances in features extraction and description algorithms: A comprehensive survey," IEEE International Conference on Industrial Technology, 2017, pp. 1059–1063.
10. J. Hosang, R. Benenson, and B. Schiele, "Learning non-maximum suppression," 30th IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 6469–6477.
11. Z. Wu, X. Chen, Y. Gao, and Y. Li, "Rapid target detection in high resolution remote sensing images using YOLO model," International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives, 42(3), 2018, pp. 1915–1920.
12. I. Gogul and V. S. Kumar. "Flower species recognition system using convolution neural networks and transfer learning," IEEE 4th International Conference on Signal Processing, Communication and Networking, 2017, pp. 1-6.
13. J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 779-788.
14. J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 7263-7271.
15. J. Redmon and A. Farhadi, Yolov3: An incremental
16. improvement. 2018, Available: https://arxiv.org/pdf/1804.02767.pdf.
17. AlexeyAB, AlexeyAB/darknet-GitHub repository. Available: https://github.com/AlexeyAB/darknet#yolo-v3-in-other-frameworks.