Convolutional Neural Network Based Malaysian Sign-Language Recognition Web Application

Riskhan Basheer * , Mochamad Azkal Azkiya Aziz , Habiba Arifa

Posted Date: 24 May 2024

doi: 10.20944/preprints202405.1601.v1

Keywords: CONVOLUTIONAL NEURAL NETWORK; MALAYSIAN SIGN-LANGUAGE; RECOGNITION WEB APPLICATION

Article

Convolutional Neural Network Based Malaysian Sign-Language Recognition Web Application

Basheer Riskhan *, Mochamad Azkal Azkiya Aziz and Habiba Arifa

Albukhary International University; azkal.aziz@student.aiu.edu.my, habiba.arifa@student.aiu.edu.my
* Correspondence: b.riskhan@aiu.edu.my

Abstract: The deaf-mute and individuals with hearing disabilities who communicate through sign language should not be overlooked or excluded. It is a social responsibility to strive for an inclusive world without inequality. Advances in technology offer various solutions to address the communication gap, one of which is the use of computer-based sign language recognition systems. This project proposes a Malaysian Sign Language recognition model utilizing MobileNetV1, a convolutional neural network-based algorithm, to interpret sign language in real-time settings through a website application using Streamlit. The outcome of the model test shows that it has achieved a recognition rate of 96 percent. Although the proposed model has demonstrated the ability to recognize sign language, it is suggested that further improvement can be achieved by enhancing the quality and diversity of the dataset. The implementation of a more comprehensive dataset would lead to improved performance and increased accuracy of the model. This project underscores the importance of considering and accommodating the needs of the deaf-mute and individuals with hearing disabilities and highlights the potential of technology to bridge communication barriers and achieve a more inclusive society.

Keywords: CONVOLUTIONAL NEURAL NETWORK; MALAYSIAN SIGN-LANGUAGE; RECOGNITION WEB APPLICATION

Introduction

1.1. Overview

Communication disorder is commonly known to be one of the barriers for society to socialize and interact with each other. Besides different languages, disability can also be one of the communication gaps. Many people have difficulty in verbal communication due to hearing loss or speech impediments. In South-East Asia's countries, there are 401 million people with hearing loss and approximately to be projected to be 660 million in 2050 (World Health Organization, 2021). Hence, the urgency of creating an inclusive environment for the disabilities becomes higher. Sign language is used to overcome this particular communication barrier. Unfortunately, the presence of sign language interpreters and the urgency of learning sign language in Malaysia is still minimal. The distribution of the interpreter also leaves the gap remaining unclosed where the majority of the interpreters are only centered in Kuala Lumpur (Ministry of Communications and Multimedia Malaysia, 2021). Moreover, the number of Malaysian sign language interpreters still leaves a significant gap compared to the numbers of the disabilities in which there are only 95 interpreters for 40,000 deaf and mute disabilities across Malaysia (Awaludin, 2021). One of the solutions to bridge this communication gap is to provide a sign language translation platform. Sign language is the language that bridges the barriers to socialization.

As the main idea, the utilization of computer technology is the key to this proposed system (Shah, 2023). The proposed system will collaborate the function of the sign language interpreters with computer technology-based methods into a digital product that is called sign language recognition. Sign language recognition is a term a method to process sign language as the input to be recognized and to be interpreted into readable and meaningful information. Necessarily, the web application

 doi:10.20944/preprints202405.1601.v1

should be compatible with multiple types of end-user devices. The web application will require the device's camera to capture the sign language gesture. Using a convolutional neural network (CNN), the captured sign language gesture will be recognized and translated into readable text (Airehrour, 2015).

The proposed project will have a social impact, especially for people with verbal communication disabilities. The existence of this platform will help people with disabilities to feel more confident in socialization, while also creating more inclusion space for society to include more people with disabilities, especially in the work environment. Therefore, it also benefits people with disabilities to get border opportunities to find employment.

1.2. Problem Statement

In overall, the identified problem of the project can be described as follows. Understanding sign language is a key to understanding the people with oral communication disabilities, unfortunately the number of sign language interpreters in Malaysia is insufficient (Awaludin, 2021). The urge of the society to learn and to understand sign language becomes the challenge for the sign language to be accepted and widely implemented.

In order to build a convenient model, the project requires an appropriate system model to implement the sign language recognition in a web application environment. There are various types of CNN algorithm that can be applied, further research is required to find the most appropriate and suitable for the project. As part of the test of the proposed model, a set of Malaysian Sign Language datasets are also required.

1.3. Objectives

Accordingly, this project aims to propose solutions to address the problem mentioned, thus the objectives of the project can be defined as follows:

1. To identify the appropriate sign language recognition CNN based algorithm.
2. To prepare Malaysian Sign Language dataset.
3. To test the performance of the proposed model in a website application environment.

1.4. Scope of The Study

The aim of this project is to create suitable models for a web application that can translate the meaning of sign language in real-time. The study focuses on using Convolutional Neural Network (CNN) algorithms to interpret Malaysian sign language during use. The success of the model is evaluated by the success rate of recognizing sign language gestures. The model's performance is assessed by conducting tests within a web application environment. The goal is to develop a system that can accurately interpret sign language and translate its meaning in real-time.

1.5. Project Limitation

This research project has certain limitations that are worth mentioning. Firstly, the project is not meant to perform a deeper analysis of the accuracy and performance of the algorithm. The focus is only on detecting and recognizing Malaysian Sign Language using a limited type of dataset. Secondly, the model is not perfect and may not be accurately detect sign language gestures in all scenarios. Lastly, the limitations of the project will be explained in more detail in the following section. Despite these limitations, the project still aims to make progress in the field of sign language recognition and contribute to the development of better systems in the future.

Literature Review

2.1. Convolutional Neural Network

Convolutional Neural Network (CNN) is a subset of artificial intelligence under the deep learning area. It is actually a deep learning algorithm that has the ability to learn filters without

 doi:10.20944/preprints202405.1601.v1

enough pre-trained input that mimics the human brain's neural network (Saha, 2018). In image processing and classification, CNN appears with feature learning that makes it fast and robust in classification. As mentioned earlier, CNN has capability to train unprocessed images without manual hand-engineered inference. (Zeng et al., 2019).

CNN is advantageous in retrieving local spatial patterns. (Masood et al., 2018). Concurrently, a study found that CNN is also capable of capturing Temporal dependencies of the data (Saha, 2015). In image processing, CNN extracts the images' features through input layer, output layer, and hidden layers. (Das, 2020). Another deeper study on CNN architecture stated that CNN commonly consists of Convolution layers, pooling layers, and fully connected layers. In convolution layers, convolution, a linear operation is performed to obtain image features from the two-dimensional grid. The efficiency of CNN in image processing lies in its ability to apply feature extraction at each image position (Yamashita et al., 2018).

2.2. Sign Language Recognition

Sign language is a visual-modality communication system language. Sign language can be categorized as human's communication by using gestures which are mostly using the hands. Gesture, specifically sign language, is a set of body motions that involves expression, and meanings in order to deliver meaningful information and to interact with the environment. Sign language recognition is a term of a method to process a sign language as the input to be recognized and to be interpreted into readable and meaningful information. The performance of sign recognition is mostly influenced by hand gesture location, position configuration, intensity, and types of the motions. In sign language, the motion can be categorized into dynamic and static motions. (Mitra & Acharya, 2007). There are various approaches in order to interpret the meaning of sign language. It can be acquired using additional hardware that is attached to the user's hand or using a vision-based approach. (Simei et al., 2002). Another method proposed using infrared sensors, electric-field sensors, and wireless signals to extract gesture information by measuring signal amplitude (Balakrishnan et al., 2023). This project focuses on the use of a vision-based method, hence more review on related work will be delivered in another section. (Kellogg et al., 2014)

According to (Fujiyoshi et al., 2019), the sign recognition process in overall can be divided into features extraction and classification. In sign language recognition, feature extraction is a crucial step that involves extracting relevant information from raw visual data. This information includes features like color, texture, and shape, which are used to represent objects in images or videos. The purpose of this step is to capture important characteristics of the objects in the visual data, which can be used for further analysis. Once the features have been extracted, the next step is classification. This step involves assigning class labels to the objects in the visual data based on the extracted features. This is done using algorithms such as neural networks. The class labels are then used to recognize the objects in the image or video and make predictions about their properties.

Sign language recognition is a challenging task in the field of computer vision and pattern recognition. Yet it is one of the most important features that can enhance overall system performance. Nowadays it is being more difficult to increase security from different types of attacks (Riskhan et al., 2023). Using sign language for this can be a huge leap in technology. Despite advances in machine learning algorithms and computer vision technology, the recognition of sign language still poses several challenges that need to be addressed. One of the challenges is the variability in the environment and the angle of the camera (Humayun et al. 2022). According (Leksut et al., 2020), the lighting conditions, background, and camera angle can all affect the appearance of the sign language gestures, making it difficult for computer vision systems to recognize them. This can lead to low accuracy and high error rates in sign language recognition systems(Ray et al., 2015) and (Lim et al. 2019). Another challenge is the variability in sign language gestures. Sign languages can vary greatly from country to country, and even within a single country, the same gesture can be performed differently by different individuals. This makes it difficult for computer vision systems to recognize and interpret the gestures accurately.

 doi:10.20944/preprints202405.1601.v1

2.3. Related Work

There are various approaches and works that are related to sign language recognition. The common idea is by utilizing the existing vision-based computing algorithm. (Oudah et al., 2021) Proposed a vision hand recognition-based system on *Microsoft Kinect v2* to be used by deaf-mute elderly folks for searching specific items using hand signals. However, the sensor capability limits the hand gesture extraction in real time performance. Similar research conducted using Red Green Blue (RGB) modality shows that the process in detecting the gesture increased latency and failed to generate the gesture detection. Delays in the synchronization rate also occurred, leading to some interruption(Jhanjhi et al., 2020). The limitations that are being highlighted is the latency during the detection, however it achieved 90 percent accuracy (Meng et al., 2020).

Different vision-based systems proposed by (Mujahid et al., 2021) using You Only Look Once v3 (YOLO v3) and *DarkNet-53* that use neural network architecture. The main challenge for this real-time gesture recognition application is to recognize and classify a hand gesture, however the models performed in high accuracy even in a complex environment. Similarly (Alnaim, 2020),(John et al., 2016), and (Neethu et al., 2020) utilized Artificial Neural Network (ANN) and CNN for hand classification and segmentation. The proposed system achieved adequate performance including in video gesture recognition settings. However, it takes high complex computation, and it requires a large amount of data sets.

Sign language recognition is a challenging task in computer vision, particularly in real-time settings (Singhal et al., 2015). In recent years, researchers have made significant progress in developing algorithms for recognizing sign languages. Many researchers have focused on using Convolutional Neural Network (CNN) algorithms to develop sign language recognition systems. For instance, Wadhawan & Kumar et al. (2020) conducted a study and proposed a CNN model to learn and recognize Indian Sign Language. They demonstrated the effectiveness of the CNN model in real-time sign language recognition. Similarly, Ji et al. (2022) conducted a study using a Two-Stream Mixed (TSM) algorithm to be used in American Sign Language Recognition. The study mentioned that to achieve real-time, high-accuracy, and relatively low-cost sign language recognition, more research is required. Another example is Alsaadi et al. (2022) who used a CNN-based algorithm AlexNet in Arabic Sign Language recognition in real-time settings. The accuracy of the model was 95 percent. Podder et al. (2022) conducted a CNN-based Bangla sign language classification research using multiple CNN algorithms. The study showed that the overall accuracy of the CNN model could achieve around 99 percent accuracy. It also highlights that the MobileNet based model had fewer parameters with similar accuracy compared to other CNN algorithms (Gouda et al., 2022).

Furthermore, Ahmed et al. (2020) conducted a study on real-time hand gesture recognition using the MobileNet based model on mobile devices, and it was found that MobileNet achieved the best trade-off between accuracy and computational efficiency in their experiments. Other studies have also shown that MobileNet is a more balanced approach in terms of accuracy, performance, and computational cost, and it is feasible to implement in mobile devices (Jang et al., 2020) (Kim et al., 2020). In conclusion, the studies mentioned above demonstrate that there has been significant progress in the development of algorithms for recognizing sign languages, particularly using CNN algorithms. However, more research is required to achieve real-time, high-accuracy, and low-cost sign language recognition systems.

Methodology

3.1. Model Framework

This chapter focuses on discussing the outline of the system to achieve the objectives according to Chapter 1. Theoretically, the proposed method will be based accordingly with the selected literature review based on Chapter 2 using CNN algorithm-based model. The overall system data flow is represented in the below figures:

Figure 3.1 shows the overview of the overall system in a use case where a sign language is being detected and recognized by the system. The sign language recognition system depicts the concept of

 doi:10.20944/preprints202405.1601.v1

the system in recognizing the sign language by comparing the sign language that is being detected with the trained dataset of the system. Container-based virtualization (CBV) and hypervisor-based virtualization are among the most well-known virtualization of modern technology (Riskhan & Muhammad, 2017). More Container-based virtualization is depicted in Figure 3-2, which shows the system in a state diagram form.

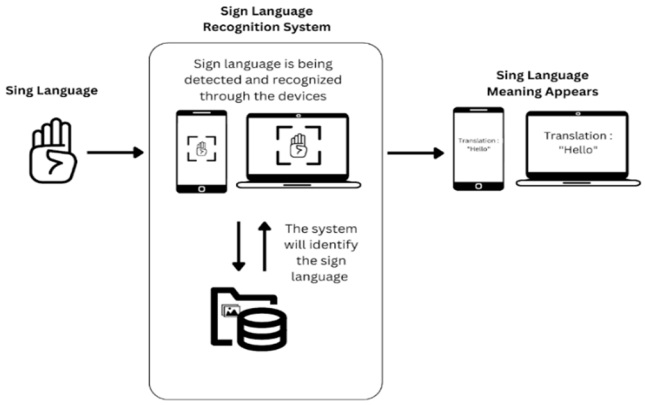

Figure 3.1. Proposed Model Concept.

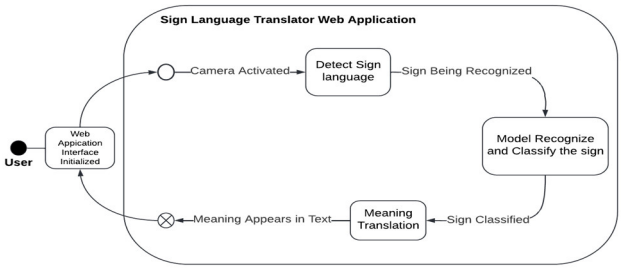

Figure 3.2. Proposed System State Diagram.

The diagram above illustrates the flow of the system and its internal states. The key processes that are highlighted include the detection, recognition, classification, and translation processes. The detection process involves capturing the sign language via the device's camera, followed by recognition and classification by the embedded system's model, which has been trained using a dataset. It is important to note that the dataset itself is not stored within the system environment, but rather the trained model that was created using the dataset. The following section will provide a more in-depth explanation of these processes.

At this current stage, a gesture recognition system is proposed using MobileNet models. Chapter 2 shows that the MobileNet model is suitable to be implemented in this project, considering its efficient performance and its feasibility to be implemented in multiple devices. MobileNet is a convolutional neural network based that uses depth-wise separable convolutions to build light weight deep neural networks with low computational requirements (Howard et al., 2017). The use of CNN is advantageous in this project due to its ability in creating internal representation of a two-dimensional image that makes the model learn the input image efficiently. In other words, CNN is advantageous for spatial purposes such as image recognition. MobileNets is commonly used and provided in the TensorFlow library. TensorFlow is an end-to-end open source platform for machine learning and deep learning (Goldsborough, 2016).

3.2. Project Development

This chapter focuses on discussing the outline of the system to achieve the objectives according to Chapter 1. Theoretically, the proposed method will be based accordingly with the selected

 doi:10.20944/preprints202405.1601.v1

literature review based on Chapter 2 using CNN algorithm-based model. The overall system data flow is represented in the below figures:

3.2.1. Malaysian Sign Language Dataset

This project requires a dataset that consists of sign language images with the annotations of the images. The dataset consists of 330 images that are divided into five classes. The classes are divided based on the meaning of the sign. In this project, the classes are “hello”, “yes”, “no”, “thanks”, and “sorry” which represent the meaning of sign language. The images that are used are the pictures of the sign language gesture that demonstrate and depict the sign language that refers to the Malaysian sign language. The images of the datasets were extracted from the videos that were previously recorded by the author using the author as the object demonstration. The extracted images are then labeled using Label Studio in order to create the annotations. As a point of clearance, although the sign is classified and translated into English, the sign language still refers to the Bahasa Isyarat Malaysia (Official Malaysian Sign Language). The reference of the sign language that is used is Bahasa Isyarat Malaysia from Pocket Handbook Malaysian Sign Language by Malaysian Federation of the Deaf.

Figure 3.3 shows the collaged images that are extracted and labeled from the recorded videos that the author took. Besides the images, the dataset also contains the annotation files in the same number as the number of images. Inside the annotation files contain the information about the region or the bounding box that locates the object that the model is supposed to take as the training input. The bounding boxes are annotated using x and y coordinates within the picture. Figure 3.4 depicts the inside of an annotation file of “yes” class.

Figure 3.3. Malaysian Sign Language Dataset Images.

```
<annotation>
	<folder>labels</folder>
	<filename>yes_12.jpg</filename>
	<path>C:\Users\Desktop\labels\yes_12.jpg</path>
	<source>
		<database>Unknown</database>
	</source>
	<size>
		<width>640</width>
		<height>640</height>
		<depth>3</depth>
	</size>
	<segmented>0</segmented>
	<object>
		<name>yes</name>
		<pose>Unspecified</pose>
		<truncated>0</truncated>
		<difficult>0</difficult>
		<bndbox>
			<xmin>205</xmin>
			<ymin>454</ymin>
			<xmax>332</xmax>
			<ymax>580</ymax>
		</bndbox>
	</object>
</annotation>
```

Figure 3.4. Dataset Annotation.

 doi:10.20944/preprints202405.1601.v1

3.2.2. Sign Language Recognition Model

This section provides the description of the sign language recognition model that is used which in this project, MobileNetV1 model is used. This section covers the explanation about pretrained model and its architecture, as well as the related component such as the deployment model and its API.

3.2.2.1. MobileNetV1 Pretrained Model

The project focuses on using MobileNetV1 algorithm. To be specific, this project used a MobileNetV1 pretrained model that is embedded with a Single Shot Detector (SSD) and Feature Pyramid Network (FPN) from Tensorflow library. SSD is a single convolutional neural network object detection algorithm that is used to predict the class labels and the bounding boxes of particular objects within the images (Liu et al., 2016). Whilst FPN is an algorithm that is used to optimize the integration of information from convolutional multiple layers which helps the model to detect the object accurately (Lin et al., 2016). The combination of the MobileNetV1 with SSD and FPN is to produce a better and efficient pretrained model. This pretrained model is publicly distributed by Tensorflow.

3.2.2.2. MobileNetV1 Architecture

As it has been discussed earlier in Chapter 2, in CNN, the input data is extracted by the CNN architecture consisting of multiple layers of filters and a set of kernels to obtain its features along the learning process called convolutional layers. The same concept is applied within the MobileNetV1 algorithm in which the convolutional layers are applied to its architecture. The MobileNetV1 uses depthwise separable and pointwise convolution as convolution operations. In contrast to standard CNN layers, MobileNetV1 uses a single filter instead of multiple layers by using the depth wise separable and pointwise convolution.

The above diagram depicts the filters of standard convolutional and MobileNetV1 according to its original documentation (Howard et al., 2017). Figure 3.5 shows the regular convolutional filters that apply in the standard convolutional neural network structure. Figure 3.6 shows depthwise filters using 3 x 3 filters that later will be summed with 1 x 1 pointwise filters as it is shown in Figure 3.7.

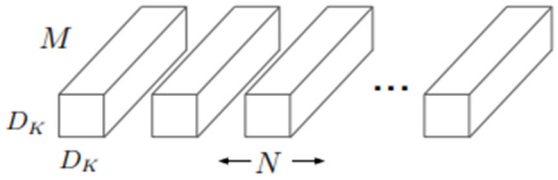

Figure 3.5. Regular Convolution Filters.

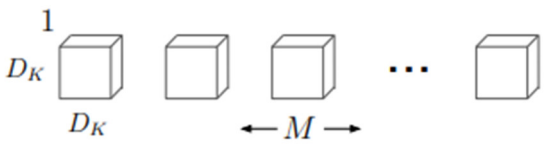

Figure 3.6. Depthwise Convolution Filters.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 24 May 2024 doi:10.20944/preprints202405.1601.v1

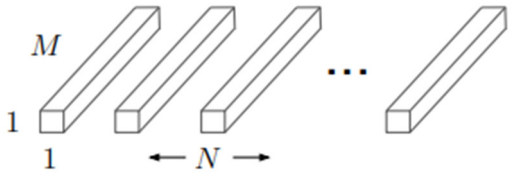

Figure 3.7. Pointwise Convolution Filters.

The computational cost of the standard convolution can be expressed as follow:

$$D_K \cdot D_K \cdot M \cdot N \cdot D_F \cdot D_F \qquad \text{E.q 1}$$

In this case, M is the number of the input also called as input depth, D_K is the spatial dimension, D_F is the spatial height and width and the N is the output number also called output depth. Meanwhile, the computational cost of the summed of the depthwise and pointwise convolution can be expressed as shown below, the summed of these two layers called as separable convolutions (Radford et al., 2018).

$$D_K \cdot D_K \cdot M \cdot D_F \cdot D_F + N \cdot M \cdot D_F \cdot D_F \quad \text{E.q 2}$$

As it is highlighted that MobileNetV1 has less computational cost by expressing two convolutional filters, the reduction of the computational can be expressed by comparing the two expressions of standard convolution and the separable convolution given as below:

$$\frac{D_K \cdot D_K \cdot M \cdot D_F \cdot D_F + N \cdot M \cdot D_F \cdot D_{Fb}}{D_K \cdot D_K \cdot M \cdot N \cdot D_F \cdot D_F} = \frac{1}{N} + \frac{1}{D^2{}_K} \quad \text{E.q 3}$$

The given expression interprets that the computational cost of the separable convolution is less compared to the standard convolution filters. In addition, according to the original publication of MobileNetV1, every layer of the convolution is followed by Batch Normalization (BN) and Rectified Linear Unit (ReLU). BN is a deep learning technique that is used to accelerate network training by normalizing the activation layers within the range of the input data batch (Ioffe & Szegedy, 2015). ReLU is a neural network activation function that allows the model to train on complex patterns as well as improving the training duration. Both BN and ReLU are used to stabilize and optimize the training of the model. The below figure shows the distinction between the structure of the standard convolution layers (left) and the separable convolution layers (right) followed by BN and ReLU after every layer according to Khasoggi et al. (2019).

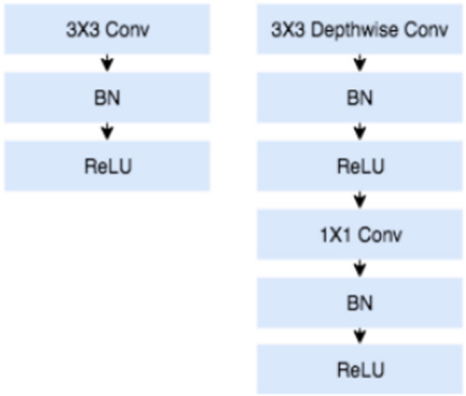

Figure 3.8. Training Structure with BN and ReLU.

3.2.2.3. Tensorflow Object Detection API

To use the specified pretrained model, the project requires a Tensorflow object detection model repository and object detection model API. The repository of the object detection is required as the set of dependencies that are used to run and perform the task of the model. The use of object detection

 doi:10.20944/preprints202405.1601.v1

API is to help the Tensorflow model to run properly as in the training and the deployment. On the other hand, it gives more feasibility for the users to run the model. The API also includes a set of tools and utilities for working with the models, such as data preparation and evaluation tools, and visualization functions (Hongkun Yu et al., 2020).

3.2.2.4. Model Deployment Using Streamlit

To use the specified pretrained model, the project requires a Tensorflow object detection model repository and object detection model API. The repository of object detection is required as the set of dependencies that are used to run and perform the task of the model. The use of object detection API is to help the Tensorflow model to run properly as in the training and the deployment. On the other hand, it gives more feasibility for the users to run the model. The API also includes a set of tools and utilities for working with the models, such as data preparation and evaluation tools, and visualization functions (Hongkun Yu et al., 2020).

3.3. Testing and Evaluation Metrics

According to the objectives and the scope of the study from Chapter 1, the evaluation of the model is determined by its ability to interpret the Malaysian sign language that refers to Bahasa Isyarat Malaysia. The measure of the performance in this project centers on the success of the recognition rate of the model. The success rate of the model or so called as score refers to the detection score that is generated within the pretrained model. The score indicates the ability of the model to identify and locate the object which in this case is the sign language. It is calculated using the Intersection over Union (IoU) score. The IoU score measures the overlap between two sets of bounding boxes, and it is calculated as the ratio of the intersection of the two bounding boxes to the union of the two bounding boxes. To conclude, the success rate comes from the functions that are generated by the pretrained model such as *detection_scores* function. The model was trained on data with specific characteristics of angle, subject, and background complexity, hence the testing will include various scenarios with different characteristics from the datasets to evaluate the model's ability and limitations in interpreting the sign language.

In this test, the characteristics of the parameters test are divided into; the subject's background, the environment background, and with the respective angle of the subject and object orientation towards the device's camera. The subject's background refers to the complexity of the pattern of the clothes that are worn by the subject, whilst the background refers to the environment background of the test.

The inclusion of various types of backgrounds aims to observe the effect of background of the subject and the environment on the model's performance in recognizing the object, however this study is not discussing the metrics that affect the performance in detail. The test can be described into several scenarios as follow:

Table 3.1. Testing Scenarios.

Test	Environment Background	Subject Background	Angle 1	Angle 2	Angle 3
Test 1	Similar	Similar	Lean Left	Straight	Lean Right
Test 2		Different			
Test 3	Different	Similar			
Test 4		Different			

Under Test 1 and Test 2, the model recognizes the object over a similar environment background to the dataset environment background, whereas the subject's background in Test 2 is differentiated.

 doi:10.20944/preprints202405.1601.v1

Under Test 3 and Test 4, the model performs the recognition of the object using a different environment background from the dataset environment background, whereas under Test 4, the subject's background is also differentiated. Each test covers all classes of the dataset, which are "hello," "yes," "no," "thanks," and "sorry." The test results are based on the recognition score of every attempt of each class. Each class has a maximum of five test attempts using the real time sign language recognition model in the web application using a desktop device.

Results and Discussion

4.1. Testing

The test was conducted using a Windows based desktop device within the web application. The model is successfully exported and implemented into a web application. This project also aims to build a web application that supports mobile devices. However, that is not the main aim of this project. The results will be explained more in the next section. The testing involved 330 annotated images of the dataset within 5000 steps of training. The testing was conducted within the Google Collaboratory environment using SSD MobileNetv1 FPN pretrained model and TensorFlow Object Detection API and models.

4.2. Results

As it has been mentioned in Chapter 3, this project performed four tests with at least 60 attempts being done in testing the model. The results of the test are shown in Table 4.1, Table 4.2, Table 4.3, and Table 4.4. The number of the scores are based on the score that is calculated by the model in measuring the successful rate of its model in recognizing the given sign language which is based on the detection scores. Each table represents the results of Test 1, Test 2, Test 3, and Test 4 sequentially with the respective angle of the subject and object orientation towards the device's camera.

Table 4.1. Test 1 Results.

Class	Orientation			Average Score
	Left	Straight	Right	
hello	87.00	95.20	49.60	77.27
no	83.40	88.60	61.00	77.67
yes	72.20	93.80	54.00	73.33
sorry	45.80	92.40	71.20	69.80
thanks	52.40	96.00	73.60	74.00
Total Average Score				**74.41**

Table 4.2. Test 2 Results.

Class	Orientation			Average Score
	Left	Straight	Right	
hello	66.40	83.00	67.20	72.20
no	0.00	73.00	13.00	28.67
yes	59.80	91.80	59.80	70.47

 doi:10.20944/preprints202405.1601.v1

sorry	15.00	16.60	12.20	14.60
thanks	24.60	96.00	84.40	68.33
Total Average Score				**50.85**

Table 4.3. Test 3 Results.

Class	Orientation			Average Score
	Left	Straight	Right	
hello	52.60	60.40	12.20	41.73
no	64.20	87.60	12.00	54.60
yes	0.00	92.00	18.40	36.80
sorry	64.80	86.80	44.80	65.47
thanks	0.00	93.00	27.40	40.13
Total Average Score				**47.75**

Table 4.4. Test 4 Results.

Class	Orientation			Average Score
	Left	Straight	Right	
hello	78.40	76.00	0.00	51.47
no	26.60	50.60	0.00	25.73
yes	0.00	80.20	14.80	31.67
sorry	27.20	86.60	0.00	37.93
thanks	90.20	96.00	82.80	89.67
Total Average Score				**47.29**

According to the previous chapter, Test 1 in Table 4.1 tested the model on five classes where the environment and the subject backgrounds are similar to the background from the dataset. The test also involves the various orientations of the object towards the camera. Table 4.2 shows the results of Test 2, which similarly uses a similar environment background from the dataset where the subject background is different. Test 2 also includes the various orientations of the object.

Table 4.3 and Table 4.4 show the results of Test 3 and Test 4 sequentially. It shows the results of the test where the model is tested using different environment background in various orientation. As discussed earlier, the scores are taken from the average value of the detection scores of the models from five attempts of each class.

From multiple tests conducted and the results were recorded in tables. The results show that Test 1 had the highest average score of 74 percent successful rate among all the tests. Test 2, Test 3, and Test 4 had a lower average score, but the exact percentages were not specified. It is also mentioned that the class “thanks” with a straight orientation produced the highest score across all the tests. This suggests that the model or algorithm being tested performed particularly well when recognizing the “thanks” class, and that the orientation of the object had a positive impact on the

 doi:10.20944/preprints202405.1601.v1

performance. Across the results, the highest score achieved is 96 percent. It indicates that under certain conditions the model can accurately predict and recognize the sign language.

4.3. Evaluation

Test 1 and Test 2 indicate that the model still can perform well in regard to the suitability of the environment and the orientation of the object. Across all the test, however, shows that the class with straight orientation outperforms other classes' scores. On the other hand, the worst performance was observed in Test 4, particularly when the test was conducted with a right orientation. This implies that the model or algorithm had difficulty recognizing objects in the "right" orientation, or that the training data for this orientation was not as diverse or representative. According to (Rosebrock, 2018) and (Bengio et al., 2016), there are multiple factors that drive the performance of deep learning models including the quality and the quantity of data, the architecture of the model, data preprocessing and augmentation, and optimization.

To elaborate further, the statement suggests that the majority of the collected dataset is homogeneous in terms of orientation. Specifically, most of the images in the dataset were taken with a straightforward orientation. This means that the majority of the images in the dataset depict the objects in a similar way, with the object facing directly forward. This homogeneity in the orientation of the images in the dataset can have a significant impact on the performance of a deep learning model that is trained on this data. Having a diverse training dataset ensures that the data provides more descriptive and distinct information for the model to learn, thus allowing the model to make more accurate predictions on new, unseen data (Gong et al., 2019). If the majority of the images in the dataset depict the objects in a straightforward orientation as depicted on the results table, the model may perform well when recognizing objects in this orientation but may struggle to recognize the same objects in other orientations. In addition to improving the accuracy of the model, there is also the potential to bring this technology into the field of cybersecurity (Hussain et al., 2021).

Author (Shorten & Khoshgoftaar, 2019) highlights the importance of data diversity in deep learning, specifically in the context of computer vision tasks, and how data augmentation can be used to improve the performance of these models by increasing the diversity of the training data. It also mentions that a significant challenge in using deep learning for computer vision tasks is to enhance the generalization capability of the models. This means that the ability of the model to produce accurate outcomes on new and unseen data (Mallick et al., 2023). Data augmentation is presented as an effective solution for this challenge by reducing the difference between the training and validation sets and any future testing sets (Anwesa et al., 2015).

Accordingly, it can be seen that the trained model performance is dependent on the diversity of the dataset. For instance, according to the results of the test, when the model is trained on a dataset where most of the images are taken with a straightforward orientation, the model may be less trained with many examples of objects in a right orientation. This means that the model will not have learned to recognize in a certain orientation. Consequently, the model may struggle to identify objects in unfamiliar orientations in real-world scenarios, leading to less accurate recognition. Additionally, this model may also need a larger dataset for training.

As previously mentioned, multiple factors impact the performance of the model, including the quantity, quality, and diversity of the data as well as the model architecture, optimization, preprocessing, and hyperparameter tuning. However, this paper will only discuss the impact of data and architecture on model performance. MobileNetV1 is considered a lightweight model with a low number of parameters, making it suitable for applications that require low computational cost while still maintaining high efficiency and performance. However, as highlighted by (Sandler et al., 2018), the model has certain limitations such as difficulty in handling high-dimensional feature maps which results in poor accuracy and performance due to its separable convolution architecture. Additionally, it also has a tendency to lose spatial information, leading to lower performance.

In concurrence with previous studies, (Ma et al., 2018) and (Kasera et al., 2019) have both emphasized that the utilization of separable convolution can result in memory constraints and negatively impact the computational efficiency of the model, thus reducing its overall performance.

 doi:10.20944/preprints202405.1601.v1

To address this limitation, the authors of these studies have proposed optimizations to the pointwise convolution component of the model. It is acknowledged that the efficient nature of MobileNetV1 presents a compelling advantage; however, a trade-off between efficiency, as characterized by the reduced use of computational resources, and performance is a prevalent issue that can be addressed through modifications to the model in the future work.

In concurrence with previous studies, (Ma et al., 2018) and (Kasera et al., 2019) have both emphasized that the utilization of separable convolution can result in memory constraints and negatively impact the computational efficiency of the model, thus reducing its overall performance. To address this limitation, the authors of these studies have proposed optimizations to the pointwise convolution component of the model. It is acknowledged that the efficient nature of MobileNetV1 presents a compelling advantage; however, a trade-off between efficiency, as characterized by the reduced use of computational resources, and performance is a prevalent issue that can be addressed through modifications to the model in the future work.

Conclusion

In conclusion, this research project has successfully developed a model for recognizing Malaysian Sign Language within a web application environment. The results of several tests conducted on the model show that it has performed well, with an accuracy of up to 96% in controlled environments. The performance of the model was found to depend on various factors such as the characteristics of the dataset and the model used. The main objective of this project was to propose a model that could be implemented as a Malaysian Sign Language Web Application and this goal has been successfully achieved. However, further research and improvement is needed to produce a better model that can produce more accurate results in recognizing Malaysian Sign Language. The evaluation of the performance of the model is not in-depth, and the author suggests that future work should be done in order to improve the model's performance.

In addition to the findings already discussed, it is important to note that the proposed model has the potential to greatly benefit the community of Malaysian Sign Language users by providing them with a tool for communication and accessibility. The web application environment in which the model is implemented also ensures that it can be easily accessed and used by a wide range of users.

Furthermore, the high accuracy achieved by the model in certain controlled environments indicates that it has the potential to perform well in real-world scenarios. However, it is important to note that the model's performance may be affected by factors such as lighting conditions and background noise, which were not considered in the tests. Therefore, further research should focus on evaluating the model's performance in more diverse and realistic environments.

Moreover, the author suggests that future work should focus on improving the model's performance by experimenting with different architectures and pre-processing techniques (Taj et al., 2022). Additionally, the model could be trained on a larger and more diverse dataset to improve its generalization capabilities. Furthermore, the author suggests to evaluate the model's performance on a larger number of test samples to achieve a more accurate evaluation of the model's performance.

In summary, this research project has successfully developed a model for recognizing Malaysian Sign Language within a web application environment. While the model has performed well in controlled environments, further research is needed to improve its performance and to evaluate its performance in more diverse and realistic environments. The proposed model has the potential to greatly benefit the community of Malaysian Sign Language users by providing them with a tool for communication and accessibility.

5.1. Social Impact

The existence of this platform will help the disabled to be more confident in socializing, while on the other hand it will help the society to give more space for the disabled to be more included. Furthermore, it gives more space for the disabled, especially in the working environment. Thus, this will not only be limited in giving the disabilities more opportunity in socializing but also provide wider opportunity to get employment. Accordingly, this project is also part of achieving the 10th

 doi:10.20944/preprints202405.1601.v1

SDG (Social Development Goals) of the United Nations in reducing the inequalities especially among the sign language users. Therefore, it also benefits people with disabilities to get border opportunities to find employment.

5.2. Future Work

The proposed model for sign language recognition has vast potential for future development and improvement. This model has the potential to become a more accurate sign language recognition web application. To achieve this, one possible area of focus would be to incorporate a larger and more diverse dataset of Malaysian Sign Language. This could greatly improve the accuracy of the model by allowing it to better recognize and interpret a wider range of sign language gestures. Another potential avenue for improvement would be to utilize more powerful graphics processing units to train the model. This would allow for a more robust training process and could lead to higher accuracy in sign language recognition.

Along with implementing cybersecurity in this field, sign language recognition technology could be used as a biometric authentication method. This will offer a unique and more secure alternative to traditional alphanumeric passwords. Utilizing sign language in this manner would address the vulnerabilities associated with traditional passwords, such as susceptibility to brute-force attacks. This could be a significant step forward in the development of more secure and effective authentication systems. Overall, the proposed model for sign language recognition has enormous potential for future development and growth, offering exciting possibilities for the future of both sign language interpretation and cybersecurity.

References

1. Alnaim, N. (2020). *Hand gesture recognition using deep learning neural networks*. Brunel University London.
2. Airehrour, D., Gutierrez, J., & Ray, S. K. (2015, November). GradeTrust: A secure trust based routing protocol for MANETs. In *2015 International Telecommunication Networks and Applications Conference (ITNAC)* (pp. 65-70). IEEE.
3. Anwesa Chaudhuri, A. C., & Sanjib Ray, S. R. (2015). Antiproliferative activity of phytochemicals present in aerial parts aqueous extract of Ampelocissus latifolia (Roxb.) Planch. on apical meristem cells.
4. Balakrishnan, S., Ruskhan, B., Zhen, L. W., Huang, T. S., Soong, W. T. Y., & Shah, I. A. (2023). Down2Park: Finding New Ways to Park. *Journal of Survey in Fisheries Sciences*, 322–338.
5. Fujiyoshi, H., Hirakawa, T., & Yamashita, T. (2019). Deep learning-based image recognition for autonomous driving. In *IATSS Research* (Vol. 43, Issue 4). https://doi.org/10.1016/j.iatssr.2019.11.008
6. Goldsborough, P. (2016). A Tour of TensorFlow Proseminar Data Mining. *Arxiv*.
7. Gouda, W., Almurafeh, M., Humayun, M., & Jhanjhi, N. Z. (2022, February). Detection of COVID-19 based on chest X-rays using deep learning. In *Healthcare* (Vol. 10, No. 2, p. 343). MDPI.
8. Hussain, S. J., Irfan, M., Jhanjhi, N. Z., Hussain, K., & Humayun, M. (2021). Performance Enhancement in Wireless Body Area Networks with Secure Communication. *Wireless Personal Communications*, *116*(1). https://doi.org/10.1007/s11277-020-07702-7
9. Humayun, M., Ashfaq, F., Jhanjhi, N. Z., & Alsadun, M. K. (2022). Traffic management: Multi-scale vehicle detection in varying weather conditions using yolov4 and spatial pyramid pooling network. *Electronics*, *11*(17), 2748.
10. Ioffe, S., & Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. *32nd International Conference on Machine Learning, ICML 2015*, *1*.
11. John, V., Boyali, A., Mita, S., Imanishi, M., & Sanma, N. (2016). Deep Learning-Based Fast Hand Gesture Recognition Using Representative Frames. *2016 International Conference on Digital Image Computing: Techniques and Applications, DICTA 2016*. https://doi.org/10.1109/DICTA.2016.7797030
12. Jhanjhi, N. Z., Brohi, S. N., Malik, N. A., & Humayun, M. (2020, October). Proposing a hybrid rpl protocol for rank and wormhole attack mitigation using machine learning. In *2020 2nd International Conference on Computer and Information Sciences (ICCIS)* (pp. 1-6). IEEE.
13. Kellogg, B., Talla, V., & Gollakota, S. (2014). Bringing gesture recognition to all devices. *Proceedings of the 11th USENIX Symposium on Networked Systems Design and Implementation, NSDI 2014*.
14. Leksut, J. T., Zhao, J., & Itti, L. (2020). Learning visual variation for object recognition. *Image and Vision Computing*, *98*. https://doi.org/10.1016/j.imavis.2020.103912
15. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C. Y., & Berg, A. C. (2016). SSD: Single shot multibox detector. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *9905 LNCS*. https://doi.org/10.1007/978-3-319-46448-0_2

 doi:10.20944/preprints202405.1601.v1

16. Lim, M., Abdullah, A., Jhanjhi, N. Z., Khan, M. K., & Supramaniam, M. (2019). Link prediction in time-evolving criminal network with deep reinforcement learning technique. *IEEE Access*, *7*, 184797-184807.
17. Mallick, C., Bhoi, S. K., Singh, T., Swain, P., Ruskhan, B., Hussain, K., & Sahoo, K. S. (2023). Transportation Problem Solver for Drug Delivery in Pharmaceutical Companies using Steppingstone Method. *International Journal of Intelligent Systems and Applications in Engineering*, *11*(5s).
18. Mitra, S., & Acharya, T. (2007). Gesture recognition: A survey. *IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews*, *37*(3). https://doi.org/10.1109/TSMCC.2007.893280
19. Mujahid, A., Awan, M. J., Yasin, A., Mohammed, M. A., Damaševičius, R., Maskeliūnas, R., & Abdulkareem, K. H. (2021). Real-time hand gesture recognition based on deep learning YOLOv3 model. *Applied Sciences (Switzerland)*, *11*(9). https://doi.org/10.3390/app11094164
20. Neethu, P. S., Suguna, R., & Sathish, D. (2020). An efficient method for human hand gesture detection and recognition using deep learning convolutional neural networks. *Soft Computing*, *24*(20). https://doi.org/10.1007/s00500-020-04860-5
21. Oudah, M., Al-Naji, A., & Chahl, J. (2021). Computer Vision for Elderly Care Based on Deep Learning CNN and SVM. *IOP Conference Series: Materials Science and Engineering*, *1105*(1). https://doi.org/10.1088/1757-899x/1105/1/012070
22. Riskhan, B., & Muhammad, R. (2017). Docker in Online Education: Have the Near-native Performance of CPU, Memory and Network? *International Journal of Computer Theory and Engineering*, *9*(4), 290–293. https://doi.org/10.7763/ijcte.2017.v9.1154
23. Riskhan, B., Safuan, H. A. J., Hussain, K., Elnour, A. A. H., Abdelmaboud, A., Khan, F., & Kundi, M. (2023). An Adaptive Distributed Denial of Service Attack Prevention Technique in a Distributed Environment. *Sensors*, *23*(14). https://doi.org/10.3390/s23146574
24. Ray, S. K., Sinha, R., & Ray, S. K. (2015, June). A smartphone-based post-disaster management mechanism using WiFi tethering. In *2015 IEEE 10th conference on industrial electronics and applications (ICIEA)* (pp. 966-971). IEEE.
25. Saha, S. (2015). A Comprehensive Guide to Convolutional Neural Networks — the ELI5 way. *Towards Data Science*.
26. Singhal, V., Jain, S. S., Anand, D., Singh, A., Verma, S., Rodrigues, J. J., ... & Iwendi, C. (2020). Artificial intelligence enabled road vehicle-train collision risk assessment framework for unmanned railway level crossings. *IEEE Access*, *8*, 113790-113806.
27. Taj, I., & Zaman, N. (2022). Towards industrial revolution 5.0 and explainable artificial intelligence: Challenges and opportunities. *International Journal of Computing and Digital Systems*, *12*(1), 295-320.
28. Yamashita, R., Nishio, M., Do, R. K. G., & Togashi, K. (2018). Convolutional neural networks: an overview and application in radiology. In *Insights into Imaging* (Vol. 9, Issue 4). https://doi.org/10.1007/s13244-018-0639-9
29. Zeng, Z., Gong, Q., & Zhang, J. (2019). CNN model design of gesture recognition based on tensorflow framework. *Proceedings of 2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference, ITNEC 2019*. https://doi.org/10.1109/ITNEC.2019.8729185