UNIVERSITI TEKNOLOGI MARA | Cawangan Melaka

PCMJ

Progress in Computing and Mathematics Journal

volume 1

https://fskmjebat.uitm.edu.my/pcmj/

Progress in Computing and Mathematics Journal
College of Computing, Informatics, and Mathematics
Universiti Teknologi MARA Cawangan Melaka, Kampus Jasin
77300, Merlimau, Melaka Bandaraya Bersejarah

PCMJ

Progress in Computing and Mathematics Journal

volume 1

Cawangan Melaka

Progress in Computing and Mathematics Journal (PCMJ)
College of Computing, Informatics, and Mathematics
Universiti Teknologi MARA Cawangan Melaka, Kampus Jasin
77300, Merlimau, Melaka Bandaraya Bersejarah

EDITORS

PCMJ
Progress in Computing and Mathematics Journal
volume 1

PREFACE

Welcome to the inaugural volume of the **Progress in Computing and Mathematics Journal (PCMJ)**, a publication proudly presented by the College of Computing, Informatics, and Mathematics at UiTM Cawangan Melaka.

This journal represents a significant step in our commitment to fostering a vibrant research culture, initially providing a crucial platform for our undergraduate students to showcase their intellectual curiosity, dedication to scholarly pursuit, and potential to contribute to the broader academic discourse in the fields of computing and mathematics. However, we envision PCMJ evolving into a beacon for researchers both nationally and internationally. We aspire to cultivate a space where groundbreaking research and innovative ideas converge, fostering collaboration and intellectual exchange among established scholars and emerging talents alike.

The manuscripts featured in this first volume, predominantly authored by our undergraduate students, are a testament to the hard work and dedication of these budding researchers, as well as the guidance and support provided by their faculty mentors. They cover a diverse range of topics, reflecting the breadth and depth of research interests within our college, and set the stage for the high-quality scholarship we aim to attract in future volumes.

As editors, we are honored to have played a role in bringing this journal to fruition. We extend our sincere gratitude to all the authors, reviewers, and members of the editorial board for their invaluable contributions. We also acknowledge the unwavering support of the college administration in making this initiative possible.

We hope that PCMJ will inspire future generations of students and researchers to embrace research and innovation, to push the boundaries of knowledge, and to make their mark on the world of computing and mathematics.

Editors
Progress in Computing and Mathematics Journal (PCMJ)
College of Computing, Informatics, and Mathematics
UiTM Cawangan Melaka

TABLE OF CONTENTS

PCMJ

Progress in Computer and Mathematics Journal (PCMJ)
volume 1 [October, 2024]
e-ISSN: 3030-6728
Website: fskmjebat.uitm.edu.my/pcmj

MOBILE APPLICATION FOR REAL TIME BABY SIGN LANGUAGE RECOGNITION USING YOLOV8

Siti Aishah Binti Idris
Universiti Teknologi Mara Kampus Jasin
aishahidris31@gmail.com

Dr. Ahmad Firdaus Bin Ahmad Fadzil
Universiti Teknologi Mara Kampus Jasin
firdausfadzil@uitm.edu.my

Article Info | Abstract

Nowadays, the use of baby sign language has increased among parents in recent years. This language utilizes simple hand gestures and motions as a medium to express the baby's needs, wants, and feelings. This is particularly important because babies have not yet developed the ability to speak. However, this language is still neither widely used nor a generally recognized mode of communication, especially for caregivers. This can lead to tantrums when the baby's needs are not understood or met promptly. Additionally, some parents have expressed difficulties in learning and remembering the sign. Hence, the development of real-time baby sign language recognition on mobile platforms could help to address these issues. This research will focus only on six basic baby sign languages that are commonly used in daily life. The model will be designed and developed using a deep learning algorithm, which is YOLOv8, the latest version of YOLO. This model can recognize and interpret baby signs based on visual input from the mobile phone's camera in real time, and also includes a dictionary feature that can be a learning tool for parents and caregivers. In the development phase, the dataset is pre-processed before the modeling process is done using YOLOv8, and deployed on Android platforms using Java and Kotlin languages in Android Studio. Functionality testing and accuracy testing have been conducted. The functionality test produced successful results for all test cases, while for accuracy testing, the model achieved an accuracy of 99.50% for Mean Average Precision, indicating its proficiency in recognizing each of the classes.

Received: February 2024
Accepted: August 2024
Available Online: October 2024

Keywords: Baby sign language; Sign Language Recognition; Real-time; Deep learning; YOLOv8; Mobile Application

INTRODUCTION

Nowadays, baby sign language is popular among hearing parents as they use this tool to enhance communication with their hearing children. This sign used simple hand gestures and motions to express the baby's needs, wants, and feelings before they were able to talk (Nadgeri & Kumar, 2022). Figure 1 shows the 25 most popular baby sign languages used by parents (Neuman, 2022). According to Dinh and Mueller (2018), parents speak to their babies in the usual way, which is by voicing the words, and at the same time make use of these signs. It is believed that by using these sign languages consistently, babies and toddlers can develop the ability to communicate before they speak.

Although the use of baby sign language has increased in recent years, it is still neither widely used nor a generally recognized mode of communication. Not all caregivers will be familiar with sign language. So, it may be difficult for them to anticipate their baby's requirements, which may result in tantrums as the baby's wishes are unmet. According to previous studies, the researcher claimed that it was challenging for parents to teach caregivers to understand the baby sign, especially if the caregivers did not know enough sign language to incorporate it into their children's daily routine (Perotta, 2017). It is only effective for the baby and the parents themselves if the parents have time, energy, and patience to teach sign language. In addition, according to the researcher, she reports that 50% of participants who are parents from the survey expressed frustration about the challenges of learning a new language and remembering the signs to keep up with the child's vocabulary development. In addition, some of the participants have reported that they find it difficult to learn baby signs due to the amount of time and energy required to consistently teach their children. Hence, due to a lack of understanding of the baby's wishes, this could lead to frustration.

Developing a baby sign language recognition model on a mobile platform can address this issue. It could help to overcome some of the difficulties associated with learning and remembering signs, while also recognizing and interpreting commonly used baby signs. The model will be designed and developed using a deep learning algorithm which is the YOLOv8 that can recognize and interpret baby signs based on visual input from the mobile phone's camera in a real time, providing caregivers with real-time feedback on their child's communication. Additionally, it will include vocabulary baby sign features as a learning tool for parents and caregivers. This research will focus on recognizing 6 commonly used and relevant baby signs, including “more”, “all done”, “change diaper”, “milk”, “eat” and “drink”.

PCMJ
Progress in Computer and Mathematics Journal (PCMJ)
volume 1 [October, 2024]
e-ISSN: 3030-6728
Website: fskmjebat.uitm.edu.my/pcmj

Figure 1: Baby Sign Language (Source: Neuman, 2022)

LITERATURE REVIEW

This section includes a literature review covering various aspects of the project, such as sign language recognition and related works utilizing deep learning methods. It also discusses the chosen deep learning method, which demonstrates the highest accuracy.

Sign Language Recognition

Sign language recognition provides a solution to the problems faced by sign language users, where it can recognize and interpret a series of hand gestures. According to Jiang and Ahmad (2019), there are two primary approaches in recognizing sign language: one involves the use of electromagnetic gloves and sensors that can detect hand shape, movements, and the orientation of the hand, while the other uses camera vision-based gesture recognition, known for its natural approach. Based on Nadgeri and Kumar (2019), the signs, tension, and pressure between fingers will be captured with sensor-based gloves, while for vision-based recognition, cameras are used to capture the signs, which are then processed and analyzed using image processing techniques. Camera vision-based gesture recognition is the most effective method in recognizing sign language, as it is not required to wear any equipment like gloves. It only relies on the camera to recognize and interpret hand sign gestures. This approach offers a more natural and intuitive interaction, as it eliminates the need for specialized equipment. Hence, in this project, camera vision-based will be used to recognize sign language.

Related Work

There is a significant amount of similar work that has already been done using a deep learning technique to recognize baby sign language. Each of the similar works will be compared to determine the most accurate and suitable method that can be used for this project. The dataset, method used and accuracy will be discussed.

Enhancing Infants Early Communication Through Sign Language Recognition System Using Raspberry Pi and Convolutional Neural Network

In 2023, Cerna and other researchers built and tested an automated sign language recognition system that can enhance early interaction between children and caregivers, thus enhancing children's quality of life and overall language development. The dataset was obtained from a baby sign language tutorial that featured participants with different hand sizes and skin tones. It also includes 8 different images that were taken from two hours of video. The method used is Internet of Things (IoT) and Raspberry Pi combined with deep learning technique which is Convolutional Neural Network (CNN). It achieved 94.6% accuracy for the system.

Deep Learning Based Framework for Dynamic Baby Sign Language Recognition System

This work was done in 2022 by Nadgeri and Kumar, where the researchers aimed to develop continuous baby sign recognition. The dataset consists of 34 baby signs. The researchers utilize deep learning techniques which are CNN and Long Short Term Memory (LSTM). It uses optimizer where Adam optimizer achieves 99%, while SGD achieves 75% of accuracy.

Deep-Hand: A Deep Inference Vision Approach of Recognizing of Hand Sign Language Using American Alphabet

This work was conducted by Alon et al. in 2021, where the researchers developed a system that is fast and more accurate for static hand gesture detection using the American

Sign Language (ASL) alphabet dataset. For the dataset, 7200 images of ASL were obtained in Kaggle, where each of the alphabet contains 300 images. It applies YOLOv3 and achieves 95.18% for training and 90.82% for validation.

Based on the three related works, it is indicated that works by Alon et al. (2021) that used YOLO as their network, provides the highest accuracy where it got 95.18% for training and 90.82% for validation with 7200 datasets compared to other methods. In addition, the researchers stated that, by using YOLO, it can detect hand sign gestures more accurately and faster. They added that YOLO can make predictions faster with a single network evaluation compared to systems using R-CNN that require thousands of network evaluations for a single image. Hence, YOLO will be used as a network in this project.

YOLO

According to Alon et al. (2021), You Only Look Once (YOLO) uses Darknet-53 which is a CNN that serves as its backbone and features extractor. This method is able to develop a system that can detect hand sign gestures more accurately and faster as the input images are only seen once through the neural network to recognize the image. This method separates the input image into different grids according to a specified grid size, and then it predicts the probability of the 14 desired objects in each grid. In a single run of the algorithm, it will predict every class and object bounding that are in the image (Dima & Ahmed, 2021). This method provides several advantages. Firstly, YOLO is known for its ability to make predictions with a single network evaluation, hence making it incredibly fast compared to other methods. Additionally, YOLO is capable of simultaneously detecting and classifying objects, providing efficient analysis. YOLO has evolved through several versions, each with its own set of enhancements and improvements. The latest version of YOLO at the time of this research is YOLOv8, where it includes numerous architectural and developer experience changes and improvements over the previous YOLO version. It comes with well-structured Python packages, developer-friendly features, and also an intuitive Command Line Interface (CLI). Hence, this project will utilize the benefits by training the sign language recognition model using YOLOv8. It will be used as this network model is significantly faster, more efficient and provides high accuracy in recognizing sign language compared to the previous versions.

METHODOLOGY

This section explains the methodology that will be used throughout the project. It thoroughly describes how the Mobile Application for Real-Time Baby Sign Language Recognition was developed using YOLOv8. There are several steps, including data collection, data pre-processing, modeling, and deployment.

Data Collection

A primary dataset that is taken in the form of images will be used for the model training, comprising various basic sign languages, including all done, eat, drink, more, change, and milk. Each sign comes with its own set of unique motions, adding complexity to the recognition process. The data is gathered from different age groups, ranging from kids to adults, as each individual may have produced different styles of gestures, especially for kids who have smaller hands. As this model was also built as a learning platform for caregivers and parents, a dataset from adults will also be used to ensure that the model can accurately understand and interpret sign language used by them. A total of 2400 images have been collected, with each class comprising 400 images. The images were taken from every angle and in every motion to ensure the model learns to recognize the sign from different viewpoints. The images have been stored in respective folders.

Data Preprocessing

Data preprocessing is a crucial step in the deep learning process, as real-world data is often messy and must be cleaned to enable the model to achieve high accuracy. For instance, in the real world, the sizes of images are frequently not uniform, which can impact both training time and performance. Performing this process can decrease training time and improve performance. The file name can be renamed in File Explorer by selecting all files and renaming them according to their class name to maintain consistency and standardization in the dataset, especially when dealing with 2400 images in this project. Roboflow was utilized for data annotation, resizing images, and splitting the dataset. In Roboflow, the images are randomly divided into training, validation, and testing sets based on a ratio of 70:20:10, respectively.

Modeling with YOLOv8

After data pre-processing has been completed, and the dataset has been exported to YOLOv8 format, the modeling process will be performed to train the YOLOv8 model on the annotated dataset. The goal of this process is to build a model that is capable of classifying new, unseen data based on the patterns learned during the training phase. For this project, it will be trained using the Python API as the syntax is simple and easy to understand. The data modeling was done in Google Colab, taking advantage of its cloud-based computing resources and GPU capabilities to expedite the training process. The model undergoes the training process, using 1680 images to train with the YOLOv8 model.

YOLOv8 can be divided into two main parts which are the backbone and the head. Figure 2 shows the layer of the model. The role of the backbone is to extract features from the input image, while the head is to get features from the previous section and predicts the bounding-box area, objectness scores, and class probabilities for the objects detected in the input image. The head of YOLOv8 consists of several convolutional layers followed by a series of fully connected layers.

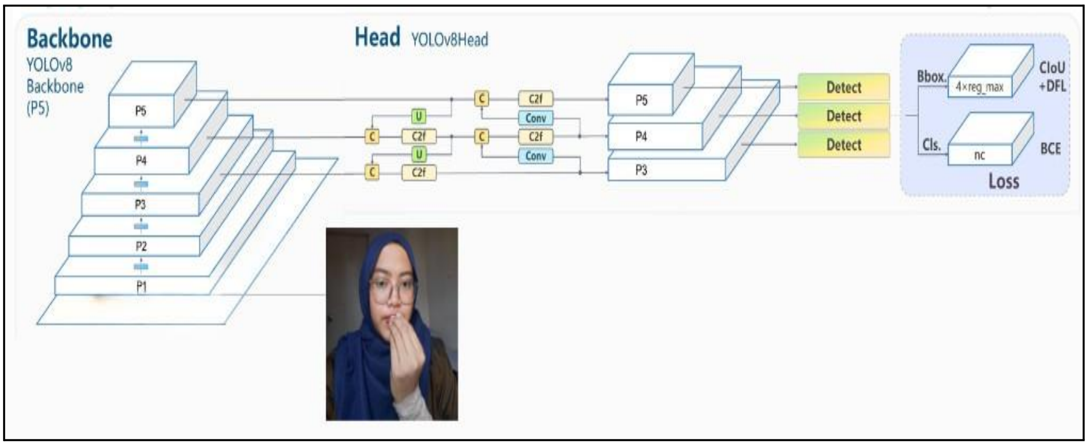

Figure 2: YOLOv8 Architecture

This model's recognition process begins by dividing the input image into NxN grid cells of equal shape. Each grid cell is responsible for predicting the class of objects within it, along with their probability values. The next step involves determining bounding boxes, which are rectangles that highlight the objects in the image. Each object will have its own bounding box, resulting in multiple bounding boxes within a single image. Yolo utilizes a single regression module to determine the attributes of these bounding boxes. Since a single object can have multiple grid box candidates for prediction, Intersection over Union (IoU) is used to identify relevant grid boxes. Grid boxes with an IoU value that below a specified threshold 28 are discarded. To select the boxes with the highest probability score, Non-Maximum Suppression (NMS) is used. Lastly, the created and trained model will determine the class of the sign in the image, returning its name and the accuracy of classification. Subsequently, the model undergoes a validation process, utilizing a dataset of 480 images to assess the quality of the trained model. The final evaluation step involves testing the model, where 240 test data are used after the model has been trained and validated. This step aims to assess how well the model performs on unseen data.

Deployment

Model deployment is the process of transforming a model into a system or application that is capable of making recognitions on new or unseen data. In this case, the model will be integrated into an Android mobile application with real-time features using Android Studio, along with additional features such as a dictionary and manual for the app. The app uses Java and Kotlin languages, where Java is employed to develop pages for the main menu, dictionary, and manual of the app, while Kotlin is used to develop the page for real-time sign recognition. The process of integrating a PyTorch-trained model for real-time sign language recognition into an Android app involves several steps. First, the model (.pt) must be converted to TensorFlow Lite (TFlite) format. Once converted, the TFlite model and label text file consist of 6 classes are imported into the Android app's assets folder and parsed in an object named 'Constants'. Next, TFlite and TFlite Support dependencies are added to the 'build.gradle' file in Android Studio to enable model execution on mobile devices. Following this, classes such as 'Detector.kt', 'OverlayView.kt', and 'MainActivity.kt' are created in Kotlin to handle setup, detection, bounding box processing, rendering bounding boxes and labels on live camera preview, and implementing real-time object recognition respectively. Additionally, the model's precision is reduced using INT8 quantization in Google Colab to enhance performance on mobile devices without compromising accuracy.

RESULT AND DISCUSSION

This section presents the results and findings obtained throughout the project. The analysis includes the result of training, validation, and testing to assess the model's performance. Additionally, accuracy testing and functionality testing are discussed in this section.

Training and Validation Analysis

Once the model has been trained and validated, learning curves, which are plots or graphs that show changes in learning performance over the training iterations, are displayed. Figure 3 shows the learning curves for training and validation performance. Better model performance is indicated by lower loss values, and worse performance is indicated by higher values. In this case, the training and validation objective of minimizing the loss value has been met. Precision, recall, and mAP50 (mean Average Precision) graphs also show higher prediction accuracy, indicating better performance.

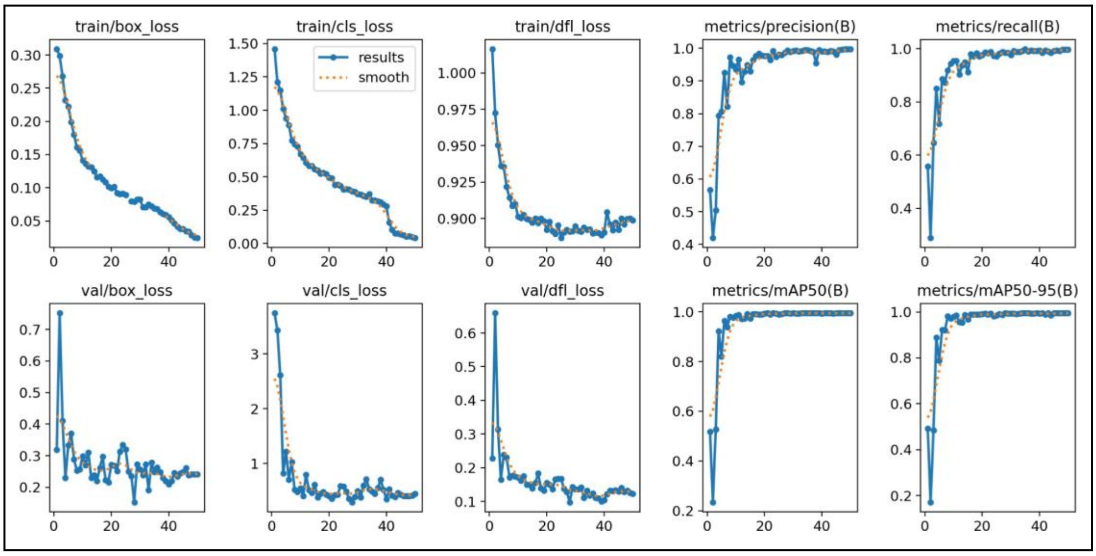

Figure 3: Learning curves of the model

Accuracy Testing

In this section, the results of the accuracy testing phase will be described. Additionally, confusion matrix and performance metrics will be discussed to evaluate the testing performance.

Confusion Matrix

A confusion matrix is a chart that basically shows how the model handles different classes. This chart includes an additional column and row representing the 'background', which does not belong to any existing class. Each cell of the matrix, excluding the background, indicates the number of detections for a given class that were assigned to each

PCMJ
Progress in Computer and Mathematics Journal (PCMJ)
volume 1 [October, 2024]
e-ISSN: 3030-6728
Website: fskmjebat.uitm.edu.my/pcmj

other class, providing insight into where the model might be confused. Figure 4 shows the confusion matrix for the testing dataset. As can be seen, class 'More' has 40 cases where its detections were correctly assigned to class 'More', indicating correct detection for all test images in this class, as the test set for this class contains a total of 40 images. Similarly, this applies to the other classes, demonstrating the model's capability to correctly recognize or detect signs across all six classes.

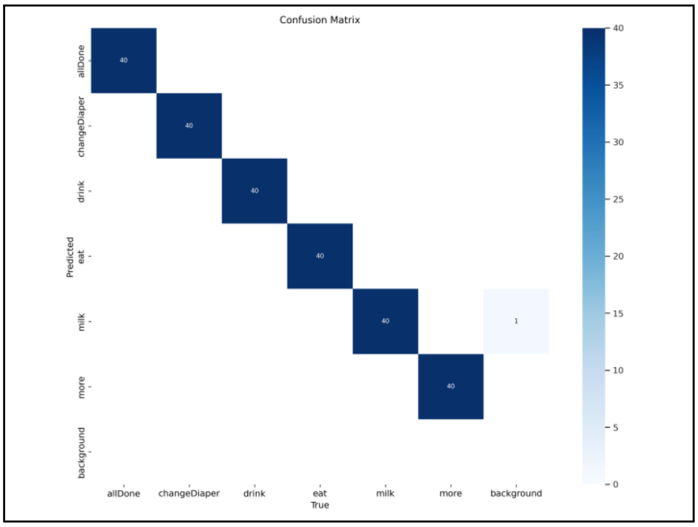

Figure 4: Confusion matrix

Evaluation Metrics

Additionally, after processing the testing data, the model returns a variety of performance metrics, including mean average precision (mAP), recall, and precision. Table 1 shows the function of the evaluation metrics for this model, and comes with the result. It shows that the model has achieved high accuracy, making it reliable for use.

Table 1: Evaluation result metric for all classes

Metric	Function	Result
Precision	Accuracy of the detected object	99.80%
Recall	Ability of the model to identify all instances of objects in the images	99.99%
Mean Average Precision	Average precision calculated at an IoU (Intersection over Union)	99.50%

Real Time Accuracy

Each class has undergone a real-time testing process to assess its ability to correctly predict the class. Table 2 shows the accuracy test being performed 20 times for each of the classes. The 'Drink', 'Eat', and 'More' classes receive lower accuracy due to lighting conditions, where it makes the model unable to detect the sign accurately. The overall average accuracy across all classes stands at 95.83%, indicating the model's high performance in class prediction tasks.

Table 2: Evaluation result metric for all classes

Class	Total number of test	Total number of correct test prediction	Percentage Accuracy (%)
All Done	20	20	100
Change Diaper	20	20	100
Drink	20	18	90
Eat	20	20	100
Milk	20	18	90
More	20	19	95
Total	120	115	Average(95.83)

Functionality Testing

Functionality testing is performed to test whether all of the features and components in the application are working as expected without any errors and run smoothly. Table 3 shows the functionality testing for three features in the app, including View About App, Sign Recognition, and View Dictionary. It shows that all of the features successfully passed the test and performed as expected.

Progress in Computer and Mathematics Journal (PCMJ)
volume 1 [October, 2024]
e-ISSN: 3030-6728
Website: fskmjebat.uitm.edu.my/pcmj

Table 3: Result of test cases

Features	Expected Output	Result
View About App	About App page can be displayed and the button to go to the next page is working well.	Passed
Sign Recognition	The camera can be successfully run without any lagging issue, and the model can recognize the hand gesture and display its bounding box along with the label sign. If the model does not recognize it, the bounding box and the label sign will not be displayed.	Passed
View Dictionary	The sign language dictionary contains images of each sign, which can be clicked to navigate to another page with a demonstration video showing how to make the sign. The video plays automatically upon accessing the page.	Passed

Summary

In summary, the project has successfully achieved its objectives in designing, developing and testing a real-time baby sign language recognition model using YOLOv8 in mobile platform. The results obtained showed a consistently high accuracy rate of 99.50% for Mean Average Precision, and overall average for all classes with 95.83% indicating that the model performs well and is reliable for use. However, there are some limitations to this model, including a limited class of sign language, covering only 6 signs, and environmental factors where the accuracy and reliability of the recognition model are heavily dependent on the quality and consistency of the lighting sources in the user's surroundings. This model can be further improved by increasing the diversity of baby sign classes, applying data augmentation and noise reduction techniques. Hopefully, this model can benefit caregivers, parents, and infants by facilitating communication and enhancing accessibility to sign language interpretation in everyday situations.

REFERENCES

Adeyanju, I. A., Bello, O. O., & Adegboye, M. A. (2021). Machine learning methods for sign language recognition: A critical review and analysis. *Intelligent Systems with Applications*, *12*, 200056. https://doi.org/10.1016/j.iswa.2021.200056

Alon, H. D., Ligayo, M. A. D., Melegrito, M. P., Franco Cunanan, C., & Uy, E. E. (2021). Deep-Hand: A Deep Inference Vision Approach of Recognizing a Hand Sign Language using American Alphabet. *Proceedings of 2nd IEEE International Conference on Computational Intelligence and Knowledge Economy, ICCIKE 2021*, 373–377. https://doi.org/10.1109/ICCIKE51210.2021.9410803

Al-Shaheen, A., Çevik, M., & Alqaraghuli, A. (2022). American Sign Language Recognition using YOLOv4 Method. *International Journal of Multidisciplinary Studies and Innovative Technologies*, *6*(1), 61. https://doi.org/10.36287/ijmsit.6.1.61

Aman, P., Avinash, K., Priyam, Priyanshu, G., & Gunjan, C. (2022). Real Time Sign Language Detection. *International Journal for Research in Applied Science and Engineering Technology*, *10*(5), 3522–3525. https://doi.org/10.22214/ijraset.2022.42961

Bragg, D., Koller, O., Bellard, M., Berke, L., Boudreault, P., Braffort, A., Caselli, N., Huenerfauth, M., Kacorri, H., Verhoef, T., Vogler, C., & Morris, M. R. (2019). Sign language recognition, generation, and translation: An interdisciplinary perspective. *ASSETS 2019 - 21st International ACM SIGACCESS Conference on Computers and Accessibility*, 16–31. https://doi.org/10.1145/3308561.3353774

Cerna, P. D., Ututalum, C. S., Evangelista, R. S., Darkis, A. T., Asiri, M. S., & Muallam-Darkis, J. A. (2023). Enhancing Infants Early Communication through Sign Language Recognition System using Raspberry Pi and Convolutional Neural Network. *Proceedings of the 2023 2nd International Conference on Electronics and Renewable Systems, ICEARS 2023*, 979–983. https://doi.org/10.1109/ICEARS56392.2023.10085246

Chen, H., Wang, Y., Guo, T., Xu, C., Deng, Y., Liu, Z., Ma, S., Xu, C., Xu, C., & Gao, W. (2021). Pre-trained image processing transformer. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 12294–12305. https://doi.org/10.1109/CVPR46437.2021.01212

Christian, J., Patrick, Z., & Kai, H. (2021). Machine Learning and Deep Learning. *Ingeniare*, *29*(2), 182–183. https://doi.org/10.4067/S0718-33052021000200180

Dayana, L., & Rana, M. E. (2022). Recommendations for Developing a Sign Language Recognition Application for Malaysia. *2022 IEEE 2nd International Conference on Mobile Networks and Wireless Communications, ICMNWC 2022*, *December 2022*, 1–8. https://doi.org/10.1109/ICMNWC56175.2022.10031774

Dima, T. F., & Ahmed, M. E. (2021). Using YOLOv5 Algorithm to Detect and Recognize American Sign Language. *2021 International Conference on Information Technology, ICIT 2021 - Proceedings*, 603–607. https://doi.org/10.1109/ICIT52682.2021.9491672

Dinh, N., & Mueller, V. (2018). The Efficacy of Baby Sign Language in Reducing Problem Behaviors in Typically Developing Children. In *ProQuest Dissertations and Theses*. https://www.proquest.com/dissertations-theses/efficacy-baby-sign-language-reducing-problem/docview/2055713431/se-2?accountid=15299%0Ahttps://csuc-uoc.primo.exlibrisgroup.com/openurl/34CSUC_UOC/34CSUC_UOC:VU1?genre=dissertations&atitle=&author=Dinh%2C+Nguy

Ivanska, L., & Korotyeyeva, T. (2022). Mobile real-Time gesture detection application for sign language learning. *International Scientific and Technical Conference on Computer Sciences and Information Technologies*, *2022-Novem*, 511–514. https://doi.org/10.1109/CSIT56902.2022.10000440

Jiang, X., & Ahmad, W. (2019). Hand gesture detection based real-time American sign language letters recognition using support vector machine. *Proceedings - IEEE 17th International Conference on Dependable, Autonomic and Secure Computing, IEEE 17th International Conference on Pervasive Intelligence and Computing, IEEE 5th International Conference on Cloud and Big Data Computing, 4th Cyber Scienc*, 380–385. https://doi.org/10.1109/DASC/PiCom/CBDCom/CyberSciTech.2019.00078

Nadgeri, S., & Kumar, A. (2022). Deep Learning Based Framework for Dynamic Baby Sign Language Recognition System. *Indian Journal of Computer Science and Engineering*,*13*(2), 550–563. https://doi.org/10.21817/indjcse/2022/v13i2/221302127

Nadgeri, S., & Kumar, A. (2019). An Image Texture based approach in understanding and classifying Baby Sign Language. *2019 2nd International Conference on Intelligent Computing, Instrumentation and Control Technologies, ICICICT 2019*, 854–858. https://doi.org/10.1109/ICICICT46008.2019.8993381

Neuman, A. (2022). How to Teach Baby 25 Key Words in Baby Sign Language. *www.thebump.com*. https://www.thebump.com/a/how-to-teach-baby-sign-language

Olorunshola, O. E., Irhebhude, M. E., & Evwiekpaefe, A. E. (2023). A Comparative Study of YOLOv5 and YOLOv7 Object Detection Algorithms. *Journal of Computing and Social Informatics*, *2*(1), 1–12. https://doi.org/10.33736/jcsi.5070.2023

Ong, N. T. J., Ali, S. H. A., & Yusoff, A. H. M. (2019). Employment of baby sign language in interactive learning for hearing-impaired people in Malaysia. *Indonesian Journal of Electrical Engineering and Computer Science*, *18*(1), 361–367. https://doi.org/10.11591/ijeecs.v18.i1.pp361-367

Perrotta, L. (2017). *The Parental Perspective of Perceived Benefits of Baby Sign Language in Preverbal Children* [University of North Carolina at Chapel Hill]. https://cdr.lib.unc.edu/concern/honors_theses/tm70n045d

PCMJ

Progress in Computing and Mathematics Journal

اونيۏرسيتي تيكنولوݢي مارا
UNIVERSITI TEKNOLOGI MARA
Cawangan Melaka

eISSN 3030-6728
9 773030 672004