Group O

Real-Time ASL Prediction Educational Game

Rohaan Aslam: 21017718
Arthur Milner: 21035478
Benjamin Hussey: 21022768

Abstract

Sign language recognition is a well-covered but challenging problem, primarily the challenge of this problem comes from the many similar signs with vastly differing meanings. We used our own dataset, landmark detection, and a CNN to create a model that accurately predicts a sign in real-time using a webcam. Our dataset consisted of 600 images for each respective class/sign, and included a variety of images from each member's hand included. We also heavily experimented to improve our understanding of our model to ensure it was optimal. The result of this project showcases the use of CNNs as an effective aid in sign-language detection, particularly when assisted with landmark detection, resulting in a relatively simple CNN and small dataset performing very efficiently.

1 Introduction

The basis of our research and project is to provide an educational game for the basics of ASL (American Sign-Language), primarily aimed at children. In 2021 there was an estimated 1 million speakers worldwide using ASL as their primary language **(Garcia, J., 2021)**, because of this popularity we felt it was appropriate to have ASL as the target of our project.

We believe ASL and sign-language in general is very much undertaught as "over 1.5 billion people globally" **(World Health Organisation, 2023)** suffer from hearing loss, yet the average person knows little on the subject **(Bin, L. Y. et al, 2019)**, hence our motivation for undertaking this as our project. The scope of the project is gamifying real-time hand detection, with the objective to perform the ASL equivalent of the word displayed on-screen, with a correct sign denoting a +1 to the users score. The system's application can be for the teaching of both hearing and non-hearing individuals.

This project has helped us gain vast knowledge on applying machine learning algorithms and methods to real world use cases. As a group we gained experience with CNNs, computer vision (OpenCV), landmark detection (MediaPipe), and of course the steps taken towards the actualization and optimization of the model produced by the aforementioned technologies.

The project's result is fast and accurate real-time sign-language detection, gamified to give it a very real potential for educational use within classrooms. We believe the agreed-upon scope of the project was covered successfully.

2 Related Work

2.1 ROI & CNN

One work that shares many similarities with our project is a proposed study on developing a system aiming to "recognize static sign gestures and convert them into corresponding words" (Tolentino, L. K. et al, 2019, p.821). The study suggests utilizing an ROI to capture the images, which are then converted into a HSV colour format with half of the dataset being flipped horizontally. Classification of the images is then performed using a CNN model. Our project also uses some of the same approaches as outlined in the aforementioned work such as the ROI and CNN, however, we chose not to alter the colour of our static images and we instead process them in their RGB format. We also do not flip any of our training images as the placement of the ROI means all signs must be made by the user's right hand.

Figure 1: Sample dataset from Tolentino, L. K. et al, 2019, p.823

2.2 System Evaluation

Our project is also tested using a similar approach to the work above, due to **Tolentino (2019)** also summarizing the effectiveness of his study using Likert's scale. To measure the system's effectiveness in this way, several people must score the project on selected metrics such as functionality and reliability, on the scale of "poor, fair, average, good, and excellent" (Tolentino, 2019). This evaluation method provides quantitative values for the performance of the project based off potential user's impressions.

TABLE VI: SUMMARY OF EVALUATION RESULTS

	No. of evaluator	No. of questions	Total Score	Goal Score	Approval Percentage
Functionality	50	3	642	750	85.86
Reliability	50	2	433	500	86.86
Usability	50	4	901	1,000	90.1
Efficiency	50	1	215	250	86
Learning Impact	50	1	235	250	94
Total score:					88.46

Figure 2: Evaluation from Tolentino, L. K. et al, 2019, p.826

2.3 MediaPipe

One previous work that uses MediaPipe within a sign language detection system is "A real-time sign language conversion system" **(Jamwal, et al, 2022)**. In relation to our project, MediaPipe is also used in this example to extract 21 landmarks on the detected hands, allowing the model to then be trained on these data points as opposed to the images themselves. However, in contrast to our use of a CNN classifier, **Jamwal (2022)** describes using an SVM to classify the hand gesture being made, leading to an accuracy of 93.7%.

3 Data

Figure 3: Static ASL Words (Tolentino, L. K. et al, 2019, p.822)

3.1 Data Identification

Firstly, we made sure to agree upon the words we will collect data for, using Figure 3 as reference, we decided 10 words to be an acceptable and beginner level number of signs to capture. We selected commonly used words as we considered their usefulness to learn. The selected words are as follows:

"Hello", "I Love You", "Yes", "Stop", "Calm", "Why", "Fine", "I Hate You", "Sorry" and "Money"

3.2 Dataset Collection

To collect our dataset, all three members took photographs in which they were performing the required signs, data collection was made easy using OpenCV and a region of interest. This meant the collected images consisted of just the data the model would need, in this case the hand performing the desired sign.

Figure 4: Showing an example image from the dataset before landmarking

We simply appended each member's images, collecting 200 images per member for each class totalling at 600 images per class. Collecting images with multiple hands & skin colours for each class gave our dataset variety. The hope is that the variety would combat the problem of overfitting to a single type of hand, especially since the game would be used by many different hand sizes and skin tones in a real-world application. We decided upon the size of our dataset following research and experimentation on different data sizes for training a CNN **(C. Luo, et al., 2018)**, whilst this research concluded a larger number of images consistently improved performance, the inclusion of landmark detection in our project meant we found 600 images to be suitable. A benefit of collecting our own data for the problem is that we didn't have to consider ethical data collection as we were all happy to be used as part of the dataset.

3.3 Pre-processing

To pre-process our images, we applied landmark detection on our dataset of images using Google's MediaPipe Hand Landmarker. This tool localizes the key points of the hands in an image to render visual effects over the hand **(MediaPipe, 2022)**. By processing the hand sign images through MediaPipe, the 21 predefined landmarks were identified and extracted. The x & y coordinates of each landmark were collected, and their values were then normalized to create a detailed dataset. This dataset consists of 42 values (21 x,y pairs) per image capturing the spatial information of the landmarks. The corresponding labels were then assigned based on their respective directory name (0,1,2...) for each category. This approach allowed us to precisely classify the shape of each hand due to the 21 landmarks being predefined without bias to external factors (i.e., lighting, skin colour).

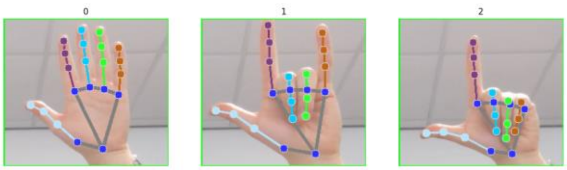


Figure 5: Examples from the data after landmarking

3.4 Pickling

Pickling enables serialization of Python variables **(Python)** such as our pre-processed data stored in a matrix and their corresponding labels in a vector for each entry in the matrix. By storing the pre-processed information in a dictionary of the format 'data': data and 'labels': label, it can be saved as a .pickle file for efficient retrieval to save pre-processing time. We used Pickling within our project for this very reason.

4 Methods

4.1 Region of Interest

One technique that we utilized to help us solve the problem more effectively was a region of interest (ROI). An ROI "is a portion of an image that you want to filter or operate on in some way" **(MathWorks, 2023)**. The main advantage associated with specifying a region of interest within the images transferred by the video feed, is to avoid processing any irrelevant data, consequently accelerating the rate of processing **(Zhang, Q and Xiao, H, 2008)**.

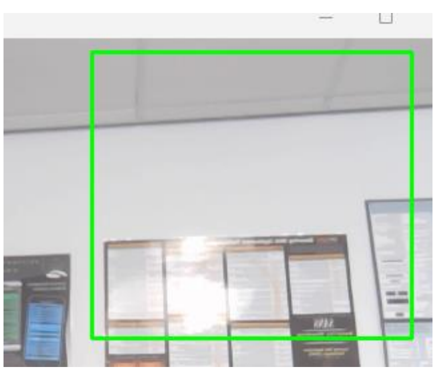


Figure 6: 1ROI as seen in OpenCV.

4.2 CNN

A Convolutional Neural Network (CNN) is a deep learning technique used for processing grid-like data, such as images or sequences **(Wu, 2017)**. It comprises multiple layers, including convolutional layers for pattern detection, pooling layers for dimensionality reduction, and fully connected layers for feature combination and output generation **(Wu, 2017)**. CNNs are used in ASL detection as they effectively process and learn patterns from the landmark data extracted by MediaPipe, enabling the recognition of distinct hand gestures and signs. Their ability to capture spatial information and specific features makes them well-suited for ASL detection.

4.3 Alternative Approaches

Before switching to landmarking, we briefly considered and experimented using thresholding/HSV on the images. During the initial data collection, we found that this approach was highly sensitive to lighting conditions, skin colour & background noise which doesn't align with our objective of real-time detection to be used anywhere and by everyone. Additionally, the similarity between certain hand signs posed a challenge to discern them requiring a more complex model. As a result, we began experimenting with hand landmarking and discovered it was more effective and suitable for addressing our problem.

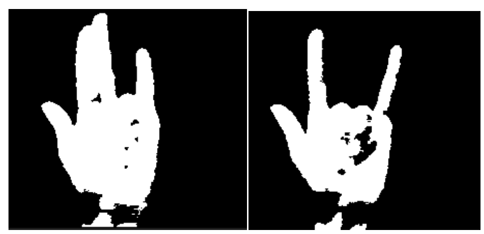


Figure 7: Displaying Similar Thresholded Signs with a plain background2

Another alternative approaches we considered to solve the problem included Random Forests and Support Vector Machines (SVMs). However, these methods often require manual feature extraction with would have been time consuming and could have potentially reduced accuracy, as opposed to the automatic feature learning capabilities of CNNs. Additionally, SVMs are often seen to struggle in high dimensional feature spaces, even with the use of kernels (Quinn and Olszewska, 2019), and random forests are prone to overfitting which could prevent our model being effective at real time classification (Ajay, et al, 2021).

4.4 Performance Metrics

Our model's accuracy was assessed using various CNN performance metrics, such as a confusion matrix, classification report (F1-score, precision, recall), and validation/training loss and accuracy graphs. These metrics offer a thorough evaluation of the model's ASL classification capabilities.

Confusion matrix and graphs provide visual insight, while the classification report shows detailed accuracy and F1-scores per class. Although the CNN lacks the displaying of real-time confidence scores for predictions, testing the model against OpenCV & MediaPipe for real-time sign detection reflects the system's effectiveness in accurate sign predictions.

4.5 Ablation Study

A further method we applied is an ablation study, an ablation study is "a scientific examination of a machine learning system in order to gain insight on the effects of its building blocks on its overall performance" **(Sheikholeslami, S., 2019)**. We used an ablation study to gain greater understanding of the individual layers within our CNN and consequently our model. The downside of performing this study is the fact it is computationally expensive to train so many different models.

5 Experiments

5.1 Initial Design and Optimisation

Our initial CNN design contained 7 layers with one being used for pooling and two convolutional layers as based off Tolentino's (2019) work and adapted to our input size. The convolutional layers are used to do the main computation within the network by performing dot product between the learnable kernel and a "restricted portion of the receptive field" while the pooling layers reduce the spatial size of the representation of the data (Mishra, 2020). We also outline the filter size which is used to determine the size of the kernel applied over the data (Sahoo, 2018).

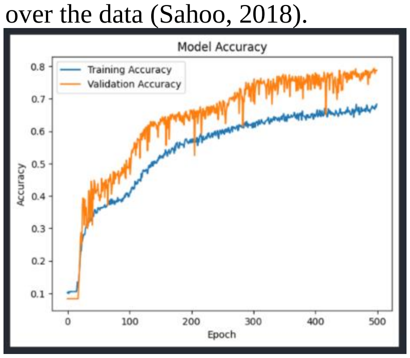
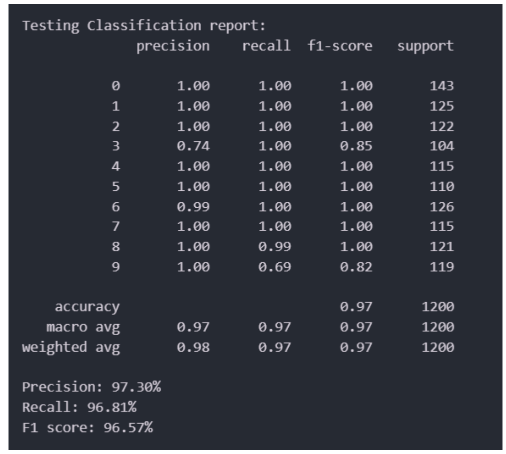


Figure 8: Showing Initial CNN's Performance

However, we found the accuracy of our initial CNN to be much lower due to the model underfitting to the dataset with scores of around 80% on the test data. To remedy this issue of a low accuracy score, we altered the size of the filters in the network from 2x2 to 3x3, which helped to prevent the model underfitting to the data. Furthermore, we then removed the second of the convolutional layers as the small input size means the model does not require a high complexity to capture the macrostructure of the dataset and we wanted to avoid any possibilities of overfitting. These adjustments improved our model's accuracy to scores of around 96%.



Testing Classification report:

	precision	recall	f1-score	support
0	1.00	1.00	1.00	143
1	1.00	1.00	1.00	125
2	1.00	1.00	1.00	122
3	0.74	1.00	0.85	104
4	1.00	1.00	1.00	115
5	1.00	1.00	1.00	110
6	0.99	1.00	1.00	126
7	1.00	1.00	1.00	115
8	1.00	0.99	1.00	121
9	1.00	0.69	0.82	119
accuracy			0.97	1200
macro avg	0.97	0.97	0.97	1200
weighted avg	0.98	0.97	0.97	1200

Precision: 97.30%
Recall: 96.81%
F1 score: 96.57%

Figure 9: Showing adapted CNN model scores

5.2 Model Challenges

A failure of our model we encountered during our experimentation was the fact that it struggled to identify signs where the hand was at a considerable distance from the webcam, this isn't a huge issue but it of course isn't ideal, we attempted to improve the accuracy from a distance by using a dataset with images both close up and far away but the model still struggled so we decided to just tune the model/dataset for closer distances. A further failure of our model is the ability to recognise two hands, we settled on using an error message if two hands are detected and focused only on single hand signs. We explored the use of two models and selectively activating the appropriate model depending on one or two hands being present. However, despite our efforts we were unable to achieve a satisfactory level of effectiveness that would warrant its integration in our final application.

Despite our model's great accuracy, there were still some slight issues with similar signs, as shown by the confusion matrix in figure 10. Classes 1 and 2 (figure 11) are very similar, and consequently discovered it was hard to get an accuracy relative to the other more distinct classes, the accuracy remains high, but it was something we found a struggle throughout our experimentation.

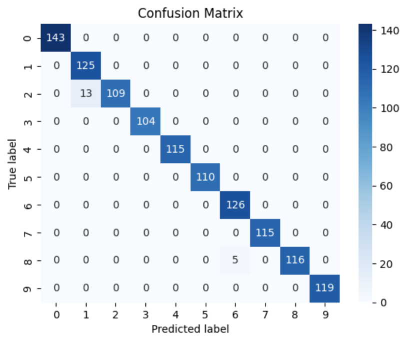


Figure 10: Confusion matrix of current model

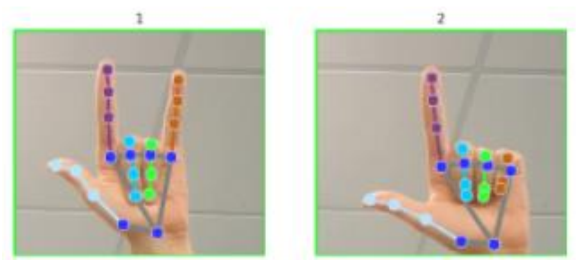


Figure 11: Similarity of classes 1 & 2

5.3 Applying Ablation Study

Once we had a model we were happy with, we performed an ablation study. To do this we removed the dropout layer, convolutional layer and pooling layer one at a time and observed the impact of each respective removal.

The removal of the dropout layer saw great results, the model now performs with 100% train and test accuracy, suggesting that the dropout layer was causing the model to underfit by dropping out too many useful neurons. This is also evidenced by the fact the training accuracy was lower than the test accuracy before removing this layer, because of this improved performance we decided to move forward with the dropout layer removed after concluding our ablation study.

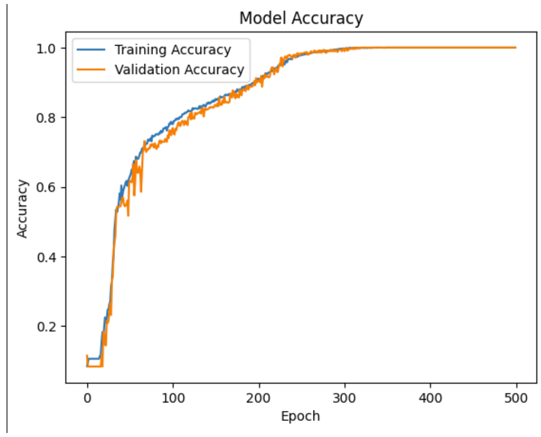


Figure 12: Graph showing improved performance after removing dropout layer

After removing the convolutional layer, as perhaps expected, performance takes a huge dip in accurate prediction. The confusion matrix below shows the extent of this with its overall failure to correctly predict most of the time, concluding a test accuracy of just 46.67%. This suggests the convolutional layer is key in allowing our model to accurately extract the features of the dataset and therefore will remain in our final model.

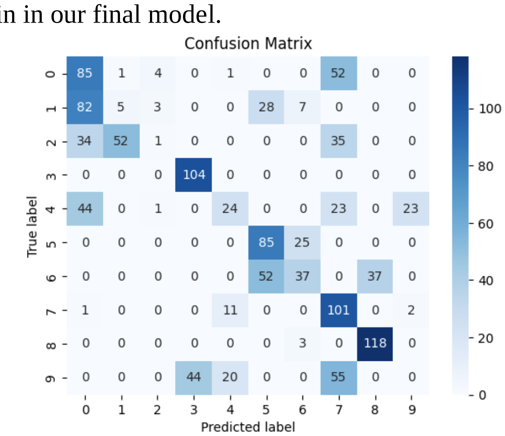


Figure 13: Confusion matrix showing performance after removing convolutional layer

Removing the pooling layer caused very little change in performance, consequently we decided to keep it for the final model due to its ability to reduce the time complexity of training and there being no obvious case for its removal.

Below are the diagrams which show our final model's structure and overall performance:

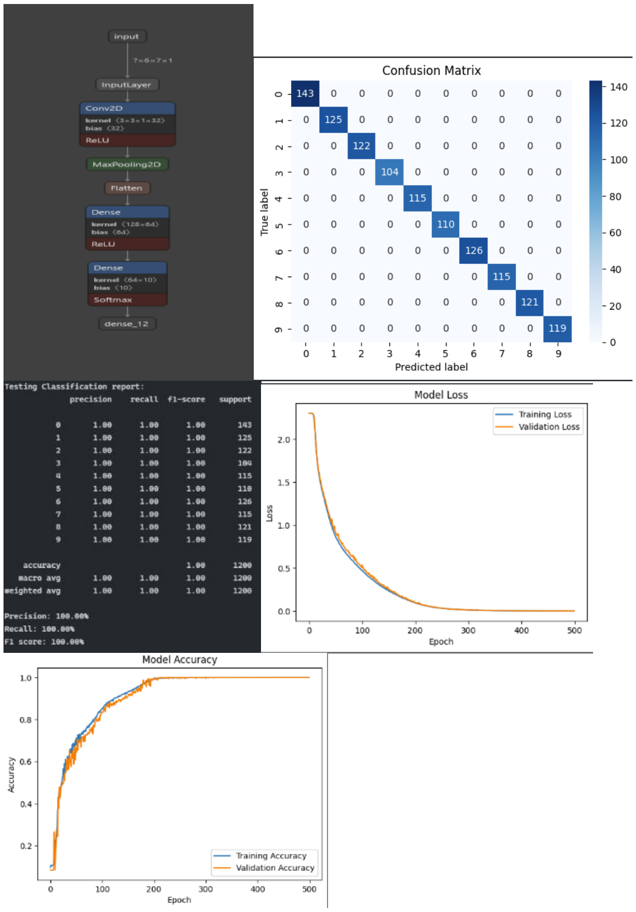


Figure 14: Diagrams showing final model's structure & performance

5.4 Success Evaluation

To gauge the effectiveness of the application we gathered individuals unfamiliar with the project to play the game and asked them to review how they found the experience. Our goal was to try identify whether they believed it effectively educated them and its ease of use. The majority believed it has potential for real-world use as an educational tool and found the game easy to use, suggesting our project was successful in its original goal. The result of this testing is found below, inspired by the approach from **Tolentino (2019)**:

Section	No. of People Questioned	Average Score (Out of 10)
Functionality	5	7
Reliability	5	8
Usability	5	7
Efficiency	5	8
Learning Impact	5	8

Figure 15: Result of testing user experience

6 Conclusion

6.1 Potential Improvements

Whilst we are happy with the project, many improvements which were not in our original scope are possible. An obvious example would be adding more words to detect, expanding upon this we could have created a separate model for detecting the ASL alphabet and made a separate mode for users to spell out words. Further improvements could include different models for different variations of sign language, such as British Sign Language, French Sign Language and more. We could have also applied more styling to the project to make it appear more visually appealing and professional.

6.2 Team Learnings

Overall, a lot of valuable insight into the training and application of machine learning methods/algorithms was gained by all members of the team, particularly the training and optimization of a CNN. More importantly, a great appreciation of the power of machine learning in real-world uses was seen by all, as we saw first-hand just how versatile this area of study is. It was not without difficulty, however, that we arrived at this appreciation as a lot of experimentation was required along the way to get a grasp of the concepts we were utilising. Finally, below are some images of the final working project, showcasing the scoring system and accurate sign prediction in an uncontrolled environment:

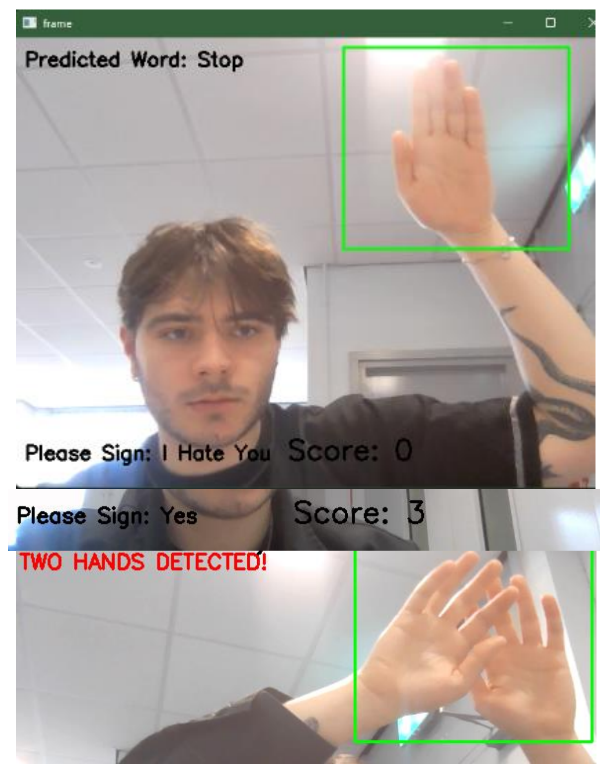


Figure 16: Screenshots of the working project

References

- Ajay, et al. (2021). *Indian sign language recognition using random forest classifier. In 2021 IEEE International Conference on Electronics, Computing and Communication Technologies,* Bangalore, 09-11 July 2021. IEEE [online]. Available from: https://ieeexplore.ieee.org/abstract/document/9622672 [Accessed 03 May 2023].
- Bin, L. Y. et al. (2019) Study of Convolutional Neural Network in Recognizing Static American Sign Language, *2019 IEEE International Conference on Signal and Image Processing Applications (ICSIPA),* IEE, Kuala Lumpur. IEEE Xplore [online]. Available from: https://ieeexplore.ieee.org/document/8977767 [Accessed 22 April 2023].
- C. Luo, X. Li, L. Wang, J. He, D. Li and J. Zhou (2018), How Does the Data set Affect CNN-based image Classification Performance?. *5th International Conference on Systems and Informatics (ICSAI)* [online]. 2018, pp. 361-366. [Accessed 29 April 2023]
- Garcia, J. (2021) How many people know sign language?, *Sign Station.* Available from: https://signstation.org/how-many-people-know-sign-language/ [Accessed 23 April 2023]
- Jamwal, A. et al. (2022) *Real Time Conversion of American Sign Language to text with Emotion using Machine LearningIn: Proceedings of the Sixth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC-2022).* [online]. 2022 Sixth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), Dharan 12 November 2022, IEEE. Available from: https://ieeexplore.ieee.org/document/9987362 [Accessed 29 April 2023].
- MathWorks (2023). *ROI-Based Processing – MATLAB & Simulink*. Available from: https://www.mathworks.com/help/images/roi-based-processing.html [Accessed 23 April 2023].
- Mishra, M (2020) *Convolutional Neural Networks, Explained.* Available from: https://towardsdatascience.com/convolutional-neural-networks-explained-9cc5188c4939 [Accessed 01 May 2023].
- MediaPipe (2022) *Hand landmarks detection guide.* Available from: https://developers.google.com/mediapipe/solutions/vision/hand_landmarker#:~:text=The%20MediaPipe%20Hand%20Landmarker%20task,visual%20effects%20over%20the%20hands. [Accessed 26 April 2023].
- Python (no date). Python Object Serialization. Available from: https://docs.python.org/3/library/pickle.html. [Accessed 26 April 2023].
- Quinn, M., and Olszewska, J. I,. (2019). *British sign language recognition in the wild based on multi-class SVM. In 2019 federated conference on computer science and information systems* [online] Leipzig, 1-4 September 2019. IEEE [online]. Available from: https://ieeexplore.ieee.org/abstract/document/8860003 [Accessed 03 May 2023].
- Sheikholeslami, S. (2019). Ablation Programming for Machine Learning (Dissertation). [Accessed 29 April 2023].
- Sahoo, S (2018) *Deciding optimal kernel size for CNN.* Available from: https://towardsdatascience.com/deciding-optimal-filter-size-for-cnns-d6f7b56f9363 [Accessed 01 May 2023].
- Tolentino, L. K. et al (2019) Static Sign Language Recognition Using Deep Learning. *International Journal of Machine Learning and Computing* [online]. 9 (6), pp. 821-827. [Accessed 28 April 2023]
- World Health Organisation (2023) *Hearing loss.* Available from: https://www.who.int/health-topics/hearing-loss#tab=tab_1 [Accessed 22 April 2023].
- Wu, J. (2017). Introduction to convolutional neural networks. *National Key Lab for Novel Software Technology. Nanjing University. China* [online], 5(23), 495. Available from: https://cs.nju.edu.cn/wujx/paper/CNN.pdf [Accessed 30th April 2023].
- Zhang, Q and Xiao, H. ed. (2008) Extracting Regions of Interest in Biomedical Images, *2008 International Seminar on Future BioMedical Information Engineering* [online], IEE, Wuhan, 2008. IEEE Xplore. Available from: https://ieeexplore.ieee.org/document/5076670 [Accessed 23 April 2023].