A CNN sign language recognition system with single & double-handed gestures

Neil Buckley
AI Laboratory, School of Mathematics, Computer Science and Engineering
Liverpool Hope University
Liverpool, UK
bucklen@hope.ac.uk

Lewis Sherrett
AI Laboratory, School of Mathematics, Computer Science and Engineering
Liverpool Hope University
Liverpool, UK
17000026@hope.ac.uk

Emanuele Lindo Secco
Robotics Laboratory, School of Mathematics, Computer Science and Engineering
Liverpool Hope University
Liverpool, UK
seccoe@hope.ac.uk, 0000-0002-3269-6749

***Abstract*— This work aims at presenting a novel Computer Vision approach in the development of a real-time, web-camera based, British Sign Language recognition system. A literature review focused on current (1) state of sign language recognition systems and (2) techniques used is conducted. This review process is used as a foundation on which a Convolutional Neural Network (CNN) based system is designed and then implemented. A bespoke British Sign Language dataset - containing 11,875 images - is then performed to train and test the CNN which is used for the classification of human hand performed gestures. Finally, the CNN architecture recognized 19 static British Sign Language gestures, incorporating both single and double-handed gestures. During testing, the system achieved an average recognition accuracy of 89%.**

Keywords*— *sign language recognition, AI, CNN, human-machine interaction

I. Introduction

Research in the area of systems that are capable of recognising sign language has received substantial attention over the past few decades, fuelled in particular by the rapid evolution of artificial intelligence techniques [1-3]. In turn this has led to the development of many Sign Language Recognition Systems (SLR), which shall be referred to as SLR systems throughout the remainder of this chapter. These systems though varying in sign language dialect, share the common goal of correctly recognising hand gestures performed by a signer. However the varying proposed approaches to achieving this goal has produced a diverse area of research and development encompassing areas of computer science such as *Computer Vision* (CV), *Sensor Processing*, *Human-Computer Interaction*, and *Pattern Recognition* [1-4]. Of these SLR systems there are two main types of design and implementation, which are those that use wearable sensors, and those that use video footage and images. Both shall be discussed below.

SLR systems that utilise sensors worn on the body to capture sign language gestures usually comprise of sensor-embedded gloves that are worn on the hands. These types of SLR systems are one of the two main approaches to capturing gestures to be classified [1]. There have many sensor-based SLR system developed. Of these developed systems many rely on sensor fusion to achieve an accurate recognition rate; such as the system proposed by Kim et al [5] in which 'bi-channel sensor fusion' is used to combine data from an accelerometer and electromyogram embedded glove that covers the hand and upper wrist to recognise German Sign Language gestures [6, 7].

SLR systems designed to use video footage or image data to capture a gesture performed by a user can be further categorised into two methods. The first method involves capturing gesture data using a 3D camera, and the second method involves capturing gesture data using a 2D camera. Therefore the literature to be reviewed in this sub-chapter shall be represented in two further sub-chapters discussing SLR systems which use 3D cameras, and SLR systems which use 2D cameras respectively.

An advantage to using a 3D camera over a 2D camera is that the depth capturing capabilities of a 3D camera allow for easier image pre-processing as everything registered as greater in depth than the signer can quickly be removed, solving the issue of complex environmental backgrounds and lighting as seen with 2D cameras [8].

Other SLR systems are only capable of recognising sign language gestures in uniform background and lighting conditions, such as Tolentino et al work [9], which achieved a recognition accuracy of 93.67% in recognising gestures of the American Sign Language in uniform lighting and background conditions when tested with 30 individuals. The system operated in real time, and used a CNN to classify the gestures performed. However, the system modified some of the gestures because their similarity to other gestures would have caused misrecognition and affected the accuracy of the system. The SLR system proposed by Sawant and Kumbhar [10] also avoided the issue of complex backgrounds by capturing all testing footage on a white background which not only worked to limit background interference but also limited ununiformed lighting. This SLR system was able to recognise 26 Indian Sign Language gestures and used Principal Component Analysis (PCA) during the classification stage. A paper authored by Berru-Novoa et al [11], tested a host of classification methods for a SLR system using a uniform background and lighting approach. This study found that amongst the classification methods of a Support Vector Machine, or SVM, a KNN, and an ANN, that the SVN performed best achieving a recognition accuracy of 89.17%. However the KNN and ANN were less than 2% behind. This system also used HOG which stands for Histogram of Orientated Gradients in the feature extraction stage.

II. Materials and Methods

The design of the sign language recognition system itself is a multi-step process. All of the steps involved are detailed below.

A. The Complexity of Sign Language

All sign languages are gesturally complex, which some signs being performed with one hand and others being performed with two hands. The position and rotation of the hands is important in the gestures, as is the position of the

fingers. British Sign Language is no different and the 19 gestures which have been chosen for this work reflect these diverse variations in hand and finger position. This is to ensure that the system which is produced has the capacity to deal with these variations. The 19 gestures which will be used by the system can be seen being performed in Fig.1.

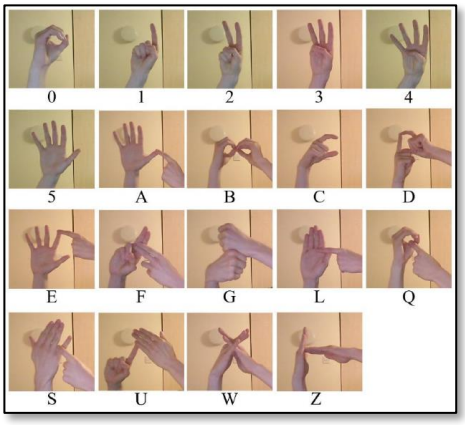

Fig. 1. The British Sign Language gestures to be used in the system.

Of the signed gestures seen in Fig.1 a diverse range can be identified. For example, signs such as *0* and *C* are performed one-handed, whereas signs such as *Q* and *W* are performed double-handed. Furthermore, the importance of finger position is evident with signs such as *A* and *L* in which the splayed fingers and position of the fingers on the secondary hand of *A* make the difference between gesturing an *L*. However, there are also clear similarities between some gestures such as *5* and *E* in which the only defining factor of the *E* gesture is the secondary index finger pointing to the index finger of the main hand, without which the *E* gesture would become a *5* gesture. These will be challenging cases for the system to deal with, and they have been incorporated to test the robustness of the system.

B. A web-camera based system

The system will make use of a web-camera for capturing the gesture performed by the user. This is because in the real-world web-cameras are common, portable, and cheap, making them accessible for users of a sign language recognition system. The web-camera based approach was discussed in the literature review chapter, and the SRL systems develop using this approach proves the viability of web-camera based systems. The web-camera used in this implementation will be the *Advent AWC72015* which contains a 12 mega pixel camera, with a 720 p resolution, which captures at 30 frames a second [12]. However, the system is not web-camera model specific, and therefore can be used with any web-camera.

C. System Architecture Design

The system will be designed around two main components. The first of these components is the sign language recognition system itself. This component will capture frames from the web-camera, pre-process these frames, and send these frames to the classifier for recognition. This component will also handle the graphical user interface, and user input. The other component will be used to house the classifier, and will consist of two smaller parts. Of which the first will be the creator of the classifier, this part will construct the model, and perform training and testing of the model. The second will be the compiled, trained, and saved model itself. The saved model will be used with the sign language recognition system component during the feature extraction and classification stage. This system architectural design is visualised in Fig.2.

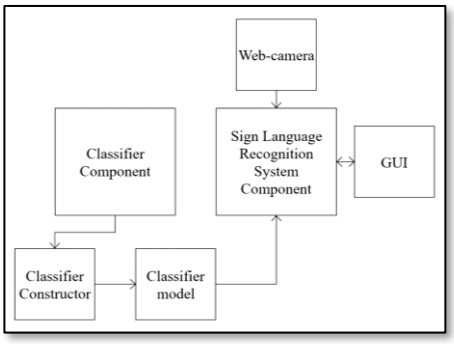

Fig. 2. The system architecture design.

D. CNN Architecture

The system will make use of a CNN for the feature extraction and classification process. This is because of the abilities of CNNs in performing feature extraction processes in CV based systems as discussed in [13]. A bespoke CNN will be created for the system which means that it will not have been pre-trained on any previous data. This is to allow for the CNN to be specifically tailored for the purpose of providing accurate gesture recognition in this system. The architecture of the CNN can be seen in Fig. 3.

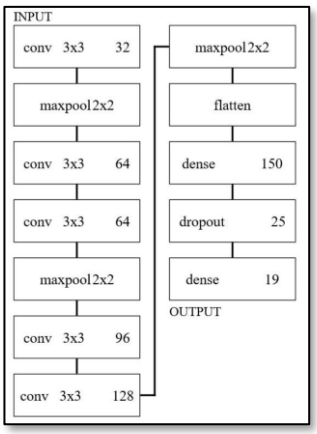

Fig. 3. The CNN set up.

E. Dataset set up

The design of the dataset involves 2 steps, of which the first is the structure of the dataset, and the second is the contents of the dataset. These steps are detailed below.

1. Dataset Structure

The dataset will be housed in a structure that is conveniently accessible for the CNN to perform training and testing with. Two main folders shall be used, one folder will be used to house the training data, and the other shall hold the testing data. Both of these folders will contain an identical set of 19 sub-folders representing each gesture recognisable by the system. Inside these sub-folders the image data shall be stored in *.jpg* format. All images will consist of the dimensions 288 x 312.

2. Gesture position and distance

The 19 gestures which are to be recognisable to the system have been displayed in Fig.1, however in this example all

signs were present at close distance to the web-camera and in a neutral position. It is important to build a dataset containing data depicting gestures being performed at varying angles, positions, and distance to the web-camera. Including this data will result in a more robust classification model, which is able to deal with fringe events. This process of collecting data will also makes the system more versatile and able to deal more efficiently with other users, who may perform gestures slightly differently [14].

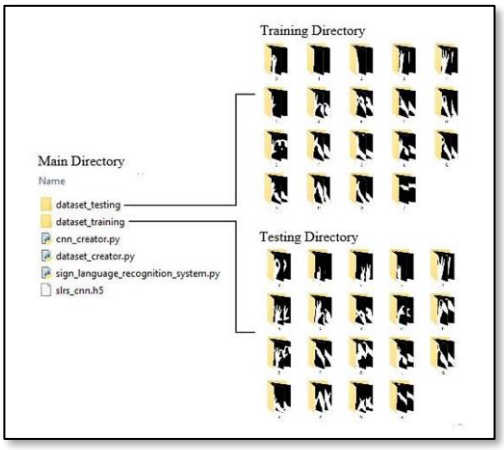

Fig. 4. The Implemented System Architecture.

F. *Implementation*

The design was implemented using the python programming language. This is due to Python's expansive libraries and suitableness in developing Artificial Intelligence based approaches [15]. The Python libraries used in the implementation of the system and the reason for their use will be discussed in the bullet-point list below:

- *Tensorflow*, which is an AI library, was used as a backend for the Keras library
- *Keras*, which is an API built on top of Tensorflow allows for the abstraction of complex Tensorflow commands with the replacement of more user-friendly Keras commands. Keras was used to construct, train, test, and save the CNN model used in the implementation.
- *openCV* is a computer vision oriented AI library to import the video footage from the web-camera, and then carry out multiple steps of image pre-processing on the captured frames. It was also used for the GUI implementation and to save the dataset images to file.
- *Numpy* is a library containing high level mathematical functions and many matrices and array functions.
- *OS module* allows for interaction with the operating system. It was used to define the file directories and saving the training CNN model.
- *Time module* provides time-based functions, and was used to pause camera capture feed while tasks such as background subtraction were performed.

Python, ver. 3.7.0, was used for the implementation as this was the most current version which was compatible will all imported libraries. The implemented system architecture can be seen in Fig. 4 below. The components which make up the sign language recognition system can all be found in the same directory, along with the training and testing datasets. As stated in the design, both of these datasets contain a folder corresponding to one of the 19 recognisable gestures.

III. Results

Each image data is captured by the web-camera and fed to the system in real-time. No pre-processing steps are taken until the user presses the *b* key on their keyboard, which runs a background subtraction task. This background subtraction process is the first of multiple steps of image pre-processing used by the system. Fig. 5 below displays the process of background subtraction.

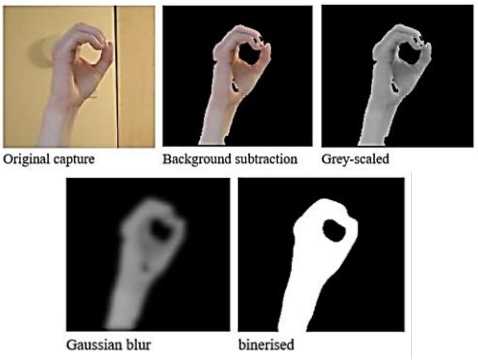

Fig. 5. Pre-processing steps.

The graphical user interface was implemented using the *openCV* library. The interface features the original frame in its entirety, with a red box in the top left-hand corner. The red box represents the region of interest and in the area in which the user must perform the gestures. If the CNN recognises a gesture with 60% or more certainty, the gesture's name is displayed on-screen to the user (Fig. 6).

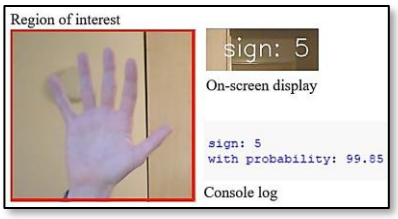

Fig. 6. The Graphical User Interface (GUI).

A. *Data set testing*

The testing carried out on the implementation using the testing dataset is discussed in this sub-chapter. The testing dataset consisted of a total of 2,375 images, which is 125 images per gesture. Examples of the training images can be seen in Fig. 7.

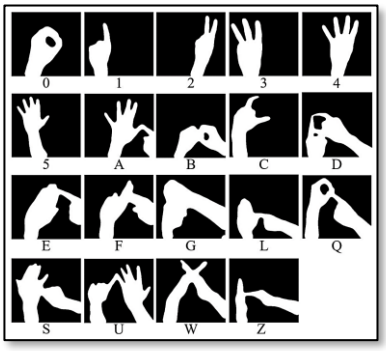

Fig. 7. Selection of testing dataset images.

The dataset was initialised, with the labels withheld, and the CNN was used to predict the gesture being performed in the images. If the CNNs prediction matched the label for that image, the test was deemed a success, if not the test was deemed a failure. Table 1 displays the accuracy of the 10 tests which were carried out on the trained mode.

TABLE I. DATASET TESTING RESULTS

Test number	1	2	3	4	5
Accuracy [%]	88.13	87.92	87.99	87.88	88.81
	6	7	8	9	10
	87.88	87.92	87.99	87.86	87.92

B. Real-time testing

The testing of the trained CNN using the testing dataset is useful for providing an insight into the effectiveness of the model, however the testing dataset contains on static images and the system's real-world function is to allow for real-time gesture recognition. Therefore a further series of tests has been carried out on the system to determine its robustness as a real-time sign language recognition system. The comments of some of these tests can be found in Table 2.

TABLE II. REAL-TIME TESTING RESULTS

Signed gesture	**Accuracy comment**
0	The *0* sign had the highest recognition accuracy of the signs and was recognized correctly almost every time, regardless of distance from the web-camera or rotation of the wrist. The placement of the hand in the region of interest also had no negative effect on the recognition accuracy. This is most likely down to the uniqueness of this gesture. The sign is one handed (no other similarity with such position of the fingers).
1	The *1* sign had a high recognition accuracy and could be distinguished by the CNN in many cases of distance and hand location inside of the region of interest. However, the rotation of the wrist to the left could cause a misrecognition of a *2* sign in some cases, especially causes where the background lighting was not highly uniformed.
2	The *2* sign had a high recognition accuracy, and could be distinguished by the CNN in most cases. This could be due to the vast diversity of testing images used for the *2* gesture.
A	The *A* sign boasted a high recognition accuracy and came in second place of the highest recognizable alphabetical gestures. A trend with all double-handed alphabetical gestures is greater reliance upon uniformed background lighting in the pursuit of high recognition accuracies.
B	The *B* sign is quite unique which lead to its relatively high recognition accuracy; on some occasions it could be misrecognized for a *D* gesture. Rotating the hands inwards slightly and therefore ensuring that the gaps between the curled fingers were visible on both hands usually resulted in the recognition being corrected to a *B* sign.
C	The *C* sign possessed the highest recognition accuracy of the alphabetical gestures and was recognized accurately in almost the same cases as the *5* sign. This is most likely due to the sign being performed one-handed, and also the unique curved finger shape of the sign. This sign was the most versatile in terms of hand position and recognition accuracy.

IV. CONCLUSION & DISCUSSION

This work aimed to research the current state of sign language recognition systems and to use this research as a foundation to develop a computer vision web-camera based British Sign Language recognition system, along with a bespoke British Sign Language gesture dataset. The system was designed to be able to recognise 19 British Sign Language gestures, and operate in real-time. This primary aim was motivated by the communication gap that exists between the hearing and hearing-impaired community, and the ability of SLR systems to bridge this gap. However further motivating was the lack of British Sign Language recognition systems, and the stark difference between those with a hearing-impairment and the number of registered users of Sign Language in the United Kingdom. Moreover a proper comparison with other state of the art SLR models should be performed.

ACKNOWLEDGMENT

This work was presented in thesis form in fulfilment of the requirements for the BSc in Computer Science for the student Lewis Sherrett under the supervision of Dr Neil Buckley from the AI Laboratory, School of Mathematics, Computer Science and Engineering, Liverpool Hope University.

References

[1] Shanableh T, Assaleh K, Al-Rousan M (2007) Spatio-Temporal Feature Extraction Techniques for Isolated Gesture Recognition in Arabic Sign Language. *IEEE Trans on Cybernetics*, 37(3), pp.641-650.

[2] Setiawardhana Hakkun RY, Baharuddin A (2015) Sign Language Learning Based on Android for Deaf and Speech Impaired People. In: IEEE. *2015 Int Electronics Symposium*. Surabaya, pp.114-117.

[3] Soodtoetong N, Gedkhaw E (2018) The Efficiency of Sign Language Recognition using 3D Convolutional Neural Networks. In: IEEE. *2018 15th Int Conf on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology*. Chiang Rai, 70-73.

[4] Chu TS, Chua AY, Secco EL, A Wearable MYO Gesture Armband Controlling Sphero BB-8 Robot, HighTech and Innovation Journal, 1(4), 179-186, http://dx.doi.org/10.28991/HIJ-2020-01-04-05

[5] Kim J, Wagner J et al (2008) Bi-Channel Sensor Fusion for Automatic Sign Language Detection. In: IEEE. *8th IEEE Int Conf on Automatic Face and Gesture Recognition*. Amsterdam, 17-19 September, pp.1-6.

[6] Maereg AT, Lou Y et al (2020), Hand Gesture Recognition Based on Near-Infrared Sensing Wristband, *Proc. 15th Int Joint Conf on CV, Imaging & Computer Graphics Theory and Applications*, 110-117.

[7] Myers K, Secco EL (2020) A Low-Cost Embedded Computer Vision System for the Classification of Recyclable Objects, *Lecture Notes on Data Eng and Comms Tech*, 61.

[8] Galicia R, Carranza O. et al (2015) Mexican Sign Language Recognition using Movement Sensor. In: IEEE. *2015 IEEE 24th Int Symposium on Industrial Electronics*. Buzios, 3-5 June. pp.573-578.

[9] Tolentino LK, Juan RS et al (2019) Static Sign Language Recognition Using Deep Learning. *Int J of ML and Computing*, 9(6), pp.821-827.

[10] Sawant SN, Kumbhar MS (2014) Real Time Sign Language Recognition using PCA. In: IEEE. *2014 IEEE Int Conf on Advanced Comms, Control, & Computing Technologies*. 8-10 May, 1412-1415.

[11] Berru-Novoa B, Gonzales-Valenzuela R et al (2018) Peruvian Sign Language Recognition using Low Resolution Camera. In: IEEE. *2018 IEEE XXV Int Conf on Electronics, Electrical Engineering and Computing*. Lima, 8-10 August, pp.1-4.

[12] Advent (2015*) Instruction Manual PC Webcam Model AWC72015*.

[13] Pigou L, Dielemna S et al (2015) Sign Language Recognition Using Convolutional Neural Networks. *Computer Vision - ECCV 2014 Workshop*. Zurich, Switzerland, 6-7 September. Springer, pp.572-578.

[14] Kanwal K, Abdullah S et al (2014) Assistive Glove for Pakistani Sign Language Translation. In: IEEE. *17th IEEE International Multi Topic Conference 2014*. Karachi, 8-10 December, pp.173-176.

[15] Sodhi P, Awasthi N et al (2018) Introduction to Machine Learning and its Basic Application in Python. *Proceedings of 10th Int Conf on Digital Strategies for Organizational Success*, India, 5-7 January. pp.1-22.