American Sign Language Fingerspell Detection Using Convolution Neural Network Within A 2-Way Translator

Jihan Angrila[1], Neno Ruseno[2], and Normalisa[3]

[1,2,3]Engineering Faculty, International University Liaison Indonesia, Intermark, BSD City, 15310
e-mail: [1] jihan.angrila@gmail.com; [2]neno.ruseno@iuli.ac.id; [3]normalisa@iuli.ac.id

Abstract. With time, computing and neural network has become more sophisticatedand can be applied for various uses. One common and vital use of this technology is forcomputer vision. Just like a human brain, which the neural network imitates, computervision can be used to classify and detect objects for various reasons. This ranges from smart systems to quality control in factories. An up-and-coming usage of computer vision is to help those with impaired hearing and users of sign language communicate with those who can't. This is done by detecting the sign language and converting it to other forms of communication such as text, audio output, and more. In this project, a two-way translator application is made with 3 features:gesture detection, text-to-speech, and speech-to-text translator. Convolution Neural Network (CNN) is used as the model for the gesture detection and the language used isEnglish for verbal and American Sign Language (ASL) for the signed. Confusion matrix and loss and accuracy evolution were analysed to see the effectivity of the model. The accuracy of the model was found to be 82.6% with certain independent variable factors (bright lighting, single-coloured background, using front-facing left hand, and fully covering the ROI box for gestures). The application Graphical User Interface (GUI) was made using Tkinter in python. Although not directly translated to speech due to coding limitations, having the detection feature is an important part as it shows potential of what the translator could be.

Keyword: Convolution Neural Network, American Sign Language, Python, Object Classification, Object Detection, Impaired Hearing Translator, Tkinter, Graphical User Interface.

1. INTRODUCTION

Those suffering from hearing loss can be divided into 2 categories; hard of hearing and deaf. Those with hard of hearing suffers from mild to severe hearing loss. While still being able to verbally communicate, they need hearing aids, cochlear implants, and other devices. Deaf people on the other hand have very little to no hearing. They mostly communicate in sign language.

To understand how people use sign language, understanding why they need to use it is necessary to clarify the significance of sign language. Oxford Languages defined 'Deafness' as the condition of lacking the power of hearing or having impaired hearing. This means that those with impaired hearing are still within the deaf group, just less extreme. In Koffler et al. (2015), it is stated that different degrees of hearing can be classified into:

- Mild: 26dB – 40dB
- Moderate: 41db – 55dB
- Moderately severe: 56dB – 70dB
- Severe: 71dB – 90dB
- Profound: more than 91dB

The last first 3 categories can fall under hard of hearing while the last 2 would be deaf.

There are many causes of hearing loss and deafness; this can range from genetics to injuries to natural progression of hearing loss as one age. One can be born deaf due to genetics or even infection during pregnancy. Sheffield & Smith (2019) article titled 'The Epidemiology of Deafness' explains in detail the different categories of hearing loss based on anatomical defects and circumstances on how someone can have them. The paper states that the said (broad) categories are called Conductive Hearing Loss (CHL) and Sensorineural Hearing Loss (SHL). A third category is also present where both CHL and SHL is present. CHL is caused by pathologies that result in defects anywhere from the external ear to the ossicles (Sooriyamoorthy & Jesus, 2021), meaning soundwave cannot be properly received/conducted due to the loss of organ function in the ear. SHL is when the hair cell in the inner ear, vestibulocochlear nerve, or the brain's central processing centers are damaged

(Tanna, Lin, & Jesus, 2021). Back to Sheffield & Smith (2019), the paper stated that factors affecting congenital hearing loss are 50% genetics and 50% environmental. Genetics part refers to the mutations in the genes of the child as it develops which causes syndromes such as Usher Syndromes, Waardenburg Syndromes, and more, while environmental refers to factors such as infections, head trauma, and noise-induced (exposure to very loud noise). Lastly, a very common factor: old age. The paper stated presbycusis (age-related hearing loss), can be caused by 4 factors: cochlear aging, environment, genetic, and medical comorbidities (other health conditions). Below shows the correlation between hearing loss and age in the US (taken from the Sheffield & Smith paper).

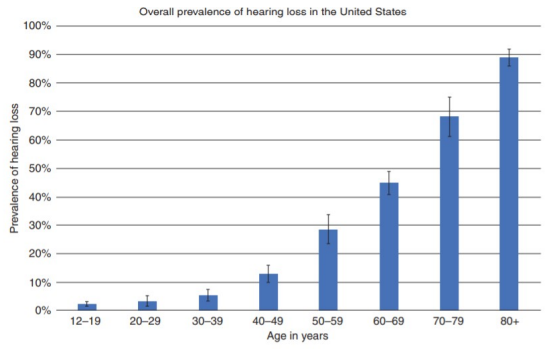

Figure 3. Prevalence of individuals in the United States with hearing loss by age. Data based on National Health and Nutritional Examination Surveys 2001 through 2008 (see Lin et al. 2011). Hearing loss is defined as thresholds of 25 dB or greater in at least one ear. Error bars represent 95% confidence interval.

Figure 1 Bar char showing the correlation between age and hearing loss in the US by Sheffield & Smith. (2021).

The issue with sign language is not everyone is able to use it. Sign language systems differ from one country to another but one thing they all possess is the fact that they're different from their spoken counterpart. American Sign Language (ASL) has different word formation, pronunciation, and other language fundamentals from American English. Learning ASL would be learning a whole different language, not just "English but sign". This fact might be why sign language, even though gaining popularity, is still not common enough in everyday life for those with healthy hearing. Although in a specific language, different regions have different sign language structures. This can be differentiated by countries such as English-speaking countries like the US, Britain, Australia have ASL, British Sign Language (BSL), and Australian Sign Language (Auslan) respectively. Another thing to note would be the fact that sign language is not the same as body language, which is another system of non-verbal communication often done subconsciously. Although crucial for those with impaired or lost hearing, sign language is also used for those who have disabilities that affect their speech. An interesting usage of sign language is for communication between animals and humans as proven by primates such as Koko the gorilla and Chantek the orangutan who learnt sign language to speak with their caretaker.

The vast usage of sign language, especially advanced by technology and digital aid such as translating apps that are easily made and available, can create a positive result for those who need it most. Not only that, it would increase accessibility for the disabled, creating an equal opportunity for them to function in society as well as 'normal' people as they should not be measured by their physical limitations, but rather their minds.

2. LITERATURE REVIEW

National Center for Health Statistics in the US estimates that there are 28 million Americans who have some degree of hearing loss with 2 million of that statistic classified as deaf. Although increasing in popularity as social awareness of the disabled increases over time, American Sign Language (ASL) still lack the normalization that it needs. Not just ASL, but all systems of sign language. This is where digital aid, using technology to create accessibility for the disabled, will greatly benefit those with impaired hearings. This literature review will examine the accuracy and efficiency of two machine learning (ML) methods, Long Short-Term Memory and Convolutional Neural Network used to create an ASL fingerspell translation through different modifications made to it. LSTM and CNN are selected due to their usage frequency related to language translation application, especially sign language translation.

LSTM is a Recurrent Neural Network (RNN) architecture with memory blocks that has input, output, and forget gates which provides the model to write, read, and reset the operations within each cell. In most research, it seems LSTM is more used in sign language recognition rather than translation itself. 3 research are listed for this section of the literature review.

A simple search will show that CNN can be broken down into 3 layers: convolutional, pooling, and fully connected (FC) layers. A simple explanation is CNN creates a hierarchy where different parts of an image (the pixels) are separated using matrix

multiplication where it then gets down sampled to become a summary of the features detected in the image. The pixels are then classified through (usually) softmax activation function; producing probability between 0 and 1. CNN is commonly used in computer vision to process images to be classified and/or recognized (IBM, 2020). For this section, 4 research are listed.

3. METHODOLOGY

To clear any confusion, this sub-part will discuss the difference between image classification and object detection as the latter is used for this project. Although sounding similar at a first glance, image classification is not the same as object detection. Both falls under object recognition, but classification is more straightforward while detection requires more complex algorithm. Image classification is simply detecting the type or class of an image (for example, labelling a picture of a dog as 'Dog') while object detection is finding the location of the object within a bounding box and the type or class that it is (for example, labelling an orange, apples, and peaches in a picture of various fruits). While image classification is a single method, object detection consists of image classification and object localization (finding the coordinate of the object with class within an image). The bounding box used in object detection is created through object localization.

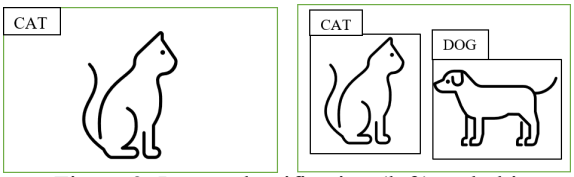

Figure 2. Image classification (left) and object detection (right).

The following flowchart shows the flow of the program with diagram X showing the creation of the CNN model and diagram X the creation of the GUI. The flowcharts are made on app.diagrams.net.

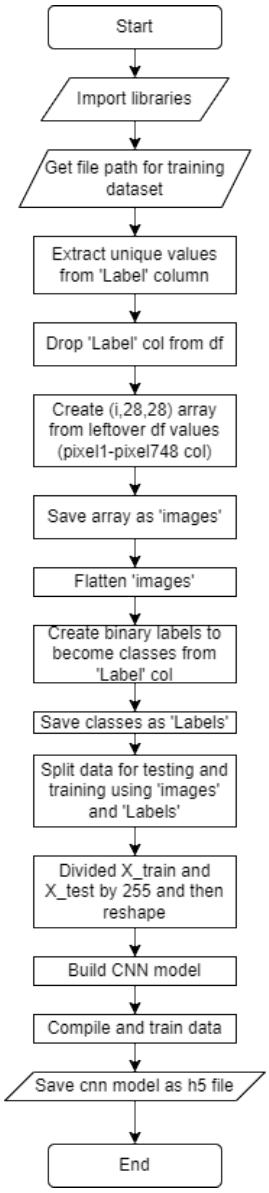

Figure 3. Flowchart for creating the CNN model.

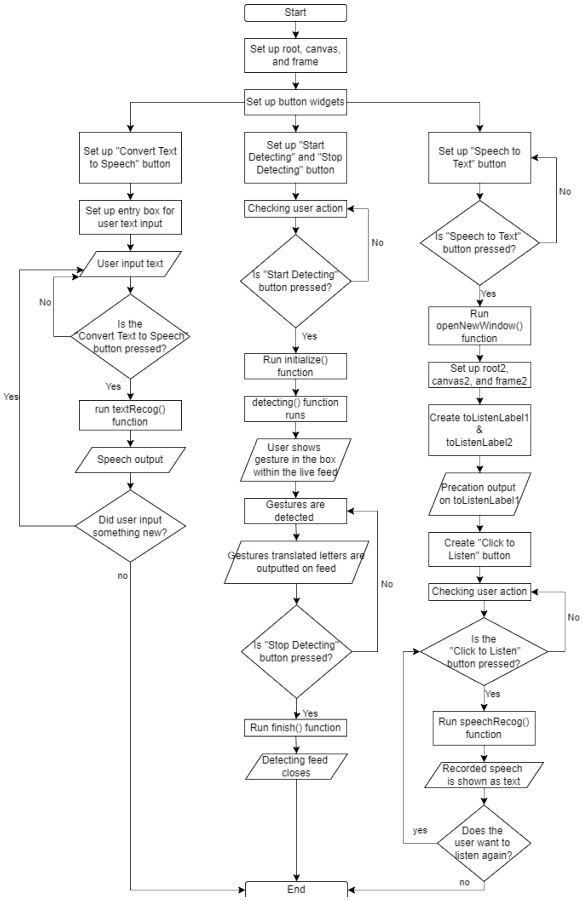

Figure 4. Flowchart for the GUI

4. RESULT

The figure above shows loss and accuracy evolution diagram. One obvious characteristic the two shares is how they are inversely proportional. The X-axis of both graph represents the epoch of the training while the y-axis of the loss evolution representing the summation or errors and the y-axis of the accuracy evolution represents accuracy with 1.0 equates to 100%. The relative shape of both evolutions is relatively inversely proportional as well with converging point happening at the same time.

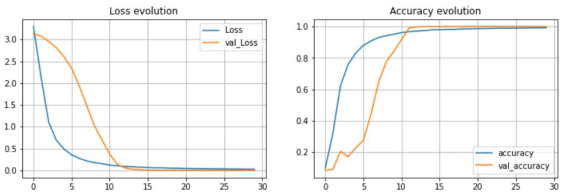

Figure 5. Loss and accuracy evolution of the model during training.

While the loss evolution decreases, the accuracy evolution increases. This shows that the model is built and learning properly. This is because loss evolution is simply the progress of the model's inaccurate prediction throughout training while accuracy evolution is the prediction after the parameters are learned from the loss evolution.

```
In [41]: from sklearn.metrics import accuracy_score

         accuracy_score(test_labels, y_pred.round())

Out[41]: 0.826547685443391
```

The model created shows an accuracy of 0.826547685443391 or roughly 82.6%. This result is obtained from the function accuracy_score imported from sklearn.metrics, a function that measures the accuracy of the model by comparing classes from dataset made specifically for testing with the detected classes made by the model. The two variables must exactly match for it to count. In the MNIST ASL dataset, this testing dataset is proved so no data splitting from the training dataset is needed. An accuracy reading of 82.6% is very good as most gestures are detected however, this accuracy can be increased by altering the structure of the CNN model either in layers or the layers that deal with standardization (BatchNormalization() and Dropout() layers). Altering the latter might give better reading as they deal with how the output of each layers are learnt (non-overfitted, normalized, etc).

2 different types of windows will be displayed for the user to interact with. The first is the main window where the live camera feed and detection box is present. This is the ASL-to-Speech section of the interface for the ASL user. Under the camera feed is a text box that display the translated sign which has been recorded for conversion to speech. Above the camera feed is a button that when clicked open anew window. This window is the Speech-to- Text translator for the non-ASL user. At the top of the window is a button where when clicked, it records the person's voice and gets printed as text on the label box below it.

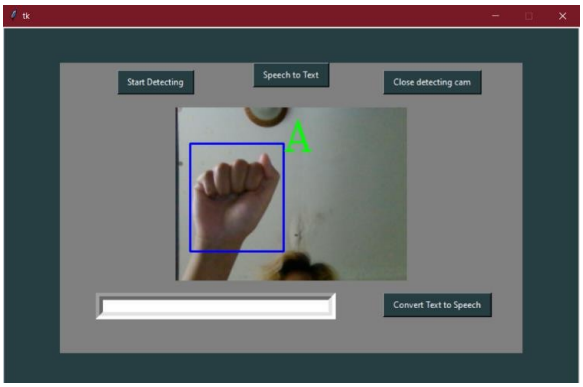

Figure 6. GUI for ASL User

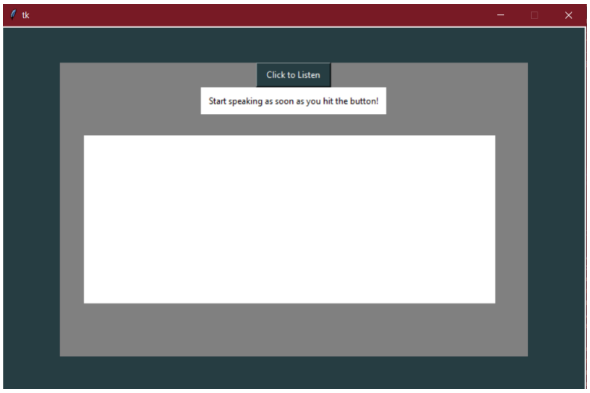

Figure 7. GUI for non ASL User

5. CONCLUSION

The objective of this research is to design an application, a two-way translator for those with impaired hearing, which has a real-time ASL gesture to letter detector. Two-way translator meaning ASL users can use text to be converted to speech and non-ASL user can use their voice (speech) to be converted to text. The language used to develop the application is python and IDE used to create the CNN model for the ASL gesture detection is Jupyter Notebook while creation of the GUI itself is done in Spyder using Tkinter library.

Accuracy result of the model created were evaluated in three ways: loss and accuracy evolution during training, using accuracy_score() imported from sklearn.metrics, and finally, a confusion matrix showing non-normalized result of the prediction. Different independent variables (lighting, placement and angle of hand, left and right hand, and background) were also observed in terms of accuracy (dependent variable). Observations was made by measuring and analysing the average incorrectness and full detection of the independent variable in different conditions. It was found that for best detection result, left hand should be used with good, clear, and natural lighting along with a single-coloured background. The hand should also be front-facing the camera and fully cover the ROI box. As a result, the application with the 3 features (ASL gesture detection, text-to-speech, and speech-to-text translator) are able to perform as intended.

Minor problem, detection camera freezing, occurs when detection camera is open while ASL user tries to translate text-to-speech. However, this is only minor as text-to-speech translator is fast and happens in real time. Detection camera unfreezes once audio speech is finished being outputted.

REFERENCES

Abiyev, R.H., Arslan, M., & Idoko, J.B. (2020). Sign Language Translation Using Deep Convolutional Neural Networks. KSII Transactions on Internet and Information Systems, vol. 14, no. 2. Korean Society for Internet Information (KSII).

Albawi, S., Bayat, O., Al-Azawi, S., & Ucan, O. N. (2018). Social Touch Gesture Recognition Using Convolutional Neural Network. Computational intelligence and neuroscience, 2018, 6973103. https://doi.org/10.1155/2018/6973103

Amos, D. (2018). The Ultimate Guide to Speech Recognition with Python. https://realpython.com/python-speech-recognition/#picking-a-python-speech-recognition-package

Anaconda. (2022). Anaconda Individual Edition. https://docs.anaconda.com/anaconda/

Arora, S., Gupta, A., Jain, R., & Nayyar, A. (2021). Optimization of the CNN Model for Hand Sign Language Recognition Using Adam Optimization Technique. 10.1007/978-981-33-4687-1_10.

Brownlee, J. (2019). A Gentle Introduction to Computer Vision. https://machinelearningmastery.com/what-is-computer-vision/

Brownlee, J. (2017). Gentle Introduction to the Adam Optimization Algorithm for Deep Learning. https://machinelearningmastery.com/adam-optimization-algorithm- for-deep-learning/

Gholamalinezhad, H., & Khosravi, H. (2020). Pooling Methods in Deep Neural Networks, a Review. Computer Vision and Pattern Recognition (cs.CV). arXiv:2009.07485

Google Developers. (2020). Classification: Accuracy. https://developers.google.com/machine-learning/crash- course/classification/accuracy

Google Developers. (2020). Descending into ML: Training and Loss. https://developers.google.com/machine-learning/crash-course/descending- into-ml/training-and-loss

Grif, M.G. & Kondratenko, Y.K. (2021). Development of A Software Module for Recognizing the Fingerspelling of The Russian Sign Language Based on LSTM. J.Phys.: Conf. Ser. 2032 012024.

Gupta, A. (2021). A Comprehensive Guide on Deep Learning Optimizers. https://www.analyticsvidhya.com/blog/2021/10/a-comprehensive-guide-on- deep-learning-optimizers/

IBM Cloud Education. (2020). Convolutional Neural Networks. https://www.ibm.com/cloud/learn/convolutional-neural-networks

IBM Cloud Education. (2020). What is computer vision? https://www.ibm.com/cloud/learn/convolutional-neural-networks

Jatu, N. (2019). Python Text To Speech | pyttsx module. https://www.geeksforgeeks.org/python-text-to-speech-pyttsx-module/

Johnson, D. (2021). What is Tensorflow? How It Works? Introduction & Architecture. https://www.guru99.com/what-is-tensorflow.html

Jupyter.org. (2022). Jupyter. https://jupyter.org/

Kaggle.com. (2017). Sign Language MNIST. https://www.kaggle.com/datamunge/sign-language-mnist

Khosla, S. (2021). CNN | Introduction to Pooling Layer. https://www.geeksforgeeks.org/cnn-introduction-to-pooling-layer/

Koffler, T., Ushakov, K., & Avraham, K.B. (2015). Genetics of hearing loss: Syndromic. Otolaryngol Clin North Am 48: 1041–1061.

Masood, S., Thuwal, H., & Srivastava, A. (2018). American Sign Language Character Recognition Using Convolution Neural Network. 10.1007/978-981-10-5547- 8_42.

Muthu, M.H. & Gomathi, V. (2021). Indian Sign Language Recognition through Hybrid ConvNet-LSTM Networks. EMITTER International Journal of Engineering Technology, 9(1), 182-203. https://doi.org/10.24003/emitter.v9i1.613

OpenCV.org. (2022). About. https://opencv.org/about/

Python.org. (2021). What is Python? Executive Summary. https://www.python.org/doc/essays/blurb/

Pypi.org. (2022). Imutils 0.5.4 Project Description. https://pypi.org/project/imutils/

Pypi.org. (2022). Pillow 9.0.0 Project Description. https://pypi.org/project/Pillow/#description

Pypi.org. (2017). SpeechRecognition 3.8.1. https://pypi.org/project/SpeechRecognition/

Qin, Z., Yu, F., Liu, C., & Chen, X. (2018). How convolutional neural network see the world-A survey of convolutional neural network visualization methods. arXiv preprint. arXiv:1804.11191.

Rockikz, A. (2021). How to Convert Speech to Text in Python. https://www.thepythoncode.com/article/using-speech-recognition-to-convert- speech-to-text-python

Sanghvirajit. (2021). A Complete Guide to Adam and RMSprop Optimizer. https://medium.com/analytics-vidhya/a-complete-guide-to-adam-and- rmsprop-optimizer-75f4502d83be

Sharma, A. (2019). Introduction to GUI With Tkinter in Python. https://www.datacamp.com/community/tutorials/gui-tkinter-python

Sharma, H. (2020). Hands-On Guide To Pillow – Python Library for Image Processing. Developers Corner. https://analyticsindiamag.com/hands-on-guide-to-pillow- python-library-for-image-processing/

Sheffield, A.M., & Smith, R.J.H. (2019). The Epidemiology of Deafness. Cold Spring Harb Perspect Med. ;9(9):a033258. doi: 10.1101/cshperspect.a033258. PMID: 30249598; PMCID: PMC6719589.

Sooriyamoorthy, T., & Jesus, D.O. (2021). Conductive Hearing Loss. Treasure Island (FL): StatPearls Publishing; 2021 Jan-. Available from: https://www.ncbi.nlm.nih.gov/books/NBK563267/

Spyder-ide.org. (2021). Overview. https://www.spyder-ide.org/

Suharjito, S., Gunawan, H., Thiracitta, N., & Nugroho, A. (2018). Sign Language Recognition Using Modified Convolutional Neural Network Model. 1-5. 10.1109/INAPR.2018.8627014.

Tang, Z., Wang, S., Yang, Z., Chen, G., Huang, X., Sun, M., & Liu, Y. (2020). Neural Machine Translation: A Review of Methods, Resources, and Tools. https://doi.org/10.1016/j.aiopen.2020.11.001

Tanna, R.J., Lin, J.W., & Jesus, D.O. (2021). Sensorineural Hearing Loss. StatPearls Publishing; 2021 Jan-. Available from: https://www.ncbi.nlm.nih.gov/books/NBK565860

Rakun, E., Arymurthy, A., Stefanus, L., Wicaksono, A., & Wisesa, I. (2018). Recognition of Sign Language System for Indonesian Language

Using Long Short-Term Memory Neural Networks. Advanced Science Letters. 24. 999-1004. 10.1166/asl.2018.10675.

Voulodimos, A., Doulamis, N., Doulamis, A., & Protopapadakis, E. (2018). Deep Learning for Computer Vision: A Brief Review. Computational Intelligence and Neuroscience, vol 2018, Article ID 7068349.