Emergence of Sign Language Recognition System into Text

Ankit Pandey [1], Aashika Dhakal [2], Prajin Khadka [3], Bishnu Bhandari [4], P. B. Jha [5], A. K. Mishra [6] & P. S. Aithal [7]

[1,2,3,4] B.Sc. CSIT Scholars, Madan Bhandari Memorial College, Kathmandu, Nepal: pbjha2039315@gmail.com

[5] Assistant Professor and Head, School of Science and Technology, Madan Bhandari Memorial College, Kathmandu, Nepal: prod@mbmc.edu.np

[6] D.Sc. Research Fellow, Srinivas University, India, and Associate Professor, Madan Bhandari Memorial College, Kathmandu, Nepal,
OrcidID: 0000-0003-2803-4918; Email: anjaymishra2000@gmail.com

[7] Professor, Institute of Management & Commerce, Srinivas University, Mangalore, India,
OrcidID: 0000-0002-4691-8736; E-mail:psaithal@gmail.com

Subject Area: Computer Science.
Type of the Paper: Research Paper.
Type of Review: Peer Reviewed as per |C|O|P|E| guidance.
Indexed In: OpenAIRE.
DOI: https://doi.org/10.5281/zenodo.7915843
Google Scholar Citation: IJAEML

How to Cite this Paper:
Pandey, A., Dhakal, A., Khadka, P., Bhandari, B., Jha, P. B., Mishra, A. K., & Aithal, P. S., (2023). Emergence of Sign Language Recognition System into Text. *International Journal of Applied Engineering and Management Letters (IJAEML)*, *7*(2), 91-101. DOI: https://doi.org/10.5281/zenodo.7915843

International Journal of Applied Engineering and Management Letters (IJAEML)
A Refereed International Journal of Srinivas University, India.

Crossref DOI: https://doi.org/10.47992/IJAEML.2581.7000.0176

Received on: 17/02/2023
Published on: 10/05/2023

Emergence of Sign Language Recognition System into Text

Ankit Pandey [1], Aashika Dhakal [2], Prajin Khadka [3], Bishnu Bhandari [4], P. B. Jha [5], A. K. Mishra [6] & P. S. Aithal [7]

[1,2,3,4] B.Sc. CSIT Scholars, Madan Bhandari Memorial College, Kathmandu, Nepal: pbjha2039315@gmail.com

[5] Assistant Professor and Head, School of Science and Technology, Madan Bhandari Memorial College, Kathmandu, Nepal: prod@mbmc.edu.np

[6] D.Sc. Research Fellow, Srinivas University, India, and Associate Professor, Madan Bhandari Memorial College, Kathmandu, Nepal,
OrcidID: 0000-0003-2803-4918; Email: anjaymishra2000@gmail.com

[7] Professor, Institute of Management & Commerce, Srinivas University, Mangalore, India,
OrcidID: 0000-0002-4691-8736; E-mail:psaithal@gmail.com

ABSTRACT

Purpose: *Communication is the passport of success. The objective of this undertaking was to fabricate a neural network ready to group which letter of the American Sign Language (ASL) letters in order is being marked, given a picture of a marking hand.*

Design/Methodology/Approach: *It was developed using Python programming with the help of TensorFlow for all AI-related tasks and Deep Learning.*

Findings/Result: *This examination is an initial move towards building a potential communication via gestures interpreter, which can take correspondences in communication through signing and make an interpretation of them into composed and oral language. Such an interpreter would enormously bring down the hindrance for some hard of hearing and quiet people to have the option to more readily speak with others in everyday collaborations. This objective is additionally propelled by the seclusion that is felt inside the hard of hearing local area. Dejection and sorrow exist at higher rates among the hard of hearing populace, particularly when they are drenched in a conference world.*

Originality/Value: *Enormous obstructions that significantly influence life quality originates from the correspondence disengagement between the hard of hearing and the conference. A few models are data hardship, impediments of social associations, and trouble coordinating in the public eye.*

Paper Type: *Research paper*

Keywords: Python programming, American Sign Language (ASL), Kaggle A-Z dataset, System Design, Deaf, Quality Life.

1. INTRODUCTION :

By 2050 almost 2.5 billion individuals are projected to have a few levels of hearing misfortune and something like 700 million will require hearing restoration. More than 1 billion youthful grown-ups are in danger of extremely durable, avoidable hearing misfortune. A yearly extra speculation of not exactly US$ 1.40 per individual is expected to increase ear and hearing consideration benefits all around the world. More than a 10-year time frame, these commitments and the arrival of almost US$ 16 for each USA dollar contributed [1]. Correspondence happens orally but its message passes through every nerve and cell of the body [2, 3, & 4]. It is this plurality of perspectives in concentrating on the very object that makes language a ceaseless peculiarity [2, 3, 4, and 5]. The issue we are effective financial planning is gesture based communication acknowledgment through solo component learning. Having the option to perceive communication via gestures is an intriguing PC vision issue while at the same time PC vision issue while all the while being very helpful for hard of hearing individuals to associate with individuals who don't have the foggiest idea how to figure out American

Sign Language (ASL) with help of data obtained from Standard MNIST 0-9 dataset and Kaggle A-Z dataset.
Notwithstanding the work we did on static pictures, we likewise made a live demo form of the task which can be run at somewhat less than 2 seconds for each edge to characterize marked hand signals from any individual [6, 7].
This paper centers on trying different things with various division draws near and unaided learning calculations to make a precise communication via the gestures acknowledgment model. To all the more effectively approach the issue and acquire sensible outcomes. Our methodology was to initially make an informational index showing the different hand tokens of the ASL letter set that we wished to order. The subsequent stage was to fragment out just the hand area from each picture and afterward involve this information for solo element learning [8, 9, 10].

In spite of the way that the not too sharp individuals can grant without issues among themselves, there is serious test for the conference impeded individuals attempting to speak with ordinary individuals. This is on the grounds that few out of every odd single average individual can appreciate their signal based correspondence. Most of the customary people have not been shown about the gesture based communication. As correspondence is basic, this issue definitely makes an impediment for the debilitated people to relate with the ordinary. Thusly, a communication through signing interpreter should be created to handle those issues through user-friendly interfaces, giving clickable access to its resources assuring the system can be accessed at any time; 24/7. This system can be useful to learn ASL Alphabets. This system can help to make communication with hearing impaired persons easier [11, 12, & 13].

2. OBJECTIVES :

The paper is aimed to generate a gesture recognizing system that can recognize sign gesture of Sign Language and translate it into text for Correspondence.

3. METHODOLOGY :

The present work entitled 'Emergence of Sign Language Recognition System into Text' is theoretical but techno-managerial in nature. It is developed using Python programming with help of TensorFlow for all AI related tasks and Deep Learning, NumPy for mathematical and logical operations, Pyttsx3 for specifically for converting text into speech and OpenCV which is a python library that handles real-time computer vision [14,15,16,17,18, & 19].

4. SYSTEM ANALYSIS AND DISCUSSION :

4.1 Functional requirements:

Technology is very important for existence in Society 5.0 [20]. Functional requirements show that the fact that two people are involved, only one of them interacts directly with the system. This is the hearing-impaired person that performs sings in front of the system responds by translating them to speech. The other person can be considered as passive user since it does interact directly with the system, but only listens/reads to recognized signs by describing the interaction that different actors (users, external systems) have with the system and where the: System (presented as a black box), Hearing Impaired (person that performs the signs) and Normal hearing (passive user of the system). Therefore, the system requirements can be specified as: Capture the real time Sign Language performed by impaired person, Predict the provided Sign language and convert to Text and Dictating the Text to Speech. Image should be of jpg, jpeg, and png format with a size of less than 4 Mb and resolution should be below 800*700 pixels. Figure 1 is very descriptive on how the actor (user) interacts straightforwardly with the framework by starting a utilization case, giving contribution to the framework and getting yield from the framework. Users upload an image which is first preprocessed.

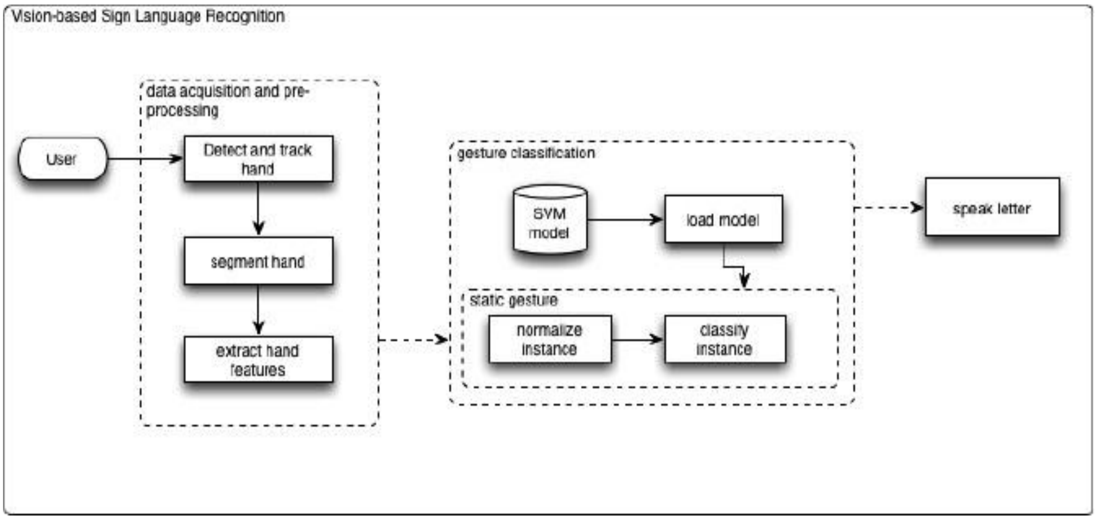

Fig. 1: Use Case Diagram of system

The preprocessing step includes conversion of the image into grayscale then into binary. Then the result is further normalized. Then the system initiates a segmentation process in which the image is segmented into subparts so that it can be recognized individually. Then the recognized characters are obtained as an output to our users [21].

4.2 Non-functional Requirements:

The Non-Functional Requirements for Sign Language Recognition System include Clean UI: The user interface should be clean and beginners friendly. Security: The application would not store any data extracted from documents permanently. Performance: The application would be fast and fetching of the data would take less time. Maintenance: Timely updates would be provided for bug fixes, and adding new features. Reliability: The features, information and data provided would be reliable and authentic. Scalability: The system is expected to be scalable so that it would be suitable for a very large data set. Efficiency: The performance of the system should be efficient. Accuracy: The system should be able to provide accurate results. Reliability: The system should be reliable with the results produced. To be reliable all the bugs and errors that may appear while running the system must be fixed. Requirements: This functionality requires Hardware as Android Phone, RAM: of at least 2GB, Memory: 2GB and Camera followed by at least Android Version 6 as software [14, 15, 16, 17, 18 & 21].

4.3 Feasibility of the System:

Feasibility study is one of the most important steps during project development in which a project is studied whether it is feasible to develop or not. The feasibility study is to check the viability of the project under Consideration. Theoretically, various types of feasibilities are conducted, but we have conducted three types of feasibilities;

4.3.1 Technical Feasibility: This may be done for a number of in this topic we deal with whether the project is technically feasible or not. Here, we deal with the capability of building the project with current available tools and resources. Our system is feasible in all senses. The system will be developed using Python programming. Libraries of python like Tesseract, NumPy and OpenCV will be used.

4.3.2 Operational Feasibility: In order to run our system basic knowledge of computers is sufficient. The operator does not require any knowledge of programming whatsoever. so, our system is operationally feasible.

4.3.3 Economic Feasibility: Economic Feasibility deals with the economic impact faced to implement a system. The cost of conducting a full system including the software and hardware cost for the development of the system must be considered. As our project is developed using a personal computer connected to the internet with no such sophisticated hardware devices being used and fewer human resources being involved so the proposed system is economically feasible [22, & 23].

5. SYSTEM ARCHITECTURE AND ALGORITHM :

5.1 System Design:

Let us explain Object-Oriented Analysis with help of a state diagram and activity diagram as shown in figures 2 and 3.

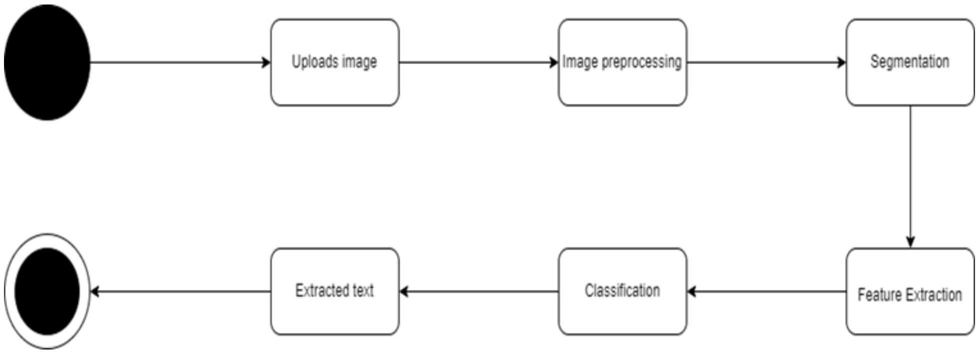

Fig. 2: State Diagram

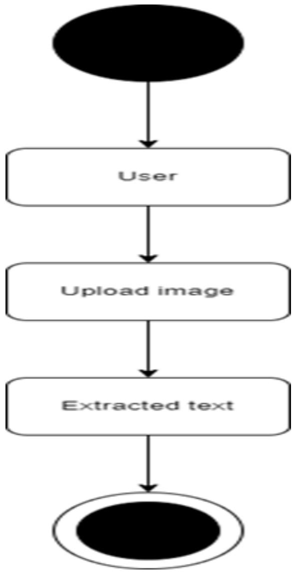

Fig. 3: Activity Diagram

5.2 System Architecture:

The architecture of our system is quite easy to understand. As shown in figure 4, users can upload an image with handwritten text on it. Then the system will extract features of the image, and segment the image. Segmentation is nothing but breaking the whole image into subparts to process them further. The characters are then recognized from the segmented subparts.

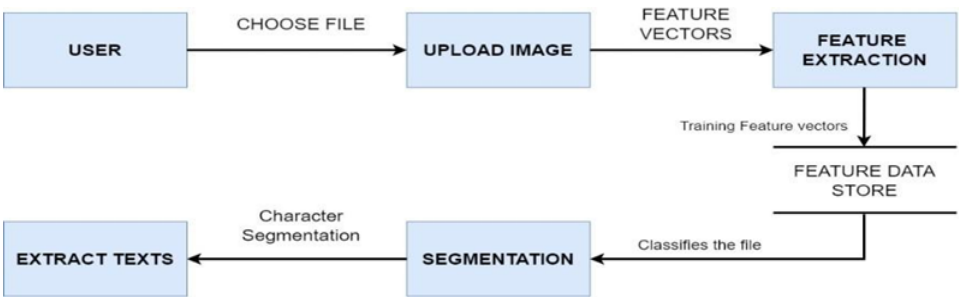

Fig. 4: System architecture of system

5.3 Convolutional Neural Network Algorithm:

Image classification is the process of taking an input image and determining which class or set of classes best describes it. In CNN, we take an image as input, give importance to the image's numerous aspects/features, and be able to distinguish one from the other. A CNN typically has three layers: a convolutional layer, pooling layer, and fully connected layer.

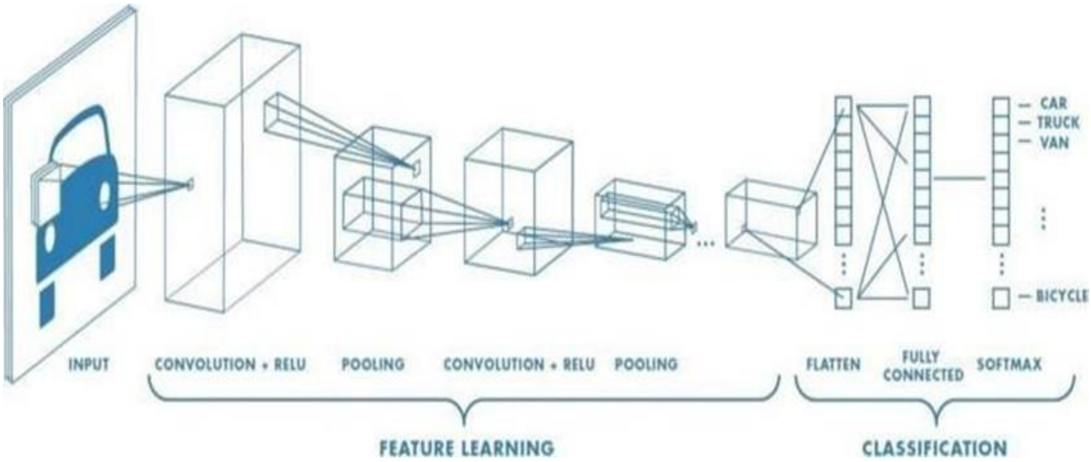

Fig. 5: Different layers in CNN

The convolution layer is the core building block of CNN and its main objective is to extract features such as edges, colors, and corners from the input. As we go deeper into the network, the network begins to recognize more complex features such as shapes, digits, and face parts [14,15,16,17,18 & 19].

This layer performs a dot product between two matrices, where one matrix (known as filter/kernel) is the set of learnable parameters, and the other matrix is the restricted portion of the image. At the end of the convolution process, we have a featured matrix which has fewer parameters (dimensions) than the actual image as well as more clear features than the actual one.

This layer is solely to decrease the computational power required to process the data which is done by decreasing the dimensions of the featured matrix even more. In this layer, we try to extract the dominant features from a restricted amount of neighbourhood.

There are two types of pooling techniques: Average pooling and Max pooling. In Average pooling, we take the average of all the values of the pooling region and in Max pooling, we take the maximum amongst all the values lying inside the pooling region. In this layer, the image is flattened into one column vector. The flattened output is fed to a feed-forward neural network and back propagation applied to every iteration of training.

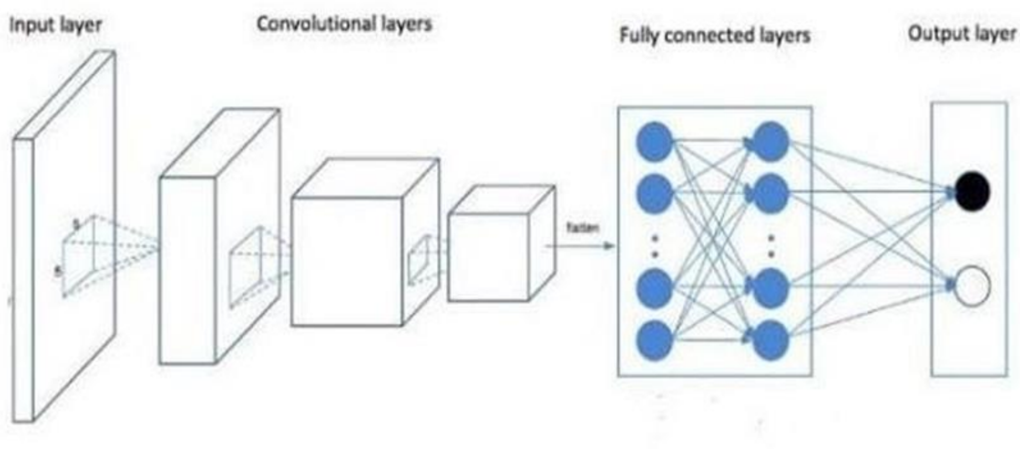

Fig. 6: Fully connected layer inside CNN [17,18,19]

5.4 Phases in Sign language Recognition:

First of all, datasets from two different sources, i.e., Standard MNIST 0-9 dataset & Kaggle A-Z dataset have been obtained. The standard MNIST 0-9 dataset can be seen on the left side of Figure 7.

The MNIST dataset contains 0-9 handwritten digits, each of which is contained in a 28 x 28 grayscale image. It has a training set of 60,000 examples, and a test set of 10,000 examples. The Kaggle A-Z dataset contains of A-Z capital letters taken from NIST Special Database 19 and rescales them to be 28 x 28 grayscale image, so it is in the same format as our MNIST data. It consists of 3,70,000+ alphabets in CSV format.

Fig. 7: Datasets

5.4.1 Pre-processing the User Input: We loaded the input image and converted it to grayscale, and then used Gaussian blurring to reduce the noise. Following that, we used the Canny module of cv2 library to detect the borders of our blurred image. We used contour detection to find the contours for each character. Then, we sorted the contours from "left to right".

5.4.2 Loading the Datasets: To train the model, first of all we will have to load the datasets into our program that we have previously collected. To load the MNIST 0-9 dataset, we can simply use the keras library to import the mnist module and then call its load data method. Likewise, to load the Kaggle A-Z dataset, first of all we have to download the dataset file which is in CSV. format and place it in the project folder. Then, we process and receive the data and labels from it.

5.4.3 Initializing the Model: The parameters we used for training the model are as follow:

- Epochs: 50
- Initial Learning Rate (LR): 0.1
- Batch Size (BS): 128

We partition the data into training and testing splits using 80% of the data for training and the remaining 20% for testing purposes. We construct our CNN architecture using the SGD optimizer and a standard learning rate decay schedule. The first three parameters of CNN's build technique show that each character/digit is represented as a 32x32 pixel grayscale picture. Then, we compile our model with the "categorical cross entropy" loss and the SGD optimizer we've used before.

5.4.4 Training the Model: We train our model using the model. Fit method. The following are the parameters:

- Aug. Flow: It establishes inline data augmentation
- validation data: Tests input images (testX) and test labels (testY).
- Steps_per_epoch: the numbers of batches that are run for each pass of the entire training data.
- Epochs: The number of complete runs over the entire data set during training.
- Class weight: Weights due to the imbalance of data samples for various classes.

- Verbose: Displays a training progress bar.

Following that, we assigned labels to each individual character. Then, we concatenated all of our digits and letters to generate an array with a single digit or number as each element. We generated predictions on the test set and produced our classification report to evaluate our model.

5.4.5 Saving the Model: Now, we need to save the model that includes the architecture and final weights after we've completed our training. Our model will be saved to disk as a Hierarchical Data Format version 5 (HDF5) file. Then we convert that .H5 file to .tFlite format so that we can use the converted to for out Flutter Application.

6. IMPLEMENTING AND TESTING :

Initially, our model had less accuracy and more loss. In the epoch 1, the training loss was 1.8590 and the training accuracy was 87.32%, meanwhile the testing loss was 0.6085 and the testing accuracy was 85.88%. Then, gradually training our model until epoch 10, the statistics were improved i.e., training loss was 0.6968 and training accuracy was 95.01%, meanwhile the testing loss was 0.4228 and testing accuracy was 94.92%. Clearly, the statistics obtained up to epoch 10 was not enough and the model certainly had more room for improvements. After training our model up to epoch 50 as seen from figure 12 our statistic was further improved. The training loss was 0.5696 and the training accuracy was 96.76%, meanwhile, the testing loss was 0.3586 and the testing accuracy was 97.17%. Clearly, there wasn't much room for improvement by increasing the epoch size after 50 epochs as the statistics were not improving much in the latter cycles. Also, there was a chance of overfitting if we had trained our model for more epoch cycles, which would decrease the accuracy of our model.

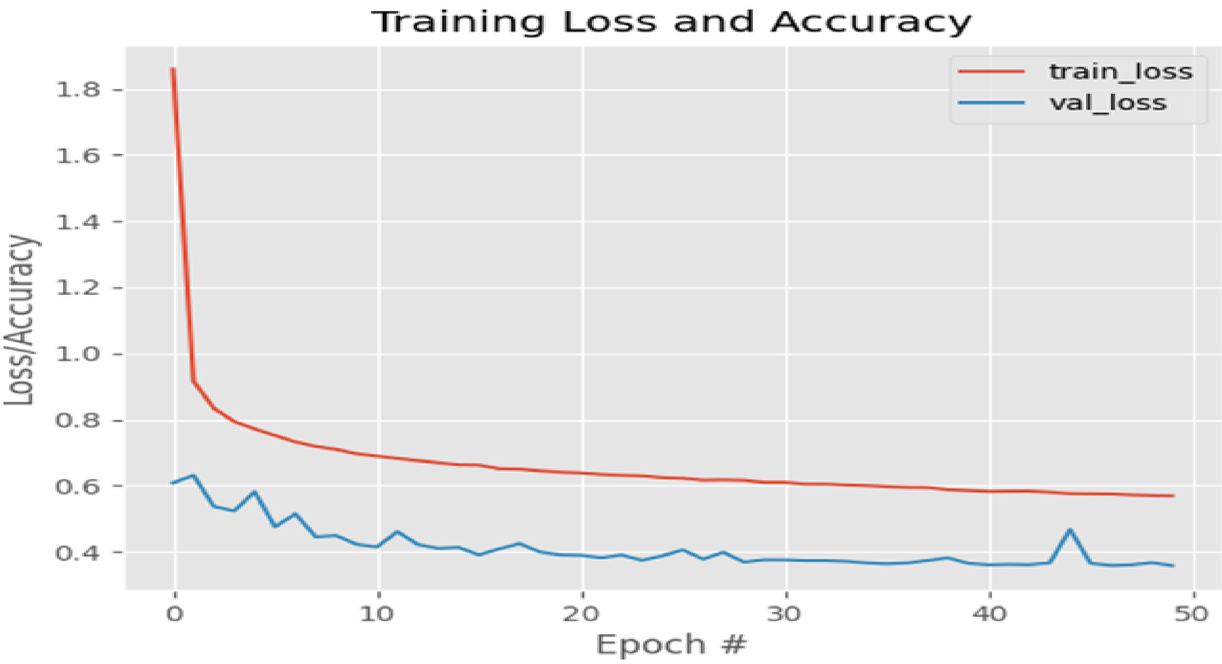

Fig. 8: Loss/Accuracy vs. Number of Epoch

From the plot in figure 8, we can clearly visualize the changing statistics of our model along with increase in number of epochs. It shows little sign of over fitting, which means that our model is performing well. We can also see that the training and testing accuracy is increasing and loss is decreasing from the graph. In figure 9, the performance measures of all the characters that have been tested using our model can be seen. All the scores obtained of the various performance measures (precision, recall, f1-score, support) for all the characters look pretty good. It can conclude that the accuracy of 97% is obtained in our model. The figure 10 represents the sample output from the testing set of data. The ASL model is performing well, but not perfect. It has successfully predicted the alphabet C from the hand gesture.

[INFO] evaluating network...

	precision	recall	f1-score	support
0	0.71	0.38	0.50	1381
1	0.98	0.98	0.98	1575
2	0.94	0.94	0.94	1398
3	0.99	0.99	0.99	1428
4	0.96	0.96	0.96	1365
5	0.88	0.90	0.89	1263
6	0.97	0.97	0.97	1375
7	0.98	0.99	0.98	1459
8	0.98	0.99	0.98	1365
9	0.99	0.98	0.99	1392
A	1.00	0.99	0.99	2774
B	0.98	0.99	0.99	1734
C	0.99	0.99	0.99	4682
D	0.89	0.99	0.94	2027
E	0.99	0.99	0.99	2288
F	0.96	1.00	0.98	232
G	0.98	0.96	0.97	1152
H	0.98	0.99	0.98	1444
I	0.97	0.99	0.98	224
J	0.98	0.98	0.98	1699
K	0.97	0.99	0.98	1121
L	0.97	0.99	0.98	2317
M	0.99	0.99	0.99	2467
N	0.99	0.99	0.99	3802
O	0.93	0.96	0.95	11565
P	1.00	0.99	0.99	3868
Q	0.97	0.99	0.98	1162
R	0.99	0.99	0.99	2313
S	0.99	0.98	0.98	9684
T	1.00	0.99	0.99	4499
U	0.99	0.99	0.99	5802
V	0.98	1.00	0.99	836
W	0.99	0.99	0.99	2157
X	0.99	0.99	0.99	1254
Y	0.98	0.97	0.97	2172
Z	0.93	0.97	0.95	1215
accuracy			0.97	88491
macro avg	0.97	0.96	0.96	88491
weighted avg	0.97	0.97	0.97	88491

Fig. 9: Classification report of all characters

Fig. 10: Sample output from the testing set

It is time to be smart through online vehicle maintenance to product identification resulting in smart village, not limited to the city for making everything consistent and applicable with proper e-governance as attempts are going on in Nepal [24, 25, 26, & 27].

7. CONCLUSIONS :

Our future will be powered by machines. We hope this report has provided some insight into how they learn. In the project, we have used image pixels as features vector and CNNas extractor and classifiers for handwritten digits recognition. CNN modeling is a time-consuming task and requires a CUDA enabled GPU for parallel processing.

We evaluated our studies using the A-Z dataset from kaggle and MNIST digits dataset, which is open to the public. From the results, it can see that our experiment result achieved 97.00% recognition accuracy. Now, we are going to integrate our Sign Language system with Node JS and build a web-based user interface where users can upload their image. In the future, we plan to work on the Devanagari script and optimize our system to obtain higher accuracy with lower implementation time. The measure limitations of the system are this system cannot recognize small case characters (a-z) as it was not available in the dataset and this system cannot recognize characters if they are connected which needs further research.

7. ACKNOWLEDGEMENT :

The author is thankful to all the professionals who took part in the discussions. The Author thanks to Saanvi Lavanya (Betkumar) for being with us as source of happiness. It is an academic exercise conducted at Madan Bhandari Memorial College,

REFERENCES :

[1] WHO, (2021). Deafness and hearing loss. https://www.who.int/news-room/fact-sheets/detail/deafness-and-hearing-loss.

[2] Anca SIRBU (2015). The Significance of Language as a Tool of Communication. *"Mircea cel Batran" Naval Academy Scientific Bulletin, 18*(2), 405-406. Available from: https://www.researchgate.net/publication/337472311_THE_SIGNIFICANCE_OF_LANGUAGE_AS_A_TOOL_OF_COMMUNICATION [accessed Feb 18 2023].

[3] Baxter, L. A. (2004). Relationships as dialogues. *Personal Relationships*, *11*(1), 1-22. DOI: https://doi.org/10.1111/j.1475-6811.2004.00068.x

[4] Braithwaite, D. O., Foster, E. A., & Bergen, K. M. (2018). Social construction theory: Communication co-creating families. In *Engaging Theories in Family Communication: Multiple Perspectives* (pp. 267-278). Taylor and Francis. Google Scholar↗

[5] Braithwaite, D. O., Waldron, V. R., Allen, J., Bergquist, G., Marsh, J., Oliver, B., Storck, K., Swords, N., & Tschampl-Diesing, C. (2018). "Feeling warmth and close to her": Communication and resilience reflected in turning points in positive adult stepchild-stepparent relationships. *Journal of Family Communication*, *18*(1), 92-109. DOI: https://doi.org/10.1080/15267431.2017.1415902.

[6] Islam M. Z., Hossain M. S., Ul Islam R., Andersson K. (2019). Static hand gesture recognition using convolutional neural network with data augmentation. In *2019 Joint 8th International Conference on Informatics, Electronics & Vision (ICIEV) and 2019 3rd International Conference on Imaging, Vision & Pattern Recognition (icIVPR)* (pp. 324-329). IEEE. Google Scholar↗

[7] Kartik, P.V.S.M.S., Sumanth, K.B.V.N.S., Ram, V.N.V.S., Prakash, P. (2021). Sign Language to Text Conversion Using Deep Learning. In: Ranganathan, G., Chen, J., Rocha, Á. (eds) Inventive Communication and Computational Technologies. Lecture Notes in Networks and Systems, vol 145. Springer, Singapore. https://doi.org/10.1007/978-981-15-7345-3_18.

[8] Suresh, R., Prakash, P. (2018). Deep learning-based image classification on Amazon web service. *J Adv Res Dyn Control Syst*, *10*(1), 1000–1003. Google Scholar↗

[9] Bantupalli K. Xie Y. (2018). American sign language recognition using machine learning and computer vision. In *2018 IEEE International Conference on Big Data (Big Data)* (pp. 4896-4899). IEEE. Google Scholar↗

[10] Beena M. V., Agnisarman Namboodiri, M. N. (2017). Automatic sign language finger spelling using convolution neural network: analysis. *Int J Pure Appl Math 117*(20), 9–15. Google Scholar↗

[11] Bheda V. and Dianna Radpour N. (2017). Using deep convolutional networks for gesture recognition in american sign language. CoRR, abs/1710.06836. Google Scholar↗

[12] Agarwal A., Thakur M. (2013). Sign language recognition using Microsoft Kinect. In *2013 sixth international conference on contemporary computing (IC3)* (pp. 181-185). IEEE. Google Scholar↗

[13] Sharma M., Pal R., Sahoo A. K. (2014). Indian sign language using neural networks and kNN classification techniques. *J Eng Appl Sci*, *9*(8), 1255–1259. Google Scholar↗

[14] https://www.tensorflow.org/api_docs

[15] https://keras.io/api/

[16] https://numpy.org/doc/

[17] https://pypi.org/project/pyttsx3/

[18] Bantupalli K. Xie Y. (2018). American sign language recognition using machine learning and computer vision. In *2018 IEEE International Conference on Big Data (Big Data)* (pp. 4896-4899). IEEE. Google Scholar↗

[19] Mandeep Kaur Ahuja, Amardeep Singh, (2015). "Hand Gesture Recognition Using PCA", *International Journal of Computer Science Engineering and Technology (IJCSET), 5*(7), 267-277. Google Scholar↗

[20] https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2093933/

[21] A. K. Mishra, Nepal Ananda, P. S. Aithal, (August 2022). Industry 4.0 Concept for Nepal - Operating Virtual Farming Industry. PP- 31-35, Proceedings on Future Trends in ICCT and its Applications in IT, Management and Education, Editors: Krishna Prasad, K., P. S. Aithal, & A. Jayanthiladevi, ISBN: 978-81-949961-8-7, DOI: https://doi.org/10.5281/zenodo.7215189.

[22] Sunitha K. A, Anitha Saraswathi.P, Aarthi.M, Jayapriya. K, Lingam Sunny, (2016). Deaf Mute Communication Interpreter- A Review. *International Journal of Applied Engineering Research, 11*(1). 290-296. Google Scholar↗

[23] Mishra AK. Project management: theory and practice from different countries. Project management: theory and practice from different countries. Tamilnadu: DKInternational Research Foundation. 2020. http://doi.org/10.5281/zenodo.4817542

[24] Mishra, A. K., Jha, P. B., & Aithal, P. S., (2023). Business Operation Using Identification of Product in Context of Developing Countries Emphasizing Nepal. *International Journal of Applied Engineering and Management Letters (IJAEML), 7*(1), 112-126. DOI: https://doi.org/10.5281/zenodo.7798162.

[25] Rajat Pokharel, Mishra, A. K., & Aithal, P. S., (2021). Practicability Assessment of Smart Village Project: A Case of Sandakpur Rural Municipality, Ilam Nepal. *International Journal of Management, Technology, and Social Sciences (IJMTS), 6*(2), 265-281. DOI: https://doi.org/10.5281/zenodo.5799392.

[26] Jha, P. B., Mishra, A. K., & Aithal, P. S., (2023). Operation of Vehicle Maintenance Systemin Context of Developing Countries Emphasizing Nepal. *International Journal of Case Studies in Business, IT, and Education (IJCSBE), 7*(1), 267-279. DOI: https://doi.org/10.5281/zenodo.7790868.

[27] Bhagat, C., Mishra, A. K., & Aithal, P. S., (2022). Model for Implementation of e-Government Services in Developing Countries like Nepal. *International Journal of Case Studies in Business, IT, and Education (IJCSBE), 6*(2), 320-333. DOI: https://doi.org/10.5281/zenodo.7139657.

****