Enhancing Sign Language Recognition using Deep Convolutional Neural Networks

Athanasios Kanavos*, Orestis Papadimitriou*, Phivos Mylonas† and Manolis Maragoudakis‡
*Department of Information and Communication Systems Engineering
University of the Aegean, Samos, Greece
{icsdd20017, icsdd20016}@icsd.aegean.gr
†Department of Informatics and Computer Engineering
University of West Attica, Athens, Greece
mylonasf@uniwa.gr
‡Department of Informatics
Ionian University, Corfu, Greece
mmarag@ionio.gr

***Abstract*—The advancements in sensing technologies and AI algorithms have opened up a wide range of possibilities for developing applications to meet the needs of individuals who are deaf or hard of hearing. Sign language plays a vital role in the lives of people with hearing and speaking disabilities. This research aims to explore digital image processing and machine learning methods for efficiently building a sign language dataset and creating a sign language interface system. The proposed system utilizes a Convolutional Neural Network (CNN) to analyze and interpret hand gestures and poses, converting them into natural language. The developed CNN model specifically focuses on improving the accuracy of predicting the American Sign Language alphabet. Despite variations in dataset conditions and size, the model achieved an exceptional accuracy rate of 98.73%. Additionally, it demonstrated a low loss value of 0.0539, indicating its robust performance.**

***Index Terms*—Sign Language Image Classification, Deep Learning, Image Processing, Convolutional Neural Networks, Sign Language Recognition**

I. Introduction

The fields of computer vision and deep learning are undergoing rapid advancements, primarily due to the emergence of new algorithms. These algorithms present an innovative way to facilitate human communication and enhance accessibility for individuals with hearing impairments [1]. Interestingly, despite the existence of numerous sign languages around the world, a single deep learning algorithm can be adapted to accommodate all of them by making minor adjustments to the training process and hyperparameters [3], [20]. Nonetheless, the creation of datasets for each language can be complex, expensive and time-consuming. Therefore, there is a need to explore techniques that can streamline this process [18].

In the realm of assisting individuals with disabilities in their communication and auditory needs, various methods have been devised. However, many individuals still encounter difficulties in obtaining appropriate assistance and effectively communicating with others in their daily lives. Fortunately, a novel approach has emerged to tackle this challenge and facilitate communication between disabled and non-disabled individuals. Significantly, there exist approximately 100 sign languages that are utilized for diverse purposes, including the categorization and comprehension of ideas expressed by individuals with disabilities [6].

Sign language plays a vital role in offering a mode of communication that transcends verbal language. This is especially crucial for individuals with hearing impairments or speech disabilities, as it provides them with an alternative means to interact and communicate with others. In order to assist individuals facing communication challenges, sign languages have been developed as a straightforward and effective method of communication, utilizing a system of signs and gestures to convey meaning [13].

While researchers strive to develop sign language recognition systems, they face implementation challenges, particularly in accurately recognizing hand gestures and poses. The complexity is further compounded by the resemblance of certain signs, making it difficult to create robust recognition systems. Moreover, sign language serves as a universal language, facilitating communication among individuals from diverse linguistic backgrounds, especially in multicultural settings or during emergencies [8]. Nonetheless, learning sign language can be challenging, as it requires memorizing numerous hand gestures and poses, some of which may appear remarkably similar. To overcome this hurdle, an automatic sign language recognition system becomes essential, enabling anyone to comprehend sign language effortlessly.

For a considerable period, researchers have recognized the importance of developing sign language technologies to assist individuals with hearing impairments in their communication and interaction with others [19]. However, the creation of these technologies poses challenges due to the diversity of sign languages and the lack of extensively annotated datasets. Despite these obstacles, recent advancements in AI and Machine Learning have played a pivotal role in automating and improving such technologies [2].

979-8-3503-1806-7/23/$31.00 © 2023 IEEE

This study aims to develop a robust and real-time system for recognizing alphabet sign language using deep learning. Deep learning has shown remarkable performance in image classification tasks, making it a promising approach for sign language recognition. The objective is to leverage the power of deep learning algorithms to accurately identify and classify alphabet signs in real-time scenarios.

The remaining sections of this paper are structured as follows: Section II presents an overview of the related literature pertaining to the problem at hand. Section III outlines the proposed model and its methodology. The research results and analysis are presented in Section IV. Finally, Section V summarizes the accomplishments of this study and discusses potential avenues for future development.

II. Literature Review

Several methods have been proposed to tackle the task of recognizing hand gestures in sign language. Initially, one approach involved the use of Support Vector Machines (SVM) for categorizing South African Sign Language (SASL), as demonstrated in the work by Naidoo [15]. Since then, researchers have been actively exploring sign language recognition for the past two decades. In this field, researchers collect input data and employ it to classify static sign language recognition systems. This literature review focuses not only on this particular class of recognition systems but also on various classifiers and their application in machine learning and deep learning-based approaches [14].

Gesture recognition systems in sign language rely on diverse methodologies employed by different researchers, leading to variations in approaches and accuracy levels [4]. However, it is worth noting that currently, no single system can achieve high accuracy across all conditions [12]. In an effort to enhance accuracy, researchers have turned their attention to Convolutional Neural Networks (CNNs) with various parameters for sign language recognition systems, leveraging the impressive performance of CNNs in image classification tasks [9]. Some studies have further augmented accuracy by combining CNNs with other techniques, while others have explored methods such as Support Vector Machines (SVM) and PCANET. Comparative analyses between CNNs and alternative approaches have consistently demonstrated the superiority and effectiveness of CNNs [16].

In a study by Pugeault [17], depth images captured by a Microsoft Kinect device were utilized, and a multi-class random forest classification approach was employed. The researchers evaluated the effectiveness of their method by conducting tests using different input types, including image-only, depth-only, and a combination of image with depth. The results indicated that the highest accuracy was achieved when depth information was combined with the image data. Notably, their system exhibited real-time classification capabilities due to its high speed. Similarly, Kang et al. [11] conducted a study where depth images were used as input, without the inclusion of color images.

In the research presented in [7], a novel model architecture called Dense Convolutional Network (DenseNet) was introduced. The primary contribution of this model was its unique approach to tackling the vanishing gradient problem that often occurs in deep networks. DenseNet addressed this issue by establishing direct connections between every layer in a feed-forward manner, enabling efficient feature reuse and reducing the number of parameters. Recognizing the advantages offered by DenseNet, we adopted this architecture as the foundation for our own network model.

In the study conducted by Daroya et al. [5], the researchers proposed a deep network specifically designed for sign language recognition. Their model achieved an impressive accuracy of 90.3%, which is comparable to the accuracy achieved by other works that utilized both RGB and depth images in their recognition systems.

III. Model

The objective of this paper is to examine different types of deep learning techniques that combine artificial intelligence principles with image classification techniques to classify sign language images. Numerous deep neural network architectures have been investigated, including convolutional neural networks, which are proficient at processing image data [10].

The three suggested designs are at first made use of with three various means, adhered to by the same specific design. Particularly, the differentation of these networks is depicted in the following Table I. All 3 networks make use of, in complying with GlobalAveragePooling2D, Flatten, Dense(256) as well as Dropout.

TABLE I
Architectures

Number	Architecture
1st	(Conv2D $\times 2$ - BatchNorm - MaxPooling2D - Dropout) $\times 3$
2nd	((Conv2D - BatchNorm) $\times 2$ - MaxPooling2D - Dropout) $\times 3$
3rd	(Conv2D $\times 3$ - BatchNorm - MaxPooling2D - Dropout) $\times 3$ - Conv2D $\times 2$ - BatchNorm - MaxPooling2D - Dropout

IV. Evaluation

A. Dataset

The dataset[1] is divided into two main folders(sign mnist train, sign mnist test) and each of these folders contains subfolders. Each training as well as examination situation works with a tag (0-25) as a one-to-one chart for every alphabetical character A-Z (as well as no scenarios for 9=J or even 25=Z due to action movements).

The training records (27,455 cases) as well as examination records (7172 cases) are actually roughly half the dimension of the conventional MNIST but otherwise comparable with a header row of tag, pixel1, pixel2 ... pixel784 which stand for a solitary 28x28 pixel picture along with grayscale worths between $0 - 255$. The authentic hand motion photo records stood for multiple customers redoing the action versus various

[1]https://www.kaggle.com/datasets/datamunge/sign-language-mnist

backgrounds. The sign language MNIST data arised from significantly extending the handful (1704) of the color images included as certainly not cropped around the hand region of rate of interest.

B. Results and Analysis

In this subsection, the speculative assessment is recommended. Specifically, Tables II to IV offer the results for the 3 designs in regards to epochs, accuracy, loss as well as time.

When examining the results for the first architecture, we observe that using a batch size of 32 leads to a noticeable decrease in loss from 0.6902 to 0.0619, indicating improved convergence. The accuracy of the model starts at 50% and steadily increases to 98%. With a batch size of 64, the model continues to converge well, with the loss decreasing from 0.7801 to 0.0948. The accuracy starts at 50% and reaches a maximum of 95%. However, when the batch size is increased to 128, although convergence is still observed, the loss slightly increases compared to the previous batch size, ranging from 0.8430 to 0.2124. The accuracy starts at 47% and reaches a maximum of 88%. Finally, with a batch size of 256, the model converges, but the loss remains higher than in previous batch sizes, ranging from 0.9515 to 0.4989. The accuracy starts at 50% and reaches a maximum of 78%.

Moving on to the second architecture, we find that using a batch size of 32 results in convergence, with the loss decreasing from 0.7495 to 0.1675. The accuracy starts at 55% and reaches a maximum of 87%. Similarly, a batch size of 64 yields good convergence, with the loss decreasing from 0.7128 to 0.2790. The accuracy starts at 55% and reaches a maximum of 85%. With a batch size of 128, the model shows convergence as well, with the loss decreasing from 0.7784 to 0.1372. The accuracy starts at 52% and reaches a maximum of 89%. However, when the batch size is increased to 256, the model still converges, but the loss is higher compared to previous batch sizes, ranging from 0.8855 to 0.6937. The accuracy starts at 48% and reaches a maximum of 53%.

Lastly, for the third architecture, using a batch size of 32 yields good convergence, with the loss decreasing from 0.7512 to 0.0716. The accuracy starts at 47% and reaches a maximum of 97%. Similarly, a batch size of 64 shows good convergence, with the loss decreasing from 0.8605 to 0.0539. The accuracy starts at 52% and reaches a maximum of 98%. With a batch size of 128, the model converges, but the loss increases compared to previous batch sizes, ranging from 0.9383 to 0.1723. The accuracy starts at 55% and reaches a maximum of 98%. Lastly, with a batch size of 256, the model shows convergence, but the loss remains higher compared to smaller batch sizes, ranging from 1.146 to 0.1494. The accuracy starts at 53% and reaches a maximum of 92%.

In summary, it can be observed that smaller batch sizes generally lead to better convergence and higher accuracy. However, it is important to consider that smaller batch sizes also increase training time. Among the three architectures, the third architecture consistently performs better in terms of accuracy and loss. Nevertheless, it should be noted that the third architecture also requires a longer training time compared to the other architectures.

V. Conclusions and Future Scope

In conclusion, this paper showcases a series of techniques employed to develop a convolutional neural network (CNN) specifically designed for multi-class sign language image classification. The proposed architectures solely rely on CNNs, leveraging their ability to process image data effectively. To assess the performance of the designs, experiments were conducted using different mini-batch sizes, including 32, 64, 128 and 256.

The results of the evaluation demonstrated the effectiveness of the CNN-based architectures in accurately classifying sign language images. The models achieved notable performance across all tested mini-batch sizes, showcasing their robustness and generalizability. It is worth noting that larger mini-batch sizes tended to yield slightly higher accuracy, indicating the potential benefits of utilizing more extensive training samples during the learning process.

Future research directions could focus on further optimizing the proposed CNN architectures by exploring additional hyperparameter tuning and network optimization techniques. Additionally, the utilization of larger-scale datasets and the investigation of transfer learning approaches could also be explored to evaluate the models' performance on diverse sign language variations and improve their overall accuracy and robustness.

References

[1] N. Adaloglou, T. Chatzis, I. Papastratis, A. Stergioulas, G. T. Papadopoulos, V. Zacharopoulou, G. J. Xydopoulos, K. Atzakas, D. Papazachariou, and P. Daras. A comprehensive study on deep learning-based methods for sign language recognition. *IEEE Transactions on Multimedia*, 24:1750–1762, 2021.

[2] S. Ameen and S. Vadera. A convolutional neural network to classify american sign language fingerspelling from depth and colour images. *Expert Syst. J. Knowl. Eng.*, 34(3), 2017.

[3] J. K. Chen, D. Sengupta, and R. R. Sundaram. Cs229 project final report sign language gesture recognition with unsupervised feature learning.

[4] D. Ciregan, U. Meier, and J. Schmidhuber. Multi-column deep neural networks for image classification. In *Proceedings of the IEEE computer society conference on computer vision and pattern recognition*, pages 3642–3649, 2012.

[5] R. Daroya, D. Peralta, and P. C. Naval. Alphabet sign language image classification using deep learning. In *IEEE Region 10 Conference TENCON*, pages 646–650, 2018.

[6] B. Garcia and S. A. Viesca. Real-time american sign language recognition with convolutional neural networks. *Convolutional Neural Networks for Visual Recognition*, 2(225-232):8, 2016.

[7] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger. Densely connected convolutional networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition CVPR*, pages 2261–2269, 2017.

[8] J. C. Isaacs and S. Y. Foo. Optimized wavelet hand pose estimation for american sign language recognition. In *IEEE Congress on Evolutionary Computation CEC*, pages 797–802, 2004.

[9] V. Jain, A. Jain, A. Chauhan, S. S. Kotla, and A. Gautam. American sign language recognition using support vector machine and convolutional neural network. *International Journal of Information Technology*, 13:1193–1200, 2021.

[10] A. Kanavos, E. Kolovos, O. Papadimitriou, and M. Maragoudakis. Breast cancer classification of histopathological images using deep convolutional neural networks. In *7th IEEE South-East Europe Design Automation, Computer Engineering, Computer Networks and Social Media Conference (SEEDA-CECNSM)*, pages 1–6, 2022.

TABLE II
EXPERIMENTAL EVALUATION FOR FIRST ARCHITECTURE: (CONV2D ×2 - BATCHNORM - MAXPOOLING2D - DROPOUT) ×3

Epochs	Loss	Accuracy	Time	Loss	Accuracy	Time	Loss	Accuracy	Time	Loss	Accuracy	Time
	Batch Size = 32			Batch Size = 64			Batch Size = 128			Batch Size = 256		
1	0.6902	0.5063	17	0.7801	0.5035	39	0.8430	0.4773	25	0.9515	0.5078	15
10	0.3424	0.8291	15	0.3791	0.7148	14	0.5944	0.6909	11	0.6601	0.6522	6
20	0.1745	0.8513	15	0.3500	0.7394	14	0.4295	0.7591	11	0.6175	0.6957	6
30	0.1495	0.9494	15	0.2762	0.8486	14	0.4139	0.7182	12	0.5833	0.7188	13
40	0.0943	0.9715	15	0.2534	0.8627	14	0.2983	0.8545	10	0.5201	0.7500	13
50	0.0845	0.9937	16	0.2026	0.9507	14	0.2684	0.8818	11	0.5492	0.7188	13
60	0.0852	0.9778	15	0.1884	0.9437	14	0.3086	0.8398	13	0.5294	0.7391	4
70	0.0611	0.9810	15	0.1157	0.9648	14	0.2288	0.8818	11	0.5087	0.7717	4
80	0.0747	0.9905	15	0.1182	0.9563	15	0.2532	0.8682	11	0.5034	0.7578	13
90	0.0684	0.9937	15	0.1161	0.9594	15	0.2816	0.8594	11	0.4739	0.7826	4
100	0.0619	0.9873	15	0.0948	0.9577	14	0.2124	0.8828	12	0.4989	0.7883	4

TABLE III
EXPERIMENTAL EVALUATION FOR SECOND ARCHITECTURE: ((CONV2D - BATCHNORM) ×2 - MAXPOOLING2D - DROPOUT) ×3

Epochs	Loss	Accuracy	Time	Loss	Accuracy	Time	Loss	Accuracy	Time	Loss	Accuracy	Time
	Batch Size = 32			Batch Size = 64			Batch Size = 128			Batch Size = 256		
1	0.7495	0.5506	17	0.7128	0.5563	16	0.7784	0.5273	15	0.8855	0.4883	16
10	0.3014	0.8513	18	0.4664	0.8380	15	0.2584	0.8516	13	0.6931	0.5000	13
20	0.1813	0.8734	15	0.4147	0.8486	14	0.2059	0.8318	12	0.6929	0.5312	14
30	0.1765	0.8861	16	0.4049	0.8556	15	0.2000	0.8455	12	0.6928	0.5435	5
40	0.1654	0.8906	16	0.3727	0.8873	15	0.1760	0.9000	12	0.6931	0.5039	13
50	0.1899	0.8797	16	0.3757	0.8592	15	0.1939	0.8818	12	0.6929	0.5234	13
60	0.1945	0.8576	16	0.3417	0.8838	15	0.1694	0.8727	12	0.6932	0.5000	14
70	0.1848	0.8544	16	0.3494	0.8687	17	0.1850	0.8633	13	0.6931	0.5039	12
80	0.1381	0.9051	16	0.3072	0.9014	15	0.1547	0.9000	12	0.6932	0.5000	14
90	0.1926	0.8734	15	0.3182	0.8803	15	0.1638	0.8727	11	0.6927	0.5326	5
100	0.1675	0.8703	15	0.2790	0.8531	17	0.1372	0.8909	12	0.6937	0.5388	14

TABLE IV
EXPERIMENTAL EVALUATION FOR THIRD ARCHITECTURE: (CONV2D ×3 - BATCHNORM - MAXPOOLING2D - DROPOUT) ×3 - CONV2D ×2 - BATCHNORM - MAXPOOLING2D - DROPOUT

Epochs	Loss	Accuracy	Time	Loss	Accuracy	Time	Loss	Accuracy	Time	Loss	Accuracy	Time
	Batch Size = 32			Batch Size = 64			Batch Size = 128			Batch Size = 256		
1	0.7512	0.4715	25	0.8605	0.5282	26	0.9383	0.5545	19	1.146	0.5326	11
10	0.5837	0.6519	24	0.4286	0.7250	25	0.6479	0.6182	17	0.6932	0.4783	7
20	0.3316	0.8544	24	0.1833	0.8662	23	0.5986	0.6091	18	0.6943	0.4239	9
30	0.1265	0.9494	23	0.0923	0.9366	21	0.4349	0.8750	19	0.6856	0.5000	8
40	0.1460	0.9747	25	0.1037	0.9401	22	0.2861	0.9045	17	0.5977	0.7188	21
50	0.1031	0.9873	24	0.0872	0.9331	24	0.2604	0.9062	19	0.5889	0.7500	23
60	0.1135	0.9557	24	0.0663	0.9469	24	0.2314	0.9364	18	0.4877	0.8478	8
70	0.1070	0.9842	23	0.1254	0.9085	23	0.1999	0.9453	19	0.3624	0.8750	19
80	0.1236	0.9620	24	0.0740	0.9507	21	0.2308	0.9258	19	0.3658	0.8594	19
90	0.0578	0.9937	24	0.0788	0.9683	22	0.1795	0.9864	19	0.1967	0.9102	19
100	0.0716	0.9778	23	0.0539	0.9812	25	0.1723	0.9855	19	0.1494	0.9258	19

[11] B. Kang, S. Tripathi, and T. Q. Nguyen. Real-time sign language fingerspelling recognition using convolutional neural networks from depth map. In *3rd IEEE IAPR Asian Conference on Pattern Recognition ACPR*, pages 136–140, 2015.

[12] D. Matthew Zeiler and F. Rob. Visualizing and understanding convolutional neural networks. ECCV, 2014.

[13] P. Mekala, Y. Gao, J. Fan, and A. Davari. Real-time sign language recognition based on neural network architecture. In *2011 IEEE 43rd Southeastern symposium on system theory*, pages 195–199, 2011.

[14] S. Nadgeri, D. Kumar, et al. An analytical study of signs used in baby sign language using mobilenet framework. In *Proceedings of the International Conference on Recent Advances in Computational Techniques (IC-RACT)*, 2020.

[15] S. Naidoo, C. Omlin, and M. Glaser. Vision-based static hand gesture recognition using support vector machines. *University of Western Cape, Bellville*, 1998.

[16] H. B. D. Nguyen and H. N. Do. Deep learning for american sign language fingerspelling recognition system. In *26th IEEE International Conference on Telecommunications ICT*, pages 314–318, 2019.

[17] N. Pugeault and R. Bowden. Spelling it out: Real-time ASL fingerspelling recognition. In *IEEE International Conference on Computer Vision Workshops ICCV*, pages 1114–1119, 2011.

[18] G. A. Rao, K. Syamala, P. Kishore, and A. Sastry. Deep convolutional neural networks for sign language recognition. In *IEEE conference on signal processing and communication engineering systems (SPACES)*, pages 194–197, 2018.

[19] A. Wadhawan and P. Kumar. Sign language recognition systems: A decade systematic literature review. *Archives of Computational Methods in Engineering*, 28:785–813, 2021.

[20] S. Zoupanos, S. Kolovos, A. Kanavos, O. Papadimitriou, and M. Maragoudakis. Efficient comparison of sentence embeddings. In *12th ACM Hellenic Conference on Artificial Intelligence SETN*, pages 11:1–11:6, 2022.