Sign Language Recognition : High Performance Deep Learning Approach Applyied To Multiple Sign Languages.

Abdellah El zaar[1,*], *Nabil* Benaya[1,**], and *Abderrahim* El allati[1,***]

[1]Laboratory of R&D in Engineering Sciences, FST Al-Hoceima, Abdelmalek Essaadi University, Tetouan, Morocco.

Abstract. In this paper we present a high performance Deep Learning architecture based on Convolutional Neural Network (CNN). The proposed architecture is effective as it is capable of recognizing and analyzing with high accuracy different Sign language datasets. The sign language recognition is one of the most important tasks that will change the lives of deaf people by facilitating their daily life and their integration into society. Our approach was trained and tested on an American Sign Language (ASL) dataset, Irish Sign Alphabets (ISL) dataset and Arabic Sign Language Alphabet (ArASL) dataset and outperforms the state-of-the-art methods by providing a recognition rate of 99% for ASL and ISL, and 98% for ArASL.
key-words: Sign Language Recognition, Machine Learning, Deep Learning, Convolutional Neural Network, Object Recognition.

1 Introduction

Object recognition is one of the active areas of artificial intelligence that has been studied in recent years. Various methods and techniques of machine learning and deep learning have been suggested and developed to solve object classification and recognition problems. Nevertheless, sign language recognition (SLR) is still a difficult task due to the diversity of languages and datasets. It is a very important area that can put an end to many problems that people with disabilities suffer from. Deaf people present an active part of society, so the main objective of our work is to help the rehabilitation of these people by implementing artificial intelligence and especially deep learning techniques to recognize the language of signs from images of static hand poses [1]. Hand gestures can be divided into poses as shown in figure1 depicted in static images [2] and dynamic poses depicted in videos [3, 4].

Among the methods used in SLR, we mention traditional methods 2 based on machine learning such as Support Vector Machines (SVM) [5], Hidden Markov Models (HMM) [6], etc. In addition, the approaches based on Deep Learning [7] is not require any feature extraction and preprocessing step. Due to their power and efficiency, especially in object recognition and image processing, deep learning techniques have attracted the attention of researchers (and industries). One of the most powerful deep learning tools is convolutional neural networks CNN [8], which outperforms many other methods in object recognition tasks, given their ability to work perfectly with very large datasets [9].

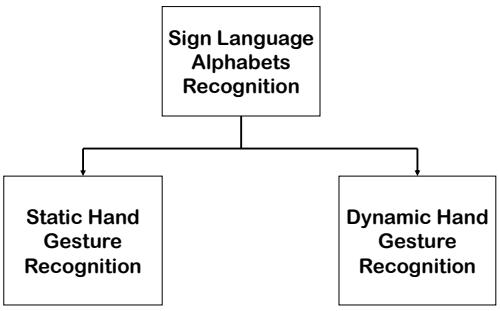


Figure 1: Sign Language Alphabets Recognition.

In order to provide a robust model, we choose to work with a large scale data set of sign language alphabets with enough classes and training examples. In practice, deaf people communicate using hand poses, for this reason we need images containing frontal views of hand poses to achieve high recognition rate. To make our work efficient, we worked with an American Sign Language (ASL) dataset which contains 29 classes and 87,000 images with a size of 200 * 200 pixels shown in figure3, an Arabic Sign Language (ArASL) dataset [10] with 54,049 images divided into 32 classes Irish Sign Language (ISL) dataset with 58,114 images [11].

Based on convolutional neural networks, we offer a high performance architecture based on the VGG model. Our architecture achieves a recognition rate of 99% on

*e-mail: abdellah.elzaar@gmail.com
**e-mail: nabil.benaya@gmail.com
***e-mail: abdou.allati@gmail.com

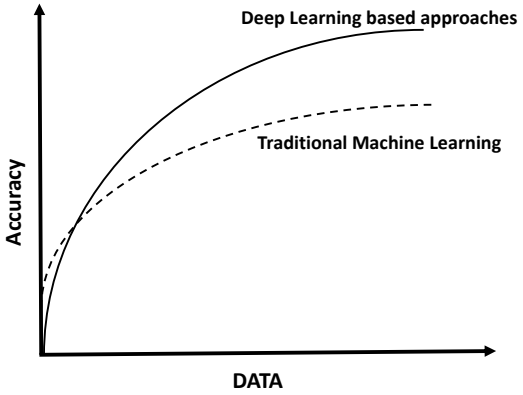


Figure 2: Deep Learning vs traditional Machine Learning based methods.

ASL and Irish datasets, 98% on ArASL2018 dataset.

Figure 3: American Sign Language Dataset.

2 Related work

In this section, we will explore some state-of-the-art works done in the field of Sign Language Recognition. We will also present some existed and open source datasets that are available to researchers. Existing Sign Language Recognition approaches are mainly divided into two subcategories: Approaches based traditional Machine Learning techniques such as Support Vector Machine (SVM), Hidden Markov Models (HMM) which uses Preprocessing, Feature extraction and classification steps, and Deep learning based approaches which does not require any preprocessing or Feature extraction steps. Among the existing sign Language, recognition approaches based Machine Learning; we find that S.Nagarajan and T.S.Subashini[2] used edge oriented histogram (EOH) based features and Multiclass Support Vector Machine (SVM) for classification to recognize static hand gestures. The system use the American Sign Language dataset as input and reaches a recognition rate of 93.75%. Fargad Yasir et *al.*[12] proposed a SIFT-based geometrically computational method to recognize Bangla sign language. To process and normalize the hand gesture images, they applied Gaussian distribution and grayscaling techniques. For the feature extraction step, they implemented scale invariant feature transform, after this step, they applied SVM classifier and they obtained a respective recognition rate with 88.33%. Aliaa A et *al.*[13] developed an automatic Arabic sign language (ArASL) recognition system based on the Hidden Markov Models (HMMs), they implemented a large dataset to recognize 20 isolated words from Arabic Sign Language. The approaches are experimented using real ArASL videos and reaches an accuracy of 82.22%. Among the works based on deep learning approaches, we find the system developed by Lean Karlo S et *al.*[14] the approach use the Convolutional Neural Network (CNN) and reaches a recognition rate of 90.04% on American Sign Language Alphabet. P.V.V. Kishore et *al.*[15] proposed the recognition of Indian Sign Language gestures using the powerful tool CNN. They employed the selfie mode on sign language video to perform the recognition process. A dataset with 200 sign in five different viewing angles used to train the CNN. They tested various architectures to obtain the better accuracy of 92.88%. To obtain an effective Sign Language Recognition and classification results, we need to train our Model architecture on a large-scale and diverse dataset that cover all hand-shape poses. The most released datasets in the literature we mention: the American Sign Language Dataset (ASL), Arabic Alphabets Sign Language Dataset (ArASL) and Irish Sign Language Dataset(ISL).

3 Proposed approach

The proposed system consists on recognition of Sign Language Alphabets using deep learning and especially Convolutional Neural Networks. Convolutional neural networks are very powerful on two-dimensional images and revolutionized the field of objects recognition given their capacity to operate with structural and defined on grid data[16]. An overview of our system is shown on figure4.

3.1 DATA

To ensure that our proposed architecture is powerful and effective, we trained it on several datasets such as the American Sign Language Dataset (ASL), Arabic Alphabets Sign Language Dataset (ArASL) and Irish Sign Language dataset (ISL). The ASL Dataset contains images of Sign Alphabets divided in 29 classes. The images are 200*200 pixel size and represents alphabets from A to Z in Addition to SPACE, DELETE and NOTHING, These three classes are very important in real-time classification. The ArASL Dataset contains 54049 images of Arabic Alphabets Sign language separated in 32 classes and

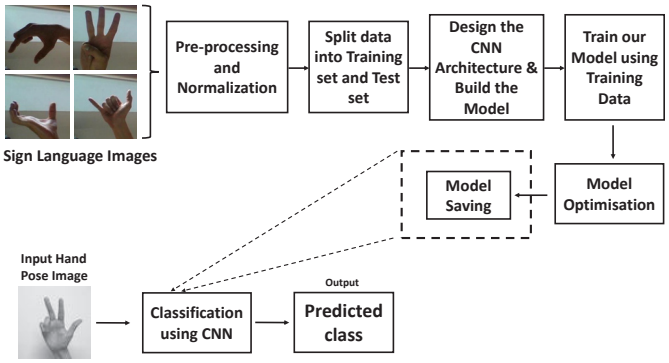
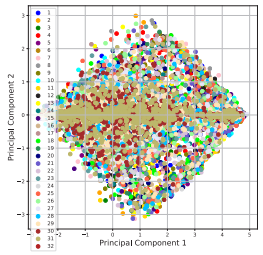
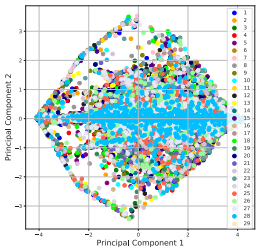


Figure 4: system overview.

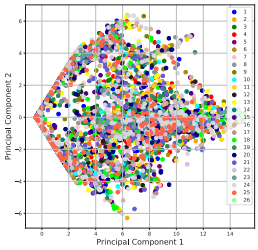
collected from 40 participants and Irish Sign Language dataset (ISL) with 58114 images for the 23 ISL hand-shapes alphabets. To visualize the given datasets we applied Principal Component Analysis (PCA)[17], which is an efficient method for data dimensionality reduction[18]. The figure 5 shows the visualization of the datasets.



(a) ArAsl Data visualisation



(b) ASL Data visualisation



(c) Irish Data visualisation

Figure 5: Sign Language Datasets Visualisation

3.2 Convolutional Neural Network

Convolutional Neural Network(CNN) is a powerful Deep Learning algorithm which performs well on image processing and objects recognition. The difference between the traditional Fully Connected Neural Network(FCNN) and CNN, is that the connection between the layers of FCNN is fully connected (the data is feeded successively from a layer to another from the input to the output), while the connection between the layers of CNN are organized with sparse (carefully designed). Each layer in the convolutional network is a 3-dimensional grid structure, which has a height, width, and depth. The Convolutional Neural Networks consists of three (03) main layers: Convolutional layer, pooling layer and fully connected layer.
The convolution operation is a dot product between an input layer and the filter. The filters(Kernels) are three dimensional parameters $K_n \times K_n \times d_n$ which represents the network parameters. The dimensions (length and width) of the output layer after performing the convolutional operation is :

$$L(n+1) = L_n - K_n + 1 \tag{1}$$

$$W(n+1) = W_n - K_n + 1 \tag{2}$$

The depth of the output layer is defined by the number of the used filters. The convolutional operation from the *nth* layer to the *nth* + 1 is defined as follows :

$$M_{ijz}^{n+1} = \sum_{r=1}^{K_n} \sum_{s=1}^{K_n} \sum_{k=1}^{d_n} w_{rsk}^{(z,n)} h_{i+r-1,j+s-1,k}^{(n)} \tag{3}$$

$$\forall i \in \{1...L_n - K_n + 1\}$$

$$\forall j \in \{1...W_n - K_n + 1\}$$

$$\forall z \in \{1...d_{n+1}\}$$

The output of the convolution layer is the input of the pooling layer. The pooling operation is defferent from the covolution, it consists of selecting a small size region of $P_n \times P_n$ and returning the maximum value of the selected region. Another parameter is included in the pooling operation which is the stride. if the stride $S_n > 1$, the dimensions of the resulting layer will be :

$$L_n - P_n + 1/S_n + 1 \tag{4}$$

$$W_n - P_n + 1/S_n + 1 \tag{5}$$

The final layer in the Convolutional Neural Network is the fully connected layer. This layer functions exactly like a traditional feed-forward neural network, and the data is forwarded from the input to the output. The main reason to employ the convolution layer is to increase the effectiveness of the Network.

3.3 Proposed Architecture

To construct our Model architecture, we based on VGGNet. VGGNet architectures conception shows that the depth of the netowork is a crucial factor to obtain better recognition and classification rate. Our Network contains four(04) convolutional layers equiped with ReLU(Rectified Linear Unit) activation function, two(02) pooling Layers and three(03) fully connected. Before feeding the images to our CNN, we need to perform the normalisation step. in every dataset we resized the images, for exemple in ASL dataset we changed the size from 200*200 pixels to 50*50 pixels not for our Network input's requirements , but to reduce dimension of the input images and to have a fast learning operation. The figure6 presents the proposed CNN architecture.

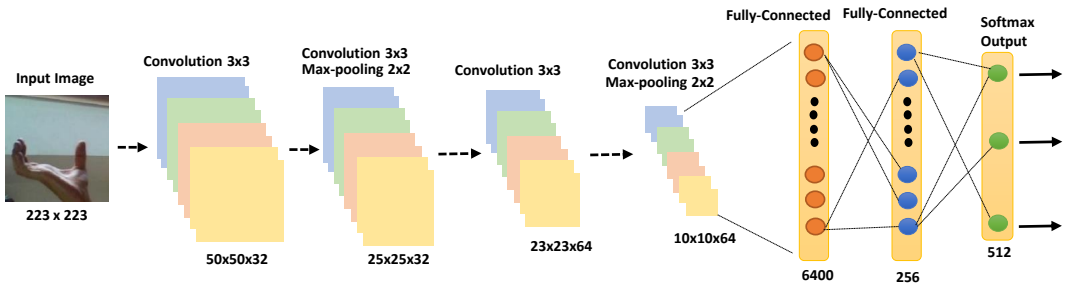


Figure 6: Proposed architecture for Sign Language Recognition

Table 1: Architecture used for ASL

Layer	Layer (type)	Details
1	Input image	$50 \times 50 \times 3$ American Sign Language images
2	(Conv2D)	Convolution operations with 32 filters of size (3×3) with a stride of 1 and paddig 0
3	(Conv2D)	Convolution operations with 32 filters of size (3×3) with a stride of 1 and padding 0)
4	(MaxPooling2D)	Max-pooing with a pool-size of (2×2) and a stride of 2
5	(Dropout)	0.23 dropout
6	(Conv2D)	Convolution operations with 64 filters of size (3×3) with a stride of 1 and paddig 1
7	(Conv2D)	Convolution operations with 64 filters of size (3×3) with a stride of 1 and paddig 1
8	(BatchNormalization())	Channel Normalisation
9	(MaxPooling2D)	Max-pooing with a pool-size of (2×2) and a stride of 2
10	(Dropout)	0.54 dropout
11	(Flatten)	Fully connected layer
12	(Dense)	Fully connected layer with 6400 neurons and ReLU
13	(BatchNormalization())	Channel Normalisation
14	(Dropout)	0.33 dropout
15	(Dense)	Fully connected layer with 512 neurons and ReLU
16	(BatchNormalization())	Channel Normalisation
17	(Dropout)	0.105 dropout
18	(Dense)	Fully connected layer with 29 neurons and ReLU

4 Experiments, comparison and discussion

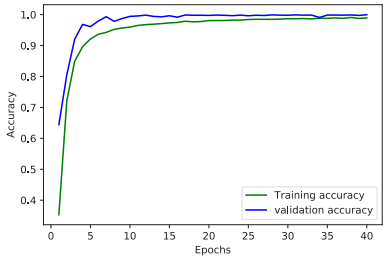
In this section we will discuss the performances and results acheived by our Model architecture using the ASL , ArASL and ISL datasets. we will also compare our work and results with other state-of-the-art methods done in the field of Sign Language Recognition using Deep Learning methods and especially Convolutional Neural Network(CNN).

4.1 experiments and results

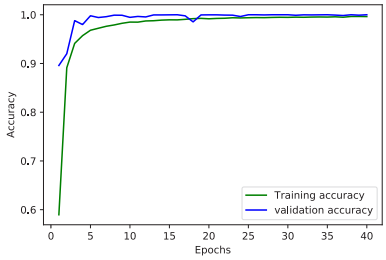
The algorithmes were implemented using Python language (we used several libraries that contains visualisation and image processing functions such as $Tensorflow^{TM}$, $Keras^{TM}$ Models and $Sklearn^{TM}$). The model run on Microsoft Arure virtual machine with six(06) core processor, 56Go of RAM. Working with Convolutional Neural Networks with a high number of epoches also visualising a high number of data can be time-consuming process, For this reason we used TESLA K80 NVIDIA GPU. The used GPU boost gives a superior performance to our Model and accelerates Libraries and training process.

4.2 experiments on ASL and ISL datasets

Before feeding the two datasets ASL and ISL into the CNN, we first splitted them into Train data and Test data. The ASL dataset is divided into 29 classes where 3000 images on each classe are dedicated for training and 200 images for testing. The ISL dataset contains 58120 images in which 200 images are dedicated for testing. BatchNormalization and dropout optimization are used to maintain the effectiveness of our Model and also to avoid the well-known overfitting problem. A batch size of 32 was selected and the number of epochs was 40. To evaluate the performances of our model, we used the accuracy metric which consists of dividing the well predicted samples by the total number of predictions. The figure 7 represents the accuracy obtained using our CNN architecture on ASL and ISL datasets, the model acheives better score with 99% on both datasets, which is a perfect recognition rate compared with other state-of-the-art-methods.



(a) Acurracy of our CNN architecture using ASL



(b) Acurracy of our CNN architecture using ISL

Figure 7: Accuracy of proposed Architecture

To see that our Model performs well on test images, used the confusion matrix represnted in figure8. We observe in the diagonal of our confusion matrix that all values are equal to 200 which is the the number of samples of each class in the testset. in other side we remarque that all other values are represented as zeros, which means that all images of each class are perfectly recognized by the model. this proves the effectiveness and the high accuracy obtained by our Model.

4.3 experiments on ArASL dataset

For ArASL dataset, we used the same methode and experiments, our CNN model is trained using 54050 images of Arabic Sign Language divided by 32 classes (compared with 29 classes for ASL Dataset and 26 for ISL) and reaches a high accuracy of 98%. the figure 9 illustrates the accuracy acheived by our model.

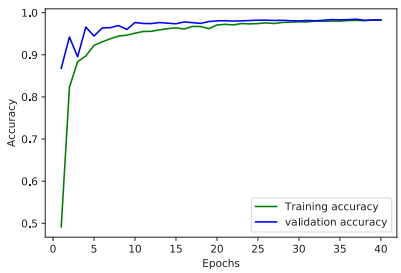


Figure 9: Acurracy of our CNN architecture using ArASL

It can be seen that the use of our CNN approche on ASL and ISL dataset gives a better results compared with it's use on ArASL. This is explained by the reduction of number of classes(29 classes in ASL, 26 on ISL against 32 in ArASL) of Sign alphabets to be recognized and classified. This is the only reason for the accuracy difference, because the two databases are already cropped and they did not require any pre-processing step. Similar to the previous subsection, we present the confusion matrix of our learning strategy, a perfect recognition rate is acheived.

5 Comparative results

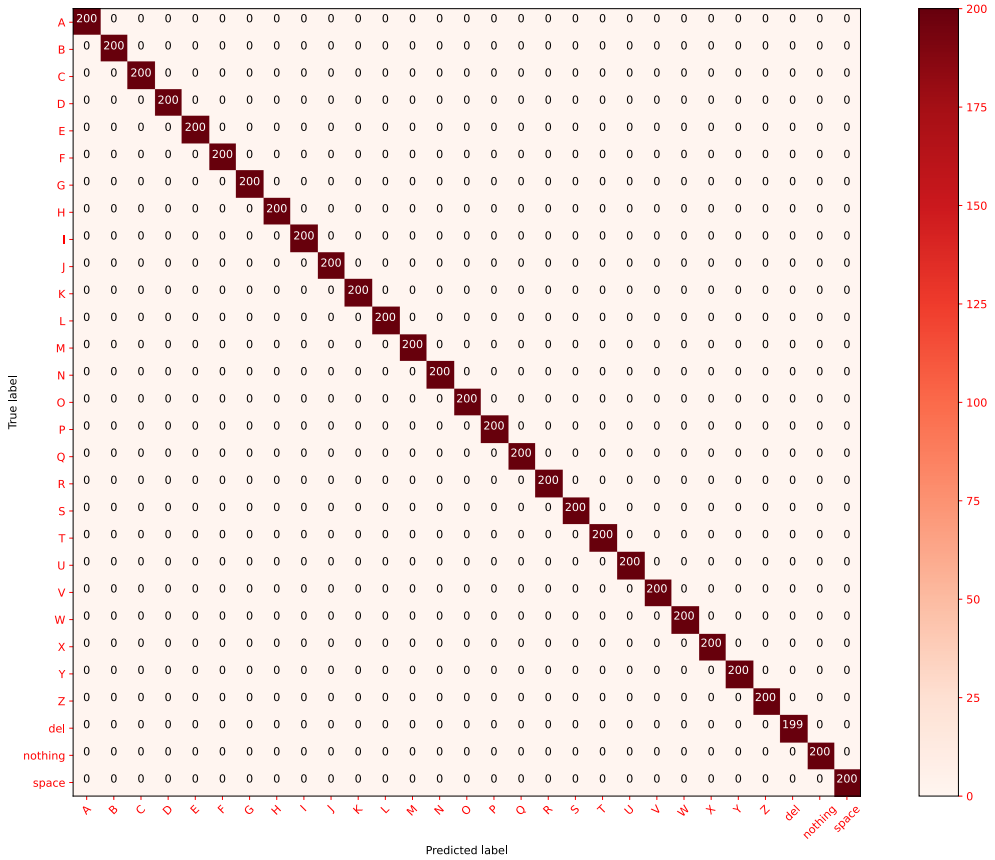
In addition to our proposed approach, Table2 presents the work done and acheived results in the state-of-the-art of Sign Language Recognition. We observe that our proposed approach shows high accuracy and outperforms the methods done in the field of Sign Language Recognition. The datasets that we used are wide and challeging as they contains divers hand poses positions and styles. Furthermore, our approach is fast and provide less complex structure than other approaches.

Table 2: Comparison with previous works on Sign Language recognition

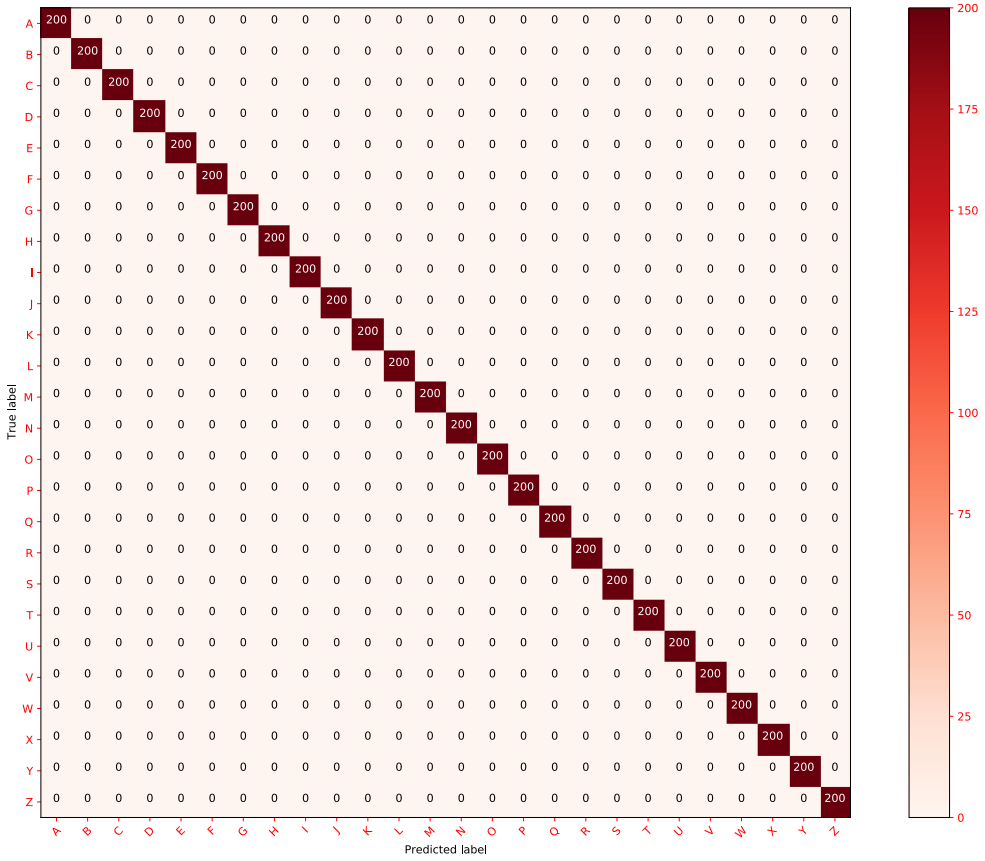
Authors	Method	Database	Train and Test data size	Recognition rate
Present approach	**CNN based VGGNet with Dropout**	**ASL ArASL IrishSL**	**58114 images 54049 images 58120 images**	**99% 98% 99%**
P.V.V. Kishore et *al.*[15]	CNN	Indian sign language(ISL)	12000	92.88%
Aliaa A et *al.*[13]	HMM	ArSL	54049 images	82.22%
Nagarajan et *al.*[2]	EOH SVM	ASL	58114 images	93.75%
Nikhil et *al.*[1]	Deep Neural Network	ASL	58114 images	83.29%
Lean Karlo S et *al.*[14]	CNN	ASL	58114 images	90.04%

6 Conclusion

In our work, we proposed a high performance Deep Learning architecture for Sign Language Recognition. We used Convolutional Neural Network to recognize static hand pose images. The recognition rate reaches 99% on American Sign Language and Irish Sign Language datasets, and 98% on ArASL dataset. The proposed methode provides less complex architecture and performs perfectly on very large datasets. Our proposed approach performs all other state-of-the-art methods on the field of Sign Language recognition. As a futur studies, we plan to investigate in CNN to performe dynamic hand poses recognition. In addition, we intend to combine several methods and classifiers for dynamic sign recognition.



(a) Confusion matrix using CNN on American Sign Language Dataset



(b) Confusion matrix using CNN on Irish Sign Language Dataset

Figure 8: Confusion Matrix of proposed Architecture

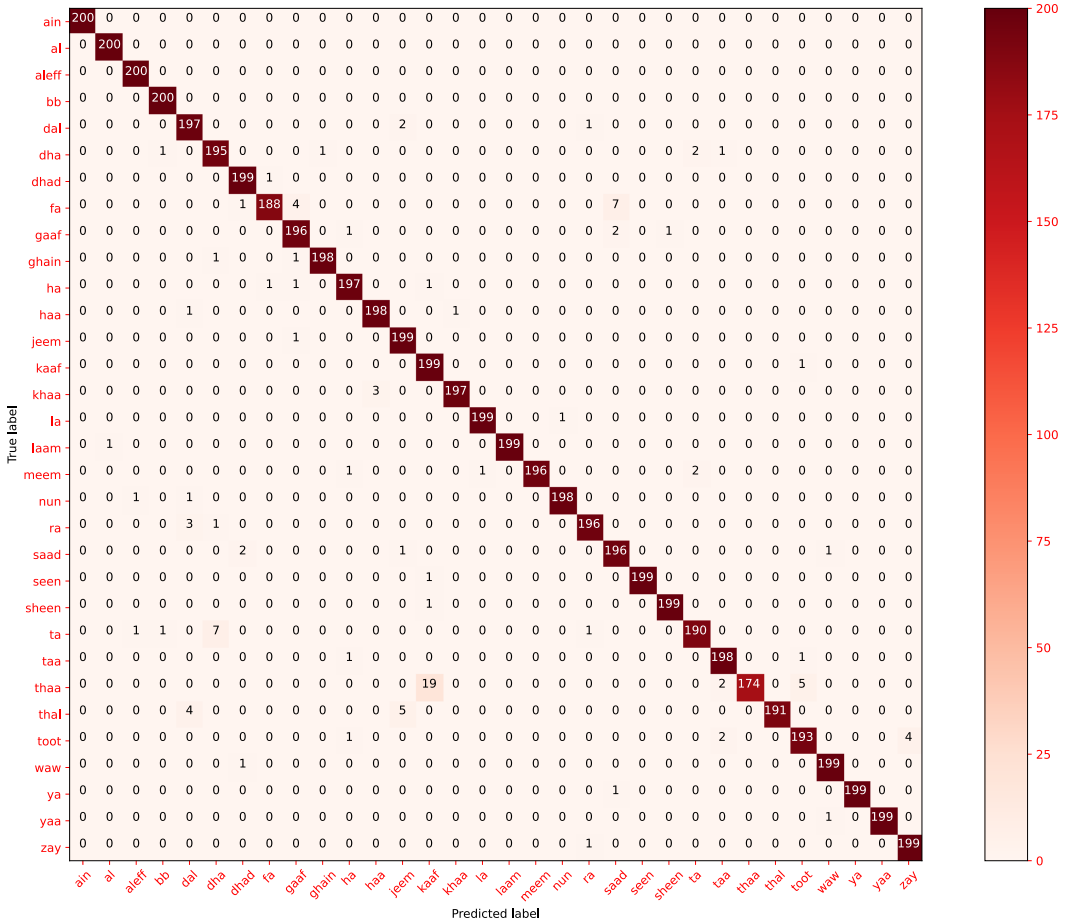


Figure 10: Confusion matrix using CNN on Arabic Sign Language Dataset

References

[1] N. Kasukurthi, B. Rokad, S. Bidani, D. Dennisan et al., arXiv preprint arXiv:1905.05487 (2019)

[2] S. Nagarajan, T. Subashini, International Journal of Computer Applications **82** (2013)

[3] D. Li, C. Rodriguez, X. Yu, H. Li, *Word-level deep sign language recognition from video: A new large-scale dataset and methods comparison*, in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision* (2020), pp. 1459–1469

[4] J. Singha, K. Das, arXiv preprint arXiv:1306.1301 (2013)

[5] J. Raheja, A. Mishra, A. Chaudhary, Pattern Recognition and Image Analysis **26**, 434 (2016)

[6] T. Starner, J. Weaver, A. Pentland, IEEE Transactions on pattern analysis and machine intelligence **20**, 1371 (1998)

[7] O. Koller, O. Zargaran, H. Ney, R. Bowden, *Deep sign: Hybrid CNN-HMM for continuous sign language recognition*, in *Proceedings of the British Machine Vision Conference 2016* (2016)

[8] C.C. Aggarwal et al., Springer **10**, 978 (2018)

[9] P. Wang, E. Fan, P. Wang, Pattern Recognition Letters **141**, 61 (2021)

[10] S.M. Shohieb, H.K. Elminir, A. Riad, Journal of King Saud University-Computer and Information Sciences **27**, 68 (2015)

[11] M. Oliveira, H. Chatbri, Y. Ferstl, M. Farouk, S. Little, N.E. O'Connor, A. Sutherland (2017)

[12] F. Yasir, P.C. Prasad, A. Alsadoon, A. Elchouemi, *Sift based approach on bangla sign language recognition*, in *2015 IEEE 8th International Workshop on Computational Intelligence and Applications (IWCIA)* (IEEE, 2015), pp. 35–39

[13] A. Youssif, A.E. Aboutabl, H.H. Ali, International Journal of Advanced Computer Science and Applications (IJACSA) **2** (2011)

[14] L.K.S. Tolentino, R.S. Juan, A.C. Thio-ac, M.A.B. Pamahoy, J.R.R. Forteza, X.J.O. Garcia, International Journal of Machine Learning and Computing **9**, 821 (2019)

[15] P. Kishore, G.A. Rao, E.K. Kumar, M.T.K. Kumar, D.A. Kumar, International Journal of Intelligent Systems and Applications **10**, 63 (2018)

[16] V. Bheda, D. Radpour, arXiv preprint arXiv:1710.06836 (2017)

[17] F. Han, H. Liu, Journal of the American Statistical Association **109**, 275 (2014)

[18] M. Partridge, R.A. Calvo, Intelligent data analysis **2**, 203 (1998)