SN Computer Science (2022) 3:481
https://doi.org/10.1007/s42979-022-01394-5

ORIGINAL RESEARCH

An Intelligent Kurdish Sign Language Recognition System Based on Tuned CNN

Hunar Abubakir Ahmed[1] · Sazgar Yassin Mustafa[1] · Sumaya Zrar Braim[1] · Razawa Mohammed Rasull[1]

Received: 8 January 2022 / Accepted: 27 August 2022 / Published online: 14 September 2022

Abstract

Hearing-impaired individuals have both hearing and speech disabilities. Therefore, they use a special language that involves visual gestures—known as "sign language"—for communicating ideas and emotions. Recognizing the gestures contained in sign language enables deaf people communicate more effectively with their interlocutor. It also helps people without such disabilities understand and identify those signs, thereby enriching the communication. However, designing a system that can automatically identify the signs of Kurdish sign language is a challenging task, especially for Kurdish sign language. This is attributable to the unavailability of a dataset and lack of standardized sign language. In this study, we investigate the problem by collecting a dataset of seven static signs and designing a model for sign recognition. The dataset consists of 3690 high-resolution images taken mostly from college students. To develop the classifier, a four-layer convolutional neural network model with a filter size of 5×5 was designed. To compare the model performance, two other pre-trained networks, namely MobileNetV2 and VGG16, were trained and fine-tuned using the same dataset. After a variety of hyperparameter fine-tuning, the proposed approach achieved the same outcome as the two pre-trained networks, with an accuracy of 99.75%. That is, the model identified 396 of the 397 images in the test set. In addition, we performed an external test using 58 images of various signs, and the model approximately classified all the images correctly. This demonstrates that our approach achieved an outstanding result, which can be considered a first in the field.

Keywords Sign language recognition · Data acquisition · CNN · Transfer learning

Introduction

Hearing-impaired individuals are those who suffer from hearing and speaking impairments. They use a special form of language to communicate their own and interpret others' thoughts and feelings. This special language is known as sign language and involves a sequence of hand/arm movements, facial expressions, as well as special body/head gestures. However, like other human languages, sign language also varies per country/region, and occasionally, even within the same country, there exist different signs for the same phrase [1]. There are about 300 sign languages used by 70 million hearing-impaired individuals throughout the world as reported by the World Federation of the Deaf [2]. In addition to that, static and dynamic signs are two types of signs that are representative of arm/hand/finger movement and shape. In the static sign, the arm/hand/finger is shown in a particular fixed shape, while in the dynamic sign, the arm/hand/finger must shift from one shape to another [3].

Besides, only a small percentage of the population employs sign language for communication, whereas the majority does not use or understand it completely. As a result, there exists a communication gap between deaf and non-deaf people. Consequently, hearing-impaired people become victims of the absence of a common approach for communication among these two groups of people. Furthermore, this gap becomes a significant barrier in their path to complete their schooling or simply becoming knowledgeable and educated. Likewise, most technology systems,

✉ Hunar Abubakir Ahmed
hunar.abubakir@uor.edu.krd

Sazgar Yassin Mustafa
sazgar.411317020@uor.edu.krd

Sumaya Zrar Braim
sumaia.411318036@uor.edu.krd

Razawa Mohammed Rasull
razawa.411318001@uor.edu.krd

[1] University of Raparin, Raniyah, Iraq

particularly those used for communication, are inaccessible to deaf individuals. This gap can be decreased through designing an intelligent system that recognizes and converts each expressed sign into voice or text data [4–6].

Recently, several scholars have been working on developing an automated system capable of classifying sign language gestures into discrete classes. In general, image- and sensor-based are two different approaches utilized by researchers for designing such systems. On the one hand, sensor-based technique requires the signer has to wear a particular type of glove with built-in sensors that collect data on the hand's movement and shape. As a result, the procedure becomes extremely complicated and time-consuming. On the other hand, the image-based technique requires a camera to capture the signer while doing the signing. This is a much simpler and quicker way, and it aids the system's recognition of the sign [1, 7, 8].

Kurdistan is an autonomous region in northern Iraq that is home to Kurdish-speaking people. Iraqi people are primarily of Arabic nation and they speak Arabic. Kurdish and Arabic are essentially separate languages. Individuals who are deaf and unable to speak in Kurdistan utilize Kurdish sign language, which is a different yet related form of Arabic sign language. The lack of standardized sign language has made this field harder; each district in the Kurdistan region has a certain degree of variance in sign language. Even deaf individuals have difficulty communicating with each other. Finally, some sources refer to the Kurdish sign language as "Zmani Hemay Kurdi," which is abbreviated to ZHK [9].

Few scholars have studied Kurdish sign language in recent years, with no more than three or four publications published in local journals. This work proposes a dataset of seven static hand gestures for simple Kurdish words and phrases in an effort to overcome this barrier. To show the reliability of the dataset, we designed a classifier based on deep learning and compared it to other classifiers.

The remaining sections of the work are organized as follows. The next section describes some of the work done in the previous studies. In the subsequent section, we describe the data collection and separation, proposed architecture, and transfer learning followed by which we describe the setup used to build and train the model. After this, we describe the results achieved through multiple experiments. Then we describe and compare achieved results. In the penultimate section, we describe how well the model will do with external data. Finally, we conclude the study.

Related Works

In the last 2 decades, many academicians have focused on classifying sign language data either from sensor-based devices or from images. Owing to the diversity in sign languages just like spoken languages, scholars worked on several kinds of sign languages such as American, British, and Indian Sign Language, etc., especially after the prevalence of and advancement in deep learning-based algorithms. For instance, in Ref. [10], the authors applied a pre-trained Inception V3 network on a dataset of 9400 images for interpreting Swedish hand alphabet signs. This work achieved 85% accuracy. In Ref. [11], the authors utilized a skeleton 3D CNN network along with 2D CNN jointly to recognize Arabic sign language. The authors built a dataset that consists of a set of 40 static and dynamic signs such as numbers, Arabic alphabets, and commonly used Arabic signs. They trained their model in three stages, namely, dependent, independent, and mixed-mode. For dynamic and static signs, in the dependent stage, the authors achieved an accuracy of 98.39% and 86.34%, respectively, while in the independent stage, they achieved an accuracy of 96.69% and 86.34%, respectively. Subsequently, in the last stage, they mixed both dynamic and static signs and attained an accuracy of 89.62% for the signer-dependent mode and 88.09% for the signer-independent mode. Following that, the researchers in [12], proposed a model for classifying Indian sign language using a video frame dataset that was prepared by themselves. The work used a combination of deep learning and long short-term memory networks to classify gestures. They gained an average accuracy of 76.21%. The authors of [13] proposed a system for classifying Czech sign language single-handed alphabet signs from an image dataset using convolutional neural network (CNN). Their method achieved an accuracy of over 87%. The authors of Ref. [14] proposed a system for recognizing two hand gestures of Indian sign language. The model is trained on a dataset that consists of 9100 images of size 50×50. The images represent 26 English alphabets with 350 images for each letter. They used different machine learning techniques to benchmark their dataset and achieved an overall accuracy of 91.2%. The authors in [15] provided Indian deaf individuals with a small tool in the form of a browser extension with 1600 words. The extension works as a dictionary for Indian sign language, and the study achieved an AUC of 85.7%. In [16], the authors proposed a model for classifying sentences in Indian sign language. They utilized three deep learning techniques to train their model, namely, multi-channel DCNN, time-LeNet, and an improved time-LeNet technique to overcome the overfitting problem. The work provides the result for each of the three algorithms, which are 83.94%, 79.70%, and 81.62% for multi-channel DCNN, time-LeNet, and improved time-LeNet, respectively. The authors of Ref. [17] proposed an Arabic sign language recognition system based on dual leap motion controllers (LMCs). The key concept of the work involved using the front and side LMCs. The features were then extracted by utilizing linear discriminant analysis and a Bayesian technique with a Gaussian

mixture model. Afterward, they described a fusion approach, which they called Dempster–Shafer, to merge information from both LMCs. The model trained on a dataset of 100 Arabic dynamic signs, and they obtained an accuracy of about 92%. The authors in [18] presented a work that demonstrates the ability of deep learning models and optimizers in recognizing Indian sign language. They explored using a three-layer built from scratch CNN along with several pre-trained networks like ResNet152V2. The proposed work trained on a dataset for Indian static signs of numbers and alphabets. The work achieved an accuracy of 96.2% and 90.8% on the number and alphabet signs, respectively, while training on ResNet architecture. On the other hand, the work gained better accuracy when trained on three-layer scratch CNN, which is 99.0% and 97.6% on the number and alphabet signs, respectively. The authors of Ref. [19] designed a framework for classifying video-level Chinese sign language based on local–global feature description using two famous datasets, namely, SLR_Dataset and DEVSIGN_D. The work presents a global residual three-dimensional network that consists of an attention layer and a target detection-based local network. The enhanced timing conversion layer has been used to investigate timing information from various eras and acquire video representations of various timings. The work attained an accuracy of 89.2% and 91% for SLR_Dataset and DEVSIGN_D, respectively. In Ref. [20], the authors proposed an edge computing end-to-end system for classifying digits of sign language from thermal images. They used a dataset of 3,200 images that divided into 320 thermal images for each sign language digit. The thermal images were low-resolution 32×32 pixels taken from a live thermal camera. The framework was trained using a state-of-the-art lightweight version of the deep learning model. The model attained an accuracy of 99.52%. The authors in Ref. [21] proposed an innovative end to end SLR approach. The work is done in three steps. In the first step, they detected a skin color from video sequences. Then, they used Camshift to track and select trajectories of hand movement. In the last step, the employed hidden Markov model classifier to recognize the sign. To prove their system performance, they tested it on American Sign Language Linguistic Research Project dataset. Their approach achieved an encouraging result.

Materials and Methods

Data Collection

A major challenge for Kurdish sign language recognition is the lack of a benchmark dataset, which prevents application development in this field. For this reason, we collected our dataset from non-deaf people, mostly college students, using both the iPhone 7 plus and iPhone XR phone cameras. The dataset contains seven static hand gestures RGB images for simple Kurdish words and phrases: "face to face," "how are you," "in front of," "love," "restaurant," "steadfast," and "system."

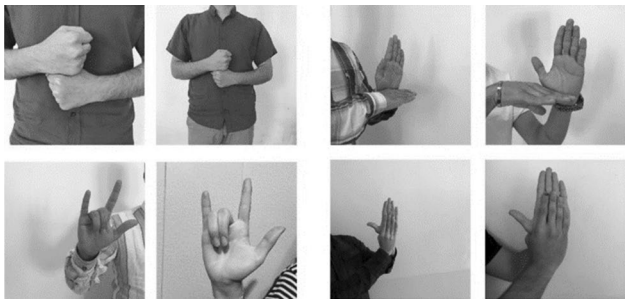

Fig. 1 The diversity of dataset images at different positions, angles, and sides

Fig. 2 Example of every word/phrase of the dataset

The images were captured with the help of 172 different individuals (117 females and 55 males) within around 3–10 shots of each person for the same sign. To ensure diversity in the dataset, the number of shots taken by every individual was captured at various positions, angles, and sides, as shown in Fig. 1. Moreover, all the images were captured under good lighting conditions.

Furthermore, the total number of images captured for each sign is 500, except for the “how are you” phrase, which can be expressed using two separate but close signs. Therefore, we opted to shoot extra images of that sign. In total, the dataset contains 3690 high-resolution images of sizes (3024×3024) and (960×960) pixels, respectively, for iPhone 7 plus and iPhone XR phone cameras.

An example of every word or phrase is shown in Fig. 2. Finally, because of the non-availability of a common form of communication sign for Kurdish deaf people, we relied on an expert and a book to determine the hand gesture and the meaning of each sign.

Data Separation

To prepare the data for the training procedure, we separated our dataset into three subgroups, which are training, testing, and a small portion for external testing. The separation process had to be done carefully because, as previously stated, each individual was captured several times for the same sign from various angles and positions. As a result, when we divided the dataset into the aforementioned groups, we had to precisely choose all the shots taken by the same person and place them in one of the sets. By doing so, we avoided the use of different shots of the same person being used for training and testing. Any image captured by the same person is included in one of the three sets.

Proposed CNN Architecture

CNN is one of the most widely used deep learning-based algorithms for classification. It comprises a vast number of hyperparameters, including kernel size, cost function, optimization function, dropout rate, activation function, batch size, epochs, and learning rate. In addition, tuning these hyperparameters affects the overall model performance as well as the classification accuracy [22]. Our proposed CNN undergoes a diverse set of experiments and hyperparameter tuning to get to the optimal hyperparameter in every section of the architecture.

The final custom-built CNN architecture has been proposed, which consists of four Conv layers each with a filter size of 5×5. After every Conv layer, there is a max-pooling layer. Following that, the output of the last CNN layer has been flattened such that it could be passed into the network's two dense layers, which contains 64 and 128 nodes, respectively, and each of the dense layers followed by a dropout layer with a drop rate of 0.5. The model architecture is shown in Fig. 3. Furthermore, ReLU is the primary activation function used by every Conv and dense layer. Then, to optimize the model weights, Adam was utilized as the main optimization function. Finally, this CNN model was trained using the collected dataset, and the model performance was compared to several popular pre-trained networks such as MobileNet and VGGNet.

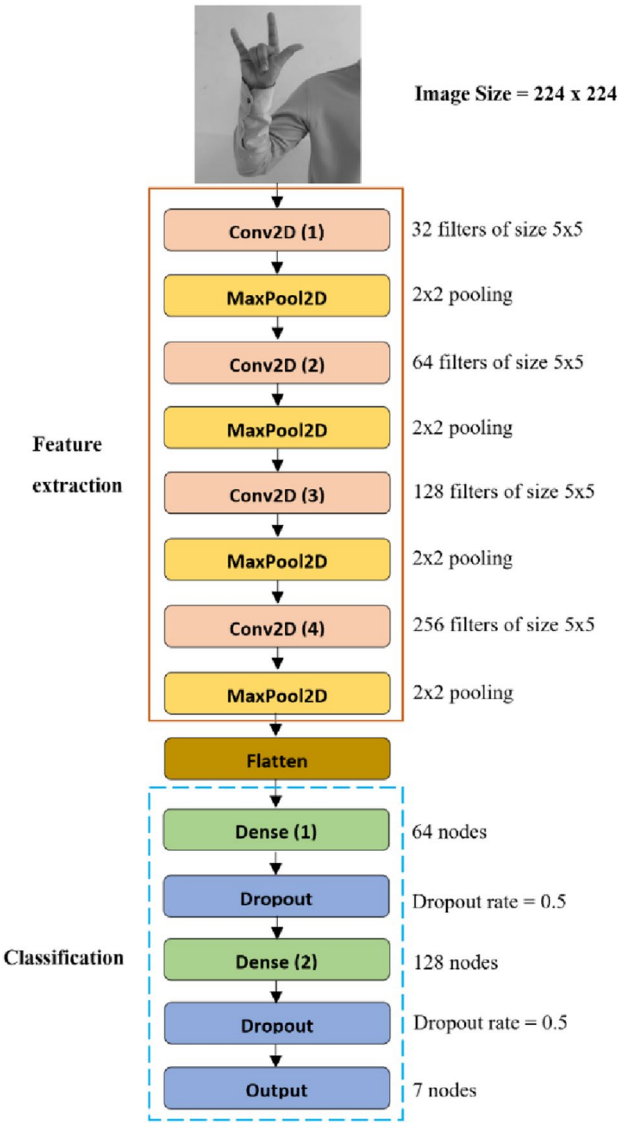

Fig. 3 The proposed model architecture with four Conv layers and a filter size of 5×5

Transfer Learning

Transfer learning (pre-trained model) refers to transferring the learning ability of a model from one task to another. For instance, if a model achieved an appealing performance on a large image dataset like ImageNet, we transfer its learning capability into sign recognition from images. In recent years, pre-trained CNN models including MobileNet, VGGNet, ResNet, Inception, and EfficientNet have gained prominence in the research community. This is because of a variety of factors. The first factor is the scarcity of large amount datasets in the field of computer vision, which is a necessity for training any CNN model.

Furthermore, the second reason is the advancement of pre-trained models through obtaining higher results in comparison to newly built from scratch models. The third factor cutting down on the time it takes to create a model from scratch and experiment with various parameters until getting to the optimal solution [23]. Finally, it is worth mentioning that pre-trained models are a promising solution for a wide range of tasks but not all of them.

MobileNet

MobileNet is a cutting-edge CNN network that focuses on building models for lightweight devices such as mobile phones. The structure of the network begins with a Conv layer that has 32 filters of size 3×3. Following that, the primary building block of the network is a special layer named residual bottleneck. The model performance has been computed using three of the most popular datasets, which are COCO, ImageNet, and VOC image segmentation. Finally, the achieved results revealed an improved performance of MobileNetV2 in comparison to MobileNetV1 as well as ShuffleNet and NasNet [24].

VGGNet

VGGNet is another state-of-the-art CNN that was proposed in 2015. The presented work mainly focused on the depth effect on the achieved accuracy in large-scale image identification tasks. VGGNet comes in two variants, namely, VGGNet16 and VGGNet19. The former contains 13 Conv layers along with three dense layers, whereas the latter contains 16 Conv layers and three dense layers. Moreover, every Conv layer has a filter size of 3×3. In addition, each dense layer has 4096 nodes. Finally, the network achieved impressive results on the ImageNet challenge and can generalize well on other datasets [25].

Experimental Setup

The proposed Kurdish Sign Language classifier was implemented using TensorFlow GPU 2.6.0 as a backend for Keras 2.6.0. The primary programming language is Python 3.8, along with various Python packages such as Scikit-learn, which is mainly used for generating classification reports. Furthermore, all experiments were run on a Dell Core i7-6700HQ CPU @ 2.60 GHz laptop with 16 GB of main memory and an Invidia GeForce GTX 960 M GPU with 4 GB of VRAM.

Table 1 Best three achieved outcomes by every utilized filter size in the proposed architecture

Filter	Epoch	Accuracy (%)	Precision (%)	Recall (%)	*F*1_score (%)
3×3	40	0.954	0.960	0.954	0.953
5×5	**75**	**0.997**	**0.997**	**0.997**	**0.997**
7×7	70	0.995	0.995	0.995	0.995

The values that are highlighted in bold represent the highest outcome out of the three that were shown

Table 2 Top three acquired outcomes by MobileNetV2

Epoch	Accuracy (%)	Precision (%)	Recall (%)	*F*1_score (%)
15	0.995	0.995	0.995	0.995
15	**0.997**	**0.997**	**0.997**	**0.997**
15	0.989	0.990	0.989	0.989

The values that have been highlighted in bold signify the result that was the best out of the three that were shown

Experiments and Results

To train the model, a total of 3235 images were fed into the network which means approximately 88% of the entire dataset. The proposed architecture was trained 25 times, each time changing at least one of the hyperparameters described in Sect. 4.3. Initially, the majority of the experiments were done by utilizing a filter size of 3×3 for every Conv layer. After that, we experimented with mixing between 3×3 and 5×5, and then, we changed all the filters to 5×5. However, each of these modifications increased the model performance, indicating that the most important parameter among those we investigated was the filter size. Following that, each trained model was tested using the test set, which contained around 11% of the data. With all these experiments, we arrived at the final model architecture that could achieve a test accuracy of 99.75%. Table 1 summarizes three of the highest achieved results throughout the training process.

To showcase the performance of our proposed model, we fed the dataset into two well-known pre-trained architectures, namely, MobileNetV2 and VGG16. Both of them have been trained using three possible ways described as follows. First, set the trainable parameter to false, i.e., do not retrain any of the feature extraction layers and utilize the saved weights of the network. Second, retrain the network architecture entirely, which takes more time to train. Third, freeze some of the Conv layers and retrain the rest of them. Additionally, we conducted several experiments

Table 3 Top three acquired outcomes by VGG16

Epoch	Accuracy	Precision	Recall	*F*1_score
10	0.995	0.995	0.995	0.995
10	**0.997**	**0.997**	**0.997**	**0.997**
10	0.972	0.973	0.972	0.972

The values that have been highlighted in bold signify the result that was the best out of the three that were shown

with the third option, increasing and decreasing the number of frozen and unfrozen layers. We experimented with both models 20 times in total, 10 times for each network. Ultimately, our experiments revealed that the outcome of the first option was very poor, so we are not mentioning its consequences, while the second and third options could achieve very high results on the test set, with accuracy of 99.75% for both networks. Tables 2 and 3 demonstrate the top three results attained by second and third network, respectively.

Finally, we stored comprehensive information on the architecture of the model and its obtained outcomes, such as loss and accuracy figures, model summary, model hyperparameter information, classification report, confusion matrix, and saving the model, after each experiment was completed.

Results and Discussion

Surprisingly, the proposed model, along with the MobileNetV2 and VGG16, could gain the exact highest result, which is 99.75% of accuracy. This indicates that each model could recognize 396 of the 397 images in the test set. The noticeable difference is the point where the classifier is confused in identifying the image. This distinction is apparent in Figs. 4, 5, and 6, which represent the confusion matrix of every model, respectively. Another noteworthy difference is that the proposed model requires more train time and epochs to reach the best outcome. Each of the MobileNetV2 and VGG16 trained for 10–15 epochs per experiment, while the proposed model trained for 40–85 epochs, as presented in Fig. 7. The figure also implies that the model loss fell significantly until it was less than 0.25, which is considered a significant improvement.

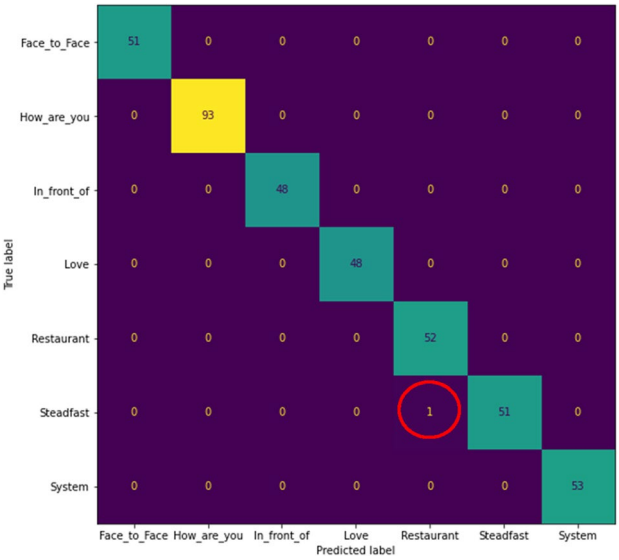

Fig. 5 Confusion matrix of the MobileNetV2 model. The location of confusion has been rounded with a red circle

In comparison to the other two pre-trained models, our proposed model performed very well and could attain the same test result as the pre-trained ones, but what makes our model better is the external test, which will be explained in the next section. Additionally, to demonstrate the model's performance, we compare the results of our final model to some of the highest achieved accuracies in related works, as shown in Table 4. The table shows that our proposed method

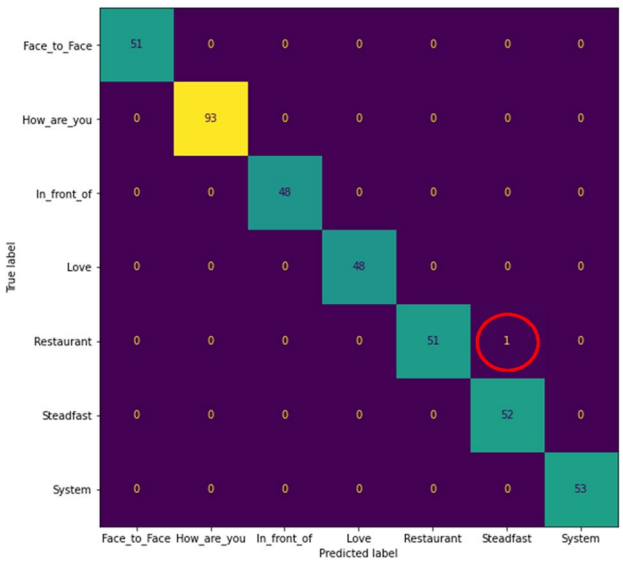

Fig. 4 Confusion matrix of the proposed model. The location of confusion has been rounded with a red circle

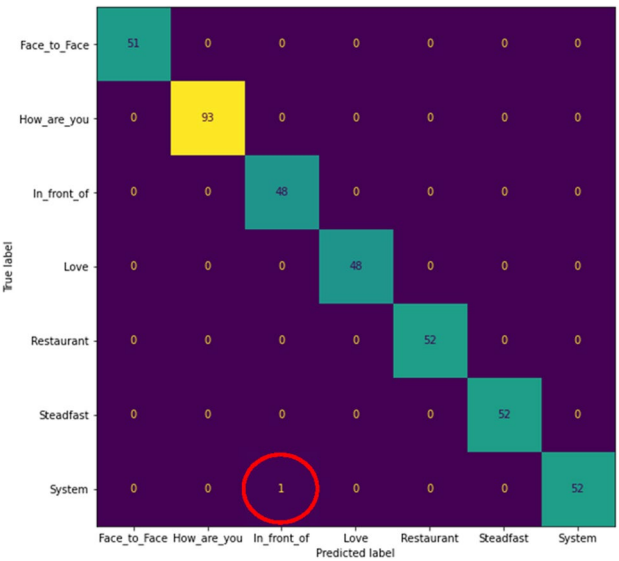

Fig. 6 Confusion matrix of the VGG16 model. The location of confusion has been rounded with a red circle

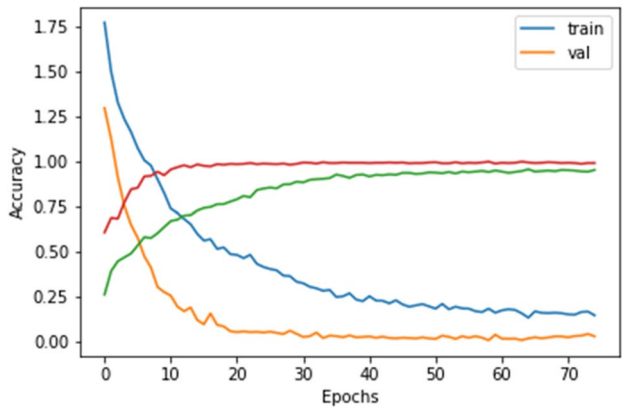

Fig. 7 Accuracy and loss per epoch of train and validation set for the proposed model

Table 4 Comparing the highest achieved accuracy of our work with the top four outcomes in the literature

#	Methods	Accuracy (%)
1	Proposed method	99.75
2	Breland et al. [20]	99.52
3	Sharma and Anand [18]	99.0
4	Bencherif et al. [11]	98.39
5	Deriche et al. [17]	92

Table 5 Number of parameters and the saved file size of our proposed method as well as VGG16

Method	Params	File size
Proposed method	**4,298,631**	**49.2 MB**
VGG16	9,472,519	128 MB

outperformed all other methods and could achieve a state-of-the-art result.

External Testing

External testing involves saving the model after training is finished, reloading the saved model, and then predicting new data that have not been seen by the model before. To accomplish this, we used an external test set of the data (around 1%) as mentioned in Sect. 4.1. We loaded the saved model using the "load_model" Keras library and then predicted the images one by one. This procedure has been carried out for all three models. The testing showed that MobileNet was somewhat weak in recognizing newer images, while our proposed method, along with VGG16, recognized nearly all the images, with VGG16 performing slightly better. What differentiates our proposed approach as better and more production-ready especially for mobile devices is the much lower number of model parameters as well as the smaller file size of the saved model when compared to the VGG16, as shown in Table 5.

Conclusion and Future Work

This paper proposed a CNN-based Kurdish sign language classifier model. The work used a dataset that was mostly collected from college students and included seven static signs. The signs are mainly interpretable through the use of hand, finger shape, and position. The model was trained employing three different CNN architectures: four-layer from scratch CNN, MobileNetV2, and VGG16. All three models obtained a comparable result, which is an accuracy of 99.75%, which is a remarkable finding in the field.

However, to achieve a novel outcome, the proposed technique undergoes a range of hyperparameter tuning, including filter size, dropout rate, and optimizer functions. The final model architecture consists of 4 Conv layers with a filter size of 5×5 and only two dense layers. The consequences of the architecture could compete with the other two mentioned pre-trained networks. Furthermore, the pre-trained networks could gain this significant outcome through fine-tuning. What makes more sense in this research is that all three models attain the same accuracy and only made one incorrect recognition out of all the images in the test set; the difference is in the location of the wrong identification.

For future work, we are in the process of capturing additional signs including dynamic signs. After that, we will consider building a model for real-time sign detection. Finally, we will attempt to develop a mobile application based on the model for recognizing the signs.

Acknowledgements The author of this work wishes to express his gratitude and appreciation to the students of the College of Basic Education\University of Raparin for helping us capture the images and collect the data. The author also thanks those who assisted us in the process of investigating and identifying signs in Kurdish sign language.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

References

1. Elatawy SM, Hawa DM, Ewees AA, Saad AM. Recognition system for alphabet Arabic sign language using neutrosophic and fuzzy c-means. Educ Inf Technol. 2020;25(6):5601–16. https://doi.org/10.1007/s10639-020-10184-6.
2. Rastgoo R, Kiani K, Escalera S. Sign language recognition: a deep survey. Expert Syst Appl. 2021;164(July 20200):113794. https://doi.org/10.1016/j.eswa.2020.113794.
3. Aloysius N, Geetha M. Understanding vision-based continuous sign language recognition. Multimed Tools Appl. 2020;79(31–32):22177–209. https://doi.org/10.1007/s11042-020-08961-z.
4. Nurena-Jara R, Ramos-Carrion C, Shiguihara-Juarez R. Data collection of 3D spatial features of gestures from static Peruvian sign language alphabet for sign language recognition. In: Proceedings of the 2020 IEEE Engineering International Research Conference, EIRCON 2020, 2020; pp. 3–6. https://doi.org/10.1109/EIRCON51178.2020.9254019.
5. Hasan MM, Srizon AY, Sayeed A, Hasan MAM. Classification of Sign language characters by applying a deep convolutional neural network. In: ICCIT 2020 - 23rd International Conference on Computer and Information Technology, Proceedings, no. November, 2020; pp. 28–29. doi: https://doi.org/10.1109/ICCIT51783.2020.9392703.
6. Hisham B, Hamouda A. Arabic sign language recognition using Ada-Boosting based on a leap motion controller. Int J Inf Technol (Singapore). 2021;13(3):1221–34. https://doi.org/10.1007/s41870-020-00518-5.
7. Wadhawan A, Kumar P. Deep learning-based sign language recognition system for static signs. Neural Comput Appl. 2020;32(12):7957–68. https://doi.org/10.1007/s00521-019-04691-y.
8. Abbas Muhammad Zakariya RJ. Arabic sign language recognition system on smartphone. 2019. https://doi.org/10.1109/ICCCNT45670.2019.8944518.
9. Jepsen JB, De Clerck G, Lutalo-Kiingi S, McGregor WB. Sign languages of the world: a comparative handbook. Ishara Press; 2015. https://doi.org/10.1515/9781614518174.
10. Halvardsson G, Peterson J, Soto-Valero C, Baudry B. Interpretation of Swedish sign language using convolutional neural networks and transfer learning. SN Comput Sci. 2021;2(3):1–15. https://doi.org/10.1007/s42979-021-00612-w.
11. Bencherif MA, et al. Arabic sign language recognition system using 2D hands and body skeleton data. IEEE Access. 2021;9:59612–27. https://doi.org/10.1109/ACCESS.2021.3069714.
12. Venugopalan A, Reghunadhan R. Applying deep neural networks for the automatic recognition of sign language words: a communication aid to deaf agriculturists. Expert Syst Appl. 2021;185(September 2020):1601. https://doi.org/10.1016/j.eswa.2021.115601.
13. Krejsa J, Vechet S. Czech sign language single hand alphabet letters classification. In: Proceedings of the 2020 19th International Conference on Mechatronics—Mechatronika, ME 2020, 2020, https://doi.org/10.1109/ME49197.2020.9286667.
14. Teja Mangamuri LS, Jain L, Sharmay A. Two hand Indian sign language dataset for benchmarking classification models of machine learning. In: IEEE International Conference on issues and challenges in intelligent computing techniques, ICICT 2019, 2019, https://doi.org/10.1109/ICICT46931.2019.8977713.
15. Joy J, Balakrishnan K, Madhavankutty S. A novel web based dictionary framework for Indian sign language. SN Comput Sci. 2021;2(3):1–7. https://doi.org/10.1007/s42979-021-00533-8.
16. Gupta R, Rajan S. comparative analysis of convolution neural network models for continuous Indian sign language classification. Proc Comput Sci. 2020;171(2019):1542–50. https://doi.org/10.1016/j.procs.2020.04.165.
17. Deriche M, Aliyu S, Mohandes M. An intelligent Arabic sign language recognition system using a pair of LMCs with GMM based classification. IEEE Sens J. 2019;19(18):1–12. https://doi.org/10.1109/JSEN.2019.2917525.
18. Sharma P, Anand RS. A comprehensive evaluation of deep models and optimizers for Indian sign language recognition. Graph Vis Comput. 2021. https://doi.org/10.1016/j.gvc.2021.200032.
19. Zhang S, Zhang Q. Sign language recognition based on global-local attention. J Vis Commun Image Represent. 2021;80(December 2019):103280. https://doi.org/10.1016/j.jvcir.2021.103280.
20. Breland DS, Skriubakken SB, Dayal A, Jha A, Yalavarthy PK, Cenkeramaddi LR. Deep learning-based sign language digits recognition from thermal images with edge computing system. IEEE Sens J. 2021;21(9):10445–53. https://doi.org/10.1109/JSEN.2021.3061608.
21. Roy PP, Kumar P, Kim B-G. An efficient sign language recognition (SLR) system using Camshift tracker and Hidden Markov Model (HMM). SN Comput Sci. 2021. https://doi.org/10.1007/s42979-021-00485-z.
22. Lee WY, Park SM, Sim KB. Optimal hyperparameter tuning of convolutional neural networks based on the parameter-setting-free harmony search algorithm. Optik. 2018;172(May):359–67. https://doi.org/10.1016/j.ijleo.2018.07.044.
23. Zhu W, Braun B, Chiang LH, Romagnoli JA. Investigation of transfer learning for image classification and impact on training sample size. Chemomet Intell Lab Syst. 2021;211(January):104269. https://doi.org/10.1016/j.chemolab.2021.104269.
24. Sandler M, Howard A, Zhu M, Zhmoginov A, Chen LC. MobileNetV2: inverted residuals and linear bottlenecks. In: Proceedings of the IEEE Computer Society Conference on computer vision and pattern recognition, 2018; pp. 4510–4520, doi: https://doi.org/10.1109/CVPR.2018.00474.
25. Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. In: 3rd International Conference on Learning Representations, ICLR 2015—Conference Track Proceedings, 2015; pp. 1–14. https://doi.org/10.48550/arXiv.1409.1556