International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING

ISSN:2147-6799 www.ijisae.org Original Research Paper

Kurdish Sign Language Recognition Based on Transfer Learning

Baraa Wasfi Salim [1]*, Subhi R. M. Zeebaree [2]

Submitted: 14/02/2023 **Revised:** 17/04/2023 **Accepted:** 10/05/2023

Abstract: Sign language is used to communicate with deaf and dumb people; it is difficult for ordinary people to communicate with them. Hence, computer vision and automatic identification can reduce the difficulties of reaching them. Deep learning algorithms were used to distinguish sign language in different languages and styles. Convolutional Neural Networks (CNNs) are used in computer vision, particularly pre-trained algorithms. This research proposes using transfer and machine learning to distinguish Kurdish Sign Language (KSL). A KSL dataset was created to characterize the Kurdish language at the level of numbers and letters, using pre-trained algorithms for feature extraction and machine learning algorithms for classification. The proposed method was tested on two data sets; KSL and American Sign Language (ASL). The algorithms (VGG19 and RESNET101) are implemented in the feature extraction phase with pre-trained weights. The algorithms: Support Vector Machine (SVM), Decision Tree (DT), and Random Forest (RF), is dependent on the classification stage, and the CNN is designed for the KSL model. The efficiency of the proposed models is evaluated using (accuracy, recall, precision, and F1 score) metrics. The proposed model's outcomes illustrated that VGG19 is better than (RESNET101 and proposed CNN) algorithms in terms of feature extraction, and the random forest is the best classifier which achieved an accuracy rate of 95% at the numbers level and 97% at the level of the letter for KSL and ASL.

Keywords: *Kurdish Sign Language; CNN, Pre-trained; Transfer Learning; VGG19 and RESNET101.*

1. Introduction

Sign language is one of the languages that use gestures and expressions to convey information and is used to communicate to and from deaf and mute people in many situations on the personal, academic, social, and television media [1][2]. There are about 70 million in the world, according to the statistics of the World Federation of the Deaf, all over the world, and they use more than 300 sign languages [3]. It is difficult for ordinary people to recognize or speak sign language with the deaf and dumb, so finding a way for him to translate sign language is one of the main issues of the deaf and dumb community and those who deal with them [4][5]. The field of computer vision has improved many recognition systems, such as distinguishing people, faces, mathematical movements, etc., and that deep learning is one of the most important pillars that have been added to this field[6][7][8]. Many researchers have used deep learning to distinguish sign language by relying on deep learning and harnessing algorithms in different ways on many languages of sign language [9][10][11][12][13]. A group of researchers used Convolutional Neural Networks (CNN) to distinguish American Sign Language at the letters, numbers, and words level, while others used pre-trained deep learning algorithms to distinguish American Sign Language.

1 ITM Dept., Technical College of Administration, Duhok Polytechnic University, Duhok, Iraq
baraa.salim@dpu.edu.krd
2 Energy Eng. Dept., Technical College of Engineering, Duhok Polytechnic University, Duhok, Iraq
subhi.rafeeq@dpu.edu.krd
** Correspondence: baraa.salim@dpu.edu.krd; Tel.: 00964-751-4947536*

In contrast, deep learning was applied to other languages [14][15][6]. Image Convolutional Neural Network algorithms work to distinguish images through two phases, the feature extraction phase, and the classification phase [17]. Usually, pre-trained neural network algorithms use fixed weights extracted previously in the feature extraction stage, while the weights of the fully linked network are found during training [18][19]. The concept of transfer learning uses pre-trained convolutional neural networks only in the feature extraction stage.

The primary aim of this article is to use a computer model to assist deaf people in learning the Kurdish sign language's alphabet, numerals, and words. One of the most challenging issues in image processing and computer vision is real-time hand gesture detection. The objective is to enable quick and precise hand gesture detection under various illumination situations. The researchers concluded that CNN could distinguish sign language efficiently, and based on the previous, it was proposed in this study to use transfer learning to distinguish Kurdish sign language in a new way from the ways of hybridizing pre-trained deep learning algorithms with classification algorithms. Moreover, the creation of a special data set in the Kurdish language for numbers and letters, in addition to its application to American Sign Language.

The main contributions of this research include: (1) Creating a data set of numbers and letters for Kurdish sign language. There is no data set for Kurdish sign language available to researchers, and the data set was created in natural and not laboratory conditions, reflecting the models' performance realistically. (2) Developing a deep learning model to detect and interpret sign language into text form using Kurdish Sign Language and American Sign Language. (3) Proposed use of the concept of transfer learning to extract the features of sign language and to use the idea of collective learning in the classification stage to create an accurate and applicable system on various data sets.

2. Literature Review

To assist those who are deaf or hard of hearing, several articles on the identification of sign languages have lately been published. A three-dimensional residual network and dilated convolutional network are the foundation of Pu et al.'s methodology for recognizing sign language [20]. To recognize 24 ASL alphabets, Oyedotun and Khashman [21] used two DL techniques: CNN and stacked denoising autoencoder (SDAE) networks. The gesture recognition database of Thomas Moeslund was used to acquire the samples. However, Aly, W. et al. [22] separate hands using depth information from RGB pictures from the ASL fingerspelling dataset. The principle component analysis network (PCANet), a specific kind of CNN, is fed these depth pictures to extract features rather than classify them. There are two convolutional layers in it. L1 filters are used in the first one to learn low-level features, while L2 filters are used in the second one to learn high-level features. The goal of the article by Karlo et al. [23] was to create a Convolutional Neural Network-based model that is trained on static sign language for American Sign Language alphabets (A to Z), numerals (1 to 10) and some basic phrases with an average testing accuracy of 93.67%. Convolutional neural networks were suggested as the design and architecture for an American Sign Language (ASL) recognition system in [24]. (CNN). For static gesture recognition in this study, a pre-trained VGG-16 architecture is used, and for dynamic gesture identification, a deep learning-based complex architecture was used to learn spatiotemporal characteristics. This architecture extracts 2D spatiotemporal characteristics and includes a bidirectional convolutional Long Short Term Memory network (ConvLSTM) and 3D convolutional neural network (3DCNN). In [25], they suggest using a CNN (Convolutional Neural Network) model that trains and classifies alphabet letters a–z and, in most cases, translates into the corresponding text. Many neural network techniques are used on the datasets, including RNN (recurrent Neural Network). In order to streamline network administration and save communication costs, Chen et al. [26]. Presented a three-tier network design with a short-term traffic forecast model created using LSTM. Only 10 American Sign language gesture alphabets were used in Hurroo et al.'s[27] proposed convolutional neural network (CNN), which achieved a 90% accuracy by combining the HSV color algorithm and many computer vision techniques.

3. Background Theory

This study used a set of deep learning algorithms to extract features in addition to machine learning algorithms for classification. The following is the background theory of the algorithms used.

3.1. Deep Learning Algorithms

This section will address three significant deep-learning algorithms that depend on this research.

3.1.1. CNN Model

The architecture of convolutional neural networks was built to model the Kurdish sign language [29]. It begins with ordinary 2D convolutions; it moves on to batch standardization and the activation function ReLU before concluding [30]. A kernel regularized is utilized to impose penalties on the layer parameters during the optimization process. Batch normalization is a technique that normalizes the activation of the preceding layer at the beginning of each batch [31][32].

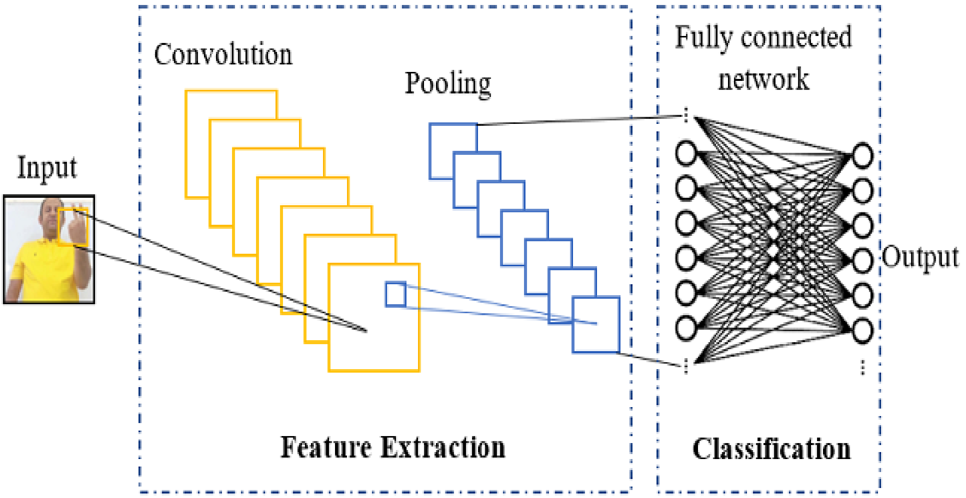

Fig. 1. feature extraction phase and the classification phase in CNN.

Depth-wise separable convolutions significantly minimize the number of parameters that must be specified [34]. Depth-wise convolutions are combined with point-wise convolutions to form these transformations. The Global Average Pooling architecture is used in this design. Calculating the average of all the components in each feature map turns each feature map into a scalar value. The final convolution layer has the same feature mappings as classes in the dataset [29].

3.2. Transfer learning

Transfer learning is a machine learning approach that uncovers hidden patterns and important information in one particular domain before applying and transferring this knowledge to an unrelated yet companion area [19][35]. Transfer learning is the process of using what the model has discovered for one job further to generalize it to a different, related issue. In this case, training one model for a particular job might be the foundation for training another for a similar task. Unlike building a model from scratch, transfer learning uses pre-trained models requiring less computing work to fine-tune [36]. Because pre-trained models have previously been trained on a very big dataset for a long period and with many optimization procedures, adopting transfer learning requires less input [37]. One of the various transfer learning methods is adding additional layers after removing the final classification layers, training the pre-trained model on the new dataset, and making necessary adjustments [18][35].

3.2.1. RESNET101 Model

A residual network (ResNet) is composed of a batch of regularized residual blocks which contain an architecture known as a "shortcut connection." With the help of the "shortcut connection" and batch normalization during the model training process, data transmission can skip some intermediate layers, which helps accelerate convergence and prevents overfitting [38][39]. The residual network mainly consists of different residual blocks called building blocks or bottlenecks [40]. In the present study, ResNet-101 was selected for the CNN architecture to extract image features to balance the computation cost and detection accuracy.

3.2.2. VGG19 MODEL

The VGG19 model (pre-trained) was modified to match the research data set in this work as a feature extractor. This model was created using a VGG network with 19 layers [41]. The VGG19 model uses nineteen convolution layers and three completely linked layers. The output of the convolution layers was obtained using the ReLU activation function, and the convolution section was separated into five consecutive max-pooling layers [42][43]. The first and second sub-regions were developed using two convolution layers with depths of 64 and 128, respectively.

Furthermore, the remaining three sub-regions were constructed using four sequential convolution layers with depths of 256, 512, and 512, respectively. The learnable parameter was then reduced using pooling layers. The suggested VGG19 model's final layer assisted in obtaining the feature vector, whereas the two hidden layers put before the feature set layer include 1024 and 512 neurons, respectively [41].

3.3. Machine learning classifier

3.3.1. Decision tree

One of the most popular and understandable machine learning algorithms is the decision tree method, often known as the Classification and Regression Tree (CART). As its name implies, a decision tree of choices divides the data space into smaller subspaces, each of which is assigned a label or a probability [44][45]. The algorithm analyzes all potential splits along all axes as the tree is built up during training to ensure that each split is done as optimally as feasible. The impurity of the resultant two partitions may be measured using a variety of metrics, including entropy, information gain, and Gini, and the optimum split point is determined to have the lowest impurity among them [44][46].

3.3.2. Random Forest

Due to their simplicity of usage and interpretation, decision tree models in Random Forest, an ensemble, are very well-liked for application in various machine-learning situations [47]. Each decision tree operates poorly when used alone because it is vulnerable to overfitting. As a bagging classifier, Random Forest employs two stochastic decision levels in its learning process [48]. Each decision tree in the ensemble selects a subset of samples and features for training [49].

3.3.3. SVM

SVM is a classifier that converts characteristics from non-linear input space to higher-dimension feature space [50]. This modification is done to simplify a higher dimensional, originally more challenging categorization task. In order to do this mapping, kernel functions are used [51][52].

4. Proposed System Implementation

From the previously trained models mentioned above, an attempt has been made to adopt two models, which are trained in parallel deep feature extractors. They are mixed and ready for the classification step after that. A multi-model network's design, which consists of two branches, is seen in Figure 2. A CNN model is used for each branch. The scenario depicted in Figure 2 uses the DenseNet101 model and the VGG19 model. Our dataset's preprocessed input color pictures, which serve as the input image for two multi-model branches in this multi-model, are 100x100 pixels in size. On their final feature extractor layers, RenseNet101 and VGG19 each create a 3x3x1024 feature map from the input picture, whereas VGG19 creates a 3x3x512 feature map. After the feature extraction phase, three classifiers were used (Support Vector Machine, decision tree, and random forest for classification).

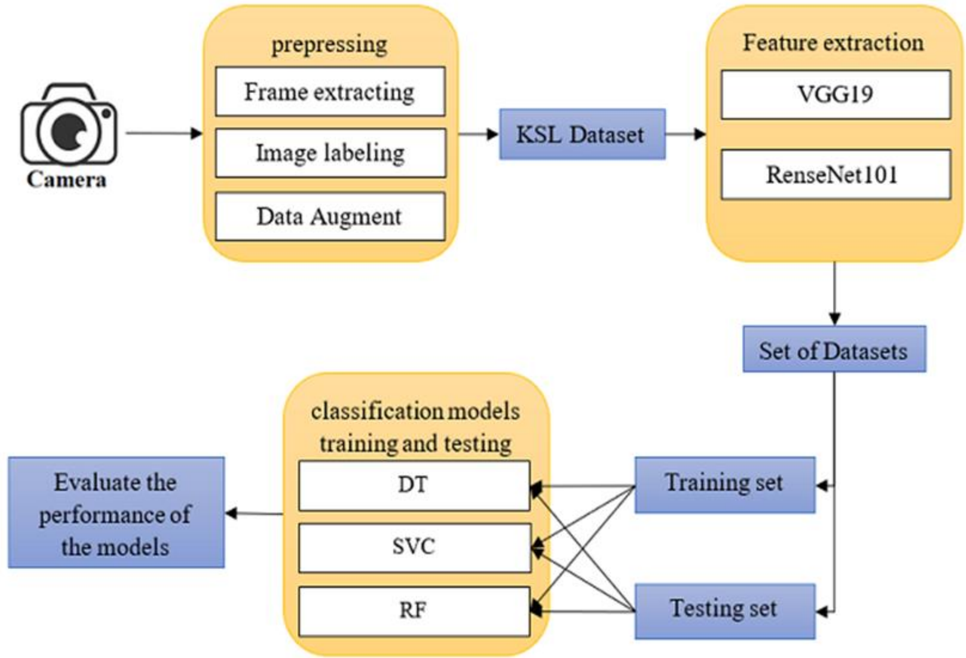

Fig. 2. Dataflow for KSL recognition using different Feature Extraction and Classification Algorithms.

After extracting the features, the data sets are divided into training and test sets. The training set is used to train the selected machine learning models, while the test set is used to evaluate and compare the performance of the trained models.

4.1. Kurdish Sign Language Dataset Creation

Data is the most critical part of machine learning and data analytics. Artificial Intelligence (AI) systems constantly learn, infer information, and make decisions to help humanity. Data mining and analytics power AI systems. An accurate and large data selection can significantly improve the generalization of machine learning performance for information extraction, prediction, and pattern recognition. Data analysis allows for knowledge discovery and decision support.

a (1) b (7) c (L) d (M)

Fig. 3. Sample of KSL letters and number signs.

There are online datasets repositories such as UCI Machine Learning Repository, Github, Google Public Datasets, etc. However, the Kurdish sign language KSL dataset is not available for study. A KSL dataset was generated from students and professors (Hiwa Institute for the Deaf and Dumb in Duhok). Volunteers were requested from different age groups; males and females each recorded a video clip ranging from 5 to 10 seconds for each word. Photographs of the still letters and numbers were taken, with 300 photographs for each letter and number. Pictures and videos were taken from various perspectives, positions, backgrounds, camera distances, and lighting conditions to increase the variety of the data set. Video frames were extracted at 30 frames per second from videos. Images are preprocessed to ensure that the size and color modes are consistent. The Kurdish language consists of 33 letters. Figure 3 gives an example of a data set.

4.2. Data Preprocessing

Resizing and normalizing the picture were preprocessing steps for sign images. Next, the picture is scaled down to 100x100. In order to balance execution speed and accuracy, this size was selected.

4.3. Data Augmentation

Several random changes will be applied to the sparse training samples to "augment" them and ensure that the model never sees the same image twice. This keeps the model from overfitting and improves generalization the Keras. Preprocessing image Image DataGenerator class in Keras used to do this. They are using the flow (data, labels) or. Flow from directory methods may construct generators of augmented picture batches (and their labels) and set random transformations and normalization procedures to be applied to image data during training (directory). These generators may then fit, evaluate, and predict generator Keras model methods, which take data generators as inputs. The number of images entered was 300 for each number and letter from the KSL data set, and the number of images generated was 1000 for each category. This study used the American Sign Language (ASL) data set to determine the proposed method's efficiency and compare it with the Kurdish sign language [28].

Table 1. This is a table. Tables should be placed in the main text near to the first time they are cited.

parameters	value	description
Rotation_range	10	range of degrees for random rotations
width_shift_range	0.1	the image shifted in any of the left or right directions.
Shear_range	0.15	Used to change angle degrees of image
zoom_range	0.1	Used for random zoom range

5. Proposed System Implementation

Accuracy, Precision, Recall, and F1-score—often employed for this type of task—were utilized to evaluate the performance of our approach. These are their definitions:

$$Acuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (1)$$

$$Precision = \frac{TP}{TP+FP} \quad (2)$$

$$recall = \frac{TP}{TP+FN} \quad (3)$$

$$F1 = 2 * \frac{Precision*recall}{precision+recall} \quad (4)$$

False Negative (FN) results occur when the model incorrectly predicts the negative class. False Positive (FP) results occur when the model incorrectly predicts the positive class. A true Negative (TN) is an outcome when the model correctly predicts the negative class. A true positive

(TP) is an outcome when the model correctly predicts the positive class.

5.1. Numbers Recognition

To distinguish the sign language of the two data sets (KSL and ASL) at the level of numbers consisting of 10 numbers from 0 to 9. 1000 images were used for each number with a total of 10,000 RGB images. The data set was divided into an 80:20 ratio, with 8000 images for training and 2,000 images for testing. After splitting the data set in the first stage, the CNN was trained and tested, and its performance was measured. In the second stage, the concept of pre-training and testing of three algorithms (vgg19 and ResNet101) was used to extract features, then classify the numbers using machine learning algorithms (SVM, DT, RF), and then individually measure the performance of each.

Table 2. Performance metrics results of depended algorithms for both (ASL and KSL).

	ASL				KSL			
methods	precision	recall	f-score	accuracy	precision	recall	f-score	accuracy
CNN	0.88	0.88	0.87	0.88	0.88	0.88	0.88	0.88
VGG19 _SVM	0.89	0.70	0.75	0.70	0.97	0.97	0.97	0.97
VGG19 _DT	0.69	0.69	0.69	0.69	0.78	0.78	0.78	0.78
VGG19_RF	0.93	0.92	0.92	0.92	0.97	0.97	0.97	0.97
ResNet101_SVC	0.89	0.65	0.71	0.65	0.94	0.93	0.93	0.93
ResNet101_DT	0.67	0.67	0.67	0.67	0.65	0.65	0.65	0.65
ResNet101_RF	0.89	0.89	0.89	0.89	0.88	0.88	0.88	0.89
CNN	0.88	0.88	0.87	0.88	0.88	0.88	0.88	0.88

By comparing the accuracy of the methods as shown in Table 2, it is noted that the highest accuracy obtained is when applying VGG19 and using SVM for the KSL group, followed by VGG19 with random forests using the KSL data group and checking the effectiveness of the methods on the two data sets.

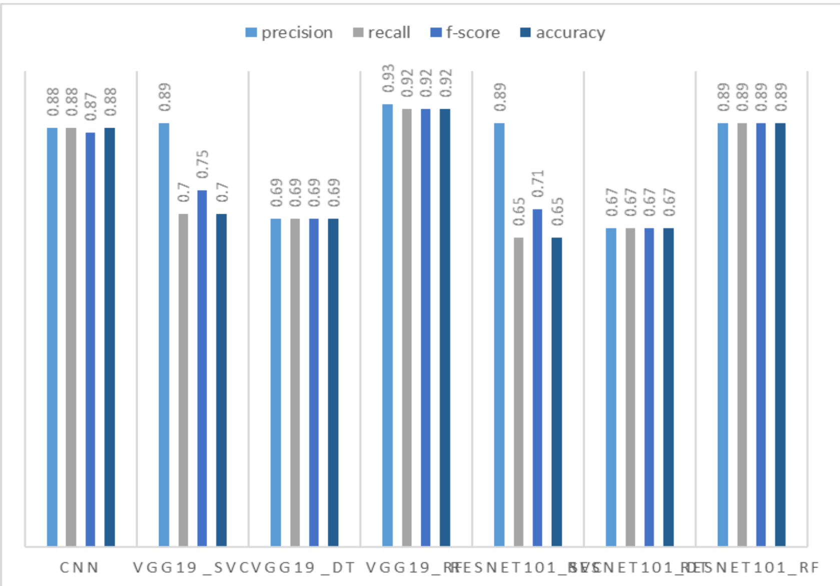

Fig. 4. Comparing the performance of models implemented on ASL at the numbers level.

Figure 4 displays the performance of models implemented ASL dataset on the numbers level. Random forests as a classifier with the VGG19 feature extraction algorithm achieved the best performance in various measures, as the accuracy was 0.93 and 0.92 in the rest of the measures (precision, recall, and f-score).

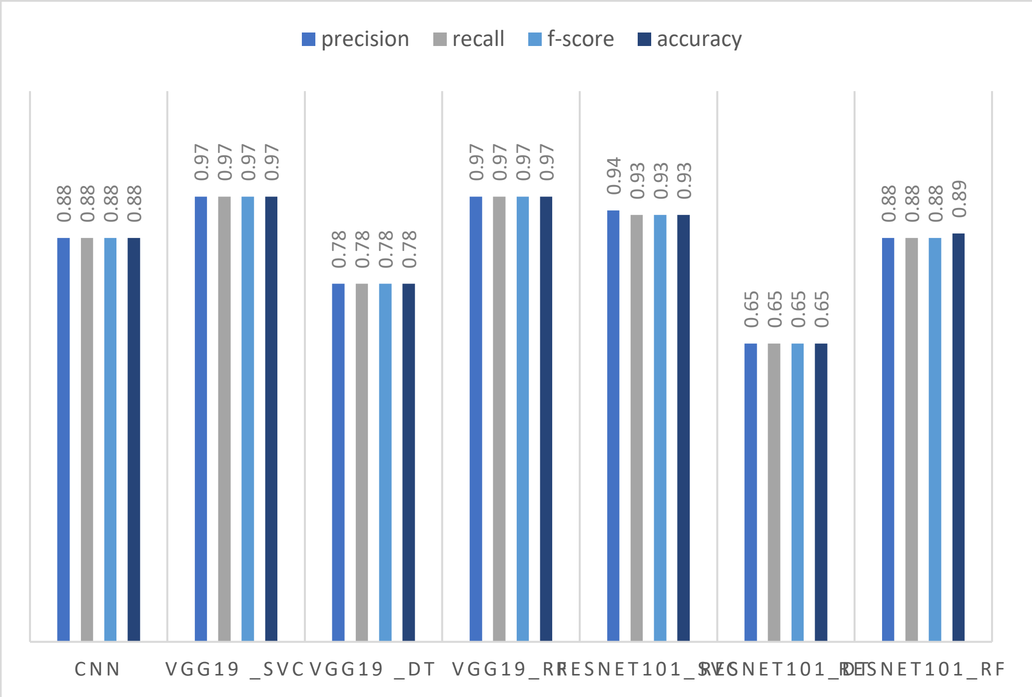

Fig. 5. Comparing the performance of models implemented on KSL at the numbers level.

Figure 5 displays the performance of models implemented KSL dataset on the numbers level. Random forests and SVC as a classifier with the VGG19 feature extraction algorithm achieved the best performance, which was 0.97 in various measures. It is also noted that the results were similar for the different measures for each model, which indicates the balance of the data in terms of classification.

5.2. Letters Recognition

To distinguish the sign language of the two data sets (KSL and ASL) at the character level. KSL consists of 33 letters, while ASL consists of 26 letters. A thousand images were used for each character, with 33,000 RGB images for Kurdish characters and 26,000 for American characters. The data set was divided in a ratio of 80:20, respectively, with a total of 26400 images for training, 6600 images for testing for the Kurdish language, 20800 images for training, and 5200 images for testing for the American language.

Table 3. Performance metrics results of dependent algorithms for both (ASL and KSL).

	ASL				KSL			
methods	precision	recall	f-score	accuracy	precision	recall	f-score	accuracy
CNN	0.91	0.99	0.94	0.94	0.98	0.77	0.86	0.88
VGG19 _SVC	0.94	0.73	0.79	0.72	0.97	0.96	0.97	0.93
VGG19 _DT	0.80	0.79	0.79	0.79	0.72	0.69	0.70	0.72
VGG19_RF	0.99	0.99	0.99	0.99	0.99	0.96	0.98	0.94
ResNet101_SVC	0.90	0.76	0.80	0.76	0.71	0.66	0.67	0.66
ResNet101_DT	0.70	0.70	0.70	0.70	0.46	0.46	0.46	0.45

ResNet101_RF	0.98	0.98	0.98	0.98	0.69	0.68	0.68	0.68

By comparing the accuracy of the methods as shown in table 3, it is noted that the highest accuracy obtained is when applying VGG19 and using SVC for the KSL group, followed by VGG19 with random forests using the KSL data group and checking the effectiveness of the methods on the two data sets.

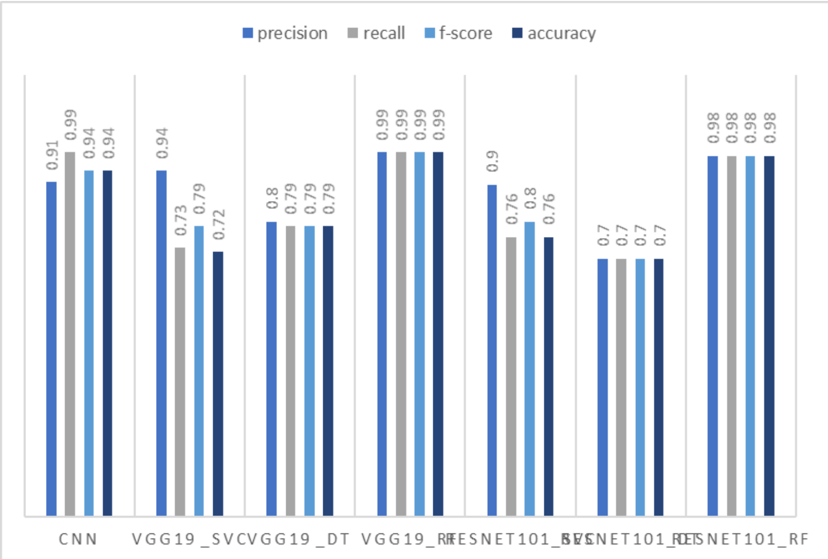

Figure 6. Comparing the performance of models implemented on ASL at the letter level.

Figure 6 displays the performance of models implemented ASL dataset on the level of the letters. Random forests as a classifier with the VGG19 feature extraction algorithm achieved the best performance, which was 0.99, Followed by Random forests with the ResNet101, which was 0.98 in various measures.

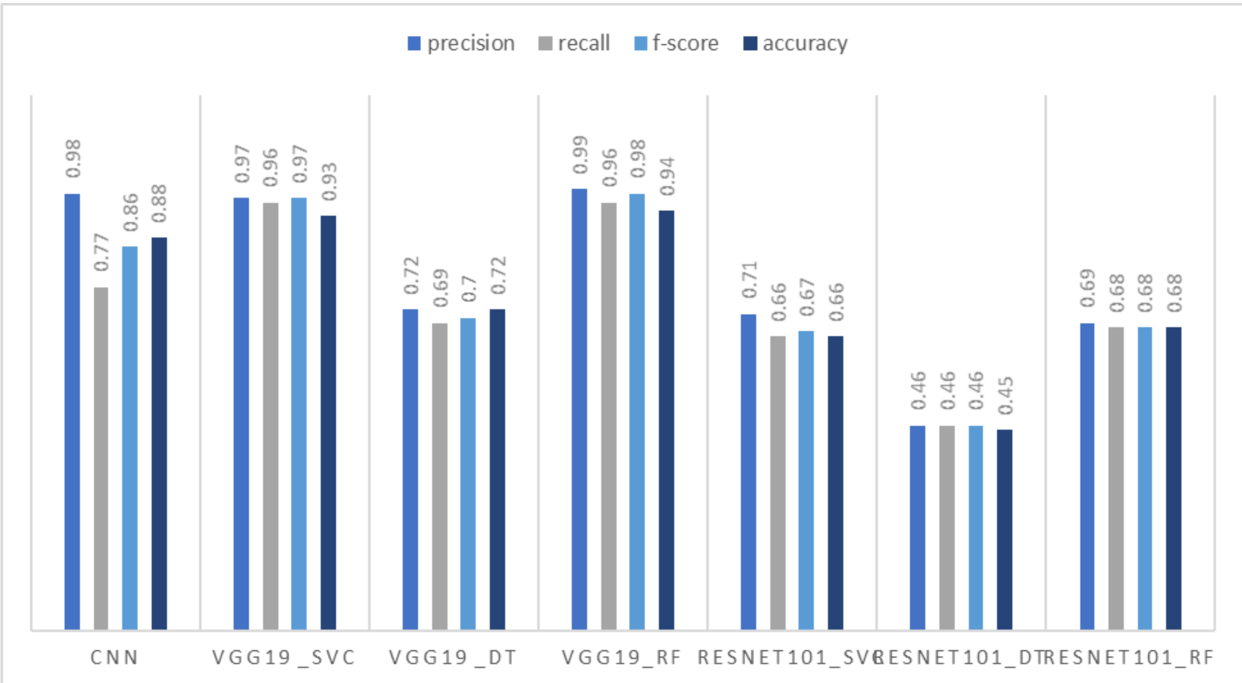

Fig. 7. Comparing the performance of models implemented on KSL at the letter level.

Figure 7 displays the performance of models implemented KSL dataset on the level of the letters. Random forests as a classifier with the VGG19 feature extraction algorithm achieved the best performance, which was 0.94 accuracy (0.99, 0.96, and 0.98) in the rest of the measures (precision, recall, and f-score), respectively.

6. Conclusion

From the proposed system and the obtained results, it can be concluded that there is no generic gray dataset nor colour dataset to identify the Kurdish sign language. We prepared more than 43,000 images (including 10 numbers and 33 letters) to generate the colour dataset. The created dataset has been built via the participation of five volunteers from different genders and ages. This research has depended on three famous CNN models (CNN, VGG19, and Resnet101) for the feature extraction stage. Adding three classifiers (RF, TD, and SVMc) for the classification stage. Based on the total number of the inaccurate-detection of the dependent images during the training and testing stages, the VGG19 CNN model is the best for feature extraction, while the random forest model is the best classifier for both Kurdish and American sign languages. Additionally, pre-trained convolutional neural networks perform better at extracting features from neural networks of which the classifier is a component in either ASL or KSL.

References

[1] Mertzani, Maria. (2022). SIGN LANGUAGE LITERACY IN THE SIGN LANGUAGE CURRICULUM. Momento - Diálogos em Educação. 31. 449-474. 10.14295/momento.v31i02.14504.

[2] Patil, Prof & Bhagwat, Ruchir & Padale, Pratham & Shah, Yash & Surwade, Hrutik. (2022). Sign Language Recognition System. International Journal for Research in Applied Science and Engineering Technology. 10. 1772-1776. 10.22214/ijraset.2022.42626.

[3] Sultan, A. (2022, March 8). International Day of Sign Languages. National Today. Retrieved November 15, 2022, from https://nationaltoday.com/international-day-of-sign-languages/

[4] Alys Young, Jemina Napier, Rosemary Oram. (2019) The translated deaf self, ontological (in)security and deaf culture. The Translator 25:4, pages 349-368.

[5] Lin, Hao. (2021). Early Development of Chinese Sign Language in Shanghai Schools for the Deaf. Frontiers in Psychology. 12. 702620. 10.3389/fpsyg.2021.702620.

[6] Aggarwal, Karan & Mijwil, Maad & Garg, Sonia & Al-Mistarehi, Abdel-Hameed & Alomari, Safwan & Gök, Murat & Zein Alaabdin, Anas & Abdul Rahman, Safaa. (2022). Has the Future Started? The Current Growth of Artificial Intelligence, Machine Learning, and Deep Learning. Iraqi Journal for Computer Science and Mathematics. 3. 115-123. 10.52866/ijcsm.2022.01.01.013.

[7] Li, Jiangong & Green-Miller, Angela & Hu, Xiaodan & Lucic, Ana & Mohan, M.R. & Dilger, Ryan & Condotta, Isabella & Aldridge, Brian & Hart, John & Ahuja, Narendra. (2022). Barriers to computer vision applications in pig production facilities. Computers and Electronics in Agriculture. 200. 107227. 10.1016/j.compag.2022.107227.

[8] Gao, Yi. (2022). Research on the Application of Artificial Intelligence Technology in the Development of Computer Vision. Highlights in Science, Engineering and Technology. 9. 80-84. 10.54097/hset.v9i.1720.

[9] Cui, Runpeng & Liu, Hu & Zhang, Changshui. (2019). A Deep Neural Framework for Continuous Sign Language Recognition by Iterative Training. IEEE Transactions on Multimedia. PP. 1-1. 10.1109/TMM.2018.2889563.

[10] Stoll, S., Camgoz, N.C., Hadfield, S. et al. Text2Sign: Towards Sign Language Production Using Neural Machine Translation and Generative Adversarial Networks. Int J Comput Vis 128, 891–908 (2020). https://doi.org/10.1007/s11263-019-01281-2

[11] BANKUR, Fatih & KAYA, Mustafa. (2022). Deep Learning Based Recognition of Turkish Sign Language Letters with Unique Data Set. Turkish Journal of Science and Technology. 10.55525/tjst.1073116.

[12] Bandyopadhyay, Shaon. (2020). A Study on Indian Sign Language Recognition using Deep Learning Approach. 10.13140/RG.2.2.25282.20168.

[13] Reshna, S. & Sajeena, A. & Jayaraju, Madhavan. (2020). Recognition of static hand gestures of Indian sign language using CNN. AIP Conference Proceedings. 2222. 030012. 10.1063/5.0004485.

[14] Alaria, Satish & Raj, Ashish & Sharma, Vivek & Kumar, Vijay. (2022). Simulation and Analysis of Hand Gesture Recognition for Indian Sign Language using CNN. International Journal on Recent and Innovation Trends in Computing and Communication. 10. 10-14. 10.17762/ijritcc.v10i4.5556.

[15] Kumar, Hemant & Sharma, Mohit & Rohit, & Bisht, Kunal & Kumar, Ashish & Jain, Rachna & Nagrath, Preeti & Singh, Pritpal. (2022). Sign language detection and conversion to text using CNN and OpenCV. AIP Conference Proceedings. 2555. 040016. 10.1063/5.0108711.

[16] Alleema, N. & Selvamani, Babeetha & Kumar, P. & chandrasekaran, Saravanan & Pandiaraj, S & Kumar, A. & Rajkumar, K.. (2022). Recognition of American Sign Language Using Modified Deep Residual CNN with Modified Canny Edge Segmentation. 10.21203/rs.3.rs-1521209/v1.

[17] Jiang, Xianwei & Satapathy, Suresh & Yang, Longxiang & Wang, Shuihua & Zhang, Yu-Dong. (2020). A Survey on Artificial Intelligence in Chinese Sign Language Recognition. Arabian Journal for Science and Engineering. 45. 10.1007/s13369-020-04758-2.

[18] Satiman, Supathep & Meesad, Phayung. (2022). Deep transfer learning base on sequenced edge grid image technique for sign language recognition. International journal of health sciences. 9982-9997. 10.53730/ijhs.v6nS5.12112.

[19] Gaye, Sakshi. (2022). Sign Language Recognition and Interpretation System. INTERANTIONAL JOURNAL OF SCIENTIFIC RESEARCH IN ENGINEERING AND MANAGEMENT. 06. 10.55041/IJSREM13428.

[20] Pu, Junfu & Zhou, Wengang & Li, Houqiang. (2018). Dilated Convolutional Network with Iterative Optimization for Continuous Sign Language Recognition. 885-891. 10.24963/ijcai.2018/123.

[21] Aly, Walaa & Aly, Saleh & Almotairi, Sultan. (2019). User-Independent American Sign Language Alphabet Recognition Based on Depth Image and PCANet Features. IEEE Access. PP. 1-1. 10.1109/ACCESS.2019.2938829.

[22] Oyedotun, Oyebade & Khashman, Adnan. (2017). Deep learning in vision-based static hand gesture recognition. Neural Computing and Applications. 28. 10.1007/s00521-016-2294-8.

[23] Tolentino, Lean Karlo & Serfa Juan, Ronnie & Thio-ac, August & Pamahoy, Maria & Forteza, Joni & Garcia, Xavier. (2019). Static Sign Language Recognition Using Deep Learning. International Journal of Machine Learning and Computing. 9. 821-827. 10.18178/ijmlc.2019.9.6.879.

[24] Bendarkar, D., Somase, P., Rebari, P., Paturkar, R. & Khan, A. (2021). Web-Based Recognition and Translation of American Sign Language with CNN and RNN. International Association of Online Engineering. Retrieved November 9, 2022, from https://www.learntechlib.org/p/218958/.

[25] Amrutha D , Bhumika M , Shivani Hosangadi , Shravya, Manoj H M, (2022), Real Time Static and Dynamic Hand Gesture Recognition using CNN, INTERNATIONAL JOURNAL OF ENGINEERING RESEARCH & TECHNOLOGY (IJERT) Volume 11, Issue 05 (May 2022),

[26] Liu, Lei & Wan, Shaohua & Hui, Xiaozhe & Pei, Qingqi. (2022). Data Dissemination for Industry 4.0 Applications in Internet of Vehicles Based on Short-term Traffic Prediction. ACM Transactions on Internet Technology. 22. 1-18. 10.1145/3430505.

[27] Mehreen Hurroo , Mohammad Elham, 2020, Sign Language Recognition System using Convolutional Neural Network and Computer Vision, INTERNATIONAL JOURNAL OF ENGINEERING RESEARCH & TECHNOLOGY (IJERT) Volume 09, Issue 12 (December 2020).

[28] "American sign language," National Institute of Deafness and Other Communication Disorders. [Online]. Available: https://www.nidcd.nih.gov/health/american-sign-language. [Accessed: 15-Nov-2022].

[29] Lewis, Buckley & Secco, Emanuele. (2021). A CNN sign language recognition system with single & double-handed gestures. 10.1109/COMPSAC51774.2021.00173.

[30] Ergen, Tolga & Sahiner, Arda & Ozturkler, Batu & Pauly, John & Mardani, Morteza & Pilanci, Mert. (2021). Demystifying Batch Normalization in ReLU Networks: Equivalent Convex Optimization Models and Implicit Regularization.

[30] Hassan, Nazmul. (2022). Bangla Sign Language Gesture Recognition System: Using CNN Model. 10.14293/S2199-1006.1.SOR-.PPUF56Q.v1.

[31] Wattamwar, Aniket. (2021). Sign Language Recognition using CNN. International Journal for Research in Applied Science and Engineering Technology. 9. 826-830. 10.22214/ijraset.2021.38058.

[32] Chen, I-Te & Shiang, Rung & Huang, Hung-Yuan. (2020). Sign Language Recognition System Using Cnn. Journal of Medicine and HealthCare. 1-6. 10.47363/JMHC/2020(2)138.

[33] Alzubaidi, Laith & Zhang, Jinglan & Humaidi, Amjad & Al-Dujaili, Ayad & Duan, Ye & Al-Shamma, Omran & Santamaría, J. & Fadhel, Mohammed & Al-Amidie, Muthana & Farhan, Laith. (2021). Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. Journal of Big Data. 8. 10.1186/s40537-021-00444-8.

[34] Srivastava, Smriti & Khari, Manju & Gonzalez Crespo, Ruben & Chaudhary, Dr-Gopal & Narula, Parul. (2021). Concepts and Real-Time Applications of Deep Learning. 10.1007/978-3-030-76167-7.

[35] Zhang, Qing & Huang, Huajie & Li, Jizuo & Zhang, Yuhang & Li, Yongfu. (2022). CmpCNN: CMP Modeling with Transfer Learning CNN Architecture. ACM Transactions on Design Automation of Electronic Systems. 10.1145/3569941.

[36] Madanan, Mukesh & Sayed, B.. (2022). Designing a Deep Learning Hybrid Using CNN and Inception V3 Transfer Learning to Detect the Aggression Level of Deep Obsessive Compulsive Disorder in Children. International Journal of Biology and Biomedical Engineering. 16. 207-220. 10.46300/91011.2022.16.27.

[37] Islam, Md & Uddin, Md & AKhtar, Md & Alam, K.M. Rafiqul. (2022). Recognizing multiclass Static Sign Language for deaf and dumb people of Bangladesh based on transfer learning techniques and the primary sign word dataset. Informatics in Medicine Unlocked. 33. 101077. 10.1016/j.imu.2022.101077.

[38] Niswati, Za'Imatun & Hardatin, Rahayuning & Muslimah, Meia & Hasanah, Siti. (2021). Perbandingan Arsitektur ResNet50 dan ResNet101 dalam Klasifikasi Kanker Serviks pada Citra Pap Smear. Faktor Exacta. 14. 160. 10.30998/faktorexacta.v14i3.10010.

[39] Zhang, Qi. (2022). A novel ResNet101 model based on dense dilated convolution for image classification. SN Applied Sciences. 4. 10.1007/s42452-021-04897-7.

[40] Ji, Li & Mao, Rongzhi & Wu, Jian & Ge, Cheng & Xiao, Feng & Xu, Xiaojun & Xie, Liangxu & Gu, Xiaofeng. (2022). Deep Convolutional Neural Network for Nasopharyngeal Carcinoma Discrimination on MRI by Comparison of Hierarchical and Simple Layered Convolutional Neural Networks. Diagnostics. 12. 2478. 10.3390/diagnostics12102478.

[41] Andrew, Andrew & Santoso, Handri. (2022). Compare VGG19, ResNet50, Inception-V3 for Review Food Rating. SinkrOn. 7. 845-494. 10.33395/sinkron.v7i2.11383.

[42] Bouhsissin, Soukaina & Sael, Nawal & Benabbou, Faouzia. (2021). Enhanced VGG19 Model for Accident Detection and Classification from Video. 39-46. 10.1109/ICDATA52997.2021.00017.

[43] Han, Baoru & Du, Jinglong & Jia, Yuanyuan & Zhu, Huazheng. (2021). Zero-Watermarking Algorithm for Medical Image Based on VGG19 Deep Convolution Neural Network. Journal of Healthcare Engineering. 2021. 1-12. 10.1155/2021/5551520.

[44] Jo, Taeho. (2021). Decision Tree. 10.1007/978-3-030-65900-4_7.

[45] Chistopolskaya, Anastasiya & Podolskii, Vladimir. (2022). On the Decision Tree Complexity of Threshold Functions. Theory of Computing Systems. 66. 10.1007/s00224-022-10084-x.

[46] McTavish, Hayden & Zhong, Chudi & Achermann, Reto & Karimalis, Ilias & Chen, Jacques & Rudin, Cynthia & Seltzer, Margo. (2022). Fast Sparse Decision Tree Optimization via Reference Ensembles. Proceedings of the AAAI Conference on Artificial Intelligence. 36. 9604-9613. 10.1609/aaai.v36i9.21194.

[47] Das, Sunanda & Imtiaz, Samir & Neom, Nieb & Siddique, Nazmul & Wang, Hui. (2022). A hybrid approach for Bangla sign language recognition using deep transfer learning model with random forest classifier. Expert Systems with Applications. 213. 118914. 10.1016/j.eswa.2022.118914.

[48] Thiruselvan, Dr. (2022). INDIAN SIGN LANGUAGE DETECTION USING ARTIFICIAL INTELLIGENCE. International Journal of Engineering and Artificial Intelligence. 3. 27-30. 10.55923/JO.IJEAL.3.3.104.

[49] Ramani, B. & Lakshmi, T. & Durga, N. & Sana, Shaik & Sravya, T. & Jishitha, N. (2022). Recognition of Hand Gesture-Based Sign Language Using Transfer Learning. 10.1007/978-981-19-1976-3_12.

[50] Tamiru, Nigus & Tekeba, Menore & Salau, Ayodeji. (2022). Recognition of Amharic sign language with Amharic alphabet signs using ANN and SVM. The Visual Computer. 38. 1-16. 10.1007/s00371-021-02099-1.

[51] Zahid, Hira & Rashid, Sheikh Muhammad & Hussain, Samreen & Azim, Fahad & Syed, Engr. Dr.Sidra & Saad, Afshan. (2022). Recognition of Urdu sign

language: a systematic review of the machine learning classification. PeerJ Computer Science. 8. e883. 10.7717/peerj-cs.883.

[52] Myagila, Kasian & Kilavo, Hassan. (2021). A Comparative Study on Performance of SVM and CNN in Tanzania Sign Language Translation Using Image Recognition. Applied Artificial Intelligence. 36. 1-16. 10.1080/08839514.2021.2005297.