Artificial Intelligence and Social Computing, Vol. 72, 2023, 131–138
https://doi.org/10.54941/ahfe1003281

Towards Kenyan Sign Language Hand Gesture Recognition Dataset

Casam Njagi Nyaga and Ruth Diko Wario

Department of Computer Science and Informatics, University of the Free State, QwaQwa Campus, Private Bag X13, Kestell 9866, Republic of South Africa

ABSTRACT

Datasets for hand gesture recognition are now an important aspect of machine learning. Many datasets have been created for machine learning purposes. Some of the notable datasets include Modified National Institute of Standards and Technology (MNIST) dataset, Common Objects in Context (COCO) dataset, Canadian Institute For Advanced Research (CIFAR-10) dataset, LeNet-5, AlexNet, GoogLeNet, The American Sign Language Lexicon Video Dataset and 2D Static Hand Gesture Colour Image Dataset for ASL Gestures. However, there is no dataset for Kenya Sign language (KSL). This paper proposes the creation of a KSL hand gesture recognition dataset. The dataset is intended to be in two-fold. One for static hand gestures and one for dynamic hand gestures. With respect to dynamic hand gestures short videos of the KSL alphabet a to z and numbers 0 to 10 will be considered. Likewise, for the static gestures KSL alphabet a to z will be considered. It is anticipated that this dataset will be vital in creation of sign language hand gesture recognition systems not only for Kenya sign language but of other sign languages as well. This will be possible because of learning transfer ability when implementing sign language systems using neural network models.

Keywords: Dataset, Kenya sign language KSL, Object recognition, Artificial intelligence, Machine learning hand image dataset, Video dataset, Gesture recognition, Image dataset, Computer vision, Feature extraction

INTRODUCTION

Sign language (SL) is a full-fledged language like any other language. According to Hulit, Fahey and Howard (2014) there are three fundamental components of any SL. These three components are form, content, and use. The form comprises of phonology (how sounds are arranged into meaningful words), morphology (how sections of a word make meaning), and syntax (structural components of a language). Content comprises of semantics (rules that dictate the meaning of words or logical meaning of sentences). Use encompasses pragmatics (rules that relate the meaning of the language with respect to the environment the communication is taking place). SL is used worldwide. However, an international SL does not exist and each country and regions in countries have their own sign languages. For instance, in America the American Sign language (ASL) is used, likewise in the Republic of South Africa the South African Sign Language (SASL) is used and in Kenya, the KSL is used (KSDC, 2001).

The KSL being of interest in this study is discussed in more detail throughout this study. Statistically, the KSL is used by approximately 1 million people and is recognized by the constitution as one of the languages in Kenya (KNAD, 2010). This shows that in Kenya, SL is taken as an important language of communication.

THE STRUCTURE OF KSL

SL utilizes idiomatic expressions, finger spelling, and gestures to communicate messages (Nyaga, C. N., & Wario, R. D.,2020). Hence, SL is considered a visual language which does not follow the grammar of spoken English (Ndurumo, 2008). Unlike the spoken language (for instance English language) where verbs are constantly preceding the object, in KSL the verbs are constantly succeeding the object. This makes KSL fundamentally very different from English which is used to explain the KSL in schools. This makes it challenging for most of the deaf students or new learners of the KSL who may use other languages like their mother tongues to explain KSL while communicating (Roald, 2002). The subject, object and verb order of arrangement is in most cases preferred and used for KSL. In order to communicate a message, KSL also relies on the facial expression to express the mood of the subject or object of the communication (Nyaga, C. N., Wario, R. D., & De Wet, L., 2021). KSL also consist of alphabets and numbers which can be finger spelled. These alphabets and numbers are illustrated in Figure 1.

There are 26 alphabets and 10 digits that make up the KSL alphabet. These alphabets are used to create the KSL dataset.

The KSL dataset was tested through a prototype system created using Convolutional Neural Network (CNN). CNN was inspired by some notable events that include the work of Hubel and Wiesel (1962) who proposed an explanation to the way a cat's visual cortex perceives the world around them using the layered architecture of the neurons in the cat's brain. This inspired researcher to attempt to develop similar pattern recognition mechanisms giving birth to algorithms like the Artificial Neural Network (ANN). Another notable event is the work of LeCun et al. (1998) who show that CNN is suitable for classification, and can outperform other 2D shapes classification methods. They also show that CNN eliminates the need for hand-crafted heuristics for feature extraction. Therefore, CNN is able to perform both feature extraction and classification automatically through the use of gradient-based learning methods. The ability to learn features and classify them automatically make CNN very desirable for image classification problems like the sign language hand gesture recognition (Nyaga, Casam & Wario, Ruth, 2020).

METHODOLOGY

The KSL dataset that was created consisted of the following alphabets A to Y excluding the two letters J and Z which require motion when their gestures are performed. The dataset was created through 10 deaf signers. Static and

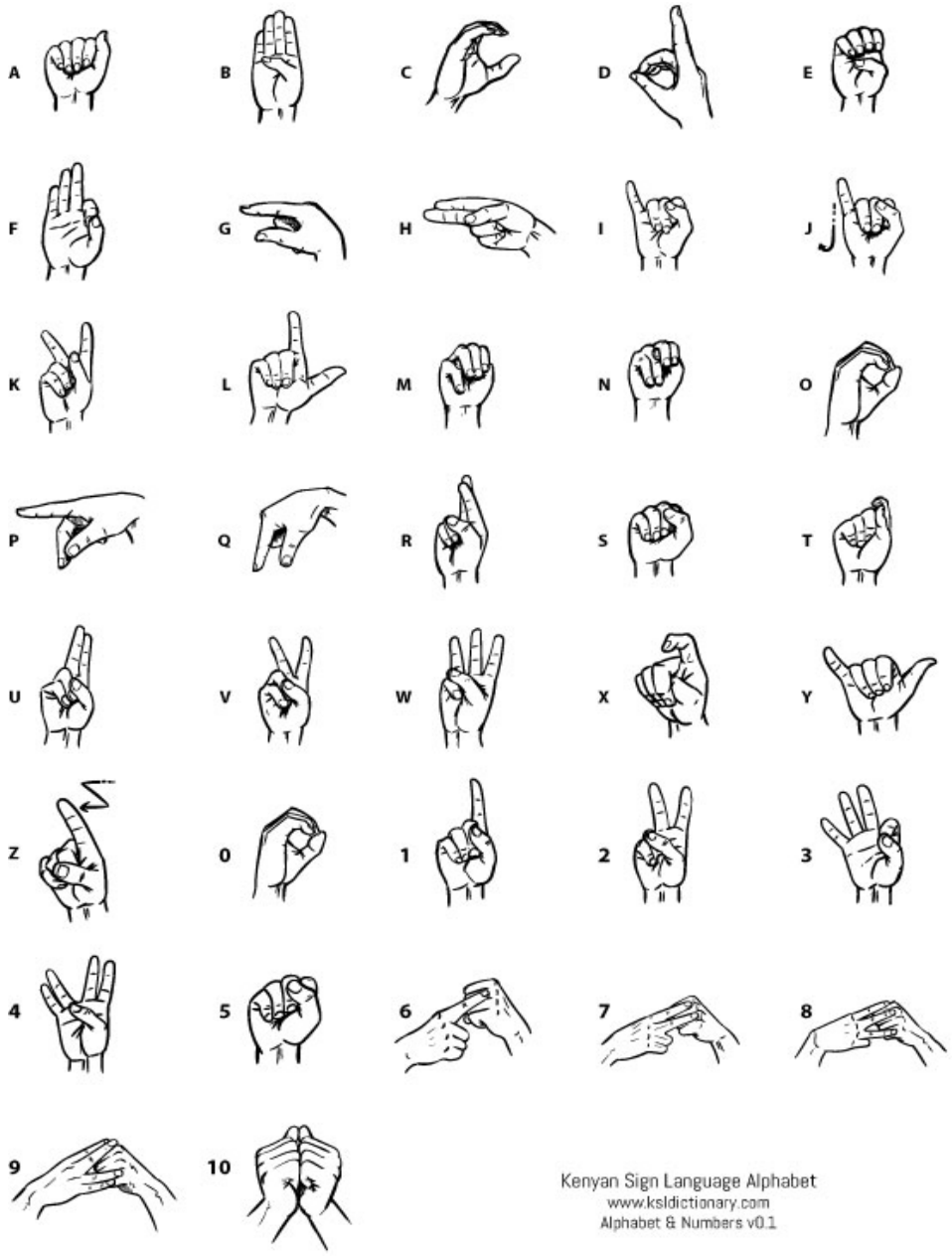

Figure 1: Kenyan sign language alphabet (KSLA, 2015).

video gestures where recorded. However the video dataset has not been finalized, hence the static dataset was tested through the CNN model which is described in Figure 4. The images where captured via a webcam and preprocessing was done on the images to remove the background and convert them to gray scale. A total of 3600 images for the KSL where used for the test set. It constituted 150 images of the 24 alphabets that where considered in

Figure 2: Sample raw KSL alphabet as captured by the webcam.

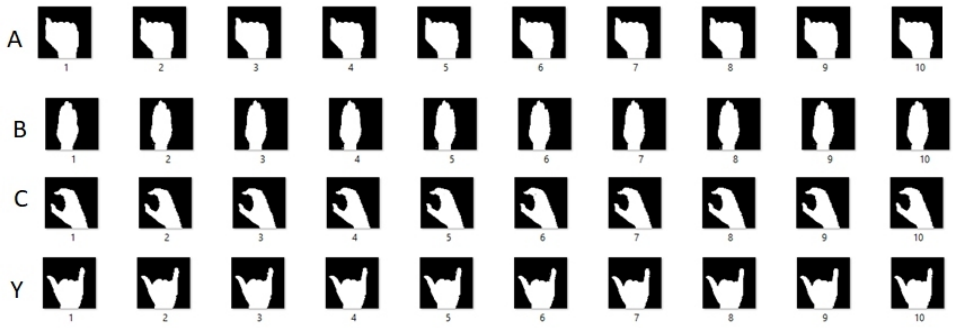

Figure 3: Sample KSL alphabet dataset after preprocessing.

the dataset. A total of 24000 images where used for the training set. This constituted of 1000 images of the 24 alphabets that where considered.

In order to test the KSL dataset a prototype KSL recognition system was proposed.

The Prototype System Design

The prototype system was developed using the Python programming language. The Python programming language was run through the Anaconda (Navigator Version 1.9.7) python distribution software. The Anaconda Python distribution software is a platform that allows one to run and integrate many programming libraries. A library is a group of programming functions that solve a specific problem. The Anaconda Python platform also allows many Integrated Development Environments (IDE) to be installed and launched from it. Therefore, the system was run using the Scientific Python Development Environment (Spyder version 3.3.3.), which is a python (IDE) with advanced editing, interactive testing, debugging and introspection features.

Flow Diagram of the Prototype System

The CNN model is diagrammatically shown in Figure 4.

Figure 4, shows the layers in the CNN model that was used in the prototype system. When running a CNN model there are two approaches that can be

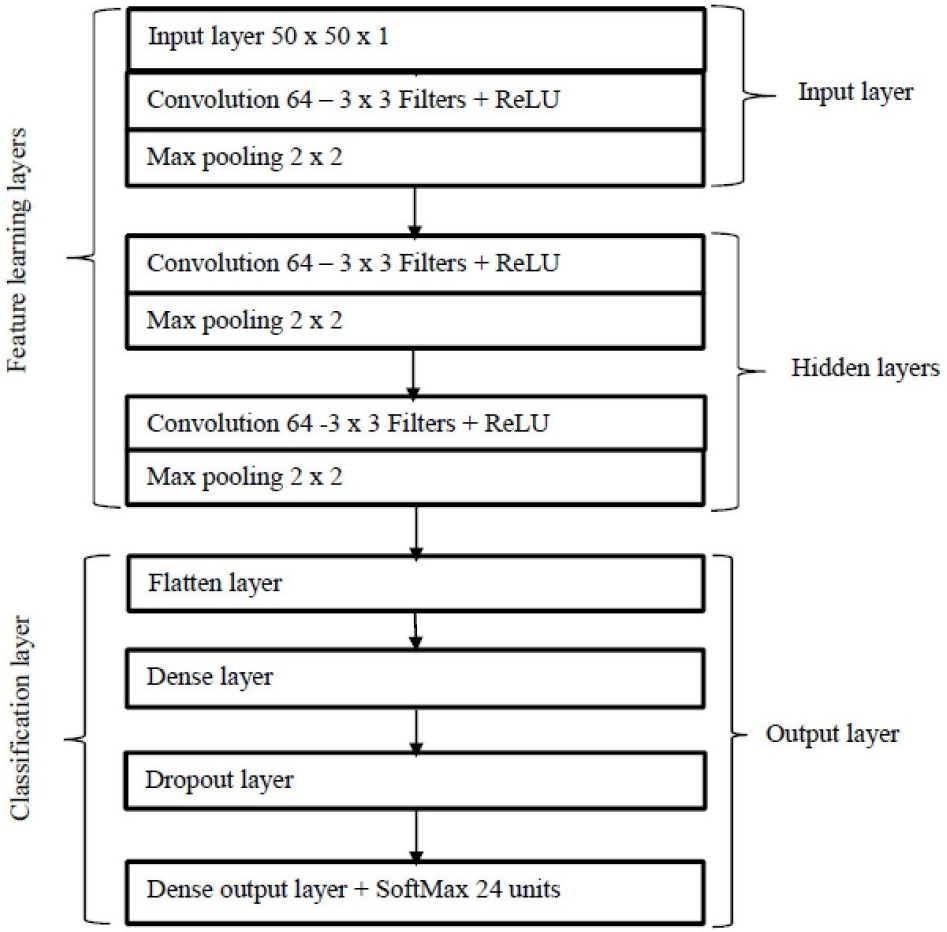

Figure 4: Convolutional neural network model for the prototype system.

used. These approaches include convolution 2D and convolution 3D. The convolution 2D is used for black and white images and the convolution 3D is used for full-colour images like those with the values of red, blue and green. The images that were used in the CNN model for the prototype system were black and white images. Therefore, the convolutional 2D approach was used (Nyaga, Casam & Wario, Ruth., 2020).

The CNN model that was used as a sequential model that is organized in layers. The first layer was the input layer that was followed by the hidden layers and finally an output layer. The input and hidden layers are used for image feature learning. Therefore, CNN is able to extract features on its own automatically. The output layers are then used for classification of the input images to the correct class labels. When compiling the CNN model there are three parameters that are required, these parameters are the optimizer, metrics, and loss. The optimizer that was used was Adam, the metrics that were used in the model were accuracy and the loss was categorical cross-entropy.

A brief explanation of each of these layers is provided in the following sections.

- Input layer

The first layer in the CNN model is referred to as the convolution layer. Convolution means to multiply the image inputs matrix with the weights to produce a convolution filter that represents a feature of interest. In some cases, the term weight can be referred to as a filter, kernel or feature detector. The weights values are also represented as a matrix of values. The multiplication may be done at the pixel level of an image. That is each pixel of the image represents a neuron. A pixel may have three channels that are red, green or blue. To represent the RGB colour space. The pixel may also be having one channel meaning that it is grayscale or simply black and white. Whatever the case, the dot products of the input and the weights is summed up and multiplied with an activation function to form the feature maps or simply an activation map.

In our system the input layer takes as input a batch of 64 images of size 50x50 pixels. Each pixel forms a neuron in the CNN input layer hence the input layer consists of 160,000 neurons. The activation function that was used in each neuron in the CNN model was the rectified liner unit ReLU. In this layer, the activation functions control the flow of signals from one layer to the next by mimicking the human brain neuron. When the activation function is multiplied by the input signal threshold value is measured. If the output meets the threshold value then the signal is propagated to the next layer. If not, it is not propagated to the next layer. Common activation functions include; Rectified Linear Unit (ReLU), Leaky ReLU, Randomized Leaky ReLU, Parameterized ReLU, Exponential Linear Units (ELU), Scaled Exponential Linear Units, Tanh, hardtail, softtanh, soft sign, SoftMax, and soft plush. The ReLU is preferred for activation at the convolution layer because it has proved to be fast when training the CNN (Nair & Hinton, 2010).

- The hidden layers

The CNN model was used consisted of two hidden layers. Each hidden layer consisted of convolution operation, rectified linear unit activation function and a max-pooling operation. Each hidden layer received batches of 64 images and used a kernel filter of size 3x3 to perform the convolution. The subsampling layer also referred to as the pooling layer or subsampling layer. Pooling reduces the inputs from the convolution layer by smoothing. Smoothing can be done to reduce noise and variations from the input signals. This can be done by Max-pooling whereby the maximum value in each filter pass is taken and repeating the process for all the filters. That is the activation map from the convolution layer is down-sampled along with the spatial dimensionality by taking in the neighborhood values. An alternative method is to take the average of every filter neighborhood values repeating the process for the entire image. Ideally, subsampling reduces the image size by half hence it reduces the complexity of the network.

- Output layer

In the prototype system CNN model, the output layer consists of several operations. These operations are; flatten layer, dense layer, dropout, and

dense layer. The flatten layer was used to convert the input into a one-dimensional array that is passed to the dense layer or the fully connected layer. The drop out operation is then applied in order to avoid the problem of overfitting in the CNN Model. Overfitting is where the network has trained and learned how to treat the input data using a forward pass hence if the network repeats the task it may be tempted to use the same neurons it used before; therefore, no learning is taking place. Some of the methods that can be used for regularization include; L1/L2 method, batch norm method, gradient clipping method, and dropout method. The dropout method was used in the system that was used in this study. The dropout method randomly switches on and off some neurons in order to control the problem of overfitting in the network. The dropout enables the input signal to be processed through different paths hence the neurons keep learning (Hinton, Srivastava, Krizhevsky, Sutskever, & Salakhutdinov, 2012).

The SoftMax function is then applied in the final dense layer. The SoftMax function is a normalized exponential function that converts the output of every neuron into a value between 0 and 1 hence; the values can be viewed as a probability distribution of values. Therefore, the CNN model is able to tell the most likely label or class of an input image.

CONCLUSION

Our model was able to predict the 24 alphabets and the numbers 0 to 5 in the KSL dataset. The characters where A to Y excluding J and Z which require motion and the numbers where 0 to 5 excluding 6, 7, 8, 9, 10 which require both hands. The CNN model was able to recognize the KSL dataset with a prediction accuracy >97%. Over fitting was observed in the model. This may be because of the small data set; generally a dataset of over 100000 samples is required to successfully optimise convolution kernels in CNNs architecture. This dataset will be enhanced by creating more vocabulary and made available for researchers.

REFERENCES

Hinton, G., Srivastava, N., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2012). Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*.

Hubel, D., & Wiesel, T. (1962). Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *The Journal of physiology, 160*(1), 106–154.

Hulit, L., Fahey, K., & Howard, M. (2014). *Born to talk: An introduction to speech and language development*: Pearson Higher Ed.

KNAD. (2010). Official recognition of Kenyan Sign language as a national and official language in the Harmonized Draft Constitution.

KSDC. (2001). Report of sign language training of trainees workshop held at KISE in Nairobi, Kenya (Unpublished Manual).

KSLA. (2015). kenya sign language alphabet and numbers. http://ksldictionary.com/dictionary/signs.

LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE, 86*(11), 2278–2324.

Nair, V., & Hinton, G. E. (2010). *Rectified linear units improve restricted boltzmann machines*. Paper presented at the Proceedings of the 27th international conference on machine learning (ICML-10).

Ndurumo, M. (2008). Sign language interpreting with special reference to Swahili. *The African Annals of the Deaf, 1*(1).

Nyaga, C. N., & Wario, R. D. (2018, May). Sign language gesture recognition through computer vision. In *2018 IST-Africa Week Conference (IST-Africa)* (pp. Page-1). IEEE.

Nyaga, C. N., Wario, R. D., & De Wet, L. (2021). Pedagogical Interface Agent for Kenya Sign Language. In *Advances in Usability, User Experience, Wearable and Assistive Technology: Proceedings of the AHFE 2021 Virtual Conferences on Usability and User Experience, Human Factors and Wearable Technologies, Human Factors in Virtual Environments and Game Design, and Human Factors and Assistive Technology, July 25-29, 2021, USA* (pp. 461–468). Springer International Publishing.

Nyaga, C. N., & Wario, R. D. (2020, May). Towards a Sign Language Hand Gesture Recognition Design Framework. In *2020 IST-Africa Conference (IST-Africa)* (pp. 1–8). IEEE.

Nyaga, Casam & Wario, Ruth. (2020). A Review of Sign Language Hand Gesture Recognition Algorithms. 10.1007/978-3-030-51328-3_30.

Roald, I. (2002). Norwegian deaf teachers' reflections on their science education: Implications for instruction. *Journal of Deaf Studies and Deaf Education, 7*(1), 57–73.