

Offline Yoruba Word Recognition System based on Capsule Neural Network

Jumoke Falilat Ajao ^{1*}, Emmanuel Olorunmaiye ², Modinat Abolore Mabayoje ³, Shakirat Ronke Yusuff ⁴ and Amos Bajeh ⁵

^{1,2,4}Dept. of Computer Science, Kwara State University, Malete, Nigeria, jumoke.ajao@kwasu.edu.ng,
emmanuel.olorunmaiye@yahoo.com, shakirat.yusuff@kwasu.edu.ng

^{3,5}Dept. of Computer Science, University of Ilorin, Ilorin, Nigeria, mabayoje.ma@unilorin.edu.ng,
bajehamos@unilorin.edu.ng

* Corresponding author

Abstract

Context: The demand for digital document archiving has prompted numerous attempts in image document analysis, where scanned historical or indigenous materials can be kept in digitized form and made readily accessible for future use. The human visual system employs a space-variant sampling, coding, processing, and interpretation of variances in human handwriting, which makes it challenging for machines to comprehend. **Objective:** Therefore, it is crucial to digitize our indigenous heritage values in order to preserve them. In this sense, the study effort develops a Yoruba Text Recognition System utilizing Capsnet, based on the alphabet's unique characteristics. **Method:** The Yoruba bird names were collected from the Yoruba dictionary and written by 20 indigenous literate writers in 50 distinct writing styles. The obtained dataset were digitized and converted from RGB to a binary image using Otsu method. The image plot profile were taken on the digitized images to see the peak value of same images with different diacritic mark and under dot. The binarized images were utilized to train and evaluate the model of the Capsule Neural Network-based for the Handwritten Yoruba text recognition system. **Results:** After training many epochs, the training time and testing time are 89.2222 sec and 0.8667 sec, respectively. **Conclusion:** The model's accuracy is 85.57%, which looks promising.

Keywords: Yoruba, Handwriting Recognition, Capsnet, Document Analysis, Deep Learning.

1. Introduction

Text is one of the cleverest and impactful works ever created by humanity. It is a written form of human languages that enables practical consistent and efficient distribution or acquisition of information over time and space. The written word is the foundation upon which human civilisation is built. Additionally, it is an important tool for communicating with others and working together. Text has taken on an increasingly significant function in our contemporary civilization in recent years. It is imperative that indigenous languages be incorporated into our digital forms, since this will play a significant part in preventing the eventual loss of the language. In spite of the incredible advances in technology, using a pen and paper is much more convenient than using a computer keyboard or mouse. Handwriting-processing computers will need to be able to cope with a broad variety of writing styles and languages, as well as work with arbitrary alphabets that are established by the user, and understand any handwritten message written by any writer (Verma & Raghuvanshi, 2018).

Handwritten text is affected by all of today's computer and communication technology, including word processors, fax machines, image search, instant translation, robots navigation among other things. Handwritten character recognition is becoming increasingly significant in a range of business areas, including document recognition systems and handwritten text input devices, as a direct outcome of the continual development and progression of modern technology (Krizhevsky et al., 2012; Iamsa-at & Horata, 2013). There are distinctive challenges in human writings, which do not pose any problem for human to read and understand, however, the variability in handwritings possess lots of challenges for machine recognition. These variations have resulted in the need to refine and reinterpret the role of handwriting and handwritten messages because these variability makes the recognition of handwriting difficult for computer to understand and interpret.

There has been numerous number of handwritten character recognition systems developed for languages such as English, Arabic, Sinhala, and Falsi, amongst

others; but, only a small number of systems had been built for the Yoruba language (Jumoke et al., 2016; Vaidya et al., 2018). This could be explained by the fact that Yoruba orthography is based on roman letters and uses diacritical marks to denote tone and under-dots to differentiate some phonemes. In spite of the significant efforts that have been put into researching handwriting recognition, additional study is still required to increase the accuracy of Yoruba word recognition. All these difficulties through the years has been handled using the conventional machine learning approach before the introduction of deep learning that showed its potential in computer vision and other fields. The deep learning was introduced to avert the problem of extracting informative feature from the data using separate algorithm. From literatures, it was observed that deep learning as surpassed the conventional feature extraction algorithm in terms of accuracy. The necessity of finding a solution to these problems prompted the development of a Yoruba Character recognition system.

The process of developing the resources that are required to solve these problems of variability's and also to improve on Human-Computer Interaction and make modern technology relevant to African languages led to this research work. Ultimately, the goal of this research work is to make modern technology relevant to African languages. As a result, it is necessary to add indigenous language into the recognition system utilizing deep learning rather than the standard machine learning approach. Therefore, there is need to incorporate indigenous language into the recognition system using deep learning approach rather the conventional machine learning approach that uses a separate feature selection algorithm.

2. Review of Related Works

This section reviews some of the research work that have been carried out in Yoruba recognition system. Ajao et al. (2018) presented Yoruba words recognition system using Hidden Markov Model (HMM). The Yoruba words were modelled using a ten (10) state left-to-right HMM. The Baum-Welch re-estimation algorithm was used to train each HMM class using the handwritten word images' with DCT feature vector. The Viterbi algorithm was used to classify the handwritten word, giving the best state sequences for the model. This experiment reported a 92 % test accuracy and a 95.6% recognition rate.

Ibrahim and Odejebi (2011) built up a framework to perceive transcribed character of Yoruba Uppercase letters. Bayesian and decision tree. It isolated character into three regions: the top, the center and the base which indicates the toner, the character and under-dot. A recognition accuracy of 94.44% was achieved.

Akintola et al. (2019) presented evaluation of an optical character recognition model for Yorùbá text over the years, optical character recognition (OCR) for several languages has been developed and implemented in numerous applications. The invention of OCR permits the digitalization of neglected paper records and serves as a type of backup for those papers. The proposed method is for isolated Yorùbá language characters. The Yorùbá language is a tonal language in which vowel alphabets are stressed. Image gray scale, binarization, de-skew, and segmentation make up the procedure. Thus, OCR allows the system to read and convert images to text data. Precision and Recall were utilized to test the proposed model for information retrieval. Recall results in the example document were 100 percent, and precision results ranged from 76% to 100%.

Ajao et al. (2020) presented implementation of Yorùbá Unicode generation for an indigenous keyboard. Due to accent and diacritic markings specific to Yorùbá, several characters were not fixed by ASCII and Unicode. Coding Yorùbá characters is necessary for keyboard circuit implementations to type Yorùbá texts. Traditional keyboard layouts do not have a key for every character. This paper suggests using Unicode to implement a Yorùbá keyboard for human-computer interaction. The Yorùbá keyboard uses binary and hexadecimal coding standards to encode characters with diacritic marks and under dot content.

Günter and Bunke (2004) combined three classifiers: Hidden Markov Model, Artificial Neural Network, and KNearest Neighbor. They build an ensemble from several base classifiers rather than just one. The results of the ensemble method were compared to the results of combining three classifiers using Bagging and AdaBoost, which was distinguished from classical ensemble. The best combination scheme increased 2.98 %. The authors' new ensemble method outperformed the best combination of base classifiers by 1.5 %, classical ensemble methods by 2.94 %, and the best base classifier by 4.48 %.

Ajao et al. (2015) proposed information theory to evaluate image preprocessing quality. This is used to test the effects of various pre-processing stages on the recognition system. Using handwritten words samples of Yoruba medical terminology collected from indigenous Yoruba literate writers. The samples were digitized at 300dpi to allow for detailed image representation. They investigated the impact of entropy on an intuitive interpretation of handwritten words. The handwritten word has a higher entropy than the typewritten word in the experiment. This means that the handwritten word's information content is affected by perturbations that need to be removed using image pre-processing tools to get the same information content as the original.

3. Methodology

Convolutional neural networks (CNNs) are widely used in computer vision; and handwritten character recognition (Oyeniran et al., (2021). While CNN architectures like LeNet and AlexNet have gained popularity in the last two decades, spatial invariance of the different kernels has remained an issue. With the advent of capsule networks, kernels can now collaborate in a way that utilizes equivariance among kernels. They used handwritten Indic digits and character datasets to compare capsule network to LeNet. The CapsNet concept can be applied to Dynamic Routing, EM Routing, and other image classification problems (Andersen 2018; Sabour et al. 2018; Mandal et al. 2018). This paper uses CapsNet to recognize Yoruba handwritten word to improve existing algorithms for Yoruba character and word recognition.

Methods adopted for the proposed model are: Data Acquisition, pre-processing and classification

3.1 Data Acquisition

Data consisting of bird names was gathered from the Yoruba dictionary. The acquired text was written in a guided format by literate indigenous writers of Kwara State University, Malete students. Each student was made to rewrite exactly from a sample copy of all the Yoruba names of birds. The offline words were scanned and digitized. Samples of the collected images are shown in Figure 1. To enhance the number of sample datasets required for training, data augmentation techniques such as flipping and rotating the available dataset, utilizing multiple angles and flipping it in different directions, were used.

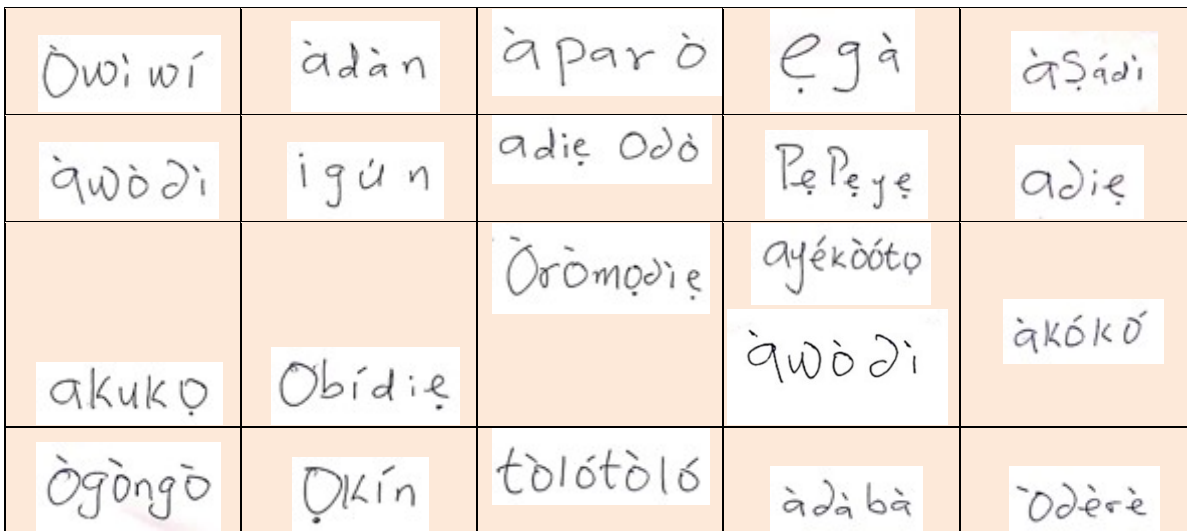


Figure 1. Original Images

3.2 Pre-processing

To make the collected data suitable for the recognition model, each word from each writer was cropped and placed in a folder so that it could become an individual entity of input, as shown in Figure 2. The cropped

images were converted to greyscale and binarization. The method returns a single intensity threshold that divides pixels into foreground and background classes. This limit is set by reducing intra-class intensity variance or, in other words, maximizing inter-class variation. These are shown in Figure 2 respectively.

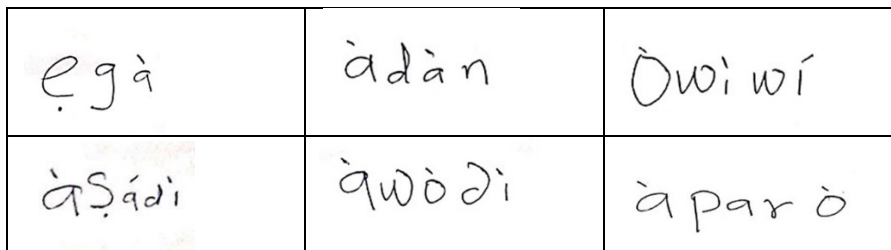


Figure 2. Binarized Images

3.3 Normalization

The (words) were re-sized into a 120 by 120 image, which is our choice because it will not just increase the

clarity but it is also important to have equal dimension because it improves the performance of the convolutional process. This also helps in calculating the

parameters we expect to generate after the activation of each letter. The preprocessed result is shown in Figure 3.

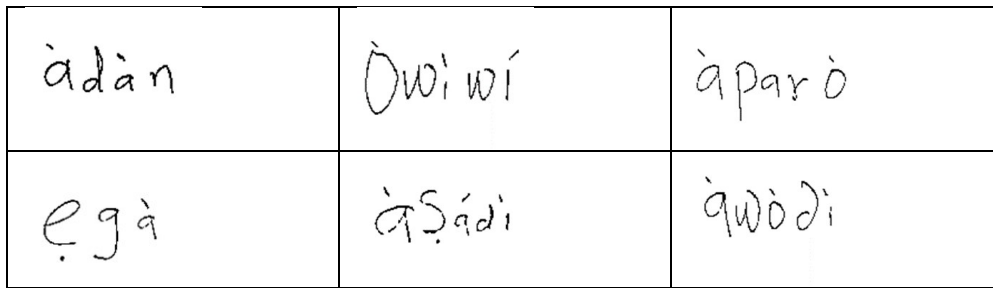


Figure 3. Normalized Images

3.4 Classification

For categorization, Capsnet, or Capsule Neural Networks, were utilized. Each layer's capsule is a group of neurons that do considerable internal calculation before producing a concise result. Capsnet was divided into two parts: an encoder and a decoder, each of which is divided into three layers.

3.4.1 Encoder

This model allow for a process in which a machine learning generates a sentence describing an image. It receives the image as the input and outputs a sequence of words. This is described in the following headings.

a. Convolutional Layer

The image that serves as input data is sent through a convolution layer, which creates outputs of feature maps in an array of 120 * 120 * 1 using an activation function called Relu. The convolutional layer multiplies the feature map using a 3 x 3 array filter before utilizing the highest value for each sector to build a new feature map that is then activated with an activation function (Relu). The number of strides used in each convolution is

completely dependent on the desired output. The process of this filter going through the pixel of the images and multiplying the values of the array with its own value is known as stride. We dealt with every bit of information recorded on the image because of the diacritic signs and under dot, A one stride filter was applied in this research work to help boost detection. Relu activation function is represented thus in Eq. (1):

$$F(x) = \max(0, x) \tag{1}$$

Where x = input value and $f(x) = \{0, \text{if } x = 0\}$ and $\{1, \text{if } x \leq 1\}$

Figure 4 shows what happens in the convolutional layer in more detail. The process of moving the filter over the pixel of the images that is multiplying the values of the array with its own value is known as stride and the number of strides used in any convolution is dependent on the desired output. In this research work a one stride filter was used because it helped to increase detection since every bit of information captured on the image has diacritic signs and under dot. Figure 4 further explains what goes on in the convolutional layer.

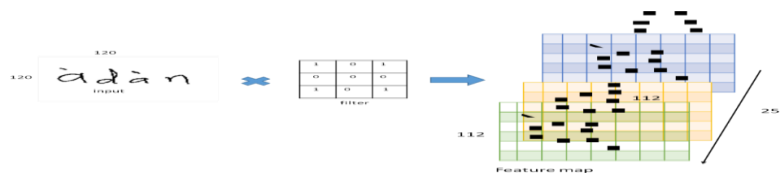


Figure 4. Process of Convolution

Figure 4 shows the size (width/height) of the feature map which is 112.

$$Output = \left((w - k + \frac{2p}{s}) + 1 \right) \tag{2}$$

Where w = input image size, k = filter size, p = padding, and s = striding = 1. Applying the formula,

$$Output = \left(120 - 9 + 2 + \frac{0}{1} \right) + 1$$

$$Output = 112 \left(\frac{width}{height} \right)$$

$$Output = 112 * 112 * 256$$

At the end of the first layer, the convolutional layer produces the image in Figure 5.

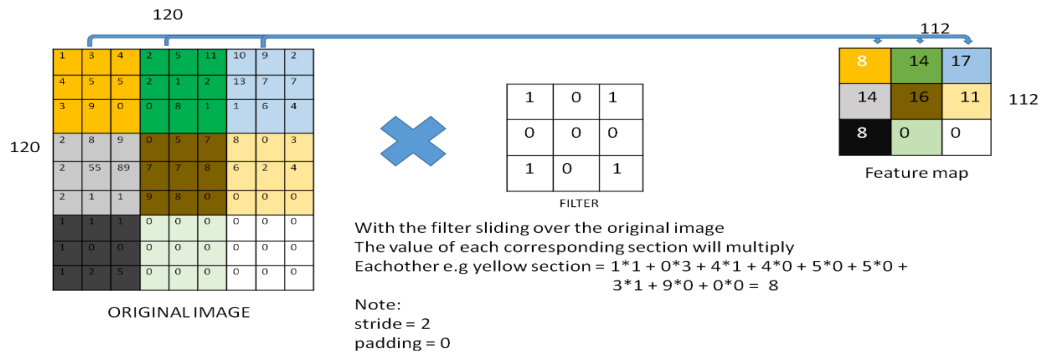


Figure 5. Sample of output from Convolutional Layer

We calculate our triangle parameter to be $(9*9+1) * 256 = 20,992$.

(i) Primary Capsule Layer

The primary capsule layer consists of a 32-primary capsule into which the convolutional layer's input parameters will be fed, and we have 20,992 trainable parameters and a $112*112*256$ dimension from the

previous layer, which was constructed using the conventional formula for output computation. As a result, new sets of dimensions will be input to both the input image, which is now the feature map, and the kernel. The formula is as follows:

Input = $112 * 112 * 256$

Kernel = $9 * 9 * 256$

Expected output is $52*52*32$

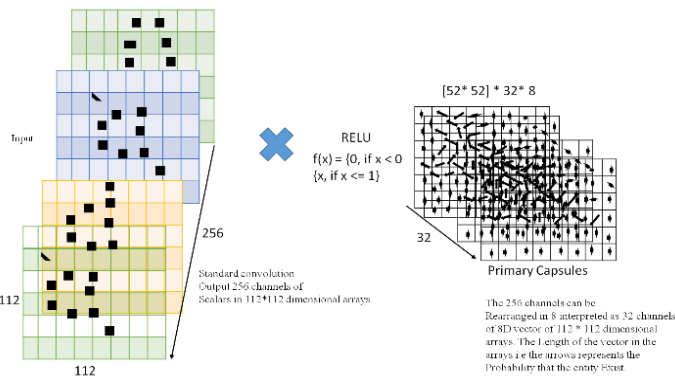


Figure 6. Primary Capsule generation

Based on the above solution, we find out that a $256, 9*9$ kernel with 2 stride this time around, works on the feature map input, which is the network's lower level capsule, to generate 32, $52*52$, 8-dimensional primary capsules as shown in Figure 5. To get the total trainable parameters, we multiply the $52*52$ dimensional array with $256/8$ dimension, which give us 32. Therefore, $52*52*32 = 86,528$ parameters, and these parameters make up the lower level capsule.

(ii) Digital Capsule

The Squashing function is the second nonlinear activation function used. The length of the output vector of a capsule represent the probability that the entity represented by the capsule is present in the current input. As a result, a nonlinear squashing function is used to ensure that short vectors were reduced to almost zero length and long vectors were reduced to a length slightly less than one. This makes it easier to put this non-linearity function to good use in classification. The Digital Capsule (DigitCaps) output matrix was created by feeding the PrimaryCaps outputs through a dynamic routing algorithm as shown in Figure 7.

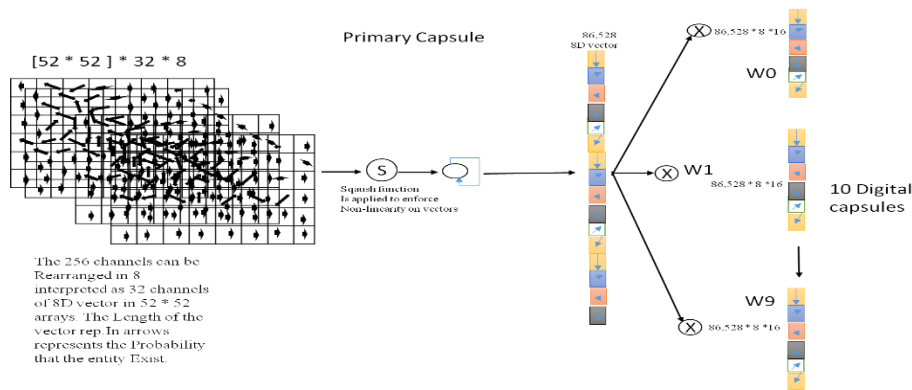


Figure 7. Digital Capsule Generation

Sabour et al. (2017) provides the squashing function used to calculate v_j . Each row in the DigitCaps matrix represents a single class's 16D instantiation parameters, and the length of a 16D vector represents the probability of that class's existence. The non-ground-truth rows are masked with zeros during training, and the matrix was passed to a reconstruction sub-network made up of two fully connected layers.

(iii) **Dynamic Routing**

Dynamic routing employs a transformation matrix containing the primary capsule's output to link the primary capsule to the digital capsule via backward and forward propagation, predicting the digital capsule's output and sending the primary capsule's output to the corresponding digital capsule. Figure 8 shows how to describe a Yoruba character using dynamic routing techniques.

P_j & P_h & P_l = Probability of existence

W_{ij} & W_{hj} & W_{lj} = weight of total output

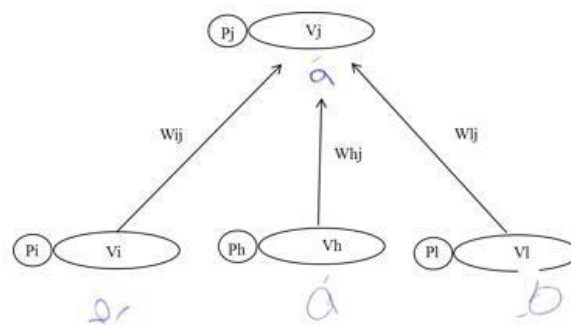


Figure 8. Dynamic routing (Source: Geoffrey et al., 2017)

The probability of existence is high between p_h and P_l we also expect the weights of w_{lj} and w_{hj} to be high, tending towards one (1). In a capsule network, neurons are grouped into capsules, and the routing algorithm uses iterations of update steps to determine the strength of the links between them. The initial value of an output capsule is computed using evenly allocated link strengths, with the exception of AttnCaps (Zhou et al., 2019), which calculates the initial output using a different input. For each routing cycle, the current link strengths determine whether the higher capsules are active. The routing

system then calculates the degree of agreement between capsules and adjusts the link strength accordingly. To determine capsule agreement, researchers can utilize the inner product (Vaidya et al., 2018; Iamsa-at et al., 2013; Shah et al., 2018) or a distance measure (Lensen et al., 2018). Once the link strengths have been changed, the cycle is repeated. In fact, routing occurs only a few times (typically three), and not until convergence occurs. The algorithm for dynamic routing is shown in List 1.

Listing 1: Dynamic Routing Algorithm((Wang and Liu, 2018)

```

Step 1: procedure Routing(^uji, r, l)
Step 2: for all capsule i in layer l and capsule j in layer (l+1): bij←0.
Step 3: for r iterations do
Step 4: for all capsule i in layer l: ci←softmax(bi)
Step 5: for all capsule j in layer (l+1): sj←∑cij^uji
Step 6: for all capsule j in layer (l+1): vj←squash(sj)
Step 7: for all capsule i in layer l and capsule j in layer (l+1):
    bij←bij+^uji.vj
    return vj
End
    
```

(iv) Vector Input/output of a Capsule

The length of a capsule's output vector, as previously stated, shows the likelihood that the entity represented by the capsule was present in the current input. As a result, we use a non-linear "squashing" function to ensure that short vectors are reduced to almost zero length whereas the big vectors are reduced to a length slightly less than one. Discriminative learning will take advantage of this nonlinearity, as shown in equation 4.

$$V_j = \frac{\|S_j\|}{1 + \|S_j\|^2} \times \frac{S_j}{\|S_j\|} \quad (4)$$

where v_j is the vector output of capsule j and s_j is its total input. The total input to a capsule s_j, except for the first layer, is a weighted sum of all "prediction vectors." ^u_ji from the capsules in the layer below, and is created by multiplying a capsule's output u_i by a weight matrix W_{ij} in eqn 5.

$$s_j = \sum c_{ij} \wedge u_j, \wedge u_j | i = W_{ij} u_i \quad (5)$$

where c_{ij} are the iterative dynamic routing process determined coupling coefficients. The coupling coefficients between capsule i and all capsules in the layer above sum to 1 and are determined by a routing softmax with initial logits b_{ij} equal to the log prior probabilities coupled to capsule j as shown in eqn (6).

$$C_{ij} = \frac{\exp(b_{ij})}{\sum \exp(b_{ij})} \quad (6)$$

The log priors can be learned discriminatively alongside the weights. They are dependent on the position and capsule type, but not on the current input image. The initial coupling coefficients are then improved iteratively by comparing each capsule's current output v_j with the prediction given by capsule i in the layer above. Each capsule in a convolutional capsule layer outputs a local grid of vectors to each type of capsule in the layer above, using unique transformation matrices for each member of the grid and each type of capsule.

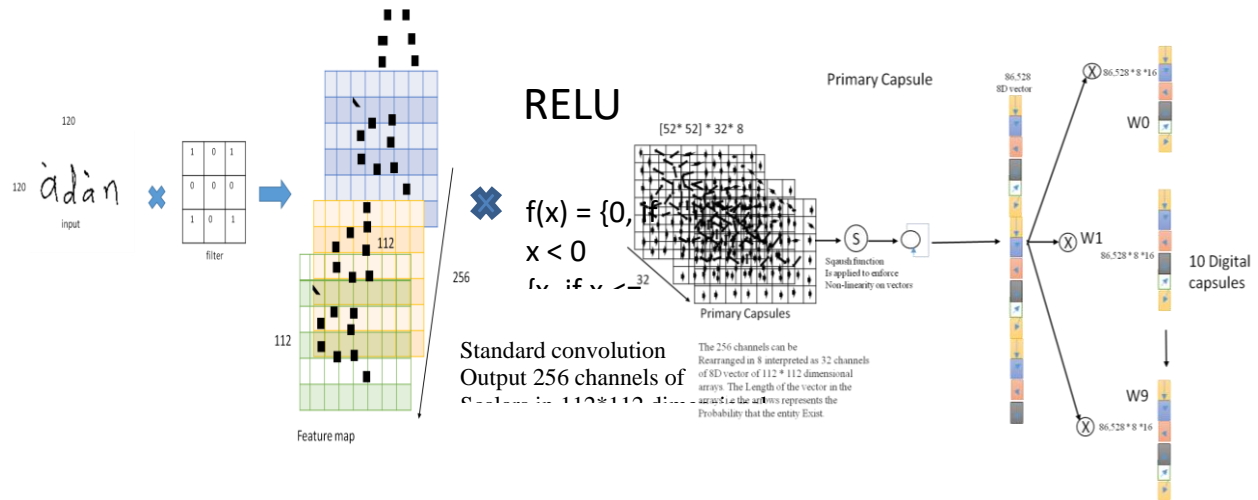


Figure 9. Capsule network layer 1-3

(v) Decoder

The Decoder maps the feature representation of the Yoruba text back into the input data space in the decoding process as shown in Figure 10.

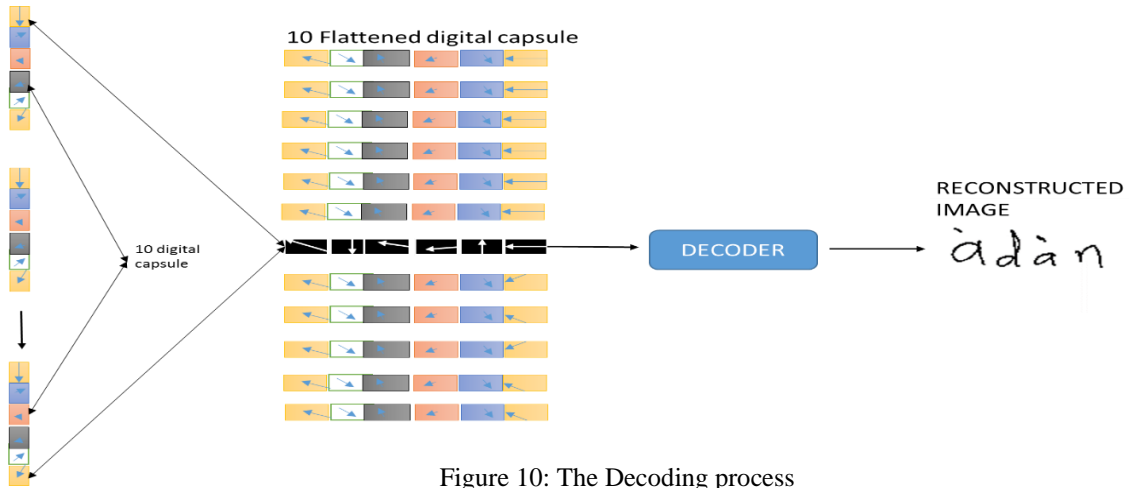


Figure 10: The Decoding process

(vi) Loss Function

A marginal loss for the existence of a digit is used in capsule networks. For digit k , the marginal loss is given as:

$$L_k = T_k \max(0, m+ - \|v_k\|)^2 + \pi (1 - T_k) \max(0, \|v_k\| - m-)^2 \tag{7}$$

$T_k = 1$ in this case if a digit of class k is present. The upper and lower bounds, $m+$ and $m-$, are set to 0.9 and 0.1, respectively, and v_k is set to 0.5. With the margin loss, a reconstruction loss is minimized.

(vii) Regularization

A network's regularization is essential in preventing models from over fitting the data. A parallel decoder network is connected to the obtained digit capsules as its input. The decoder reconstruct the original image. To ensure that the network does not over-fit the training set. However, the

reconstruction loss is scaled down by a factor of 0.0005 to avoid being dominated by the margin loss. The marginal loss function was used to track key image attributes during training and regularization to reduce over fitting. To reconstruct the original image, the encoding process was reversed using the same parameters.

4. Results and Discussion

This section shows the detailed description of the model for the Yoruba word recognition system the full description is shown in Table 1: The total parameters generated is 6,287,825 total trainable is 6,287,825 the total non- trainable parameter is 0. The training time and testing time after training different epoch produces 89.2222secs and the testing time generated 0.8667secs. The model accuracy gives 85.49%. The detail of this is presented in Table 1.

Table 1: Parameters settings for Capsule Neural Network Model

Layer Type	Output	Parameter count	Next Layer
Input layer	(None, 120, 120, 1)	0	Conv Layer 1
Conv layer 1	(None, 112,112, 256)	20,992	Conv Layer 2
Conv layer 2	(None, 52,52,256)	86, 528	Primary capsule
Primary Capsule	(None 32,32, 8)	0	Digital Capsule
Digital Capsule	(None 32, 8)	98, 547	Capsule layer
Decoder	(None, 120, 120,1)	75,758	
Training Time (500 epoch)	89.2222secs	Testing Time	0.8667secs

The plot profile shown in Figure 11-13 were distinctive indication of variations in diacritic signs and under dot of the image used. It was observed that distance 0 to 20 (pixel) shows a relatively reduced peak curve compare to Figure 11b, 12b, and 13b. The highest peaks correspond to character g, and the lowest values represent the acute accent. The measure of the image plot profile shows different curve value of the character, a character with under dot, a character with diacritic

signs. This is an indication that many instances of Yoruba characters should be trained with variant of the diacritic signs. Table 2 shows the result of the recognition accuracy. Twenty-five (25) instances of each Yoruba bird names were tested to validate the performance of the Yoruba recognition system developed.

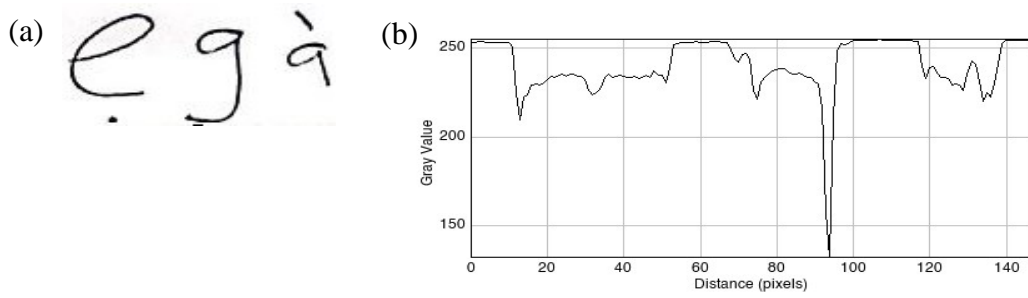


Figure 11. (a) with word (b) Plot profile of word “ega” with acute sign and acute diacunder dot ritic sign and under dot

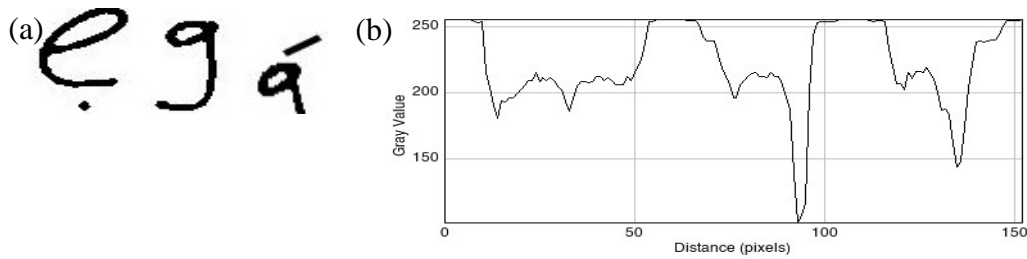


Figure 12. (a) word with grave diacritic sign. (b) The plot profile of the word ega with grave and under dot sign

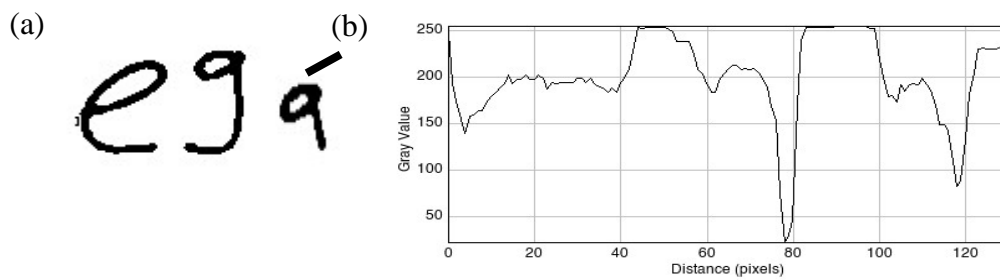


Figure 13. (a) Word ega without under dot (b) The plot profile of the word ega without under dot

Table 2: Recognition Accuracy for the Yoruba word

Names of Bird	instances	Recognized	Not Recognized
adan	25	20	5
owiwi	25	22	3
asadi	25	23	2
awodi	25	25	0
aparo	25	25	0
Owiwi	25	24	1
adaba	25	23	2
igun	25	23	2
ogongo	25	20	5
pepeye	25	20	5
odere	25	25	0
Adie odo	25	18	7
akuko	25	17	8
adaba	25	25	0

5. Conclusion

This research evaluates the effectiveness of capsule neural networks for offline Yoruba word recognition, in order to determine their suitability for recognizing characters with under dots and diacritic signs. 350 instances of the data collected were used to evaluate the developed system. 310 out of 350 words were recognized successfully. Though, the system generate 85.57% accuracy. The observed result demonstrates a substantial improvement. However, the recognition accuracy might be improved if more Yoruba word datasets could be provided for training the model.

References

- Ajao, J.F., Olabiyisi, S.O., Omidiora, E.O. & Odejobi, O.A. (2015), "Yoruba Handwriting Word Recognition Quality Evaluation of Preprocessing Attributes using Information Theory Approach", *International Journal of Applied Information Systems (IJAIS)*, Vol. 9, No. 1. pp. 18 - 23.
- Akintola, A., Ibiyemi, T. & Bajeh, A., (2019)," evaluation of an optical character recognition model for yoruba text". *Annals. Computer science series.* vol. 17 issue 1, p33-42. 10p.
- Andersen, P. A. (2018). Deep reinforcement learning using capsules in advanced game environments. *arXiv preprint arXiv:1801.09597*.
- Gunter, S. & Bunke, H. (2004, October). Combination of three classifiers with different architectures for handwritten word recognition. In *Ninth International Workshop on Frontiers in Handwriting Recognition* (pp. 63-68). IEEE.
- Iamsa-at, S. & Horata, P. (2013, December). Handwritten character recognition using histograms of oriented gradient features in deep learning of artificial neural network. In *2013 international conference on IT convergence and security (ICITCS)* (pp. 1-5). IEEE.
- Ibrahim, O. & Odejobi, O. A. 2011 "A system for the recognition of handwritten Yoruba Characters" *AGIS 2011 Ethopia.* Obafemi Awolowo University, IleIfe, Nigeria.
- Jumoke F. Ajao, Stephen O. Olabiyisi, Elijah O. Omidiora & Oladayo O. Okediran. (2016). *Hidden Markov Model Approach for Offline Yoruba Handwritten Word Recognition.*
- Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25.
- Lenssen, J. E., Fey, M. & Libuschewski, P. (2018). Group equivariant capsule networks. *Advances in Neural Information Processing Systems*, 31.
- Mandal, B., Dubey, S., Ghosh, S., Sarkhel, R., & Das, N. (2018, December). Handwritten indic character recognition using capsule networks. In *2018 IEEE Applied Signal Processing Conference (ASPCON)* (pp. 304-308). IEEE.
- Mukhometzianov, R. & Carrillo, J. (2018). CapsNet comparative performance evaluation for image classification. *arXiv preprint arXiv:1805.11195*.
- Oyeniran, O., & Oyeboode, E. (2021). Transfer Learning Based Offline Yorùbá Handwritten Character Recognition System. *Journal of Engineering Studies and Research*, 27(2), 89-95.
- Sabour, S., Frosst, N. & Hinton, G. (2018, February). Matrix capsules with EM routing. In *6th international conference on learning representations, ICLR (Vol. 115)*.
- Sabour, S., Frosst, N. & Hinton, G. E. (2017). Dynamic routing between capsules. *Advances in neural information processing systems*, 30.
- Shah, P. L., Gupta, T. K., Dhakad, J. B. & D'silva, M. R. (2018). A review paper on understanding capsule networks. *International Journal of Engineering Development and Research*.
- Vaidya, R., Trivedi, D., Satra, S. & Pimpale, M. (2018, April). Handwritten character recognition using deep-learning. In *2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT)* (pp. 772-775). IEEE.
- Verma, R. N. & Raghuwanshi, M. M. (2018, April). Comparative Study of Various Techniques on Handwriting Recognition and Analysis. In *2018 3rd International Conference for Convergence in Technology (I2CT)* (pp. 1-6). IEEE.