ISSN 2278-3091
Volume 9, No.1.1, 2020
International Journal of Advanced Trends in Computer Science and Engineering
Available Online at http://www.warse.org/IJATCSE/static/pdf/file/ijatcse0791.12020.pdf
https://doi.org/10.30534/ijatcse/2020/0791.12020

Convolutional Neural Network for Automatic Speech Recognition of Filipino Language

Felizardo Reyes Jr.[1], Arnel Fajardo[2], Alexander Hernandez[3]
[1]Technological Institute of the Philippines Quezon City, Philippines, felizardo.reyes.jr@tip.edu.ph
[2]Manuel L Quezon University, Philippines, acjajardo2011@gmail.com
[3]Technological Institute of the Philippines Manila, Philippines, alexander.hernandez@tip.edu.ph

ABSTRACT

Researches focusing on deep learning for speech recognition are integral for the successful implementation of natural language processing (NLP). Successful implementations of NLP would allow even non-technical and illiterate users to have access to technology by simply using their native tongue. Currently, there are very limited studies on the use of the Convolutional Neural Network (CNN) for Automatic Speech Recognition of Filipino Language. This paper presents a CNN model using SqueezeNet architecture for the Filipino language with 99.58% training accuracy, and 84.71% testing accuracy. Experimental method was employed to achieve this accuracy by adjusting the learning rate of the model. SqueezeNet architecture was chosen because it requires less resource but maintains the same level of accuracy as compared to other neural network architectures. Mel Frequency Cepstral Coefficient was also utilized to convert the audio inputs into Mel Spectrograms. The study also presented the precision rate of each Filipino class identified in the data set and how CNN was used to increase prediction accuracy. This CNN model in this study can be used as basis for further improvement of the speech recognition rate of other Filipino words and other new languages.

Key words: Automatic Speech Recognition, Convolutional Neural Network, CNN for Filipino language, Mel Frequency Cepstral Coefficient, Natural Language Processing, SqueezeNet architecture, Speech recognition for Filipino Language.

1.INTRODUCTION

Automatic Speech Recognition (ASR), sometimes called Automatic Voice Recognition, is the process of converting voice speech signals into text, which can be in the form of a word, word sequences, syllable, or any sub-word unit [1]. ASR being a subset of Natural Language Process (NLP) under the knowledge area of Artificial Intelligence (AI), aims to make communication between humans and computers as natural as possible. Its ultimate goal is to have human-computer communication indistinguishable to the human-human conversation.

With the advances in the field of neural networks, a series of algorithms used for recognizing relationships in a given set of data by using the human brain operation as model [2], has been continuously addressing difficulties relative to ASR. Deep Neural Networks (DNNs), a neural network with more than two layers of complexity and uses mathematical modeling to process data [3], has been used in speech recognition as early as 2010. DNNs made some significant achievements in automatic speech recognition. However, due to different speaking styles and settings, a model that can account for small shifts and perturbations in feature space leads to overfitting, and poor generalization is desired [4]. This lead to the focus on alternative neural network architecture, Convolutional Neural Networks (CNNs), to address some of these issues. CNNs, also known as ConvNet, are specific types of artificial neural network that uses machine learning unit algos or perceptrons to analyse data and perform supervised learning [5]. CNNs are implemented in computer vision, mainly in face recognition, scene labeling, image classification, action recognition, human pose estimation, and document analysis [6]. However, CNNs, when employed with a small number of convolutions, still suffer from overfitting [7].

CNNs provide significant improvement in the performance of image classification by using a large number of convolutional layers and sophisticated designs, and this similarity in CNN structures prompted CNN's use in speech recognition [8]. In order to visualize the audio signals, sounds need to be converted into spectrograms (two-dimensional representations of frequency over time) and mapped into a grid-like topology and then inputted to the CNN to be processed. This concept has been implemented in various research works on several types of audios and speeches and of different languages, but the results vary. In the research "Speech Recognition: Key Word Spotting through Image Recognition" [9], the developed CNN model was able to achieve a validation accuracy of 92% using 20% sample of input data. The low accuracy of the random sample is attributed to the limited data set that was used to train the CNN model.

CNN model for ASR implemented in the Filipino language is relatively unexplored. Previous studies on ASRs employ Hidden Markov Model [10], [11], [12], [13] while some focused only on recognition of Tagalog vowels [14]. Hence, the study is conceptualized to address this research gap. The objective of this paper is to develop a CNN model for the Filipino language, an important component in Automatic Speech Recognition of Filipino language. The paper also tested the accuracy of the developed CNN

model in terms of recognizing Filipino words. With this study, an available CNN model for Filipino words which uses processed data set was made available for automatic speech recognition.

The first section of this paper introduced the project and its objective and the use of CNN for speech recognition. The second section presents the literature associated with the concept of the paper, while the third section described the data set used in the study, including the pre-processing performed to prepare the data for deep learning. The fourth section details the process of applying a convolutional neural network to the Filipino language, including all the pre-processing requirements. Also, this paper discusses significant findings from experiments conducted. The last section discusses the conclusion and the future works related to the paper.

2.LITERATURE REVIEW

2.1 Convolutional Neural Network

Convolutional neural networks (CNNs) are commonly used in image identification and recognition problems as these CNNs offer several advantages compared to other techniques [15], as shown in Figure 2. This paper describes the various layers used. It used recognition of traffic signs where the authors discussed the challenges of the CNN implementation, and introduced Cadence-developed algorithms and software that can assess computational burden and energy for a by modifying recognition rates of traffic signs. This process of CNN layering was covered by the study.

A common implementation used in NLP is text classification. These implementations use text classifiers and rely on features designed for human use, such as dictionaries, knowledge bases, among others. The paper introduced a recurrent CNN for text classification without the above mentioned human-designed features. The model used a structure to capture the context of the information to learn how words can be represented, and reduce noise in a considerable level compared to some window-based neural networks. The authors employed a max-pooling layer that evaluates the critical roles the words play in the entire text [16]. The same principle of classification was employed in the study, but the focus is on speech or utterances.

Another study implemented CNN in mosquito imaging by extracting features from mosquito images to identify adult mosquitoes from various species and identify which are carriers of some fatal diseases like Dengue, Chikungunya, and Zika. The use of CNN provided an significant method for autonomous identification which is important for health workers and taxonomists for the identification of some insects that can transmit infectious agents to humans [17]. The study also used feature extraction for the classification of sounds. It also validated the accuracy of the CNN model for Filipino words.

Convolutional neural network (CNN) was also used is various computer vision activities and implementation to various human endeavor including radiology [18]. This paper offered the basic concepts of CNN and its application to radiological tasks. It discussed challenges and future directions in radiology field and how CNN can be used in addressing these. Similar to the use of multiple building blocks of this study, the study also experienced overfitting due to a small data set. The paper provided a complete CNN layer composed of a convolution layer, pooling layer, and fully connected layer plus the checking of non-linearity, as shown in Figure 1.

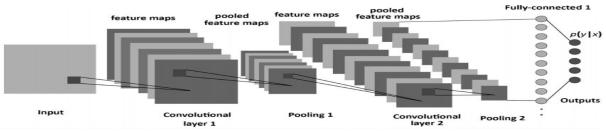

Figure 1: Structure of a CNN

CNN was also used for other scientific activities like natural history collections [19]. This study presented examples of how deep CNNs can be applied in analyzing of images of herbarium specimens. The results showed that CNN can detect mercury-stained specimens across a collection with 90% accuracy. Results also showed that CNN can accurately identify differences of two morphologically similar plant families with 96% accuracy. The way CNN modeling processes employed in this study was also adapted to this paper.

2.2 Automatic Speech Recognition

Automatic Speech Recognition (ASR) is the integration of process and the technology to convert speech signals into sequence of words or other linguistic entities using algorithms implemented in a device, a computer, or computer clusters [20]. Applications focused on consumers increasingly require ASR to be adaptable to the full range of real-world noises and other disturbances in the environment. The study is an attempt to provide a take-off point for ASR in Filipino by utilizing the same process of converting Filipino spoken words into word or sequence of words.

There are a lot of desirable aspects for the use of ASR in enhancing user experience. A sound ASR system can allow human voice command over typing, which facilitates faster communication. ASRs can also allow less knowledgeable users, through their natural language, interact with technology. In the work environment, ASRs can increase productivity, improve reliability, save time, minimize mistakes, and provide greater mobility. However, despite its many advantages, the development of ASR systems is by itself complex, as shown in Figure 2, and faces many challenges. There are challenges on the language itself like variability, characteristics, form, grammar, and vocabulary, challenges on the speakers of the language like pronunciation, speed of utterance, age, and the environment with which the speech was made like surrounding noises.

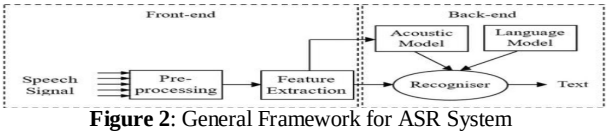

Figure 2: General Framework for ASR System

2.3 CNN for Automatic Speech Recognition

ASRs show the relationship between the acoustic speech signal and the phones in two steps: feature extraction and classifier training. Recent studies have shown that, in the CNN framework, direct modeling of the relationship

between the raw speech signal and the phones is possible, and ASR systems standardization can be built. In this paper, the authors analyzed and showed how the CNN learns (in parts) and models the phone-specific spectral envelope information of 2-4 ms speech. The robust CNN-based model yielded an ASR trend similar to standard short-term spectral based ASR system under noisy environments [21]. This paper proved the robustness of CNN-based models. It was also observed in the CNN model for the Filipino language.

Another paper focused on single-word speech recognition where CNN was used to identify a set of predefined words from a set of short audio clips. It used the Speech Commands dataset, which consists of 65,000 one-second long utterances of 30 short words of where identifying and classifying 10 words was implemented, To classify samples, the authors used one dimensional convolution on the raw audio waveform. The model achieved 97.4% accuracy on the validation set, 88.7%, and 89.4% on the two test sets. The paper showed that the model could predict samples of words it has seen during training with high accuracy. However, the model suffers from overfitting and had difficulty identifying words with extreme noises in the training data [22].

3.EXPERIMENTAL SETUP

3.1 Dataset

The data set used in this paper came from the Electrical and Electronics Engineering Institute of the University of the Philippines Diliman, a public university in the Philippines. It is composed of five (5) volumes of audio files in wav format (mono recording with sampling rate of 20 KHz) . It has a total of 140.8 hours of spoken Filipino sentences, words, and syllables with a total file size of 15.545 Gb. The gender of the speaker is 53% female, and 47% are male. Their age ranges are: 97% are 20-27 years old; 2% are 28-35 years old; and, 1% is 36-43 years old. Twenty-eight percent (28%) are spontaneous speeches, while 72% are non-spontaneous.

Each file in the volume follows a naming convention used to distinguish one file from another. The filename given indicated the gender of the speaker, the age bracket, and the type of spoken words, among others. File size ranges from 2Mb to 110Mb with an average duration of 1.1 minutes to 50.3 minutes of talk time.

Files in the data set were mostly composed of paragraphs, sentences, words and syllables in Filipino. In this unstructured format, the data set needs pre-processing to make it ready for speech recognition that is based on deep learning. To manipulate this waveform signal, the data needs to be translated into spectrograms, a visual representation of the spectrum of frequencies of a signal as it varies with time, using Adobe Audition. Adobe Audition was used as an editing tool to do simple cuts and splicing needed for this speech manipulation.

Since these were the raw data set, hours of speech segmentation with automatic labeling, shown in Table 1, was implemented to prepare Filipino corpora usable for deep learning, specifically CNN modeling. The process employed was able to extract 95% usable data set from the raw data set. However, among the extracted words, only forty-one (41) Filipino words were used in this study because a minimum limit of 200 samples of each word was set as a parameter.

Table 1: Speed Segmentation of the Dataset

Volume	Size of Data Set (in Gb)	Speed of Segmentation (in hours)
1	3.931	21.83
2	4.094	34
3	4.123	22.890
4	2.327	12.921
5	0.965	5.359

The data set used in the CNN model for the Filipino Language is composed of forty-one (41) Filipino words with over 200 samples each word, i.e., over 12,000 samples. These samples were used to train and test the model. Eighty percent (80%) of the sample Filipino words, shown in Table 2, were randomly chosen to train the CNN model and compute its accuracy. Twenty percent (20%), as shown in Table 3, Filipino words from the sample data set were used to test the CNN model for its accuracy.

Table 2: Dataset Used for CNN

Class	Training		Testing	
	Number of Samples	Size (in Gb)	Number of Samples	Size (in Gb)
tayo	261	1.55	63	0.37
may	343	1.82	86	0.46
yung	204	1.10	49	0.25
kailan	165	1.13	39	0.27
basa	171	1.05	41	0.26
tapos	165	1.12	38	0.25
nang	205	1.17	49	0.28
ang	246	1.30	59	0.32
kung	221	1.03	54	0.25
tao	255	1.50	62	0.37
nila	225	1.12	54	0.28
ngayon	229	1.46	55	0.36
ko	202	0.91	49	0.22
wala	209	1.10	51	0.27
pamilya	168	1.31	40	0.32
tama	162	0.90	39	0.21
ikaw	220	1.20	53	0.29
siya	298	1.39	73	0.34
naman	185	1.21	45	0.29
ano	177	0.98	42	0.24
maganda	173	1.35	42	0.32
hindi	274	1.59	67	0.38
iyon	169	1.03	41	0.24
dito	173	1.03	42	0.26
saan	168	1.00	40	0.24
na	602	2.88	149	0.72
sila	192	1.03	46	0.25
nung	172	0.92	41	0.24
ka	264	1.17	64	0.28
pero	177	1.06	43	0.27
po	287	1.26	70	0.30
sa	656	3.07	161	0.73
para	209	1.21	51	0.29
isang	192	1.13	46	0.27
pang	175	0.87	42	0.22
kaya	229	1.28	55	0.30
yan	282	1.53	69	0.38
buwan	204	1.23	49	0.31
ni	350	1.68	86	0.42
nasa	177	1.17	43	0.28
lang	276	1.48	67	0.37
TOTAL	9712	53.32	2355	12.97

The paper utilized the concept of CNN in Automatic Speech Recognition taking spectrograms as input as shown in Figure 3.

Figure 3: CNN for Speech Recognition

The processes done in this paper is shown in Figure 4. With audio files as input, preprocessing was done to convert these inputs into Mel Spectrograms and extract features (phonemes). Optimization using Adam was integrated before the actual modeling process was done.

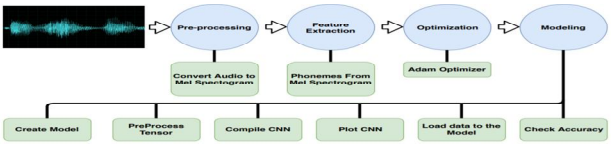

Figure 4: CNN Processes for Filipino Language

3.2 Preprocessing - Conversion of Audio Files to Mel Spectrogram

Figure 5 shows the process of converting audio input into the Mel Frequency Cepstral Coefficient (MFCC) or Mel Spectrogram. The first step in ASR is to extract features of the audio files to identify components that can be used in identifying the appropriate words and eliminating other information which does not contribute to the recognition process like background noise, emotions, and others. The phonemes [23] in a speech produced by the sound is determined by the shape of the vocal tract including the tongue, and teeth, among others, wherein the shape of a vocal tract is identified in the envelope of short-time power spectrum [24]. MFCC accurately represents this envelope.

Figure 5: MFCC Deviation Process

To generate the Mel Spectrogram, sample input with windows of size 2048 and a hop size of 512 were defined for the system to sample each time following the next window. Each window is transformed from a time domain to a frequency domain by computing the Fast Fourier Transform (FFT) for each window. Mel scale was generated by separating the entire frequency spectrum into evenly spaced frequencies of 128. Note that this is not based on distance on the frequency dimension, but the distance by which it is heard by the human ear.

Each window was processed by decomposing the magnitude of the signal into its components corresponding to the frequencies in the Mel scale. The audio input was converted to Mel Spectrogram using the function. This Mel Spectrogram was converted to decibels using the function.

3.3 Feature Extraction

The Mel Spectrograms generated will be used to extract features from the image. Phonemes from each sample of the class will be extracted to identify the appropriate Filipino word a specific phoneme is mapped. Figure 6 shows samples of spectrograms of the Filipino word "Ang", which is "the" in English. MFCC extracts these features and compares the power and decibel computed from each frame.

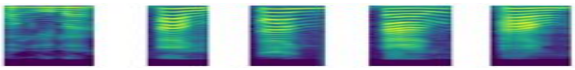

Figure 6: Sample Spectrograms of the Filipino Word "Ang"

3.4 Optimization

Adam (or Adaptive Moment Estimation) was used as an optimization algorithm rather than the classical stochastic procedure since it achieves goods results fast for training deep learning models [27]. This algorithm is appropriate for natural language processing because of its adaptive learning rate optimization method, which finds individual learning rates for each parameter. This means that sparse gradients, or those networks not receiving strong signals to tune their weights, in this case, those areas in the model which does not achieve a reasonable level of performance, will continuously be modified until the desired result is achieved. In this paper, the learning rate is the object of manipulation.

3.5 Modeling

The study used the SqueezeNet architecture for the arrangement of its neurons being the most appropriate in the study and its modelling of layers, connection patterns between layers, activation functions, and learning methods [25]. Among the many available architectures for neural networks, SqueezeNet was the most appropriate because it requires less communication across servers, less bandwidth, and less memory but achieves the same accuracy result [26]. Diagram for SqueezeNet Architecture is shown in Figure 7. This architecture of SqueezeNet is pre-built on Jupyter Notebook, an interactive computational environment for Python language.

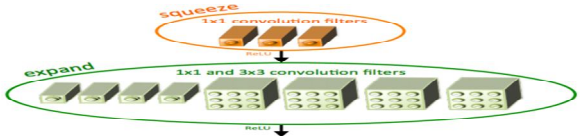

Figure 7: SqueezeNet Architecture

The initialized value of the SqueezeNet architecture for this paper is input shape = 32x32 pixels 3 channels (R, G, B); 41 Filipino words as classes; and max pooling for the Pooling Layer. The model, which was automatically generated by the script, resulted in 67 layers.

To visualize the model, Keras (open source neural network library written in Python) library was used to provide a utility function for plotting. The four arguments in the plot_model control the parameters for the graph to be visualized, such as controlling the output shapes, controlling layer names, controlling nested models into clusters, and controlling image dpi.

Image pre-processing was done using the ImageDataGenerator function to generate batches of tensor image data which will be looped over in batches. The number of images per batch of training and testing is set to 64 with a standard size of 32x32 (length x width).

In training the model, the callback function imports the values of the EarlyStopping, ModelCheckPoint, and ReduceLROnPlateau for every epoch, the number times that the learning algorithm will work through the entire training dataset [28]. The ModelCheckPoint function saves the model if its performance is better than the previous model. The EarlyStopping function stops the training of the model if the model's accuracy is not improving, given its patience value, in this case, 10. The ReduceLROnPlateau function reduces the learning rate of the model when the loss is not decreasing, given the patience value, in this case, 5. The CNN architecture was compiled using these functions and the parameter set for accuracy together with the Adam optimizer learning rate.

To cover the 41 classes, the number of training images per batch was set to 62 for both train and set images. The model will train 500 times (epoch = 500), and the steps per epoch were set to 2000 and 1000 validation steps to cover all the 9000+ training images and 2000+ test images. The CNN architecture was plotted using the Keras plot_model function model, and these were saved to update the model. These processes were iterated until the model achieved the highest accuracy. Figure 8 shows the CNN Model.

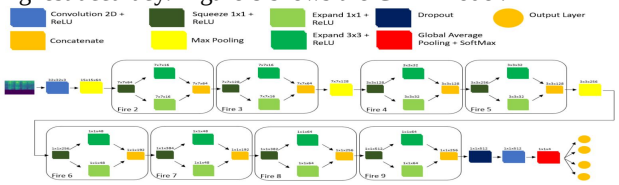

Figure 8: CNN Model for Filipino Language

3.6 Evaluation

To evaluate the CNN model, precision, recall, and f1-score were computed. Precision is the ratio of correctly predicted positive observation over the total predicted positive observations. Recall is the ratio of correctly predicted positive observations to all observations in the actual class. F1 score is the weighted average of Precision and Recall. Accuracy is the ratio of correctly predicted observation to the total observations.

4. RESULTS

Table 3 presents the training classification results. In the training results, the model achieves 100% accuracy.

Table 3: Training Classification Results

Class	Precision	Recall	F1-score	Support
tayo	0.94	0.99	0.96	261
may	1.00	1.00	1.00	343
yung	1.00	1.00	1.00	204
kailan	0.99	1.00	1.00	165
basa	1.00	1.00	1.00	171
tapos	1.00	0.99	1.00	165
nang	1.00	1.00	1.00	205
ang	1.00	1.00	1.00	246
kung	0.99	1.00	0.99	221
tao	1.00	1.00	1.00	255
tula	1.00	1.00	1.00	225
ngayon	1.00	1.00	1.00	229
ko	1.00	1.00	1.00	202
wala	1.00	0.99	1.00	209
pamilya	1.00	1.00	1.00	168
tama	0.99	0.99	0.99	162
ikaw	0.97	1.00	0.98	220
siya	1.00	1.00	1.00	298
naman	0.99	1.00	1.00	185
ano	1.00	0.99	1.00	177
maganda	1.00	0.99	1.00	173
hindi	1.00	1.00	1.00	274
iyon	1.00	0.98	0.99	169
dito	0.99	1.00	0.99	173
Saan	1.00	1.00	1.00	168
Na	1.00	0.99	1.00	602
Sila	1.00	1.00	1.00	192
Nung	1.00	1.00	1.00	172
Ka	1.00	1.00	1.00	264
pero	1.00	1.00	1.00	177
po	1.00	1.00	1.00	287
sa	1.00	1.00	1.00	656
para	1.00	1.00	1.00	209
isang	1.00	0.99	1.00	192
pang	1.00	1.00	1.00	175
kaya	1.00	0.99	1.00	229
yan	1.00	0.99	1.00	282
buwan	1.00	1.00	1.00	204
ni	1.00	1.00	1.00	350
nasa	1.00	0.98	0.99	177
lang	1.00	1.00	1.00	276
accuracy			1.00	9712
macro avg	1.00	1.00	1.00	9712
weighted avg	1.00	1.00	1.00	9712

Table 4 presents the training classification results. In the training results, the model achieves 85% accuracy.

Table 4: Testing Classification Results

Class	Precision	Recall	F1-score	Support
tayo	0.91	0.92	0.91	63
may	0.85	0.86	0.86	86
yung	0.96	0.88	0.91	49
kailan	0.79	0.77	0.78	39
basa	0.91	0.98	0.94	41
tapos	0.89	0.82	0.85	38
nang	0.74	0.76	0.75	49
ang	0.72	0.80	0.76	59
kung	0.83	0.83	0.83	54
tao	0.87	0.77	0.82	62
tula	0.83	0.74	0.78	54
ngayon	0.82	0.76	0.79	55
Ko	0.82	0.82	0.82	49
wala	0.69	0.73	0.70	51
pamilya	0.88	0.93	0.90	40
tama	0.80	0.85	0.83	39
ikaw	0.80	0.74	0.76	53
siya	0.82	0.84	0.83	73
naman	0.83	0.87	0.85	45
ano	0.88	0.90	0.89	42
maganda	0.88	0.88	0.88	42
hindi	0.92	0.90	0.91	67
iyon	0.90	0.88	0.89	41
dito	0.95	1.00	0.98	42
saan	0.78	0.70	0.74	40
na	0.87	0.86	0.86	149
sila	0.85	0.85	0.85	46
nung	0.74	0.85	0.80	41
ka	0.83	0.77	0.80	64
pero	0.93	0.95	0.94	43
po	0.84	0.91	0.88	70
sa	0.88	0.84	0.86	161
para	0.84	0.94	0.89	51
isang	0.87	0.85	0.86	46
pang	0.85	0.79	0.81	42
kaya	0.89	0.91	0.90	55
yan	0.86	0.86	0.86	69
buwan	0.89	0.84	0.86	49
ni	0.88	0.93	0.90	86
nasa	0.82	0.86	0.84	43
lang	0.76	0.81	0.78	67
accuracy			0.85	2355
macro avg	0.85	0.85	0.85	2355
weighted avg	0.85	0.85	0.85	2355

To optimize the parameter values in the CNN model, loss function was used to indicate how well those parameters accomplish the prediction of Filipino words. The desirable value for this model is to achieve 100% accuracy in prediction. The loss tables per epoch for both training and test/validation set were reflected in Table 5.

Table 5: Loss Per Epoch

Epoch	Training Loss	Testing Loss
1	0.34	0.87
2	0.26	0.80
3	0.15	0.87
4	0.12	0.95
5	0.10	0.90
6	0.08	1.07
7	0.07	1.01
8	0.02	1.06

Figure 9 shows the accuracy graph of the model for every epoch. The graph's training accuracy started at 89.85%, and validation accuracy of 79.32% due to a large amount of batch size indicated (64). Steps per epoch in which the specified value is 2000 that resulted in faster convergence of the model, but the model suffered from overfitting because of the limited amount of data.

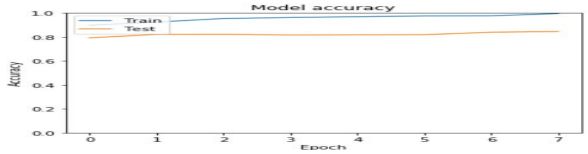

Figure 9: Accuracy Graph

Figure 10 shows the graph for the loss of the model for every epoch. The training loss started at 0.342 and ended at 0.018 while the validation loss started at 0.873 and ended at 1.060. It is clearly indicated that the model learns from the training data well in which it cannot correctly identify on the validation data.

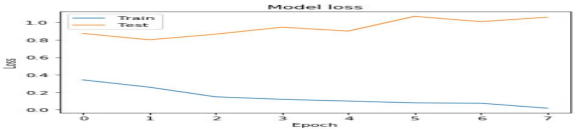

Figure 10. Loss Graph

5.CONCLUSION

In this paper, a convolutional neural network was used to recognize Filipino words automatically. The accuracy of the CNN model for the training set reached 99.6%, while for the validation, the accuracy was 84.7%. It was observed that the CNN model has overfitting, which is a common problem in any CNN model. It can be resolved by having more data sets that can be used to train the model.

It is recommended that the data set used in this study be augmented by considering other sources like those which are publicly available like YouTube, although preprocessing of these would require extensive speech segmentation. Applying Generative Adversarial Network (GAN) to increase the number of data set is also recommended since GAN applies two CNNs to create data set from an original data set. With an increased number of data set, the CNN model can be further trained to recognize more Filipino words with higher accuracy.

ACKNOWLEDGMENT

The authors acknowledge various organization/people for their invaluable contribution for the completion of this paper: the Electrical and Electronics Engineering Institute of the University of the Philippines Diliman for providing the data set; and the people from the Technological Institute of the Philippines Quezon City specifically the Office of the Vice President for Academic Affairs, Office of the Dean of Information Technology Education, colleagues and students of the College of IT Education, and, the Department of Computer Engineering, for all the help in completing this paper.

REFERENCES

1. P. S. D. Nagajyothi, “Speech Recognition Using Convolutional Neural Networks,” vol. 7, pp. 133–137, 2018. https://doi.org/10.14419/ijet.v7i4.6.20449
2. “What is a Deep Neural Network? - Definition from Techopedia.” [Online]. Available: https://www.techopedia.com/definition/32902/deep-neural-network. [Accessed: 05-Dec-2019].
3. “Neural Network Definition.” [Online]. Available: https://www.investopedia.com/terms/n/neuralnetwork.asp. [Accessed: 05-Dec-2019].
4. D. Guiming, W. Xia, W. Guangyan, Z. Yan, and L. Dan, “Speech recognition based on convolutional neural networks,” *2016 IEEE Int. Conf. Signal Image Process. ICSIP 2016*, pp. 708–711, 2017.
5. “What is a Convolutional Neural Network (CNN)? - Definition from Techopedia.” [Online]. Available: https://www.techopedia.com/definition/32731/convolutional-neural-network-cnn. [Accessed: 05-Dec-2019].
6. A. Bhandare, M. Bhide, P. Gokhale, and R. Chandavarkar, “Applications of Convolutional Neural Networks,” *Int. J. Comput. Sci. Inf. Technol.*, vol. 7, no. 5, pp. 2206–2215, 2016.
7. N. Takayashi, M. Gygli, B. Pfiste, and L. Van Gool, “Deep Convolutional Neural Networks and Data Augmentation for Acoustic Event Recognition,” *IEEE Signal Process. Lett.*, vol. 24, no. 3, pp. 279–283, 2017.
8. Y. Qian and P. C. Woodland, “Very deep convolutional neural networks for robust speech recognition,” *2016 IEEE Work. Spok. Lang. Technol. SLT 2016 - Proc.*, vol. 1, no. 16, pp. 481–488, 2017. https://doi.org/10.1109/SLT.2016.7846307
9. S. K. Gouda, S. Kanetkar, D. Harrison, and M. K. Warmuth, “Speech Recognition: Keyword Spotting Through Image Recognition,” 2018.
10. A. C. Fajardo and Y. Kim, “Test Of Vowels In Speech Recognition Using Continuous Density Hidden Markov Model And Development Of Phonetically Balanced-Words In The Filipino Language,” *Balk. Reg. Conf. Eng. Bus. Educ.*, vol. 1, no. 1, pp. 531–536, 2014.
11. A. C. Farjardo and Y.-J. Kim, “Development of Filipino Phonetically-balanced Words and phoneme-level Hmms,” *Ijarcce*, vol. 4, no. 1, pp. 1–6, 2015.
12. F. Ang, Y. Miyanaga, R. C. Guevara, R. Cajote, and M. G. A. Bayona, “Open domain continuous filipino speech recognition with code-switching,” *Proc. - IEEE Int. Symp. Circuits Syst.*, pp. 2301–2304, 2014. https://doi.org/10.1109/ISCAS.2014.6865631
13. J. L. Bautista and Y. Kim, “An Automatic Speech Recognition for the Filipino Language using the HTK System,” pp. 542–547, 2013.
14. M. A. C. Bermudo, O. J. Abesamis, and J. Addawe, “Isolated Tagalog Vowel Recognition using the Wall Street Journal Speech Corpus and Hidden Markov Models,” *C. 190 Spec. Probl. Dep. Math. Comput. Sci.*, pp. 1–10, 2010.
15. B. S. Hijazi, R. Kumar, C. Rowen, and I. P. Group, “Using Convolutional Neural Networks for Image Recognition: What Is a CNN?,” pp. 1–12.
16. S. Lai, L. Xu, K. Liu, and J. Zhao, “Recurrent convolutional neural networks for text classification,” *Proc. 29th AAAI Conf. Artif. Intell.*, pp. 2267–2273, 2015.
17. D. Motta *et al.*, “Application of convolutional neural networks for classification of adult mosquitoes in the field,” *PLoS One*, vol. 14, no. 1, pp. 1–18, 2019.
18. R. Yamashita, M. Nishio, R. K. G. Do, and K. Togashi, “Convolutional neural networks: an overview and application in radiology,” *Insights Imaging*, vol. 9, no. 4, pp. 611–629, 2018.
19. E. Schuettpelz *et al.*, “Applications of deep convolutional neural networks to digitized natural history collections,” *Biodivers. Data J.*, vol. 5, p. e21139, 2017. https://doi.org/10.3897/BDJ.5.e21139
20. “Automatic Speech Recognition - an overview | ScienceDirect Topics.” [Online]. Available: https://www.sciencedirect.com/topics/engineering/automatic-speech-recognition. [Accessed: 07-

Dec-2019].

21. D. Palaz, M. Magimai-Doss, and R. Collobert, "Analysis of CNN-based speech recognition system using raw speech as input," *Proc. Annu. Conf. Int. Speech Commun. Assoc. INTERSPEECH*, vol. 2015-Janua, pp. 11–15, 2015.
22. P. Jansson, "Single-word speech recognition with Convolutional Neural Networks on raw waveforms," 2018.
23. "Speech Recognition Lecture 12: An Overview of Speech Recognition."
24. "Practical Cryptography." [Online]. Available: http://practicalcryptography.com/miscellaneous/machine-learning/guide-mel-frequency-cepstral-coefficients-mfccs/. [Accessed: 07-Dec-2019].
25. "Neural Network Architecture - an overview | ScienceDirect Topics." [Online]. Available: https://www.sciencedirect.com/topics/engineering/neural-network-architecture. [Accessed: 07-Dec-2019].
26. "[1602.07360] SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5MB model size." [Online]. Available: https://arxiv.org/abs/1602.07360. [Accessed: 07-Dec-2019].
27. "Gentle Introduction to the Adam Optimization Algorithm for Deep Learning." [Online]. Available: https://machinelearningmastery.com/adam-optimization-algorithm-for-deep-learning/. [Accessed: 07-Dec-2019].
28. "Difference Between a Batch and an Epoch in a Neural Network." [Online]. Available: https://machinelearningmastery.com/difference-between-a-batch-and-an-epoch/. [Accessed: 07-Dec-2019].