Convolutional Neural Networks for Isolated Word Speech Recognition

Wityi Mon[1], Phyu Phyu Tar[2]
Department of Information Science[1,2]
University of Technology (Yatarnarpon Cyber City)[1,2]
wityimon.wym.utycc@gmail.com[1], thitagu7@gmail.com[2]

Abstract

Automatic Speech Recognition (ASR) is a technology that allows spoken input into system. There are two main types of speech recognition: (1) isolated word and (2) continuous speech recognition. An isolated-word system operates on single words at a time requiring a pause between saying each word. A continuous speech system operates on speech in which words are connected together and are not separated by pauses. This paper mainly focuses on isolated word speech recognition using two types of convolutional neural networks (CNN). The first one is traditional CNN using Mel Frequency Cepstral Coefficients (MFCC) and the second one is Alex-Net with the use of spectrogram which is converted from data points from audio signal. In this research work, isolated words of 20 English singer names audio files dataset is created for the classification task of two CNNs. The applied traditional CNN generates overall accuracy of over 90% and Alex-Net gives the accuracy of over 75%.

Keywords- Automatic Speech Recognition (ASR), Mel Frequency Cepstral Coefficient (MFCC), Convolutional Neural Networks (CNN), Spectrogram

1. Introduction

Speech is the physical production of sound using tongue, lips, palate and respiratory system to communicate ideas. Automatic speech recognition (ASR) is the ability of a machine or program to identify words and phrases in spoken language and convert them to a machine-readable format. There are plenty of applications and areas where speech recognition is used, including the military, as an aid for impaired persons such as a person with crippled or no hands or fingers, in the medical field, in robotics, etc. Speech is also faster than typing on a keypad and more expressive than clicking on a menu item. ASR system typically comes with two main steps: (1) Feature Extraction and (2) Classification (Pattern Matching). Feature extraction refers to the procedure of transforming the speech signal into a number of parameters, while pattern matching is a task of obtaining parameter sets from memory which closely matches the parameter set extracted from the input speech signal [8]. A fundamental problem of speech recognition is a reasonable selection of features. There are many speech feature extraction methods such as Linear Predictive Cepstral Coefficients (LPCC); Perceptual Linear Prediction (PLP); Mel-Frequency Cepstral Coefficients (MFCC); and Neural Predictive Coding (NPC) [5][6][7]. MFCC is a popular technique because it is based on the known variation of the human ear's critical frequency bandwidth. MFCC coefficients are obtained by de-correlating the output log energies of a filter bank which consists of triangular filters, linearly spaced on the Mel frequency scale [4].

Deep neural network (DNN) based acoustic models have been shown by many groups [9][10][11][12][13] to outperform the conventional Gaussian mixture model (GMM) on many automatic speech recognition (ASR) tasks. Recently, several sites have reported some successful results using deep convolutional neural networks (CNNs) as opposed to standard fully connected DNNs. There are two main properties of CNNs that can potentially improve speech recognition performance. First, pooling at a local frequency region makes CNNs more robust to slight formant shifts due to speaker or speaking style variation. Second, sparse local connections of the convolutional layer require far fewer parameters to extract low-level features which avoid over-fitting [2]. After extracting characteristic features from speech signal, recognition stage comes into the work. In this work, CNN is trained to recognize predefined keywords from short utterances where the label for each utterance is the keyword. CNNs have achieved state of the art results in various tasks in computer vision such as object detection and image segmentation, as well as machine translation, text classification and speech recognition.

In this work, two approaches of CNN are applied. The first approach is CNN model based on MFCC feature extraction and the second one is AlexNet model based on spectrogram.

2. Literature Review

Xuejiao Li and Zixuan Zhou [16] discussed that an accurate, small-footprint, low-latency speech command recognition system that is capable of detecting predefined keywords. The motivation of the paper is to build a keyword spotting system that is capable of detecting predefined keywords and helps device to interact differently based on what the command asks for. Using the Speech Commands Dataset provided by Google's TensorFlow and AIY teams, the system is implemented by different architectures using different machine learning algorithms. The models used in the paper are Vanilla Single-Layer softmax model, Deep

Neural Network and Convolutional Neural Network. The Convolutional Neural Network proves to outperform the other two models and can achieve great accuracy for 6 labels.

Anjali Pahwa and Gaurav Aggarwal proposed [17] that speech recognition system for gender recognition. Gender recognition is an important component for the application embedding speech recognition as it reduces the computational complexity for the further processing in these applications. In this paper, a gender recognition system is built in Hindi vowels to determine the gender using speech. The speech samples of 46 speakers are taken and pre-processed the signals and extracted there MFCC, first and second derivatives. The system is trained using combined Support Vector Machine (SVM) and neural network classifier using the process of stacking of classifiers in a tool named RapidMiner Studio. The speech database prepared contains two datasets with or without the first value of the Mel coefficients obtained. The model has been trained and tested using the speech samples collected from the real time environment that has a considerable amount of background noise while recording.

3. Isolated Word Speech Recognition

Experiments have been performed on both MFCC and spectrogram independently as well as together to achieve better accuracy. For different inputs, traditional CNN and Alex-Net are used. The detail of these methods and architectures are discussed in the following section.

3.1. Voice Activity Detection

Voice Activity Detection (VAD) is a very important front end processing in all speech and audio processing applications. VAD which is also called detecting silence parts of a speech or audio signal is a very critical problem in many speech/audio applications including speech coding, speech recognition, speech enhancement and audio indexing [18]. In VAD process, energy on each frame is used as feature. The full-band energy measure calculates the energy of the incoming frames. This energy, E_j is given as in (1).

$$E_j=\frac{1}{N}\sum_{i=(j-1)N+1}^{jN} x^2(i) \qquad (1)$$

Where, E_j is the energy of the j-th frame, x(i) is the i-th sample of speech and the length of the frame is N samples. Calculating the threshold value is very important as it estimates the background noise. In this study, it is assumed that the initial 100ms does not contain any speech. Therefore, the mean energy of the initial 100ms is calculated according to (2).

$$E_r=\frac{1}{v}\sum_{m=0}^{v} E_m \qquad (2)$$

Where, E_r is the threshold value, v is the number of frames whose individual size is 480 samples which is equivalent to 30ms sampled at 16000Hz frequency. The speech signal is divided into frames of 30ms duration at 16000Hz sampling frequency. This corresponds to 480 samples per frame. The energy of the incoming frame is calculated according to (1) and compared to the estimated threshold. If the energy of the frame is greater than the threshold, the frame is judged as a voice frame, otherwise, unvoiced frame. VAD is performed before MFCC process for CNN model and before visualizing spectrogram for AlexNet model. Sample VAD process performed on speech signal is shown in Figure1.

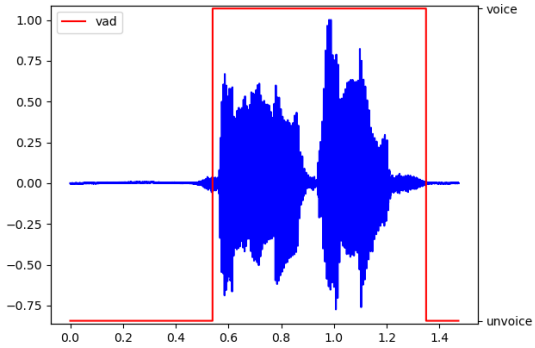

Figure 1. Sample VAD Performed on Speech Signal

3.2. Model 1: CNN model with MFCC

The extraction of the best parametric representation of acoustic signals is an important task to produce a better recognition performance. The efficiency of this phase is important for the next phase since it affects its behavior [1]. Feature extraction for the first model is done using MFCC. One of the most dominant feature extraction methods in ASR is Mel Frequency Cepstral Coefficients (MFCC). MFCC is a representation of the real cepstral of a windowed short-time signal derived from the Fast Fourier Transform (FFT) of that signal [3].

In this research work, VAD is firstly performed on input speech signal. After VAD, the output from VAD is then processed for feature extraction using MFCC. MFCC mainly works with seven steps.

1) Pre-emphasis filtering is used to obtain a smoother spectral form of speech signal frequency.
2) Framing is required as speech is a time-varying signal but when it is examined over a sufficiently short period of time; its properties are fairly stationary [14]. For analyze the speech, it get divided into frames of 30ms where it supposed to be stationary speech signal.
3) In windowing step, each frame has to be multiplied by a hamming window in order to keep the continuity of the first and the last points in the frame. The window size is 10ms.
4) Fast Fourier Transform (FFT) is usually performed to obtain the magnitude frequency response of each frame.

5) In Mel-Filterbank step, the magnitude frequency response is multipled by a set of triangular bandpass filters to get the log energy of each triangular bandpass filter. Mel-frequency is proportional to the logarithm of the linear frequency, reflecting similar effects in human's subjective aural perception.
6) Logarithm values are obtained by converting FFT values into one value and reduce the value of the mel filterbank by substituting each base log value.
7) The log Mel spectrum is converted into time domain using Discrete Cosine Transform (DCT). The result of the conversion is called MFCC coefficients [17]. DCT is calculated to obtain the first 13-dimensional coefficients vector from 40 log mel filters output.

Delta coefficients which are differential and acceleration coefficients are appended to original MFCC coefficients. The MFCC feature vector describes only the power spectral envelope of a single frame. Calculating the MFCC trajectories and appending them to the original feature vector increases ASR performance by quite a bit. 13 MFCC coefficients are appended by 13 delta and 13 delta-delta coefficients. Therefore, 39 coefficients are used for training CNN model.

A convolutional neural network is implemented using two convolution layers, one pooling layer, one fully connected layer and the output layer. Convolution layer uses multiple convolution filters to obtain different features from the input matrix. After convolution, the pooling layer serves two main purposes. The first is that the amount of parameters or weight is reduced by 65%, thus lessening the computational cost. The second is that it controls the overfitting. The dropout layer is added into the network to control overfitting as a regularization technique. The fully connected layer is just like a traditional neural network. Convolution layer and subsampling layer do feature extraction, whereas fully connected layers do classification based on the features by the previous layer. In the output layer, softmax activation function is used because of multiclass classification problem. In this study, Rectified Linear Unit (ReLU) is used as an activation function. Activation functions are non-linearities used between convolutional layers so that the neural network can model more complex than linear data. A common activation function is the Rectified Linear Unit (ReLU), which is a function that takes input x and returns max (0, x). The ReLU and variants of it such as ReLU have mostly replaced the older activation functions sigmoid and tanh. This is due to much greater training efficiency compared to the older activation functions [15].

In this study, the CNN model is implemented with the input size of [39x40] dimensional matrix. Loss function, Cross-Entropy also referred to as Logarithmic loss, is used for predicting the likelihood of an example belonging to each class. Adam which stands for Adaptive Momentum is used for training optimization. Adam is a very powerful and fast optimizer to use error correction to solve cold start problem in weighted average calculation. The epoch size for the CNN model is 100 and the batch size is 64.

In the applied CNN model, the input size is [39x40]. The first convolutional layer is 32 [2x2] kernels. It is followed by the second convolutional layer with 64 [2x2] kernels. The features generated in the convolution layers are fed to max-pool layer with the size of [2x2]. To reduce overfitting, the dropout layer is added. The extracted features are flattened and fed to the fully connected (FC) layer. Another dropout layer is also added. Finally, an output softmax layer is used to perform the twenty class classification. The dropout probability is 0.25. The parameters used for the applied CNN model are mentioned in Table1.

Table 1. Parameters Used in the Applied

	Kernel Size	Number of Kernel	Percentage
1st Convolution Layer	2x2	32	-
2nd Convolution Layer	2x2	64	-
Max Pooling Layer	2x2	-	-
Dropout Layer	-	-	25

3.3. Model 2: Alex-Net with Mel-Spectrogram

In this model, spectrogram is used as input to the Alex-Net. The spectrogram is generated by STFT (Short Term Fourier Transform) of windowed audio or speech signal. The audio is sampled at 16000Hz sampling rate. Each frame of audio is windowed using "hann" window of length 2048. 2048 length FFT windows are applied on the windowed audio samples and used 512 as the hop length for the Short-Time Fourier transform (STFT). The computed magnitude spectrogram is mapped to the mel-scale to get mel-spectrogram. Mel-frequency scale emphasizes the low frequency over the high frequency, similar to the human ears' perceptual capability. The sample mel-spectrogram is shown in Figure 2.

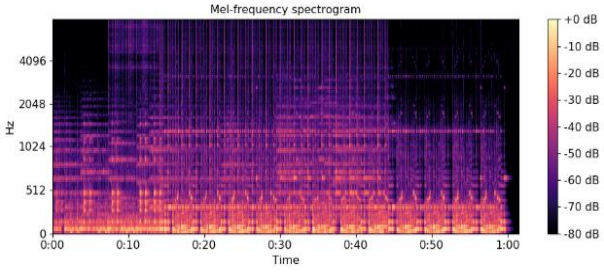

Figure 2. Sample Mel-spectrogram of Speech Signal

AlexNet contains 5 convolutional layers and 3 fully connected layers. Relu is applied after every convolutional and fully connected layer. Dropout is applied before the first and the second fully connected layer [19]. It consisted of 11×11, 5×5, 3×3, convolutions, max pooling, dropout, data augmentation, ReLU activations, SGD with momentum. It attached ReLU activations after every convolutional and fully-connected layer. The first two Convolutional layers are followed by the Overlapping Max Pooling layers. Max Pooling layers are usually used to downsample the width and height of the tensors, keeping the depth same. Overlapping Max Pool layers are similar to the Max Pool layers, except the adjacent windows over which the max is computed overlap each other. Pooling windows of size 3×3 with a stride of 2 are used between the adjacent windows.

The third, fourth and fifth convolutional layers are connected directly. The fifth convolutional layer is followed by an Overlapping Max Pooling layer, the output of which goes into a series of two fully connected layers. The second fully connected layer feeds into a softmax classifier with 20 class labels. The epoch size is 30.

The input shape of the spectrogram Image is [227x 227x3]. Spectrogram is a time-frequency representation of speech. 2D convolution filter captures a 2-dimensional feature from the spectrogram.

3.4. Datasets

The dataset of audio files of each English singer's name are created by uttering speech of each singer name. All the speech samples collected are 1.5 seconds long utterances. The data which involve singer and group names of English were collected from **20** speakers. Target keywords are "Adele", "Akon", "Beyonce", "Bruno Mars", "Celine Dion", "Charlie Puth", "Demi Lovato", "Drake", "Eminem", "John Legend", "Justin Bieber", "Katy Perry", "Lady Gaga", "Maroon 5", "Michael Jackson", "Miley Cyrus", "Rihanna", "Selena Gomez", "Taylor Swift" and "Westlife". The sampling rate of each speech sample is 44100 Hz. The sampling is downsampled to 16000 Hz for fast computation. The speech was recorded using the pyaudio library supported by python. The speech signals are recorded as the format of .wav files. The dataset is split into 80% of training set and 20% of testing set. The size of the dataset is 5,548. The audio files are collected in a background noise-free studio room with a mono microphone. The microphone brand is Remax (KO2).

4. Experiments

The experiments are set up using Python programming language and Keras deep learning framework. As system performance, precision, recall and accuracy are calculated. Precision means the percentage of the predicted results which are relevant. On the other hand, recall refers to the percentage of total relevant results correctly classified by the model. The formulas of how to calculate precision, recall and the percentage of accuracy for each class are as shown below.

$$\text{precision}=\frac{TP}{TP+FP}$$

$$\text{recall}=\frac{TP}{TP+FN}$$

$$\text{accuracy} = \frac{TP+TN}{TP+TN+FP+FN} * 100$$

Where, TP=True Positive, TN=True Negative, FP=False Positive and FN=False Negative.

The testing dataset is created in two types. The first testing dataset is created from the **10** dependent speakers in the training dataset but in a different location with background noise. The second one is created from the **10** independent speakers in different locations of training dataset. All the speakers are non-native speakers for both training and testing dataset. The same numbers of data files are tested on both CNN and AlexNet models.

The precision and recall values are calculated based on the confusion matrix. There are 20 classes for recognizing stage. Therefore, there is not enough space to fit the 20x20 confusion matrix. Example values of TP, TN, FP and FN for two classes ("Adele" and "Selena Gomez") are mentioned. After testing speaker dependent testing dataset on the CNN model, TP, TN, FP and FN values are described in Table 2. The testing dataset size is 1501. The number of audio files tested for "Adele" and "Selena Gomez" classes is 75.

Table 2. Example TP, TN, TP and FN Values for Two Classes

	Adele	Selena Gomez
TP	69	72
TN	1455	1424
FP	17	2
FN	6	3

$$\text{precision (Adele)} = \frac{69}{69+17} = \frac{69}{86} = 0.80$$

$$\text{recall (Adele)} = \frac{69}{69+6} = \frac{69}{75} = 0.92$$

$$\text{precision (Selena Gomez)} = \frac{72}{72+2} = \frac{72}{74} = 0.97$$

$$\text{recall (Selena Gomez)} = \frac{72}{72+3} = \frac{72}{75} = 0.96$$

The precision and recall values for the other 18 classes are calculated as "Adele" and "Selena Gomez". These values are illustrated in the following figures. The precision and recall values on the testing dataset for the CNN model are shown in Figure 3 and 4.

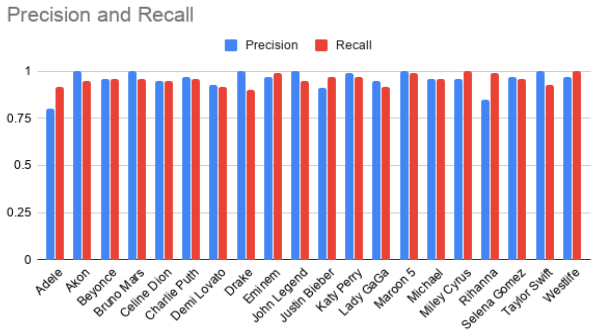

Figure 3. Precision and Recall on Speaker Dependent Testing Dataset for CNN Model

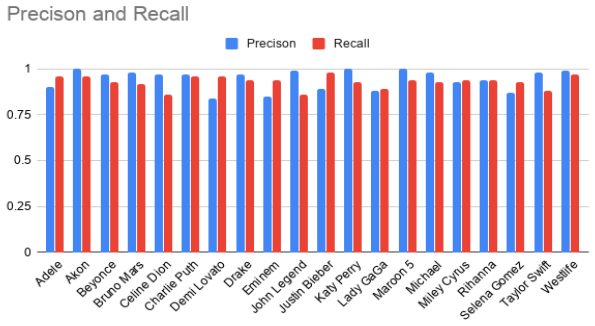

Figure 4. Precision and Recall on Speaker Independent Testing Dataset for CNN Model

The precision and recall values for the AlexNet model are shown in Figure 5 and 6.

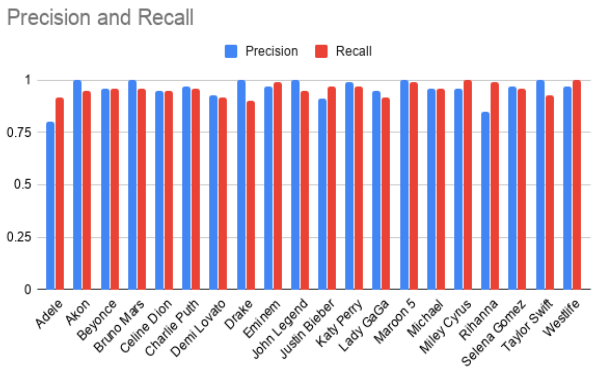

Figure 5. Precision and Recall on Speaker Dependent Testing Dataset for AlexNet Model

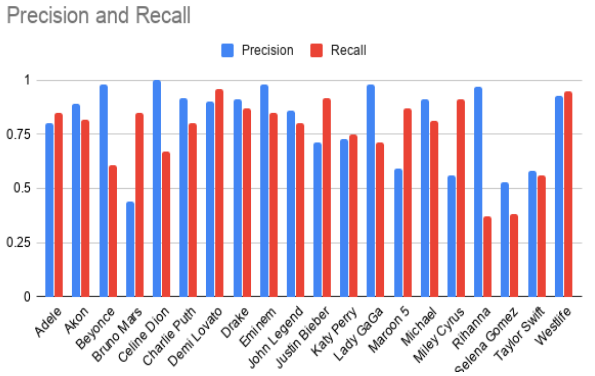

Figure 6. Precision and Recall on Speaker Independent Testing Dataset for AlexNet Model

The overall accuracy is calculated by averaging the accuracy of each class. After training the models, the prediction is performed on both testing dataset to obtain testing accuracy. The number of testing dataset size for the speaker dependent condition is 1501 for both CNN and AlexNet models. For speaker-independent case, there are 1814 audio files for both models. Training, validation and testing accuracy on speaker dependent testing dataset is shown in Table 3.

Table 3. Training, Validation and Testing Accuracy for Speaker Dependent Testing Dataset

	Training Accuracy (%)	Validation Accuracy (%)	Testing Accuracy (%)	Testing Data Count
CNN	99.11	98.38	95.00	1501
AlexNet	99.26	93.96	80.41	1501

Training, validation and testing accuracy on speaker independent testing dataset is shown in Table 4.

Table 4. Training, Validation and Testing Accuracy for Speaker Independent Testing Dataset

	Training Accuracy (%)	Validation Accuracy (%)	Testing Accuracy (%)	Testing Data Count
CNN	99.11	98.38	93.00	1814
AlexNet	99.26	93.96	77.01	1814

MFCC based model generates over 90% on both speaker dependent and independent testing dataset. Spectrogram based approaches perform better on speaker-dependent testing dataset than speaker-independent testing dataset. Therefore, the MFCC based CNN model performs better than on spectrogram based AlexNet model accordingly to the experimental results.

5. Conclusions and Further Extension

In this paper, MFCC based Convolutional Neural Network (CNN) and spectrogram based AlexNet model are used for isolated word speech recognition. The experimental results show that the CNN model gives precision and recall values above 0.75. It means the applied CNN model can generalize well on the testing data. In the AlexNet model, there are some precision and recall values under 0.50 which shows the model cannot well generalize on the testing data for some of the classes. The names of English singers are focused as speech samples with the aim of using voice interfaces in music application which can be used in the driving situation and for people with disabilities to type the keyboard. The purpose of this study is to apply deep learning approach in ASR technology and to demonstrate the idea of isolated word speech recognition with Graphical User Interface (GUI) to be useful for intended users. The data used in this study is limited to 20 target classes of English singer and group names. The experiment results show that both CNN and AlexNet models achieve great accuracy on validation

set but good accuracy on testing dataset. More training data are needed to get great accuracy on the testing dataset especially on speaker-independent situation.

6. References

[1] M.D. Lindasalwa, B.G. Mumtaj and I. Elamvazuthi, "Voice Recognition Algorithms Using Mel Frequency Cepstral Coefficient (MFCC) and Dynamic Time Wraping (DTW) Techniques, Journal of Computing, Volume 2, Issue 3, March 2010, pp. 138-143.

[2] H. Jui-Ting, L.Jinyu and G. Yifan , "An Analysis of Convolutional Neural Network for Speech Recogntion", IEEE International Conference on Acoustics, Speech and Signal Processing, 2015, pp. 4989-4993

[3] P. Vishal and K.A. Rajesh, "Convolutional Neural Networks for Raw Speech Recognition", 2018.

[4] H. Xiaohui, Z.Haolan and Z. Lvjun , "Isolated Word Speech Recognition System Based On FPGA", Journal of Computers, Volume 8, December 2013, pp. 3216-3211

[5] S. Ahmed, M. Ejaz andK. Khawar, "Speaker verification using boosted cepstral features with gaussian distributions," IEEE International Multitopic Conference, 2007. INMIC 2007, 2007 pp.1 – 5.

[6] K.P. Anup, D. Dipankar and Md. Mustafa Kamal, "Bangla speech recognition system using lpc and ann," Seventh International Conference on Advances in Pattern Recognition, 2009, pp. 171 – 174.

[7] C. Charbuillet, B. Gas, M. Chetouani and J. L. Zarader,"Complementary features for speaker verification based on genetic algorithms," IEEE International Conference on Acoustics, Speech and Signal Processing, 2007, pp. 285-288.

[8] S.A. MAJEED, H. HUSAIN, S.A. SAMAD and T.F. IDBEAA, "MEL FREQUENCY CEPSTRAL COEFFICIENTS (MFCC) FEATURE EXTRACTION ENHANCEMENT IN THE APPLICATION OF SPEECH RECOGNITION: A COMPARISON STUDY," Journal of Theoretical and Applied Information Technology, Vol.79, pp. 38-56, Sep. 2015.

[9] D. Yu, L. Deng, and G. Dahl, "Roles of pretraining and fine-tuning in context-dependent DBN-HMMs for real-world speech recognition," in Proc. NIPS Workshop on Deep Learning and Unsupervised Feature Learning, 2010.

[10] T. N. Sainath, B. Kingsbury, B. Ramabhadran, P. Fousek, P. Novak, and A. Mohamed, "Making deep belief networks effective for large vocabulary continuous speech recognition," in Proc. Workshop on Automatic Speech Recognition and Understanding , 2011, pp. 30–35.

[11] G. E. Dahl, D. Yu, L. Deng, and A. Acero, "Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition," IEEE Trans. on Audio, Speech and Language Processing, vol. 20, no. 1, 2012, pp. 30–42.

[12] G. Hinton, L. Deng, D. Yu, G. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. Sainath, and B. Kingsbury, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," IEEE Signal Processing Magazine, vol. 29, no. 6, 2012, pp. 82–97.

[13] L. Deng, J. Li, J. -T. Huang et al. "Recent advances in deep learning for speech research at Microsoft," in Proc. ICASSP, 2013.

[14] D Anggraeni , W S M Sanjaya, M Y S Nurasyidiek and M Munawwaroh, "The Implementation of Speech Recognition using Mel-Frequency Cepstrum Coefficients (MFCC) and Support Vector Machine (SVM) method based on Python to Control Robot Arm," The 2nd Annual Applied Science and Engineering Conference (AASEC), 2017.

[15] J. Patrick, "Single-word speech recognition with Convolutional Neural Networks on raw Waveforms".

[16] Zhou Z. and Li X, Speech Command Recognition with Convolutional Neural Network, 2017.

[17] Pahwa A. and Pggarwal G, Speech Feature Extraction for Gender Recognition, I.J. Image, Graphics and Signal Processing, 9 (2016), pp. 17-25.

[18] Moattar M. and Homayounpour M, A SIMPLE BUT EFFICIENT REAL-TIME VOICE ACTIVITY DETECTION ALGORITHM, 17th European Signal Processing Conference, Glasgow, 8 (2009), pp. 24-28.

[19] Hao Gao, A walk-through of AlexNet, https:// medium. com/ @smallfishbigsea/a-walk-through-of-alexnet -6cbd137a 5637, 2017.

Detecting DDoS Attacks In IoT Sensor Network

Ngu Wah Kyaw[1], Nandar Win Min[2]
Department of Information Science[1,2],
University of Technology (Yatanarpon Cyber City)[1,2]
nguwahkyaw5@gmail.com[1], nandarwinmin@gmail.com[2]

Abstract

Internet of Things (IoT) is compromised of billions of connected devices, gathering and sharing data. This data can be gathered and analysed. This changing service brings new security threads and challenges to various organizations. A distributed denial of service (DDoS) attack attempts to partially/completely shut down the targeted server with a flood of internet traffic. DDoS attack on IoT is launched by using Botnet (Mirai) and can cause significant disruptions, loss of data and information. This paper demonstrates the use of machine learning (ML) methods for detecting three DDoS attacks (udp, tcp, icmp(ping)) attacks that can be launched in IoT networks. An IoT sensor network is setup using sensors/devices to collect data for training ML models. For attack detection, network traffic are evaluated by using five ML algorithms(K-Nearest Neighbors, Gaussian Naive Bayes, Support Vector Machine, Classification and Regression Tree, and Deep Neural Network) and compare the performance of these methods.

Keywords- Cyber Attacks, Distributed Denial of Service (DDoS) Attack, Internet of Things (IoT), Machine Learning (ML).

1. Introduction

Cybercriminals have many different ways of exploiting network vulnerabilities and weak spots in cyber defenses. Considering that the number of devices that use on a daily basis is growing, more avenues of exploitation will be open to cybercriminals unless those pathways are closed. DDoS attacks on IoT networks via botnets have been especially alarming and difficult to counter. By 2020, Gartner predicts the total number of IoT devices will reach 20.4 billion. At the same time, DDoS attacks are on the rise, with Cisco estimating that the number of DDoS attacks exceeding 1 gigabit of traffic per second will soar to 3.1 million by 2021. He also predicts that that in excess of 25 percent of recognized attacks in organization will include the IoT[3].

Considering that IoT devices are used by countless companies across a wide array of industries, from trucking to insurance to communications, it isn't necessarily surprising that IoT devices are being used by criminals to facilitate DDoS attacks. In October 2016, Dyn, a company that controls much of the internet's domain name system (DNS), was under attack by Distributed Denial of Service (DDoS) attacks[7]. This was made possible through the Mirai infection of over 100,000 IoT devices, including IP cameras, DVRs, cable set-top boxes, and printers. As a result, major platforms including Twitter, PayPal, Reddit, Amazon, Airbnb, and Netflix were rendered unavailable to users across the globe, for several hours[8]. Therefore, analysis of sensor data is needed for securing IoT systems. Because of the vast amount of networks and sensing data produced by IoT devices and systems, Machine Learning methods are highly effective in analysis for the security of IoT systems.

This significant challenge drives to analyze the IoT sensor data for identifying and detecting malicious traffic from IoT botnets/attack devices. This system performs data collection, feature pre-processing, and attack detection. Six ML classifiers are utilized and compared for attack detection. Detection was performed at the local networks and at the packet level.

2. Related Work

In [1], nine parameters are selected by exploiting of DDoS attacks properties for proactive detection DDoS attacks. After parameter selection, To distinguish the normal traffic from attack packets, clustering analysis is also performed. They experiment using DARPA 2000 Intrusion Detection Data set for evaluating their methods. They cannot extract two clusters (phase3 and phase4) exactly in simulation.

In [2], HTTP DDoS attacks detection is presented based on Information Theoretic Entropy and Random Forest ensemble learning algorithm. A time-based sliding window is used to measure the entropy of the packet's header features from the incoming network traffic. To assess the proposed methodology, various experiments were performed by using the CIDDS-001 open dataset. The performance of the network entropy estimation algorithm relies mainly on the time window size.

In [5], DOS intrusion was detected by supervised neural network. For experiment, NSLKDD database was used and implementation speed was increased by using CPUs as the Parallelization Technology. DoS attacks have been classified into 7 categories and try to detect each category by one agent which is equipped with IDS.

In [6], The Radial-Basis-Function neural network (RBF-NN) was used by to recognize DDoS attacks from

the normal traffic. RBF-NN detector is a two layer neural network. It uses nine packet parameters, and the frequencies of these parameters are estimated. Based on the frequencies, RBF-NN classifies traffic into attack or normal class.

In [9] Artificial Neural Network (ANN) algorithm was used to detect and mitigate DDoS Attack. DDoS detectors are installed on different networks . Each detector registers the IP address of all neighbouring DDoS detectors to inform and send encrypted message when DDoS attacks are detected.

In [10], Cisco Systems NetFlow and two distinct data mining method are utilized to identify the different types of DDoS attacks. Seven useful features were given by the NetFlow on each data traffic that enters the system. This incorporated the source IP, destination IP, source port, destination port, layer 3 protocol, TOS byte (DSCP) and input logical interface (in Index). So the decision tree algorithm was utilized to automatically choose various features gave by the NetFlow to show the traffic example of the diverse DDoS attack types. The second method is the neural network algorithm.

There are many researches about detecting DDoS attacks in conventional system/IoT sensor network utilizing machine learning methods and open datasets. A considerable lot of these are done utilizing recreation programming for dataset creation and attack recognition. In this detection system, "Detecting DDoS attacks in IoT sensor network utilizing Machine Learning Methods", real sensor gadgets are used to gather sensor data and analyse the collected sensor dataset using six machine learning methods for DDoS attack detection.

3. Threads and Challenges in IoT

One of the many values of the IoT is that it is driving the digitalization of everything-industries and organization. In any case, the IoT carries with it new security dangers because of new innovation applications.

As the tools utilized in attacks become increasingly advanced, Machine Learning (ML) and Artificial Intelligence (A1) will compound attack protection confrontation. Although A1 can be utilized to quickly identify new security threats, it can likewise be utilized to launch attacks. The specialized hindrances for implementing attacks become lower. IOT gadgets, including water meters, vacuum cleaners, refrigerators and street lights, will become potential focuses for attack[4].

- **Scalability:** Billions of internet-enabled devices get connected in a huge network, large volumes of data are needed to be processed. The system that stores, analyses the data from these IoT devices needs to be scalable. In present, the era of IoT evolution everyday objects are connected with each other via Internet. The raw data obtained from these devices need big data analytics and cloud storage for interpretation of useful data[11].
- **Use of weak and default credentials:** Many IoT companies are selling devices and providing consumers default credentials with them — like an admin username. Hackers need just the username and password to attack the device. When they know the username, they carry out brute-force attacks to infect the devices[11].
- **Data protection and security challenges:** In this interconnected world, the protection of data has become really difficult because it gets transferred between multiple devices within a few seconds. One moment, it is stored in mobile, the next minute it is on the web, and then the cloud. All this data is transferred or transmitted over the internet, which can lead to data leak. Not all the devices through which data is being transmitted or received are secure[11].

4. Distributed Denial of Service (DDoS) Attacks

IoT has the same threats with IPv4. Additionally, IoT are subject of the unprecedented threats due to its location that is at a junction point of cyber domain and physical domain. Briefly, the expanding attack surface is a threat. An attack could manipulate the information and that can cause the unintended action in physical domain.

There are many type of attacks to IoT that are physical attacks, DoS, access attacks, attacks on privacy, cybercrimes and destructive attacks. IoT devices are a common thread in large-scale DDoS attacks. The DoS attack typically uses one computer and one Internet connection to flood a targeted system .The DDoS attack uses multiple computers and Internet connections to flood the targeted resource as shown in Figure 1**.** The target of this attack can be server, router, an Internet Service Provider (ISP) or country. DoS attack makes these targets unavailable by intended users mainly by exhausting target devices CPU and memory resources[12].

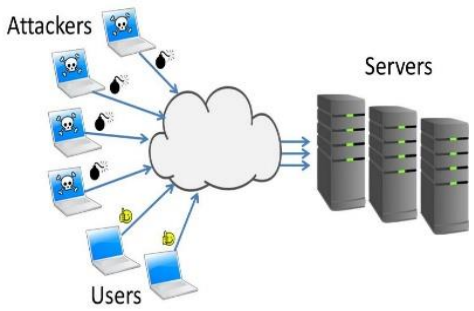

Figure 1. Structure of Distributed Denial of Service

In this system, three most commonly used DDoS attacks (udp, tcp and icmp) are detected.

- **UDP flood -** is any DDoS attack that floods a target with User Datagram Protocol (UDP) packets. The goal of the attack is to flood random ports on a remote host. It may exhaust the target resource which can lead inaccessibility of the target host and its services[12].
- **SYN flood DDoS attack -** exploits a known weakness in the TCP connection sequence (the "three-way handshake").In a SYN flood scenario, the requester sends multiple SYN requests, but either does not respond to the host's SYN-ACK response, or sends the SYN requests from a spoofed IP address[12].
- **ICMP(ping) flood -** overwhelms the target resource with ICMP Echo Request (ping) packets, generally sending packets as fast as possible without waiting for replies. This type of attack can consume both outgoing and incoming bandwidth, since the victim's servers will often attempt to respond with ICMP Echo Reply packets, resulting a significant overall system slowdown[12].

5. DDoS Attack Detection with Machine Learning Methods

Detecting DDoS attacks in the IoT system by using machine learning methods are described in this system. Firstly, how to construct training datasets from captured packets is stated. Then, the ML methods are presented to learn given datasets so as to conduct DDoS attack detection. The architecture of the DDoS detection system is shown in Figure 2.

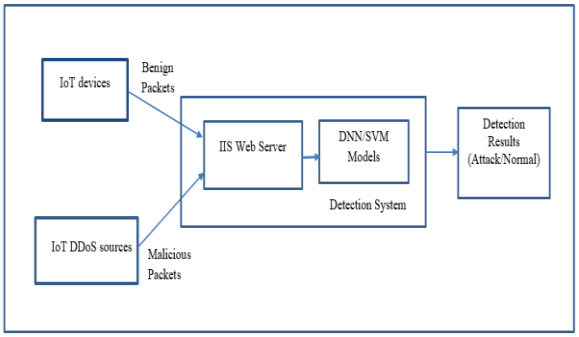

Figure 2. Architecture of the DDoS Detection System

The DDoS attack Detection processes and the interaction/data flow between the processes are shown in Figure 3. The network packets that are captured on the WiFi interface are entered as input. Then, some packets that are not from the source of IoT endpoints (such as phones, pc and other devices which are using the same WiFi interface) are filtered according to the IoT source address. From the captured packets, some features are extracted and pre-calculated as feature preprocessing step. The pre-calculated feature are tested with machine learning methods for classification of attack or normal packets as shown in Figure 3.

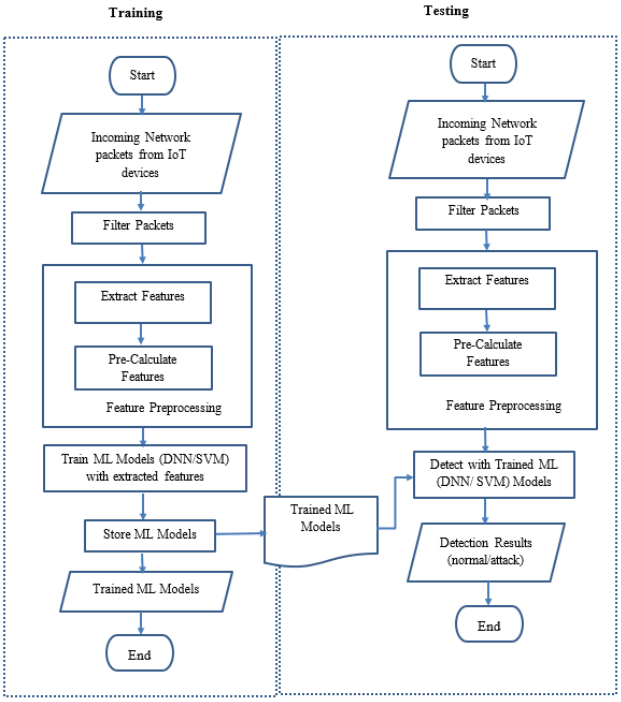

Figure 3. DDoS Detection Processing Flow

5.1. Dataset Construction

Data construction is organized with three steps(traffic capturing, grouping of packets by device and time, and feature extraction).

5.1.1. Packets Capturing: The capturing process records the incoming network packet feature (highest layer, the source IP, source port, destination IP, destination port, transport layer protocol, packet size, and arrival time of all IP packets *)* sent from the IoT systems and attacks sources in fixed time window.

5.1.2. Grouping of Packets by Source Device and Time: Packets from each device are grouped into nonoverlapping time windows by timestamps.

5.1.3. Feature Extraction: The selected features are pre-calculated from the extracted features. These features are proposed by observing the characteristics of DDoS attack packets. These features can be used to recognize and classify incoming attack packets.

- **Number of Packets:** DDoS attacks send a great number of packets to the victim server/website. Therefore, the number of packets when attacking increases in comparison to normal case.
- **Bytes:** Increase in number of bytes demonstrates launching the DDoS attacks.
- **Packet Size Variance:** According to our studies, it is found that attack packets sizes are the same. However, normal packets have different packet sizes even when they belong to the same file.

DDoS packets can be identified by using the Packet Size Variance as in equation 1.

$$Variance = \frac{1}{n}\sum_{k=1}^{n}(Xi - \overline{X}) \quad (1)$$

Where X_i = each input packet size and $\overline{X}$ = the mean value of the input packet size.

- **Arrival Time Variance:** The attacker sends attack packets in the same time span while launching DDoS attack, so arrival time variance will be lower than that in the normal case.
- **Bit Rate:** A very high rate of this feature points out launching DDoS attack.
- **Packet Rate:** This feature shows the packet rate sent from a source address to a destination in a specific time span. Packet rate increases significantly in attack time.

Table 1. summarizes the results of the Feature Extraction process. The results show that proposed features contain considerable information related to the presence of DDoS attack. For example Number of Packets increases in attack time and number of bytes relatively large in attack case. Because of high packet size similarity in attack packets, packet size variance is close to zero and the arrival time is increased. Moreover, packet rate and bit rate in attack time increase in compare to normal time.

Table 1. Feature Pre_Processing Result

ource Ip	Destination Ip	Number of Packets	Byte	Packet Size Variance	Arrival Time Variance	Bit Rate	Packet Rate	Class
192.168.8.109	192.168.8.100	5807	697798	290.21743	68	26403.27	203.9667	Attack
192.168.8.102	192.168.8.100	131	13125	7902.0645	67	444.1667	4.2	Normal
192.168.8.109	192.168.8.100	4541	481201	0	61.115	16037.8	151.3	Attack
192.168.8.105	192.168.8.100	497	20874	0	64.08	696.3	16.567	Attack
192.168.8.101	192.168.8.100	41	4144	8094.6051	63.70571	138.1333	1.333333	Normal
192.168.8.103	192.168.8.100	37	3796	7778.4	72.99403	127.4	1.9	Normal

5.2. Machine Learning Methods

For DDoS attack detection, six packet features (Number of packets, Byte, Packet Size Variance, Arrival Time Variance, Bit Rate and Packet Rate) that are extracted from the incoming traffic are used as input of five ML classifiers. The classifiers analyze these features and output the detection results (normal / attack). The ML classifiers that are used in this system are:

- Gaussian Naive Bayes classifier
- Support Vector Machine with linear kernel (LSVM)
- Deep Neural Network (DNN)
- K-Nearest Neighbors (KNN)
- Classification and Regression Tree (CART)

These machine learning models are implemented using the scikit-learn Python library[13].

5.2.1. Gaussian Naive Bayes Classifier

A Gaussian Naive Bayes algorithm is a special type of Naive Bayes algorithm. It's specifically used when the features have continuous values. It's also assumed that all the features are following a Gaussian distribution i.e, normal distribution.

$$P(x = v|C_k) = \frac{1}{\sqrt{2\pi\sigma_k^2}} e^{-\frac{(v-u_k)^2}{2\sigma_k^2}} \quad (2)$$

Where, x= continuous attribute, u_k = be the mean of the values in x associated with class C_k, $\sigma_k^2 =$ be the Bessel corrected variance of the values in x associated with class C_k, and v = collected some observation value.

5.2.2. Support Vector Machine

Support Vector Machines (SVM) are supervised learning models with associated learning algorithms that analyze data used for classification and regression analysis. It can solve linear and non-linear problems and work well for many practical problems. The main idea is to identify the optimal separating hyperplane which maximizes the margin of the training data.

The decision surface separating the classes is a hyperplane of the form:

$$w^T x + b = 0 \quad (3)$$

Where, w is a weight vector, x is input vector and b is bias.

5.2.3. Deep Neural Network

Deep Neural Network (DNN) is a kind of Neural Networks(NN) which has an input layer, an output layer and many hidden layers in between. The DDoS detection system uses input layer, three hidden layers, three dropout layers, three batch normalization layers and one output layer. The dropout rate is 0.1.

$$Y = \sum(input) * (wieght) + bias \quad (4)$$

$$Output = f(Y) \quad (5)$$

Where f is the activation function.

5.2.4. K-Nearest Neighbors Algorithm

K-nearest neighbors algorithm (KNN) is a supervised machine learning algorithm useful for classification problems. It calculates the distance between the test data and the input and gives the prediction according. In scikit learn KNN, the default distance method is Minkowski distance.

The Minkowski distance between two variables X and Y is defined as

$$distance = \left(\sum_{i=1}^{n} |X_i - Y_i|^p\right)^{1/p} \quad (6)$$

The case where p = 1 is equivalent to the Manhattan distance and the case where p = 2 is equivalent to the Euclidean distance. Although p can be any real value, it is typically set to a value between 1 and 2.

5.2.5. Classification and Regression Tree

Classification and regression trees (CART) is a term used to describe decision tree algorithms that are used for classification and regression learning tasks. It explains how a target variable's values can be predicted based on other values. CART uses the Gini method to create split points.

$$Gini = 1 - \sum_{i=1}^{n} (P_i)^2 \quad (7)$$

where p_i is the probability of an object being classified to a particular class.

5.3. Experiment Setup

An IoT sensor network is set up to collect IoT normal and attack traffic as in Figure 4. Three sensor devices (LDR, DHT11 and Soil Moisture) are configured to send normal data to local IIS server. Two Raspberry Pi 3 are used as the DDoS sources, and a machine running an Internet Information Services (IIS) Web Server as the DDoS victim. These devices are connected via WiFi. To send DDoS traffic, instead of running real Mirai botnet code for avoiding complexity, commonly used DDoS attack types that a Mirai-attack device runs :(TCP, UDP and ICMP(ping)) flood are simulated. The ICMP (ping) attack was inserted by using hping3 tool[14]. The UDP and TCP attacks were inserted using Low Orbit Ion Cannon (LOIC)[15]. To collect normal (non-DoS) and attack traffic, the packets are captured when three sensors are sending normal sensor data and two DDoS sources send attack data to the local website in fixed time window (30 seconds interval).

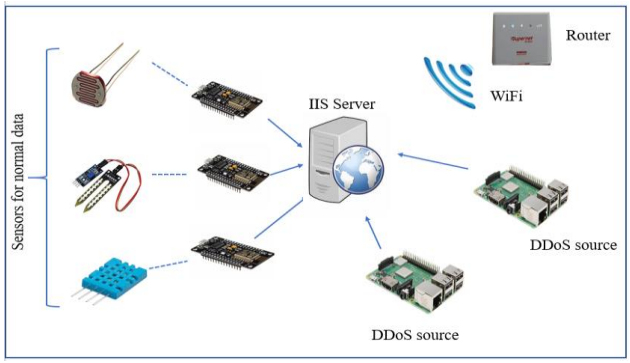

Figure 4. Configuration for the Experiment

5.4. Experimental Results

As system performance, precision, recall and accuracy are calculated. Precision means the percentage of the predicted results which are relevant. On the other hand, recall refers to the percentage of total relevant results correctly classified by the model.

$$\text{precision} = \frac{tp}{tp+fp}$$

$$\text{recall} = \frac{tp}{tp+fn}$$

$$\text{accuracy} = \frac{tp+tn}{tp+tn+fp+fn} * 100$$

Where, tp =true positive, tn=true negative, fp=false positive and fn =false negative.

The terms positive and negative refer to the classifier's prediction, and the terms true and false refers to whether that prediction corresponds to the real label of symbol.

Table 2. shows the results of precision and recall on test set. The data was collected with fixed time window (30 seconds interval). 20594 packets of normal traffic and 430206 packets of attack traffic were captured during five hours. Five classifiers are trained with the captured same data set. 80% of network traffic (combined attack and normal packets) is used as training data and the remaining 20% is used as the testing data.

Table 2. Precision, Recall and Accuracy of Test Data

	Precision	Recall	Accuracy	Detection Time (seconds)
Naïve Bayes	0.96	0.95	0.95	0.0789
SVM	0.96	0.94	0.95	0.0768
DNN	0.96	0.97	0.96	0.0722

KNN	0.97	0.97	0.97	0.0626
CART	0.95	0.99	0.97	0.0651

The overall accuracy is calculated by averaging the accuracy of each class. After training, prediction is performed on test-set to obtain test accuracy.

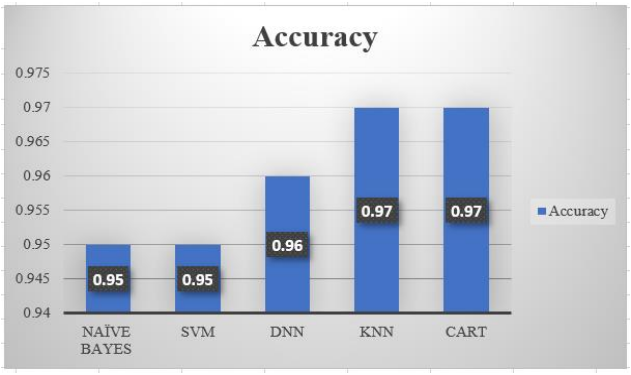

Figure 5. Accuracy Results of the ML Classifiers

Among these classifiers, Naïve Bayes and SVM achieve detection accuracy of 95% and detection time of over 0.076 seconds as shown in Figure 5. The KNN classifier and CART classifier performed surprisingly well. Both methods obtain 97% accuracy for DDoS attack detection. Their detection time is just over 0.06 seconds. DNN model also achieved 96% of detection accuracy and. The results of detection accuracy indicates that the proposed machine learning models can generalize on test set in good accuracy.

6. Conclusion

In this system, five machine learning algorithms are utilized to recognize attack and normal traffic in the IoT sensor network. Six packet features that are pre-processed from the gathered sensor based data are applied for DDoS attack detection. The selection of features are based on the behaviour of DDoS attacks traffic. Five machine learning algorithms can detect well the likelihood that a DDoS attack is ongoing from a set of features extracted from captured network traffic. This system can demonstrate that machine learning can effectively deal with IoT security. In future work, based on the proposed methodology, it is intended to detect DDoS attacks in IoT networks including attacks against Zigbee and Bluetooth-connected devices. The system is expected to predict attack possibilities on other DDoS attacks types that are not stated in this system.

7. Acknowledgement

The authors would like to express most profound gratitude to all teachers from University of Technology (Yatanarpon Cyber City) for their effective guidance and suggestions in carrying in bringing this research to fruition.

8. References

[1] K. Lee, J. Kim, K. Kwon, Y. Han and S. Kim, "*DDoS attack detection method using Cluster Analysis"*, Expert System with Applications, Elsevier, 2008, pp-1659–1665.

[2] M. Idhammad, K. Afdel, and M. Belouch, "Detection System of HTTP DDoS Attacks in a Cloud Environment Based on Information Theoretic Entropy and Random Forest*"*, LabSIV, Department of Computer Science, Faculty of Science, Ibn Zohr University, Agadir, Morocco, June 2018.

[3] Garnar, "Leading to IoT".[Online]. Available: https://www.gartner.com/imagesrv/books/iot/iotEbook_digital.pdf

[4] (2018) "IoT Security White Paper" [Online]. Available: https://www.huawei.com/minisite/iot/img/iot_security_white_paper _2018v2en.pdf

[5] M.M. Javidi and M.H. Nattaj, "A New and Quick Method to Detect DoS Attacks by Neural Networks". Department of Computer Science, Shahid Bahonar University of Kerman, Iran, January 2013.

[6] Dimitris Gavrilis and Evangelos Dermatas "Real-time detection of distributed denial-of-service attacks using RBF networks and statistical features". Computer Networks, 48(2), 235–245, 2005.

[7] S. Hilton. (2016) Dyn analysis summary of friday october 21 attack. Dyn. [Online]. Available: https://dyn.com/blog/ dyn-analysis-summary-f-friday-october-21-attack/.

[8] (2016) Threat advisory: Mirai botnet. Akamai.[Online]. Available https: //www.akamai.com/us/en/multimedia/documents/_state-of-the-internet/_akamai-mirai-botnet-threat-advisory.pdf

[9] A. Saied, R. Overill, and T. Radzik, Neurocomputing, "Detection of known and unknown DDoS attacks using Artificial Neural Networks", vol. 172, no.1, pp. 385-393, Jan 2016.

[10] Mihui Kim, Hyunjung Na, Kijoon Chae, Hyochan Bang and Jungchan Na, "A Combined Data Mining Approach for DDoS Attack Detection, Lecture Notes in Computer Science", Vol. 3090, pp. 943-950, 2004

[11] "Challenges in World of IoT". [Online]. Available: https://www.geeksforgeeks.org/challenges-in-world-of-iot/

[12] "DDoS Attacks". [Online]. Available: https:// www.imperva.com/learn/application-security/ddos-ttacks/

[13] (2017) Scikit learn: Machine learning in python. [Online].Available: http://scikit-learn.org/stable/

[14] (2017) hping3 package description. [Online]. Available: http://tools.kali.org/information-gathering/hping3

[15] "Low Orbit Ion Cannon". [Online]. Available: https://sourceforge.net/projects/loic/

Kitchen Utensil Recognition for Vision Based Domestic Service Robot

Thuzar Tint[1], Tin Myint Naing[2]
University of Technology (Yadanarpon Cyber City)[1,2]
thuzartint1984@gmail.com[1],utinmyintnaing08@gmail.com[2]

Abstract

With the coming of new innovations, service robots are widely utilized in anyplace instead of human works. The research is to develop the domestic service robot based on computer vision. The paper is aimed to implement a kitchen utensil recognition which is one part of the vision based domestic service robot processing. In this kitchen utensil recognition system, there are eight types of kitchen utensil which are whisk, tong, potato peeler, kitchen knife, fork, table knife, spoon and ladle. Firstly, background subtraction algorithm, Otsu thresholding algorithm and morphological approaches are applied to find the region of interest. Convolution Neural Networks (CNNs) is used to recognize the object type of kitchen utensil. According to the experimental results, the proposed object segmentation approach is well suitable in the clear background environment. The proposed system is intended to use in vision based domestic service robot.

Keywords- Kitchen Utensil, Service Robot, Background Subtraction, Thresholding, Convolution Neural Networks (CNN).

1. Introduction

Just as innovation progress, there are growing welfare robots in the human society. Service robots will hugely affect human life. Various service robots have just been created while some are being developed. These service robots can be utilized as assistants in workplaces and homes. Therefore, the research is directed towards the development of the vision based domestic service robot. In the system, the robot will look through the objects which is human's command if human offers order to the robot. After the robot has found the object which is human's command, the robot will pick the recognized object and place it to the target region. In the vision based domestic service robot system, there are three main systems which are speech recognition, object recognition and robot arm control processing. When the human give command to the robot such as "Please take Fork" in Myanmar language, the speech recognition process is performed to recognize the human command. After recognizing the human command, the output speech signal result will be produced from the speech recognition system to the object recognition system. In the object recognition stage, the requested object is searched with the use of the robot webcam. If the requested object is found, the object's coordinates and location will be informed from the object recognition system to the robot arm controlling system for picking and placing to target location. The overall procedure of vision based domestic service robot is shown in Figure 1.

The rest of the paper is organized as follows. Related work in object recognition is revised in section 2. The proposed method of kitchen cutlery recognition is explained in Section 3.The experimental evaluation on real time testing are shown in Section 4, followed by conclusion in Section 5.

2. Related Work

Object Recognition is one of the fundamental challenges in vision based domestic service robot. The vision based object recognition systems are needed to be fast and accurate in order to get real-time information and expose the potential for responsive robotic systems. A large number of techniques have been proposed in the process of object recognition. In [1], Canny algorithm and Artificial Neural Networks (ANN) classifier were used to detect and recognize resistors and capacitors with the use of vision sensors. The classifier yielded an accuracy of 82.7162% (the OSP as it also takes into account the FE accuracy) upon final testing given a cluttered scene.

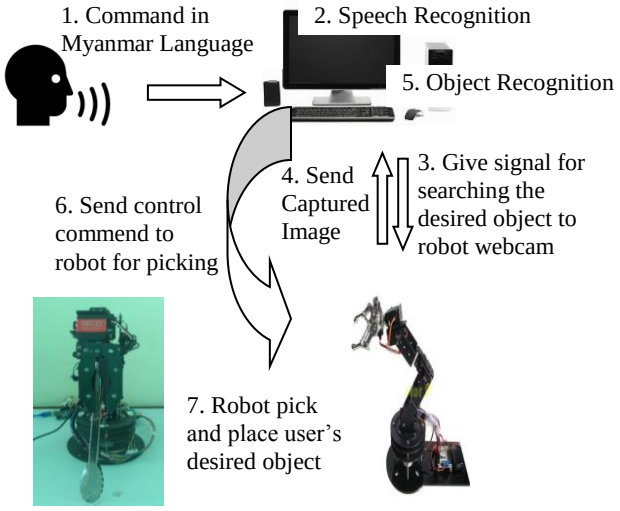

Figure 1. Overall Procedure of Vision based Domestic Service Robot

Rachmawati E. et al. [2] proposed a fruit recognition approach from RGB color image based on color histogram. In the fruit recognition system, KNN classifier was used to classify quantized histogram features of each color channel. Susovan J. et al. [3] proposed automatic fruit recognition from natural images. In the automatic fruit recognition system, images were preprocessed in order to separate the fruit in the foreground from the background. And texture features from Gray-level Co-occurrence Matrix (GLCM) and statistical color features were extracted from the segmented image. Finally, a Support Vector Machine (SVM) classification model was trained using these feature descriptors extracted the training dataset. Finally, the trained SVM model could be used to predict the category for an unlabeled image from the validation set. The method was performing with 83.33% overall accuracy. The [4] paper was proposed to recognize soft drink can objects such as "Shark", "Burn", "Sprite" and "100 Plus". In the system, template matching approach was used for object detection and Adaptive Neural Fuzzy Inference System (ANFIS) was employed based on color features for recognizing the specified object. The experimental result showed that the proposed object detection and recognition system for soft drink can object got 85% accuracy. Though the approach could detect the desired objects in real time, illumination was sensitive to detect the objects when the light was bright or dim.

The paper is intended to detect and recognize kitchen utensil for vision based domestic service robot. In the system, there are eight types of kitchen utensil which are whisk, tong, peeler, kitchen knife, fork, table knife, spoon and ladle.

3. Kitchen Utensil Recognition Methodology

In the kitchen utensil recognition system, there are two main stages. They are:

(1) Segmentation of the objects from the input scene,

(2) Classification of the segmented objects.

The overview system design of the proposed system is shown in Figure 2.

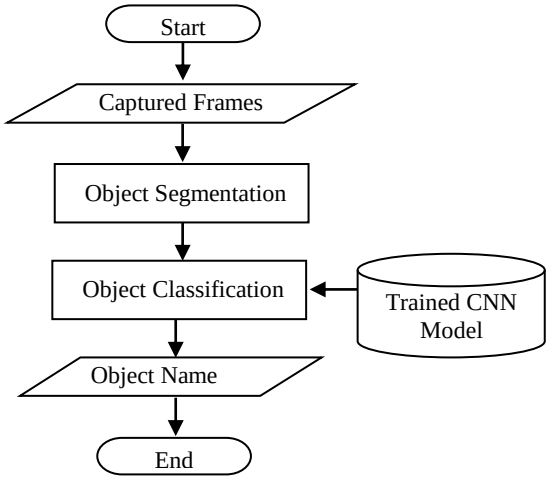

Figure 2. Overview System Design of the Kitchen Utensil Recognition System

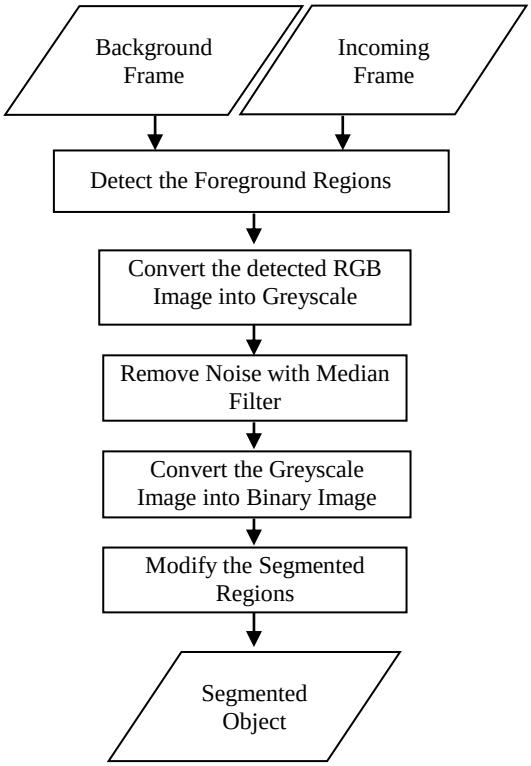

Figure 3. System Flow of the Object Segmentation

3.1. Object Segmentation

In the vision based kitchen utensil recognition system, the object segmentation is the primary step. System flow of the object segmentation is presented in Figure 3. In object segmentation, there are five steps in order to get the segmented object region.

3.1.1. Foreground Regions Detection. In object segmentation, background subtraction algorithm is firstly applied to find the foreground objects' regions. Therefore, the first incoming frame from robot webcam is set as a background frame and then robot webcam is moved to find the objects. And the next incoming frame, *F(x,y)*, is subtracted from the background frame, *B(x,y)*, to search the regions of objects, *D(x,y)*. Background subtraction equation is described in (1).

$$D(x, y) = |F(x, y) - B(x, y)| \quad (1)$$

3.1.2. Grey-scale Conversion. After detecting the foreground objects' regions, the detected RGB frame, is converted into grayscale image using following equation (2).

$$I_{grey-scale}(n,m) = 0.2989 * I_{colour}(n,m,r) + 0.587 * I_{colour}(n,m,g) + 0.114 * I_{colour}(n,m,b) \quad (2)$$

3.1.3. Noise Removal. After converting to the grayscale image, the grayscale image is filtered with 7x7 median filter to remove random noise. The median filtering process is accomplished by sliding a window over the image. The filtered image is completed by assigning the median of the values in the input window, at the location of the center of that window, at the output image.

3.1.4. Image Binarization. To segment the foreground objects, the enhanced grayscale image is converted to binary image by using Otsu's thresholding. The method computes a measure of spread for the pixel levels each side of the threshold, i.e. the pixels that either fall in foreground or background. It finds the optimal threshold value which minimizes the within-class variance of the thresholded black and white pixel.

3.1.5. Modification of the Segmented Regions. After getting the segmented foreground regions, it is necessary to modify them to get the prefect segmented results. Therefore, the morphological processing, dilation is applied to the binary image with a disk-shaped structuring element of radius 10 to connect some foreground pixels. After that, region fill processing is performed to fill holes. Finally, the regions of interest which is only one object are extracted by applying the morphological connected component algorithm [5]. Real time testing results for object segmentation is displayed in Figure 4.

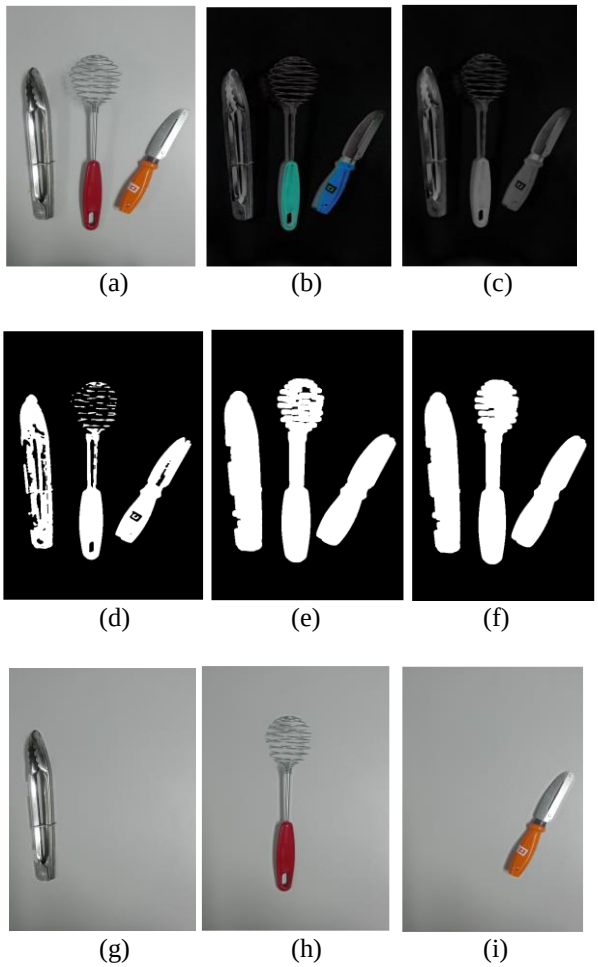

Figure 4. Real Time Testing Results for Object Segmentation (a) Original Image (b) Absolute Difference Result (c) Resultant Image Which is Filtered Media Filter After Converting the Grayscale Image (d) Binary Result by Using Otsu Thresholding (e) Dilated Result (f) Region Filling Result (g)(h)(i) Extracted Results Which is Only One Object, from the Multiple Objects

3.2. Object Classification

After getting the segmented objects, the next step is to classify these objects. In the kitchen cutlery classification step, convolutional neural network (CNN or ConvNet) is applied to extract feature and classify the object. A Convolutional Neural Network (CNN) is a powerful deep learning algorithm due to the use of multiple feature extraction stages that can automatically learn representations from the image and can classify features in images for computer vision. It is a multi-layer neural network intended to examine visual inputs and do tasks such as image classification. There are two main portions to a CNN: a convolution tool that splits the various features of the image for analysis and a fully connected layer that employs the output of the convolution layer to predict the best description for the image. A CNN is constructed with several kinds of layers: Convolutional layer, Pooling layer, Fully connected input layer, Fully connected layer and Fully connected output layer. Convolutional layer is the first layer that extracts a feature map to predict the class probabilities for each feature by applying a filter that layer performs down sampling operation that minimizes the dimensionality of each map but maintains the important information. In fully connected input layer, "flattens" the outputs created by previous layers to turn them into a single vector that can be utilized as an input for the next layer. For predict an accurate label, fully connected layer applies weights over the input created by the feature analysis. In fully connected output layer, the final probabilities is generated to predict a class for the image. There are many popular CNN architectures. GoogLeNet CNN architecture is used in the system.

3.2.1. GoogLeNet

This architecture uses 3 different size filters (i.e., 1x1, 3x3, 5x5) for the same image and combines the features to get a robust output. It consists of 22 layers and it lessens the number of parameters from 60 million (AlexNet) to 4 million. The 1x1 convolution is introduced for dimension reduction. This architecture finds the best weight during training the network and

naturally select the appropriate features. There are multiple Inception modules combined to form a deeper network by which high accuracy can be obtained. The Figure.5 illustrates the multiple convolution with 1x1filter, 3x3 filter, 5x5 filter, and max-pooling layer [6]. For GoogLeNet Network, transfer learning provides new training to recognize new objects in the kitchen cutlery.

In the training step, eight types of kitchen utensil images are firstly resized to 224x224x3 images and trained by applying the transfer learning. And Load the pretrained GoogLeNet network. Specify the training options, including 1e-4 learning rate, 10 mini-batch size and 3 validation frequency. To retrain GoogLeNet to classify new images, replace the last three layers of the network. Set the final fully connected layer to have the same size as the number of classes in the new data set.

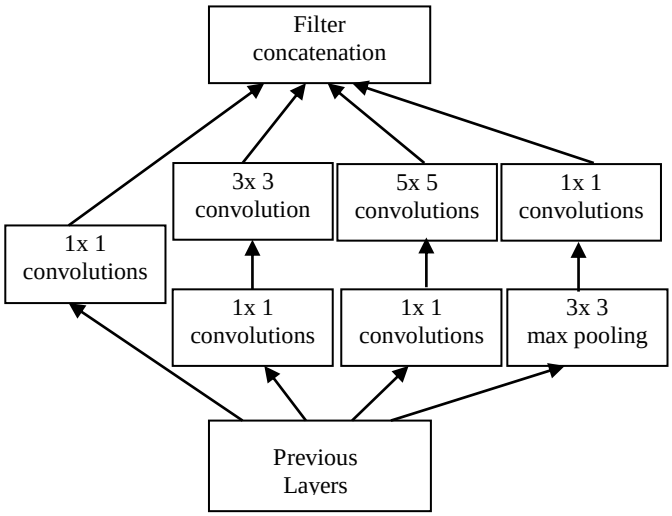

Figure 5. Inception Module of GoogLeNet Architecture[7]

4. Experimental Evaluation

In the section, the proposed system is employed and evaluated to prove the efficiency of proposed method for vision based domestic robot to pick and place the desired object.

4.1. Acquiring the Dataset

Since there is no standard data sets for kitchen utensils, eight types of kitchen utensils' images which are fork, table knife, spoon, kitchen knife, ladle, potato peeler, tong and whisk, are collected via digital cameras and online collecting from images.google.com as the search engine. In the acquisition process with digital camera, the image are captured with various angle that are 30, 45, 60, 70,80 and 90 degrees. There are 800 images for training data. Sample images from training dataset is shown in Figure 6.

4.2. Real Time Testing Results for Kitchen Utensil Recognition

In the section, experimental results of kitchen cutlery recognition with vision based domestic robot webcam will be discussed. The constructed vision based domestic robot arm is working on one color white background for testing.

In real time experiment, eight types of kitchen utensil which are whisk, tong, potato peeler, kitchen knife, fork, table knife, spoon and ladle, are tested with various conditions. At the results of Figure 4, Figure 7, Figure 8 and Figure 9, the proposed object segmentation method is correctly segmented regions.

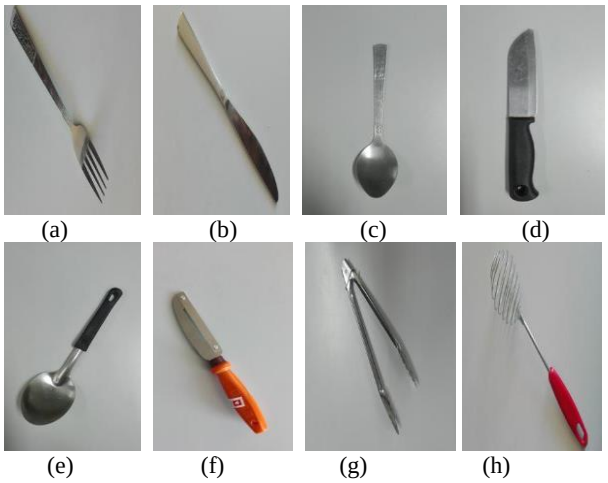

Figure 6. Sample Image from Training Dataset (a) Fork (b) Table Knife (c) Spoon (d) Kitchen Knife (e) Ladle (f) Potato Peeler (g) Tong (h) Whisk

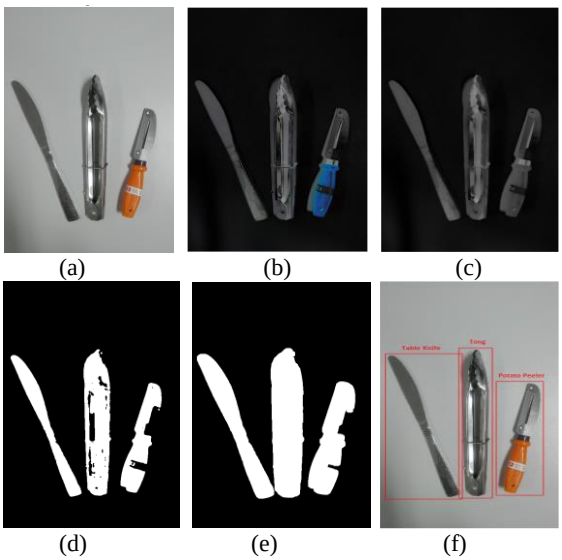

Figure 7. Real Time Testing Result 1 (a) Original Image (b) Absolute Difference Result (c) Resultant Image which is Filtered Media Filter After Converting the Grayscale Image (d) Binary Result by Using Otsu Thresholding (e) Modified Binary Result (f) Objects Classification Results

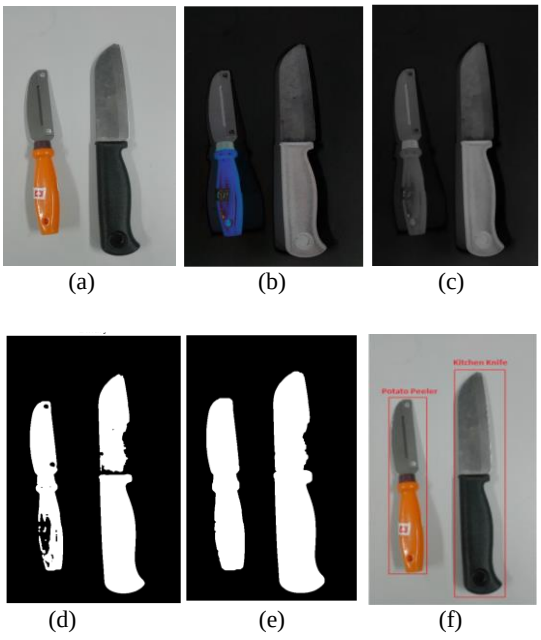

Figure 8. Real Time Testing Result 2 (a) Original Image (b) Absolute Difference Result (c) Resultant Image Which is Filtered Medium Filter After Converting the Grayscale image (d) Binary Result by Using Otsu Thresholding (e) Modified Binary Result (f) Objects Classification Results

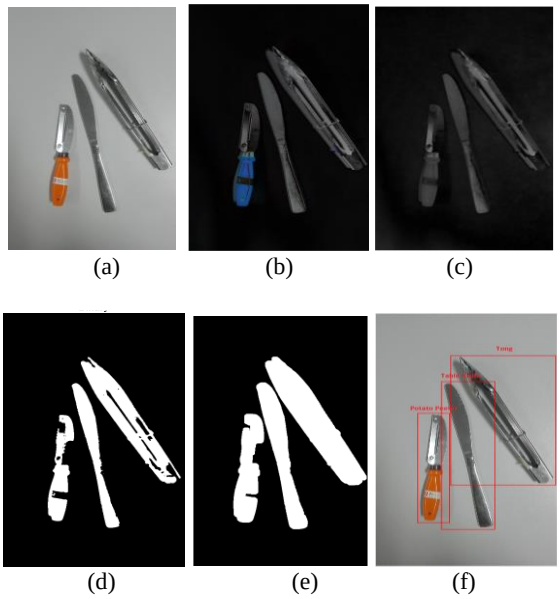

Figure 9. Real Time Testing Result 3 (a) Original Image (b) Absolute Difference Result (c) Resultant Image Which is Filtered Media filter after Converting the Grayscale Image (d) Binary Result by Using Otsu Thresholding (e) Modified Binary Result (f) Objects Classification Results

Therefore, types of kitchen utensil object is well classified based on CNN classification model. According to results of Figure 7 and Figure 8, the various rotation position of objects can be well classified by the proposed method. In Figure 10 and Figure11, dinner fork and dinner spoon region is segmented into two parts due to lighting condition. So,

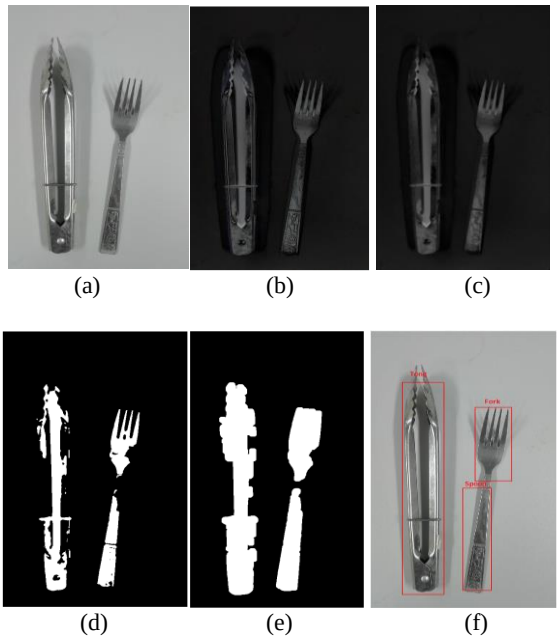

Figure 10. Real Time Testing Result 4 (a) Original Image (b) Absolute Difference Result (c) Resultant Image Which is Filtered Media Filter After Converting the Grayscale Image (d) Binary Result by Using Otsu Thresholding (e) Modified Binary Result (f) Objects Classification Results

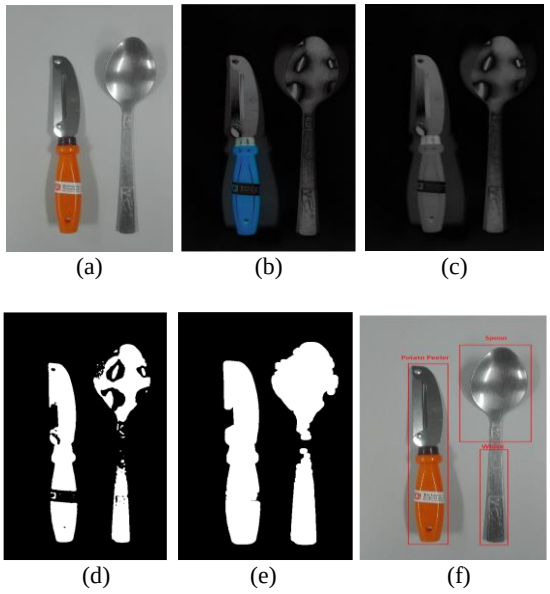

Figure 11. Real Time Testing Result 5 (a) Original Image (b) Absolute Difference Result (c) Resultant Image Which is Filtered Media Filter After Converting the Grayscale Image (d) Binary Result by Using Otsu Thresholding (e) Modified Binary Result (f) Objects Classification Results

one part region is correctly classified and other part region is missed. According to the experimental results, the proposed object segmentation method is able to segment region of potato peeler, tong in any condition. Sometime, it cannot be able to correctly segment the

region of dinner spoon, and dinner fork when lighting reflection condition.

4.3. Performance Evaluation

The performance evaluation for the proposed system is computed according to the following formulas:

$$\text{Precision}\quad (C) = \frac{Correct\quad of\quad C}{Total\quad as\quad C} \tag{3}$$

$$\text{Recall}\quad (C) = \frac{Correct\quad of\quad C}{Total\quad of\quad C}. \tag{4}$$

$$Accuracy = \frac{Correct\quad \text{Prediction}}{Data\quad Size}. \tag{5}$$

Table 1. Confusion Matrix of Testing Results

	1	2	3	4	5	6	7	8
Tong (1)	19	0	0	0	0	1	0	0
Whisk (2)	0	15	0	0	0	2	1	2
Potato peeler (3)	0	0	20	0	0	0	0	0
Ladle (4)	0	0	0	13	3	2	1	1
Kitchen knife (5)	0	0	0	1	16	1	1	1
Spoon (6)	0	1	0	0	0	16	2	1
Fork (7)	0	1	0	0	0	2	15	2
Table knife (8)	1	0	0	0	0	2	2	15

Table 2. Performance Measure of the Proposed System

	Precision	Recall
Tongs (1)	95 %	95%
Whisk (2)	88.2%	75%
Potato peeler(3)	100%	100%
Ladle (4)	92.8%	65%
Kitchen knife (5)	84.2%	80%
Spoon (6)	61.5%	80%
Fork (7)	88.2%	75%
Table knife (8)	68.2%	75%
Overall	84.76%	80.6%

The performance measure is calculated based on 20 time real time testing results for each type of kitchen cutlery object. Confusion matrix of real time testing results is shown in Table1 and Performance measure of each object in the kitchen cutlery is described in Table2. According to the testing results, the proposed system achieves 80.62 % overall accuracy, 84.76 % overall precision and 80.6% overall recall.

When the system is tested with Intel (R) Core (TM) i7-5500U CPU @ 2.40GHz, Installed memory (RAM) 4.00GB, execution time of object segmentation is approximately 4.530649 seconds and execution time for kitchen utensil classification step is approximately 5.106443 seconds.

5. Conclusions

In the paper, the kitchen cutlery recognition system is proposed for the vision based domestic service robot. Background subtraction and Otsu thresholding are applied to detect and find objects' region for segmentation. And in order to recognize the object correctly, convolutional neural network (GoogLeNet) is used. According to the testing results, the overall accuracy result is acceptable but it is not the best for vision based system. So, to gain the higher accuracy result, it is necessary to upgrade the segmentation process in the proposed system. If segmentation is more precise, then CNN (GoogLeNet) is able to recognize more correctly than as before.

6. References

[1] Rahul Kumar, Sunil Lal, Sanjesh Kumar, Praneel Chand, "Object detection and recognition for a pick and place Robot", IEEE, Asia-Pacific World Congress on Computer Science and Engineering, 4-5 Nov. 2014.

[2] E.Rachmawati, M.L Khodra, and I.Supriana, "Histogram based color pattern identification of multiclass fruit using feature selection" IEEE, 5th International Conference on Electrical Engineering and Informatics (ICEEI) , 10-11 August 2015, pp.43-48.

[3] Susovan Jana, Saikat Basak, Ranjan Parekh, "Automatic Fruit Recognition from Natural Images using Color and Texture Features", IEEE, 2017 Devices for Integrated Circuit (DevIC), Kalyani, India, 23-24 March, 2017, pp.620-624.

[4] Aung Kaung Sat, Thuzar Tint, "Object Detection and Recognition System for Pick and Place Robot", International Conference for Big Data Analysis and Deep Learning 2018, Miyazaki, Japan.

[5] Rafael C.Gonzalez, Richard E.Woods, "Digital Image Processing", Pearson Education International, Third Edition.

[6] K.K. Sudha, P. Sujatha, "A Qualitative Analysis of Googlenet and Alexnet for Fabric Defect Detection", International Journal of Recent Technology and Engineering (IJRTE), ISSN: 2277-3878, Volume-8, Issue-1, May 2019, pp.86-92.

[7] http:// towardsdatascience.com/ a-simple-guide- to-the-versions-of-the-inception-network-7fc52b863202s.

Forensics Analysis of Mobile Financial Applications Used in Myanmar

Htar Htar Lwin[1], Wai Phyo Aung[2]
Faculty of Computer Systems and Technologies[1], Department of Automation Control System[2],
University of Computer Studies, Moscow Automobile and Road Construction State Technical University[2]
Yangon, Myanmar[1], Russia[2]
htarhtarlwin@ucsy.edu.mm[1], myfamily46123@gmail.com[2]

Abstract

This paper aims to analyze digital forensics of specific Android mobile financial applications such as m-banking and m-pay applications used in Myanmar. Some applications may store customer's credentials on the phone's internal memory. As sensitive data can be recovered through mobile forensic, sensitive user information is at vulnerability. Thus, we investigated on mobile financial applications to become aware of how tons touchy statistics may be recovered. Android application usually stores data in /data/data/package_name, thus analysis focuses primarily there. The selected Android applications are three mobile banking applications and five mobile money applications which are popular in Myanmar. We used popular open source forensics tools for data extraction and analysis. After analysis, finding indicates that some applications do not store data on user's device. Some applications store encrypted user credentials on device. Some applications not only store user information on device but also upload signature and photo of customer in cleartext.

Keywords- Forensics, Android mobile financial applications, sensitive data, data extraction and data analysis.

1. Introduction

In Myanmar, the ascent in mobile subscribers started after the government opened up the telecommunications segment in 2014, however implemented strong guidelines and point-of-sale (POS) installment arrangements are as yet lingering behind. [1]

Myanmar plans to increment financial inclusion from 30% in 2014 to 40% by 2020, and grown-ups with more than one product from 6% to 15%, by supporting the advancement of a full scope of reasonable, quality and successful financial services. Since the arrangement was propelled in 2013, Myanmar has seen a noteworthy increment in financial access. A report released in 2018 as a part of the Making Access Possible (MAP) program found that grown-ups with access to in any event one formal financial product expanded from 30% in 2013 to 48% in 2018, a right around 66% expansion in financial inclusion, outperforming the underlying 2020 objective of 40%. A few components have added to rising financial inclusion in Myanmar including developing Internet and mobile phone infiltration, with the mobile phone availability rate became from under 10% in 2014 to 95% in 2019. Today, Myanmar is encountering huge innovation drove changes in its banking and finance segment, with individuals step by step moving ceaselessly from money towards setting aside cash in banks and utilizing installment cards, for example, Automatic Teller Machine (ATM) cards and Myanmar Payment Union (MPU) cards. MPU is the national payment network. Additionally, the proliferation of mobile phones and Internet access is giving people access to digital financial services via mobile technology such as mobile applications and web platforms [2].

Mobile banking is an innovation that gives banking services such as balance enquiry, money transfer, billing, and transaction statement utilizing a customer's mobile device. Mobile banking is characterized as an event when clients get to a bank's systems utilizing telephones or comparable gadgets through media transmission systems. Mobile money is an innovation that enables individuals to get, store and go through cash utilizing a cell phone. It's occasionally referred to as a 'mobile wallet' such as Wave Money, OK$, MyTel Pay, M-Pitesan and many more. Mobile money is a mainstream option in contrast to both money and banks since it's anything but easy to utilize, verify and can utilized anyplace there is a cell phone signal. The expanded utilization of cell phones in this manner, applications, has made the requirement for application security. The need to create secure applications that ensure client's data without discarding a help is basic. The most recent Android Operating System adaptation offers Application Programming Interface (API)'s and rules for designers with the aim to advance the reception of secure practices, while most of new gadgets incorporate a segregated equipment/programming framework named Trusted Execution Environment (TEE) to verify information very still even in the event that the cell phone is established. In this way, it appears to be persuading to look at whether versatile applications safely store delicate data or permit the revelation of significant proof in a scientific examination [3].

We made a forensic analysis on Android mobile financial applications in order to obtain sensitive information concerned with the mobile device's owner. The eight selected applications belong to the following

categories: mobile banking and mobile money or mobile wallet. Various open source and commercial tools were used for forensics extraction and analysis in this study. The remainder of the paper is organized as follows. In section 2 literature review on application forensics focusing on Android Operating System was presented. In section 3 we explored the methodology used to analyze the selected financial applications. In section 4 we presented implementation and analysis results. Section 5 is about future works and in section 6 we concluded our findings with discussion.

2. Related Work

In [3], the researchers performed a forensic investigation to Android mobile applications aiming to find sensitive information of the mobile phone user. These applications were chosen based on these facts: (i) popularity on Google Play Store, (ii) handling sensitive privacy information, (iii) have not been researched by past works and (iv) free to download and install. The three chosen applications categorized by bank, mobile network carrier and public transport. The assessment of the security of the applications was performed using two techniques: code and disk analysis. In light of their discoveries they concluded that these applications neglected to protect user's sensitive data and a forensic analysis can reveal crucial and significant information from a forensics perspective.

In [4], exhausted portable entrance in Africa offers unbelievable potential to quicken money related consideration through extended determination of versatile banking by individuals at the Base of the Pyramid (BOP) on the mainland. This article gives results from an efficient audit of existing examination discoveries on the troubles, points of interest and selection elements of versatile banking at the BOP in Africa. The orderly survey, which sought after preferred reporting items for systematic reviews and meta-analyses (PRISMA) rules, perceives the going with key troubles for versatile financial dissemination at the BOP on the mainland: poor portable availability; absence of attention to versatile financial administrations; ignorance; destitution; absence of trust because of saw security dangers; lawful and administrative systems; and social components. In view of investigation of these difficulties, and of the advantages and appropriation elements additionally distinguished, the article gives recommendations on how versatile financial administrations can be all the more economically actualized to help individuals at the BOP in Africa.

In [5], they investigated how much sensitive user and app-generated data are put away on the mobile device after a user registers and banking transactions are finished. As indicated by App code examination, Bank A, Bank F, and Bank G do not implement root device detection. Although Bank D, Bank F, and Bank G have built-in encryption class, the latter is not implemented to encrypt user data. Bank A, Bank D, and Bank E implement SSL printing to check trusted certificate prior to app running with hard-coded public key for Bank A and Band D. In App repackaging analysis, when they installed their repackaged apps to catch SSL traffic and install third party certificate on the rooted device, they are able to intercept SSL traffic and capture sensitive information from Bank C, Bank D, Bank F, and Bank G (e.g. clear text PIN code, account number). Being able to use the same session ID to login to Bank D while another device running this app used in this session ID.

3. Methodology

The main purpose of this work is to determine whether activities performed through smartphone financial applications are stored on the internal memory of the device and whether these data can be recovered. The goal of this study was achieved by conducting experiments on a number of popular financial applications used in Myanmar. Forensic examinations and analyses were performed on three popular mobile banking applications and five mobile money applications.

The experiments were conducted using forensically sound approaches and under forensically acceptable conditions to fulfill a crucial rule in digital forensics, which is to preserve the integrity of the original. We followed the experiment procedure laid down from the Computer Forensics Tool Testing program guidelines established by National Institute of Standards and Technology (NIST).

We used forensically sound methods for data acquisition and data analysis which are implemented as forensics tools. These tools are open source tools and commercial tools.

3.1. Experiment Environment and Requirements

Prior to conducting the experiments, a forensic workstation was set up and configured. Once the forensic workstation was ready, it was isolated from the network. Table 1 and Table 2 show a list of software and hardware used to conduct the experiment:

Table 1. Software Tools

Tool	Name	Version
Root	CF-Auto-Root	
	Odin3	3.10.6
	Busy Box.apk	v1.20.1-Stericson
Forensics Acquisition	Magnetic Acquire	2.22.0.18775
	Android SDK	3.4.1
Forensics Analysis	Autopsy	4.13.0
	DB Browser for SQLite	3.11.2
	Belksoft Evidence Center	9.9.4611

Table 2. Hardware List

Hard	Specification
Android Phone	Samsung Galaxy Note4, Model: SM-N910H, Android version: 6.0.1, Kernel version: 3.10.9-7284779
Laptop	Intel (R) Core i7 CPU, 8.0 GB RAM

3.2. Experiment Procedure

The experiment procedure consisted of three stages: scenarios, logical acquisition, physical acquisition and analysis. The following sections describe each stage in details.

3.2.1. Scenarios. This stage involved conducting common user activities on financial applications on the smartphones. The applications were installed on device if they were not already integrated with the device. Applications were chosen based on their popularity in Myanmar.

For the purpose of the experiments, our own accounts were used on each service. For each application, a predefined set of activities were conducted. The activities were chosen to represent common activities, such as balance inquiry, money transfer, mobile top up, calendar remind setting, ATM/Branch Search and Bill Payment.

3.2.2. Data acquisition. Data acquisition is the way toward imaging or generally extracting data from a digital device and its fringe hardware and media. Obtaining information from a mobile phone is not as simple as a standard hard drive forensic acquisition. The accompanying focuses separate the three sorts of forensic acquisition methods for mobile phones: physical, logical, and manual. These techniques may have some cover with a few levels talked about in the mobile forensics tool leveling framework. The sum and sort of information that can be gathered will shift contingent upon the kind of acquisition technique being utilized.

In this examination, physical acquisition of mobile phones was performed utilizing mobile forensic tools and methods. Physical extraction obtains data from the device by direct access to the flash memory. The procedure makes a bit-for-bit copy of an entire file system, like the methodology taken in PC forensic examinations. A physical acquisition can obtain the entirety of the information present on a device including the erased information and access to unallocated space on most devices.

3.2.3. Analysis. The third stage included performing forensic examinations to the obtained physical image of device, to decide if the activities conducted through these applications were put away on the device's internal memory. Provided that this is true, the amount, location, and importance of the information that could be found and recovered from the physical image were determined. The examinations were conducted manually utilizing various tools to see the gained images, search for data related to the mobile financial applications, and decide how these information were put away on device.

4. Implementation and Analysis

The first stage of the experiment involved installing the financial applications and conducting the predefined activities on each device. These applications were downloaded from the App Store and installed on the devices. For experiment purpose, all financial applications were updated to latest version.

Once the applications were installed on the devices, the predefined activities were conducted on each device. These activities included balance inquiry, money transfer, mobile top up, calendar remind setting, ATM/Branch Search and Bill Payment.

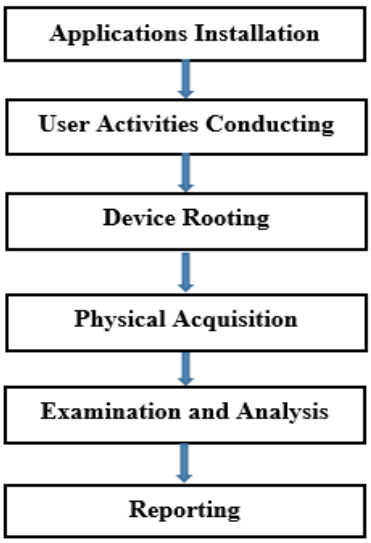

Figure 1. Stages of the Experiment

Similar activities were conducted on each application except some services which are not supported by applications. After conducting the financial activities on the tested phones, a physical image of the internal memory of device was acquired and analyzed for evidence of the conducted activities. The following sections describe the procedures used for the acquisition and analysis of each application. The stages of the experiment are shown in Figure 1.

4.1. Rooting

This section describes the procedure of the physical acquisition and forensic analysis of the Android phone (Samsung Galaxy Note4 – Kernel version 3.10.9-7284779). Except if the Android phone was rooted, many data files could not be accessed. Therefore, the tested Android phone was first rooted utilizing Odin3 (version 3.10.6) to upload the root-kit (CF-Auto-Root).

Installing a root-kit enables the user to gain privilege access the Android OS, permitting him/her to bypass a few restrictions that the manufacturers put on the device. A rooted Android phone enables the user to access protected directories on the system that hold user data (e.g., /data/data directory) and the entirety of the files in these directories. These data files can hold a lot of that may support an ongoing investigation.

4.2. Physical Acquisition

To get physical image of the phone, we used two methods. The first one is using Linux commands 'dd' (Duplicate Data) and the next one is using Magnetic Acquire commercial tool. Extraction time depends on capacity of phone.

We used the following methods to get physical image of the tested phone. Firstly, we need to know which partition holds the data. So we used 'mount' command in first command window to take a look at the location of our desire data partition.

adb –d shell
su
mount

From output of 'mount', we knew that data is located in partition 'mmcblk0p21'. In second command window, we did TCP port forwarding in order to transfer extracted data image to the forensics work station.

adb forward tcp:8888 tcp:8888

In first command window again, we used 'dd' command to get image of data partition.

dd if=/dev/block/ mmcblk0p21 | busybox nc -l -p 8888

In second command window, we used netcat.exe to transfer acquired image file to the forensics work station. Our image file was named as dd_data.dd.

C:\netcat\nc64 127.0.0.1 8888 > dd.dd

Following is alternative to transfer image file to the forensics work station instead of TCP traffic.

dd if=/dev/block/mmcblk0p21 of=/sdcard/dd.img bs=512 conv=notrunc, noerror,sync
adb pull /sdcard/dd.img

4.3. Examination and Analysis

After extracting the image, we started analyzing the image using forensics analyzing tools. We used Autopsy, Belkasoft Evident Center and DB Browser for SQLite which are open source tools. DB Browser for SQLite is used for analyzing SQLite database.

Firstly, extracted image file was copied to the forensic workstation for forensics analysis. Application data is located at /data/data/app_package_name/. Under the folder with application package name, there are four subdirectories that held relevant data for this study: databases, files, cache and shared_prefs, which contains a number of files.

The 'databases' folder held SQLite files. Viewing each file through the SQLite Database Browser and examining its content yielded interesting results. These files hold the records that included significant information for the forensic investigator, such as the users' name, phone number, National Registration Card (NRC) number and account ID, contents of exchanged messages, Uniform Resources Locaters of uploaded photos.

The 'files' folder contained files with names that consisted of letters and numbers and that did not have any extensions. In our case, images contained within these files are screen shots of the customer profile. The 'cache' folder contains pictures that the user had used to register. The 'shared_prefs' folder contains many xml file that hold a lot of information.

In our examination, the m-banking category includes three applications of major banks in Myanmar. The mobile money category comprises five applications that allow financial transactions. Analyzing results are reported as follows.

Application 1 (Bank 1): In this application, we found user ID is stored with clear text in two files of 'Shared_prefs folder' as follow.

string = "ht276***"
name = "USERID"

Application 2 (Bank 2): This application stored activate code length and pin key length in an xml file of 'Shared_prefs' folder as follow.

name = "activate_code_length"
value = "4"
name = "pin_key_length"
value = "4"

In 'database', we found the names of branches which the customer registered. Accont number is not mentioned in there. In our case, the customer registered two saving accounts in Lewe and Bogalayzay, and ATM card in Toungoo as follow.

title	**description**
SA-MMK-LWE	SA-MMK-LWE
SA-MMK-BKLZ	SA-MMK-BKLZ
ESA-MMK-TGU[ATM]	ESA-MMK-TGU[ATM]

We also found the name, phone number and NRC number of the customer in the 'database' as follow.
DAW *** +959797926*** 12/THAGANA (N)032***

Application 3 (Bank 3): In this application, we found the last updated location of the customer and last

location updated time. It was stored in 'Shared_prefs' folder. We can locate the location of the customer in Google Map as shown in Figure 2.

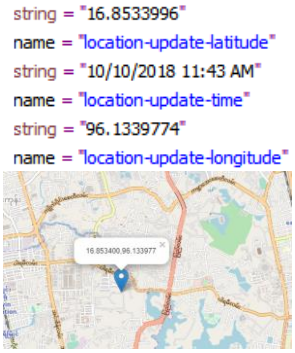

Figure 2. Location of the Customer

Application 4 (Mobile Money 1): In this application, we found the balance of the customer mentioned in 'Shared_prefs' folder as follow.
<string nam="Balance">1121.00</string.
Application 5 (Mobile Money 2): No information was stored with clear text in device.
Application 6 (Mobile Money 3): This application takes care the customer information. It stored user data in encrypted database. Although we found the database, we couldn't decrypt it. This is desirable for security purpose.

Application 7 (Mobile Money 4): This application cannot be installed on rooted device. So we couldn't analysis it. It means this application takes care of security.

Application 8 (Mobile Money 5): This application stores the customer photo in 'cache' folder. In 'files' folder, we also found two screen shots as shown in Figure 3.

Figure 3. Customer Photo in Cache Folder

This app stored a lot of user credential on local device and also uploaded signature and photo of customer to their server in clear text. URLs can be seen in an xml file under 'shared_prefs' folder as follows.

<string name="PROFILE_PIC_SIGN">https://s3-ap-southeast-1.amazonaws.com/***/Signpic00959974145***/079d4dcf-4d82-4183-9d8e-70cb8efc8339SignpicFebruary_14_2018 11_51_23 AM.jpg</string>

<string name="PROFILE_PIC">https://s3-ap-southeast-1.amazonaws.com/***/profilepic00959974145***/876a51be-6083-46d8-ad20-9ef554c67949profilepicFebruary_14_2018 11_51_23 AM.jpg</string>

We downloaded the photos using the URLs obtained from above as shown in Figure 4.

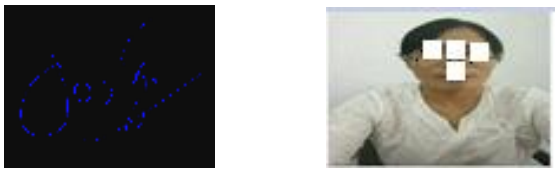

Figure 4. Signature and Photo of Customer

Another credential data were also found in 'shared_prefs' folder. There were a lot of information such as Name, Phone, mail, address, Balance, Location, NRC, Security Question (English and Myanmar) as follows.
<string name="number">09974145***</string>
<string name="walletbalance">8225.00</string>
<string name="SENDMONEYBANK_NRC">
12/THAGAKA(N)032***</string>
<string name="security_question_english">Who is your *** name?</string>
<string
ame="MOBILENO">00959974145***</string>
Table 3 summarizes the report of our analysis for each application.

Table 3. Analysis Report

App Name	Database	Files	Cache	Shared _prefs
App 1				User ID
App 2	Branch, name, phone, NRC			Code/ key length
App 3				Locatio n, time
App 4				Balance
App 5	No user information found			
App 6	Encrypted			
App 7	Cannot install on rooted device			
App 8		Photo	Photo	URLs

5. Future work

We analyzed non-volatile memory in this work. As a future work we will do on volatile memory where user credentials are usually stored. We will also analyze network traffic of application data in next work.

6. Discussion and Conclusions

In other countries, there are forensics investigations and analysis on financial applications used in their country. However there is no such works in Myanmar. This is the reason why we performed this work. This study focused on the recovery of artifacts and traces related to the use of applications of eight financial services used in Myanmar. The forensic analysis determined the amount, significance, and location of user credential data that could be found and retrieved from the physical image of device. The tested financial applications were three mobile banking applications and five mobile money applications. According to professional ethic, we could not mention the names of banks and payment services used in our investigation and analysis.

We used forensically strong methods and followed Computer Forensics Tool Testing program guidelines established by the National Institute of Standards and Technology. The results showed that most applications stored data in 'Shared Preferences' folder. Applications (1 to 4) stored a significant amount of valuable data that could be recovered and used by the forensic investigator. Application 5 did not keep any user information on the device. Application 6 stored data in encrypted database. Application 7 cannot be installed on rooted devices for security reason. The worse application which violence customer privacy is Application 8. It not only stored credential data such as profile photo and signature on device but also uploaded to server with clear text. Our analysis shows the nature of the credential data that could be recovered from device and their locations from physical image file. As indicated by our analysis, we can infer that these applications failed to secure user's sensitive information and a forensic analysis can uncover critical and noteworthy data from forensics perspective.

We would like to suggest financial application developers to follow privacy policies and have great concern on security matters. Department of Consumer Affairs in Myanmar needs to take care of this issue because people from Myanmar is increasingly using mobile financial applications and they are still less awareness of security to protect their properties.

7. References

[1] Kyaw Soe Htet, "Growing mobile penetration gives Myanmar fintech a big boost", Myanmar Times, Myanmar, 21 August 2019.

[2] Thiha, "With 70% Unbanked Population Myanmar Bets on Fintech to improve Financial Inclusion", Consult-Myanmar, Myanmar, 19 November 2019.

[3] Theodoula-Ioanna Kitsaki, Anna Angelogianni, and Christoforos Ntantogian, "A Forensic Investigation of Android Mobile Applications", Proceedings of the 22nd Pan-Hellenic Conference on Informatics, Association for Computing Machinery, New York, NY, United States, November 2018. pp. 58-63.

[4] Richard Pankomera and Darelle van Greunen, "Challenges, Benefits, and Adoption Dynamics of Mobile Banking at the Base of the Pyramid (BOP) in Africa: A Systematic Review", The African Journal of Information and Communication (AJIC), LINK Centre University of the Witwatersrand (Wits), Johannesburg, South Africa, Issue 21, 23 November 2018. pp. 21-49.

[5] Rajchada Chanajitt, "Forensics Analysis of m-banking Apps on Android Platforms".