



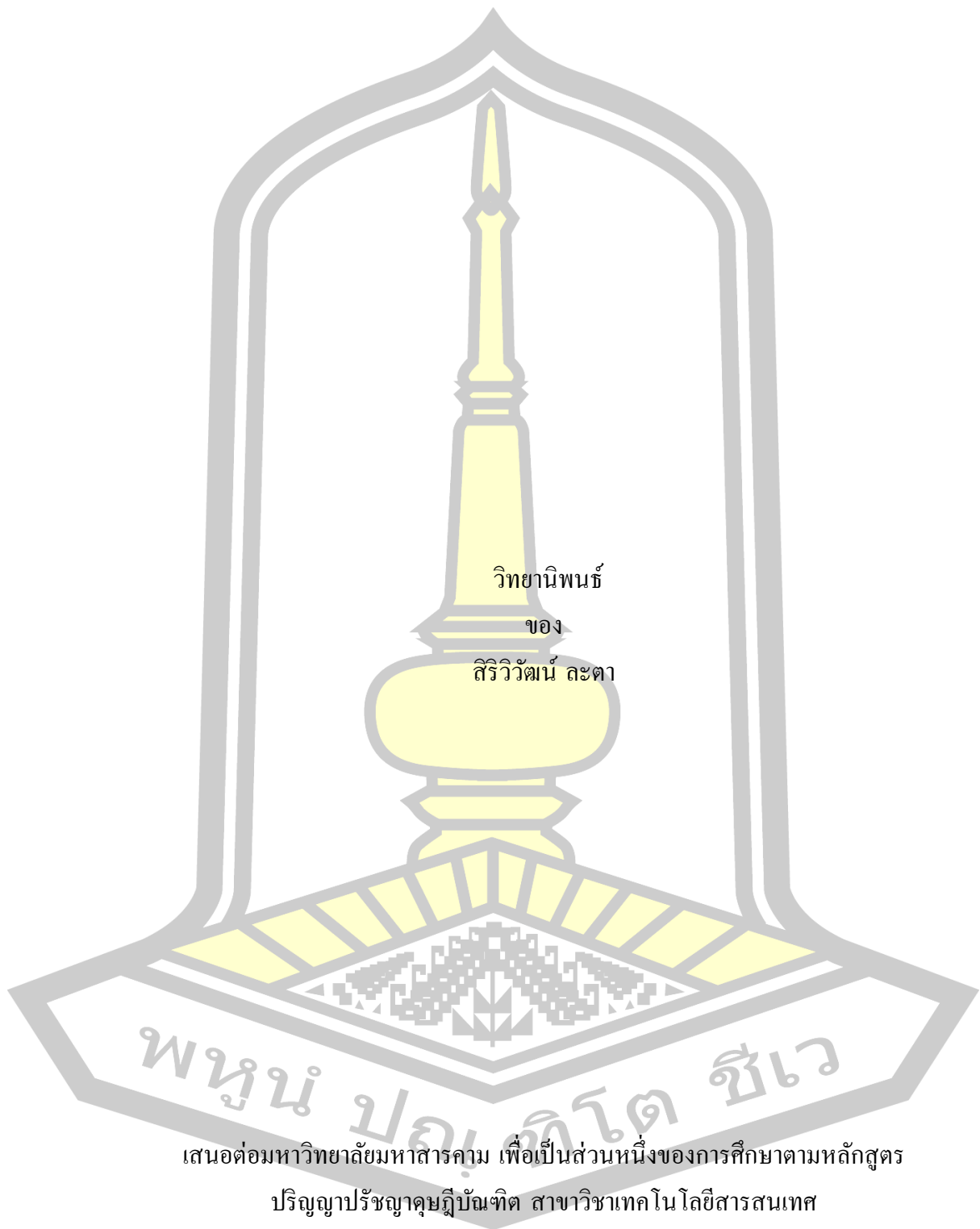
The Automated Sign Language Translation System Using Deep Learning

Siriwiwat Lata

A Thesis Submitted in Partial Fulfillment of Requirements for
degree of Doctor of Philosophy in Information Technology
May 2023

Copyright of Mahasarakham University

ระบบแปลภาษาเมื่ออัตโนมัติโดยใช้การเรียนรู้เชิงลึก



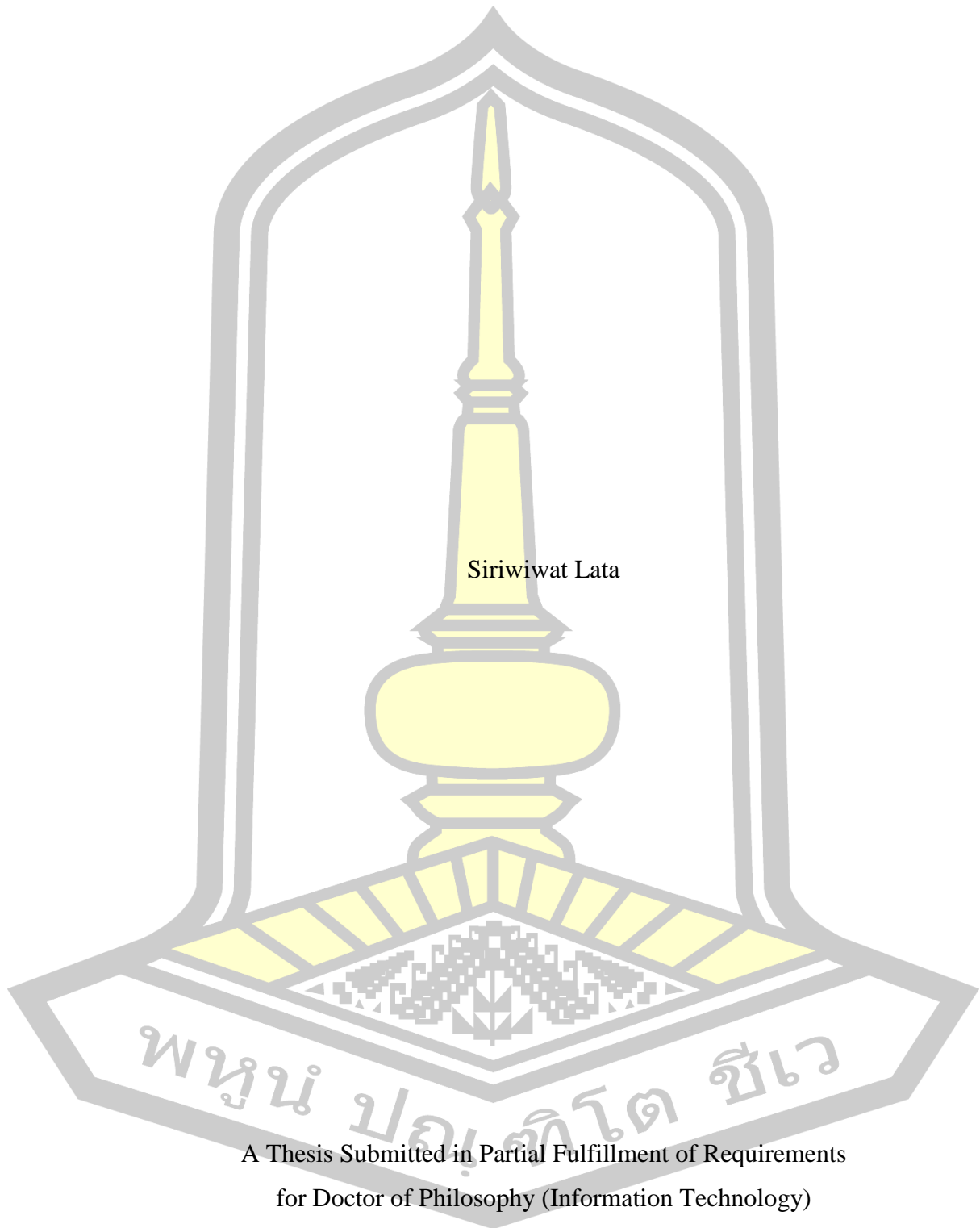
เสนอต่อมหาวิทยาลัยมหาสารคาม เพื่อเป็นส่วนหนึ่งของการศึกษาตามหลักสูตร

ปริญญาปรัชญาดุษฎีบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ

พฤษภาคม 2566

ลิขสิทธิ์เป็นของมหาวิทยาลัยมหาสารคาม

The Automated Sign Language Translation System Using Deep Learning



A Thesis Submitted in Partial Fulfillment of Requirements
for Doctor of Philosophy (Information Technology)

May 2023

Copyright of Mahasarakham University



The examining committee has unanimously approved this Thesis, submitted by Mr. Siriwiwat Lata , as a partial fulfillment of the requirements for the Doctor of Philosophy Information Technology at Mahasarakham University

Examining Committee

Chairman

(Prof. Rapeepan Pitakaso , Ph.D.)

Advisor

(Asst. Prof. Olarik Surinta , Ph.D.)

Committee

(Asst. Prof. Rapeeporn Chamchong ,
Ph.D.)

Committee

(Asst. Prof. Chatklaw Jareanpon ,
Ph.D.)

Committee

(Asst. Prof. Phatthanaphong
Chompoowises , Ph.D.)

Mahasarakham University has granted approval to accept this Thesis as a partial fulfillment of the requirements for the Doctor of Philosophy Information Technology

(Assoc. Prof. Jantima Polpinij , Ph.D.)
Dean of The Faculty of Informatics

(Assoc. Prof. Krit Chaimoon , Ph.D.)
Dean of Graduate School

TITLE	The Automated Sign Language Translation System Using Deep Learning		
AUTHOR	Siriwivat Lata		
ADVISORS	Assistant Professor Olarik Surinta , Ph.D.		
DEGREE	Doctor of Philosophy	MAJOR	Information Technology
UNIVERSITY	Maharakham University	YEAR	2023

ABSTRACT

Sign language is essential for communication with the hearing impaired. It's difficult for normal people to understand so that people can communicate or interpret sign language. This thesis purposes to invent an automatic Thai sign language system that can translate sign language using the proposed deep learning techniques. This method recognized Thai Sign Language covering both Static and Dynamic spellings in Thai Sign Language, including the words in the sentence.

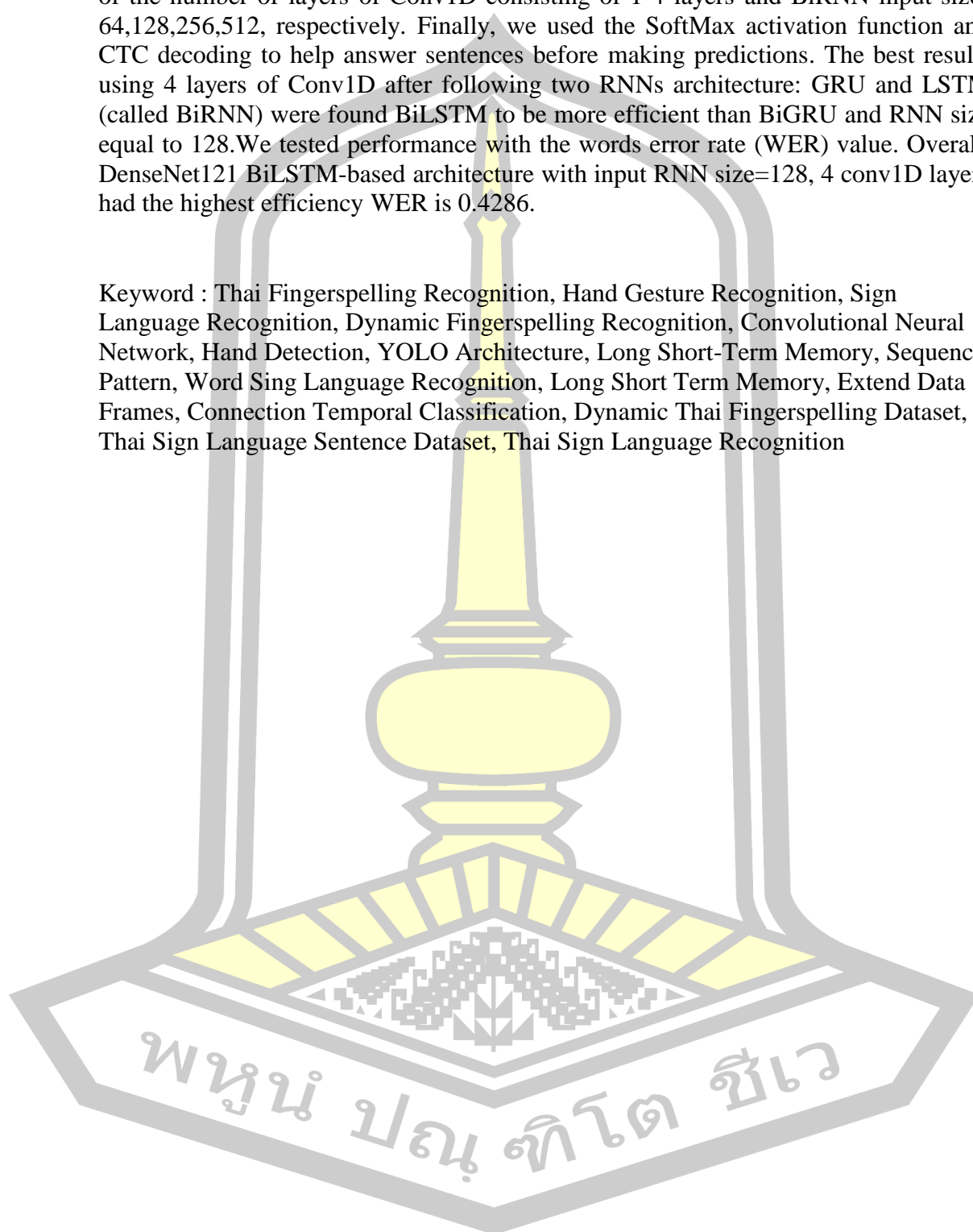
In the first approach, we proposed an end-to-end method of recognizing sign language with deep learning based on static Thai sign language (1-stage) from complex images. This approach was based on hand detection from the YOLO v3 architecture to detect the region of interest (ROI) in hand only. We performed feature extraction and quantified the recognition efficiency of five CNNs: MobileNetV2, DenseNet121, InceptionResNetV2, NASNetMobile, and EfficientNetB2 for training. The results showed that DenseNet121 and MobileNetV2 outperformed other CNN models that have high accuracy in image recognition of Thai Sign Language.

In the second approach, Thai Sign Language required a variety of gestures (from 2-Step up) to interpret meaning. We proposed Dynamic Sign Language Recognition based on the deep learning approaches: the YOLOv5 algorithm for human detection and a combination of two deep learning methods of the Convolutional Neural Network and Recurrent Neural Network to aid sequence-based image recognition. The RNN used two different LSTM and GRU and numbers of units: 32 and 64. We created various CNN models based on three architectures: MobileNetV2, ResNet50, and DenseNet201. It was called the CNN-LSTM architecture. We decided to add the extra densely connected layer with 64 units between the GAP layer and softmax activation function. The results obtained from the ResNet50-LSTM architecture achieved the highest accuracy on the validation set.

In the third approach, Thai Sign Language was generally used to communicate in words that express gestures in a continuous sentence. To increase the number of datasets, we selected 32 and extended data frames 100 times to obtain different datasets. In addition, Thai sign language sentence recognition was used for predicting words in sentences of the Thai sign language base on three CNN: RestNet50, DenseNet121, and VGG16 architectures. The sequence patterns were given to the

Conv1D and LSTM to classify words in sentences. Also, we compared the performance of the number of layers of Conv1D consisting of 1-4 layers and BiRNN input sizes 64,128,256,512, respectively. Finally, we used the SoftMax activation function and CTC decoding to help answer sentences before making predictions. The best results using 4 layers of Conv1D after following two RNNs architecture: GRU and LSTM (called BiRNN) were found BiLSTM to be more efficient than BiGRU and RNN size equal to 128. We tested performance with the words error rate (WER) value. Overall, DenseNet121 BiLSTM-based architecture with input RNN size=128, 4 conv1D layers had the highest efficiency WER is 0.4286.

Keyword : Thai Fingerspelling Recognition, Hand Gesture Recognition, Sign Language Recognition, Dynamic Fingerspelling Recognition, Convolutional Neural Network, Hand Detection, YOLO Architecture, Long Short-Term Memory, Sequence Pattern, Word Sign Language Recognition, Long Short Term Memory, Extend Data Frames, Connection Temporal Classification, Dynamic Thai Fingerspelling Dataset, Thai Sign Language Sentence Dataset, Thai Sign Language Recognition



ACKNOWLEDGEMENTS

This thesis was accomplished with excellent assistance and advice from Assistant Professor Dr. Olarik Surinta, thesis advisor, who transfers research knowledge and skills, sacrifices time to give advice, and allows me to develop the necessary skills while studying the Ph.D. I greatly appreciate this opportunity and would like to express my deep gratitude. Thank you to the Doctor of Philosophy program in Information Technology and Multi-agent Intelligent Simulation Laboratory Research Unit Maha Sarakham University for providing scholarships in research and support for research work. Thanks to the Bachelor of Education Program in Special Education at Rajabhat Maha Sarakham University, students assisted with the dataset for creating a dataset. In addition, I would like to thank the Department of Computer Rajabhat Maha Sarakham University for my work, classmates, colleagues, and the research teams who have spent time together in the research room. Also, my family has always supported and assisted me in various fields until graduation.

Siriwivat Lata

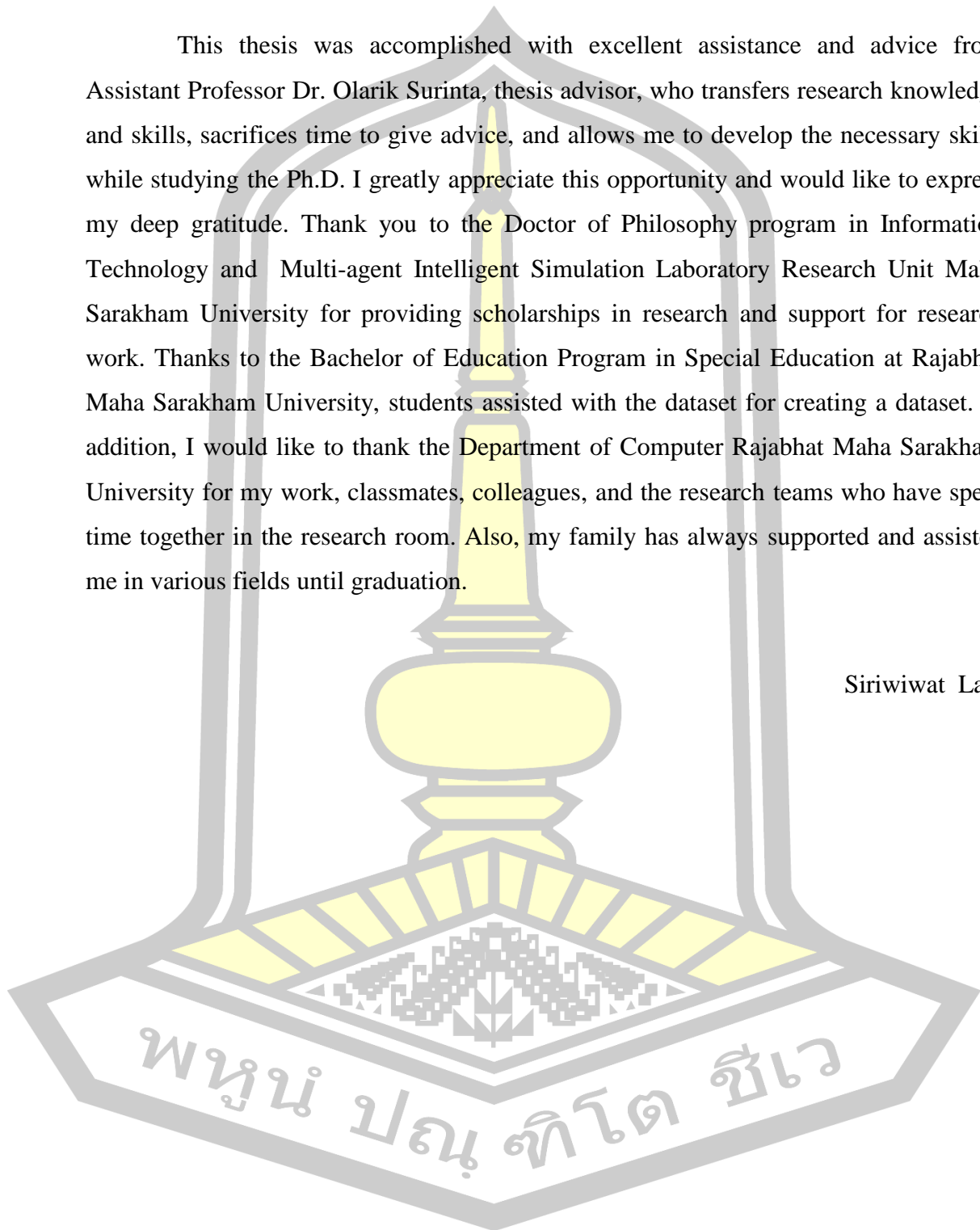
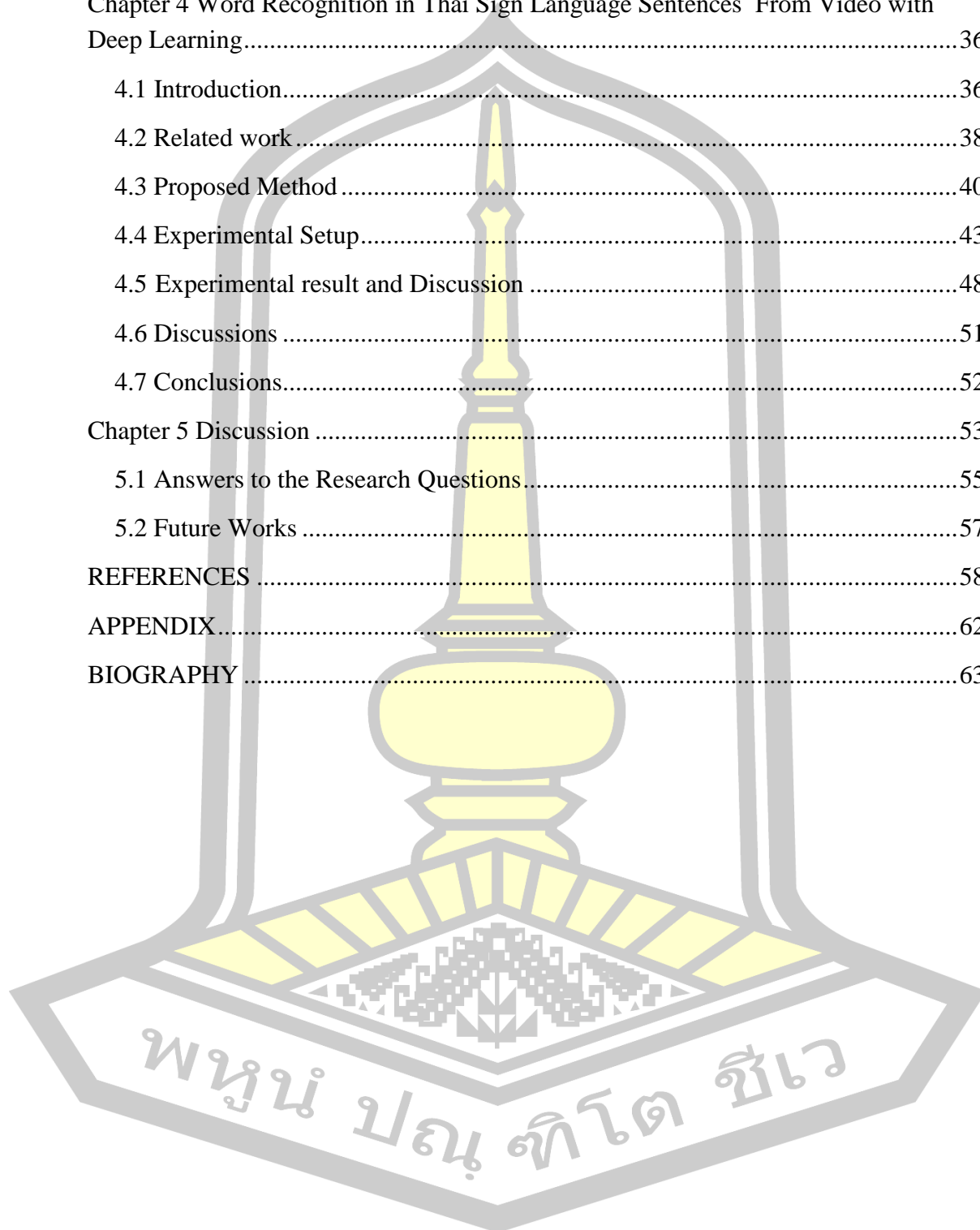


TABLE OF CONTENTS

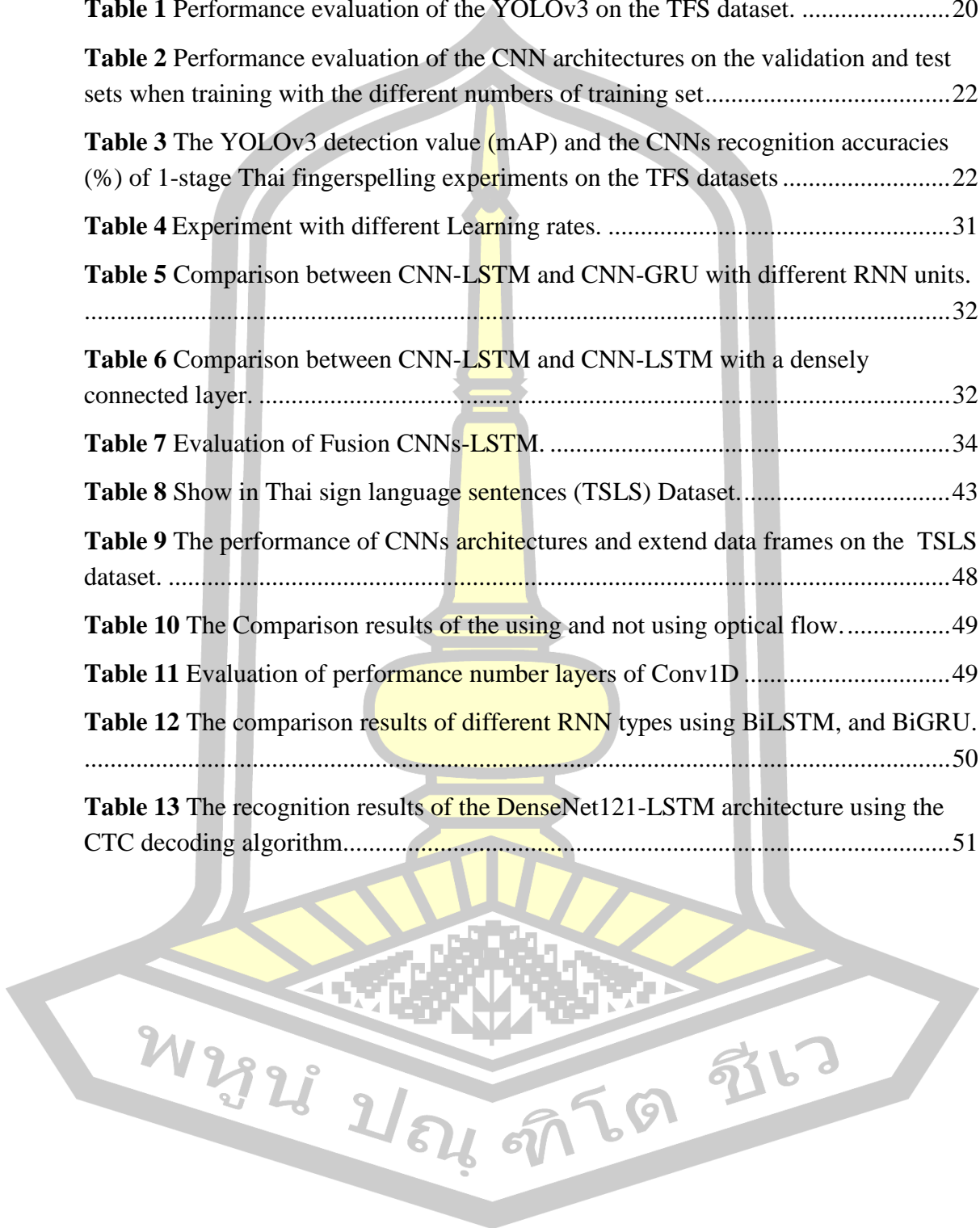
	Page
ABSTRACT.....	D
ACKNOWLEDGEMENTS.....	F
TABLE OF CONTENTS.....	G
LIST OF TABELS.....	I
LIST OF FIGURES.....	J
Chapter 1 Introduction.....	1
1.1 Introduction.....	1
1.2 Research Aim.....	4
1.3 Research Question.....	4
1.4 Contributions.....	5
1.5 The automated Thai sign language recognition system.....	6
Chapter 2 An End-to-End Thai Fingerspelling Recognition Framework with Deep Convolutional Neural Networks.....	14
2.1 Introduction.....	14
2.2 Related work.....	15
2.3 End-to-End Thai Fingerspelling Recognition Framework.....	16
2.4 Fingerspelling Datasets.....	18
2.5 Experimental Results.....	20
2.6 Conclusions.....	23
Chapter 3 Dynamic Fingerspelling Recognition from Video using Deep Learning Approach: From Detection to Recognition.....	25
3.1 Introduction.....	25
3.2 Related work.....	26
3.3 Dynamic Fingerspelling Recognition System Architecture.....	27
3.4 Dynamic Thai Fingerspelling Dataset.....	30
3.5 Experimental Results and Discussion.....	31

3.5 Conclusions.....	35
Chapter 4 Word Recognition in Thai Sign Language Sentences From Video with Deep Learning.....	36
4.1 Introduction.....	36
4.2 Related work.....	38
4.3 Proposed Method.....	40
4.4 Experimental Setup.....	43
4.5 Experimental result and Discussion.....	48
4.6 Discussions.....	51
4.7 Conclusions.....	52
Chapter 5 Discussion.....	53
5.1 Answers to the Research Questions.....	55
5.2 Future Works.....	57
REFERENCES.....	58
APPENDIX.....	62
BIOGRAPHY.....	63



LIST OF TABELS

Table 1 Performance evaluation of the YOLOv3 on the TFS dataset.	20
Table 2 Performance evaluation of the CNN architectures on the validation and test sets when training with the different numbers of training set.....	22
Table 3 The YOLOv3 detection value (mAP) and the CNNs recognition accuracies (%) of 1-stage Thai fingerspelling experiments on the TFS datasets	22
Table 4 Experiment with different Learning rates.	31
Table 5 Comparison between CNN-LSTM and CNN-GRU with different RNN units.	32
Table 6 Comparison between CNN-LSTM and CNN-LSTM with a densely connected layer.	32
Table 7 Evaluation of Fusion CNNs-LSTM.	34
Table 8 Show in Thai sign language sentences (TSLs) Dataset.....	43
Table 9 The performance of CNNs architectures and extend data frames on the TSLs dataset.	48
Table 10 The Comparison results of the using and not using optical flow.....	49
Table 11 Evaluation of performance number layers of Conv1D.....	49
Table 12 The comparison results of different RNN types using BiLSTM, and BiGRU.	50
Table 13 The recognition results of the DenseNet121-LSTM architecture using the CTC decoding algorithm.....	51



LIST OF FIGURES

Figure 1 Types of gesture recognition Static and Dynamic	2
Figure 2 Static gestures for all 15 alphabets.	7
Figure 3 Examples of dynamic gestures, alphabets $\text{๓}/\text{ng}/$, and $\text{๔}/\text{ch}/$, respectively.....	7
Figure 4 Examples of gestures: words, and sentences in Thai Sign Language	7
Figure 5 illustration of the convolutional neural network architecture.....	9
Figure 6 Illustration of the convolutional operation	9
Figure 7 The network architecture of YOLO.....	12
Figure 8 The internal structure of the LSTM.....	13
Figure 9 The proposed framework of the 1-stage Thai fingerspelling recognition ...	16
Figure 10 Hand detection with Yolov3 Network Architecture.....	17
Figure 11 Examples of the 1-stage Thai fingerspelling consonants that recorded in (A) non-complex and (B) complex backgrounds.....	18
Figure 12 Examples of the TFS datasets. (A) Unseen-TFS and (B) KKU-TFS datasets	19
Figure 13 Illustration of the output that was detected using the YOLOv3.....	21
Figure 14 Frame Selection method.....	28
Figure 15 Human Detection with YOLOv5.....	28
Figure 16 Illustration of the dynamic fingerspelling recognition system	28
Figure 14 Examples of dynamic Thai fingerspelling dataset.....	31
Figure 18 Illustrated the validation loss values of CNN-LSTM models.	33
Figure 19 Illustration of the ROC curves for dynamic fingerspelling recognition using different CNNs.....	34
Figure 20 The Proposed Framework for Thai sign language recognition.....	41
Figure 21 Class distribution of word frequency the TSLS dataset.	43
Figure 22 Example Thai sign language sentences Dataset	44
Figure 23 The sample of image frames show	45
Figure 24 The Process for selecting frames from a video.....	46
Figure 25 The sample of image frames show (A) Original sequence images, (B) Optical flow on sequence images.	47

Chapter 1

Introduction

1.1 Introduction

Humans are perceived as the most valuable resource in the world. However, several factors result in human imperfections which can cause human disabilities, such as accidents, congenital disabilities, etc. In 2021, the World Health Organization's Disability Situation Report found that more than 1500 million people worldwide, or about 15 percent of the world's population, are people with disabilities. It also found that there are at least 430 million people with hearing impairments as children as 34 million. By 2050, it is estimated that there will be more than 700 million people with hearing impairments (World Health Organization, 2021). These groups need to be taken care of, especially when it comes to communicating with normal people who lack an understanding of sign language gestures, causing difficulty in communication between people with disabilities and normal people and misunderstandings (Watcharin, 2015). This means that these groups of people need to communicate with each other by the hand, known as sign language.

Sign language is a nonverbal language that is communicated by gestures and movements of the hands, body, and the use of lips to convey meanings or representations for people with hearing impairments. The use of hand and finger gestures in spelling and gestures to convey the meaning of words and sentences. Therefore, Thai sign language is a common language used for communication among deaf people and is important for the development of deaf people's knowledge and abilities (Boonya, 2008). However, the main problem with using sign language is the inability to communicate with normal people or people who do not know sign language. The importance of sign language affects these people with disabilities, therefore, various methods or techniques have been introduced to assist in communication, especially in computer vision, to recognize gestures from a variety of sign languages, both the shape of the hand movements of the hands, arms, including the face to learn gestures and interpret results to get the most accurate value in the meaning. Modern technology has been applied in various areas to improve hand gesture recognition. It can be classified according to two types of recognition (Rautaray & Agrawal, 2012): Static, and Dynamic (Rautaray & Agrawal, 2012). Static refers to the gesture of the hand with only one gesture to convey meaning. Also, dynamic refers to the gesture of the hand, having many strokes to convey meaning, as shown in Figure 1.



Figure 1 Types of gesture recognition Static and Dynamic

The diverse gestures of each sign language in the world are unique, so each sign language is different. Likewise, Thai Sign Language is the national language and needs to be learned as the basis of education for the hearing impaired in Thailand and as a language of communication. For instance, if we want to converse with English people, we need to learn English. Therefore, teaching Thai sign language, alphabet, and Thai tones is necessary for learning and affecting writing. In addition, specific meanings or spelling to tell meanings such as names of people places names and specific words that are not in the words provided in Thai Sign Language. Thai Sign Language uses various complex words and sentences that are difficult to learn. Due to the need to interpret the gestures and hand movements, it still doesn't get the attention of normal people. As a result, people with disabilities have problems with communication and become incapable of living with normal people and with the problem of limited communication. Therefore, technology and equipment are needed to help communicate with these groups.

Many researchers have invented and applied technologies to improve the efficiency of sign language communication, such as the introduction of sensor technology to facilitate communication such as Accelerometers, Gyroscope (Bajpai, 2015), Flex Sensor (Shukor et al., 2015), and Depth Camera (Huang et al., 2018), (Bantupalli & Xie, 2018). Such an expensive technology has led researchers to focus on developing artificial intelligence-powered image or video recognition since only a webcam could be used to communicate with people with disabilities. The technique was able to detect movement (Zhao et al., 2019) of the hand and was able to interpret hand gestures or gestures expressed by people with disabilities due to the complexity of sign language.

The researchers started by designing and developing a process for static-sign language recognition. That means one gesture has only one meaning (Salian, 2017),(Zisserman, 2015), for example, American sign language recognition (C. M. Jin & Omar, 2016), number recognition from sign language (Thalange & Dixit, 2016) by research by Pariwat & Seresangtakul (2017) (Pariwat & Seresangtakul, 2017). In addition, Nakjai & Katanyukul (2018) (Nakjai & Katanyukul, 2018) has created a static sign language image dataset by setting the background to a color that contrasts with the color of human skin. However, if the dress is black, it will be able to detect the hand area. In a study by Oliveira et al (2017) (Oliveira et al., 2017), a dataset of the Irish sign language static alphabet with a total of 23 static gestures was generated. This dataset

stores only the hand and the image have a black background and then takes the specific area of the hand to recognition with different methods, including support vector machines, linear discriminant analysis (LDA), multi-layer perceptron (MLP), K-nearest neighbor (KNN), and convolutional neural network (CNN). However, in actual situations where sign language interpretations may not be in the provided space, this may result in the designed algorithm being unable to detect the hand and recognize it correctly.

Currently, there are deep learning algorithms for object detection that can be effectively divided into two types: 1) One-Stage: Suitable for detecting small objects with high detection speed, including Single Shot Detector (SSD) (Liu et al., 2016) and YOLO. YOLO algorithm was developed from version 1 to version 3, improving the algorithm to be faster and more accurate (Redmon & Farhadi, 2018). 2) Two-Stage: An algorithm that uses a Selective Search method to find an object localization. As it is the most difficult method of object detection, the process must find the desired objects from many regions of interest. For example, the R-CNN method (Girshick et al., 2014), Fast-RCNN (Girshick, 2015), Faster-RCNN (Shaoqing Ren, Kaiming He Ross Girshick Jian, 2015), and RetinaNet (Lin et al., 2017).

The object detection method mentioned above can be used to detect the hand area where the background doesn't need to be black, so it can be detected in any environment. In addition, deep learning is also used in sign language recognition by a popular algorithm using a convolutional neural network (CNN). Using this technique, it can recognize illustrations that are effective and suitable for sign language (Salian, 2017), (Zisserman, 2015). It is used in two ways, static and dynamic, which can be interpreted to communicate with a normal person. The dynamic nature of the motion or video gesture recognition requires CNN deep learning. In addition, it is implemented for the learning of sign language sentence translation using the Recurrent neural network (RNNs) architecture pioneered by hierarchical learning.

In particular, the long short-term memory (LSTM) (Hochreiter & Schmidhuber, 1997) neural networks are well suited for use with sequences such as video work, and word separation from the text in the book. It has a hierarchical recognition feature that makes it possible to recognize multi-stroke gestures for different hand gestures. However, in everyday life, Thai sign language speakers use a large number of words in succession to form sentences. People who study sign language need to learn both the various gestures in one word and link them into the correct sentences because sign language has complicated before and after gestures that are difficult to learn.

Therefore, we utilized the effective decoding of the connectionist temporal classification loss (CTC) (Graves et al., 2006) technique to learn hierarchy to help locate the appropriate word placement and interpret the results. It can be a sign language

sentence from the animation. This will cover a fully automated Thai sign language translation system.

1.2 Research Aim

To develop an automated sign language translation system for the hearing impaired using deep learning that can translate both consonant and verbal sign language.

1.3 Research Question

Due sign language is a unique and communicative language characterized by using hands to interpret the meaning with both single and multi-gesture interpretations which complicate the use of both rhythm and gestures (Boonya, 2008) therefore, this thesis presents an efficient automatic sign language translation system using for deep learning.

It consists of the following research questions:

RQ1: In general, the nature of the use of sign language seen through a conventional device is to see, not just the area of the hand. We see it as a half-body image, most of which is illustrated with a complex background. How does the deep learning method for the detection and recognition improve on Static Sign Language recognition efficiency? Which consists of 15 alphabets using YOLO architecture for hand detection and CNN architecture for recognition. We experimented with five architectures of CNN: MobilNetV2, DenseNet121, InceptionResNetV2, NASNet-Mobile, and EfficientNetB2. This research tests three Thai Fingerspelling (TFS) datasets: TFS, Unseen-TFS, and KKU-TFS.

RQ2: Since the learning illustrations in the TFS datasets contain only static datasets, it is not possible to learn the Dynamic of Thai Fingerspelling. Do Deep learning neural networks in combination with LSTM learning increase the efficiency of dynamic Sign Language recognition efficiency? Therefore, we created the Dynamic Thai Finger Spelling Dataset (DTFS) as a video of all Thai Sign Language alphabet. It continues to reduce the background of unrelated images by using the state-of-the-art YOLO v5 architecture to detect person images before sending them to CNN and using LSTM in sign language recognition that enables them to recognize the Memorize alphabet with multiple strokes or animated form.

RQ3: In addition to the two forms of recognition, Thai Sign Language in everyday life often uses words and sentences with contiguous gestures forming sentences. Therefore, learning words in sentences is an important part of using Thai Sign Language. This study created the Thai sign language sentence (TSLs) Dataset. After that, how can we increase the dataset to train from low data? We chose a method to reduce the background of the image in the unrelated section using the YOLO v5 architecture. Followed by using CNN deep learning in conjunction with the LSTM

architecture, Conv1D, and using Connectionist Temporal Classification (CTC) to decoding the output in the sentence to be classified words to translate the sentences in sign language automatically.

1.4 Contributions

In the research paper in the thesis on an automatic sign language translation system using deep learning, the research team tried to develop a complex Thai sign language recognition system. The researcher presented an effective Thai sign language recognition process by categorizing the research process as follows:

The first section focuses on an end-to-end Thai sign language recognition by focusing on non-specific images of the hand and images with complex backgrounds by using deep learning to perform detected in the hand area.

Then, the researchers took the images of the detected hands to perform the sign language recognition. It focuses on the Static Thai Sign Language data set consisting of 15 alphabets (ก/k/, ด/d/, ต/t/, น/n/, บ/b/, พ/p/, ฟ/f/, ม/m/, ย/y/, ร/r/, ล/l/, ว/w/, ส/s/, ห/h/, and อ/o/) by using the YOLOv3 architecture for hand detection. Next, we found a suitable CNN architecture in the dataset by experimenting with five different CNN architectures: MobilNetV2, DenseNet121, InceptionResNetV2, NASNetMobile, EfficientNetB2. In the resulting model, we tested three Thai Sign Language datasets consisting of one learning-related dataset and two independent data sets, Unseen-TFS and KKU-TFS of Pariwat and Seresangtakul in 2017 (Pariwat & Seresangtakul, 2017) to test the performance of the developed model.

This thesis is based on the following publications.

Lata, S., & Surinta, O. (2022). An end-to-end Thai fingerspelling recognition framework with deep convolutional neural networks. *ICIC Express Letters ICIC International c*, 2022(5), 529–536. <https://doi.org/10.24507/icicel.16.05.529>

Part 2, we focus on solving a single gesture recognition problem by presenting a method for recognizing informative information in multiple gestures. We will use the whole set of Thai Sign Language consonant VDO while still minimizing the irrelevant portion of the image by using the state-of-the-art YOLO v5 architecture to detect person images before sending them to learn with CNN. We will compare the MobileNet V2, ResNet50, and DenseNet201 architectures and use Fusion CNNs of the three architectures which are MobileNetV2+DenseNet201, MobileNetV2+ResNet50 and ResNet50+DenseNet201. In addition, we will conduct a comparative experiment in conjunction with the RNN architectures GRU and LSTM to assist in hierarchical visual recognition enabling single and multiple gesture recognition alphabet or VDO formats.

This thesis is based on the following publications.

Lata, Siriwiwat, et al. (2022). Dynamic Fingerspelling Recognition from Video using Deep Learning Approach: From Detection to Recognition. ICIC Express Letters ICIC International B, 2022, 949–957. 13(9). 10.24507/icicelb.13.09.949

Finally, we focus on providing the system with comprehensive recognition of Thai Sign Language, alphabet, words, and sentences to communicate meanings by using multiple and continuous sign language gestures to achieve the proper architecture for learning words and sentences in the perfect Thai sign language. The researcher chose to extract specific features in the interpretive area. The area of the image was reduced to the unrelated areas with person detection using the YOLO v5 architecture. Consequently, this research requires CNN-based deep learning in combination with an LSTM architecture with hierarchical learning properties. In addition, this study used the properties of CTC as an output type neural network and a recognition function for training recurrent neural networks (RNNs) to solve the problem of variable gesture sequences of sentences in Thai sign language. As a result, a system that can automatically translate sign language is comprehensive.

1.5 The automated Thai sign language recognition system

Automated sign language recognition systems can provide tangible benefits and improve quality of life for people who rely on sign language to communicate daily (Wadhawan & Kumar, 2021). The main objective of this research was to study deep learning methods and optimization model selection algorithms to select the best model. It's a way to automatically interpret the results of sign language gestures. In this chapter, we present the basic knowledge to understand the basic concepts of the Automatic Thai Sign Language Translation System as follows.

1.5.1 Thai Sign Language (TSL): Thai Sign Language is the official sign language of Thailand and is widely used in Thailand. Therefore, Thai Sign Language is the official language of Thailand for the deaf. In August 1999, it was signed by the Minister of Education on behalf of the Thai government (Reilly, Charles & Suvannus, 1999) to provide Thai Sign Language learning. Thai Sign Language includes gestures, hand movements, and finger spelling in alphabets, words, and sentences. It can be divided into two types: static and dynamic which uses only one hand and both hands.

1) Thai Sign Language consonants are characterized by gestures that consist of gestures with fingers in various shapes representing the alphabet, vowels, tones, and other symbols. It is divided into Static 15 alphabets (Figure 2) and Dynamic 27 alphabets for a total of 42 alphabets, as shown in Figure 3.



Figure 2 Static gestures for all 15 alphabets.
n/k, d/d, t/t, n/n, b/b, p/p, f/f, m/m, y/y, r/r, l/l, w/w, s/s, h/h, and o/o
 respectively.



Figure 3 Examples of dynamic gestures, alphabets *ng*, and *ch*, respectively.

2) Thai Sign Language Sentences and Words: Sign language is not a universal language that can be used in all countries as the use of sign language differs depending on the nature of the language in the country. Thai sign language gestures are characterized by the continuity of gestures to convey the meaning of several words to form sentences that are complex and understandable. There are both Static and Dynamic gestures, making interpretation difficult to understand as shown in Figure.4

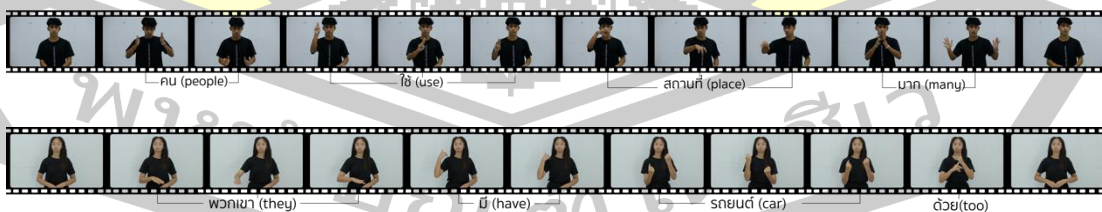


Figure 4 Examples of gestures: words, and sentences in Thai Sign Language

Thai sign language is like the common language of the hearing impaired, therefore communication needs to be perceived by both parties, especially the recipients who are normal people. Most of them are people who do not have the knowledge of sign language, so each communication requires an interpreter, which are very few nowadays. Therefore, using technology to help the hearing impaired communicate with

normal people requires image processing and machine learning techniques based on gestures to recognize alphabets, word, and sentence gestures. Several countries, such as Wenjin Tao et al. (Tao et al., 2018), use CNNs to recognize American Sign Language (ASL) characters from Microsoft Kinect machines. Farhad Yasir et al (Yasir et al., 2017) performed Bengal Sign Language recognition using CNN network to recognize sign language usage data from Leap motion controller (LMC). However, in-depth detector learning, depth vision remains a matter of adoption in real-world environments but with the popularity and efficiency of CNNs, researchers are focusing on developing these architectures. However, the popularity of effective deep learning has resulted in researchers focusing on architecture development, as shown in the next section.

1.5.2 Deep learning for sign language recognition

Deep learning is the use of automated computer software based on the concept of neural networks in the human nervous system. In addition, it is a form of machine learning that uses processing on a complex structure. Deep learning has been used to translate continuous sign language communication quickly and easily. The proposed models are quite effective for a wide range of tasks, but none currently has the potential to be used for both problem-solving and commercialization (Al-Qurishi et al., 2021). Sign language recognition entails pattern matching, computer vision, linguistics, and other aspects of natural language processing (Yin et al., n.d.). According to the network structure, two architectures are commonly used: recurrent neural networks (RNNs) with at least one recurrent layer and convolutional neural networks (CNNs) with at least one convolutional layer for different tasks depending on the number and type of layers. These networks contain training procedures that affect the performance of the algorithm. Specific data sets will help strengthen network training, so the quality of training sets is an important factor. Further customization of the model can be made by changing some of the associated hyperparameters that are defined in the training process. (Lecun et al., 2015). The application of deep learning techniques is widely applied to effectively solve problems such as detection, classification, and grouping. In this research, the unique properties of deep learning are used in the detection of objects as well, which will be explained in the next section.

1.5.3 Convolutional Neural Networks (CNNs)

Convolutional neural networks are architectures that learn based on neural networks that receive an input image (Al-Qurishi et al., 2021). Its highlight is that it can directly extract features and classify information which is used to recognize or classify things from complex data and is popular for image recognition. A convolutional neural network is a multilayer perceptron. (Y. LeCun, B. Boser, J.S. Denker, D. Henderson, R. Howard, W. Hubbard, 1990) CNN's work uses complex mathematical methods that contain a large amount of data. Deep learning, in addition to popular architectures, includes Long Short-Term Memory (LSTM), Recurring Neural Networks (RNN), and Deep Belief Networks (DBN) (Shrestha & Mahmood, 2019) In image

recognition, data is automatically extracted with convolution-layer learning generated by the weight of feature maps. A typical CNN structure starts with the input image and then features extraction of the image designed with the Convolutional layer and Pooling layer (LeCun, n.d.) It is a mathematical computation when the characteristics are fed to the Fully connected layer's neural network, the prediction layer, with the computation for classification or image recognition, shown in Figure5.

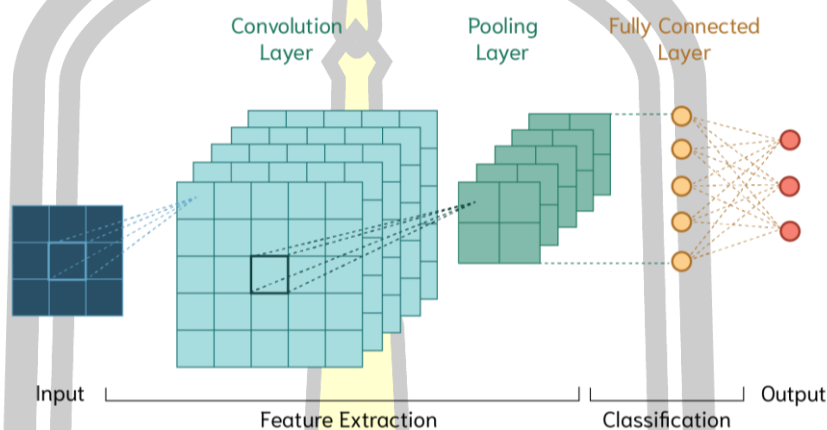


Figure 5 illustration of the convolutional neural network architecture

Convolutional Layers calculate the value of each kernel image and its weight through a function called Activate Function to send data to pooling, which is a computational layer to reduce image detail. It will select the largest value in each kernel derived from the image to represent the next generation of data to compute the kernel from the source image from the top left to the right key area, the convolution operation is computed by multiplying the corresponding values from the source image and the kernel and adding them together, as shown in Figure 6

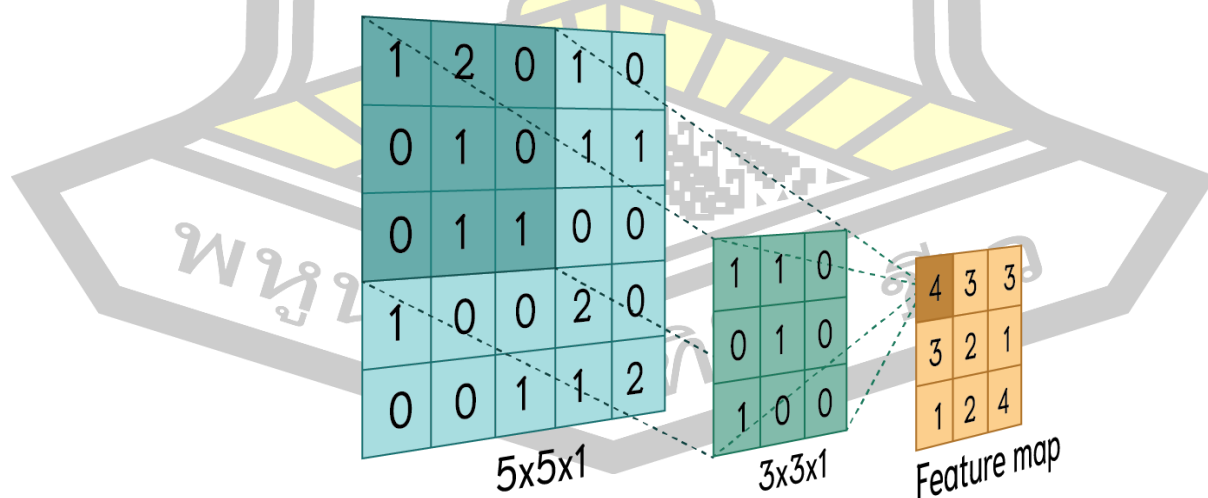


Figure 6 Illustration of the convolutional operation

Hand shape recognition is a process created by the subject's gestures. There is an unclearly high intra-class ambiguity that results in the increased burden of acquiring training data in many cases (Al-Qurishi et al., 2021). CNNs require a much lower level of pre-processing compared to other deep learning algorithms (Koller et al., 2016). Labeled training data (Agathe Balayn et al., 2018). The methodology applied an end-to-end CNN architecture to a training dataset for comparison purposes American Sign Language (Nguyen & Do, n.d.), as well as Ameen and Vadera (Ameen & Vadera, 2017) developed a CNN focused on grouping fingerspelling images using a mixture of image intensity and depth data.

In addition, it has also been found that hand detection is an important preprocessing step, and it can still be difficult to correctly recognize certain hands in cluttered environments, and the intricate rendering of the human hand is agile and has a wide range of movements. Therefore, the use of object detection architecture detects objects to extract the relevant areas into consideration can increase the efficiency of the model.

1.5.3 Object Detection

Object detection is a system for detecting still images or videos to find and recognize the object of an image. The result of detection is displayed in the area of the object in the image and the desired position is determined from the detected object in the image, such as face detection, people detection, object detection, etc. The detection is displayed as a rectangular box indicating the presence of an object in the image, providing important information for learning image classification (Zhao et al., 2019). Most detectors are used to extract image data in the form of feature vectors, which face the problem of tracing and detecting hand parts from a variety of environments both complex backgrounds and hand obscuring (Dewi & Juli Christanto, 2022). However, there are various attempts at hand detection. In the early stages of detection, the theory of Michael J. Jones and James M. Rehg (Jones et al., 1998) was applied to the color of the skin detection, as well as Dardas & Petriu (Dardas & Petriu, 2011). However, this research still has problems when the hand moves to cover other skin areas of one's own. Additionally, Xingbao Meng et al (Meng et al., 2012) performed hand detection using a histogram of oriented gradients (HOG) and a skin color histogram of oriented gradients (SCHOG) are used to help visualize and detect areas of the hand. In addition, the use of border features from hand detection assisted in the analysis of gestures and the various equalizer histograms of Regina Lionnie et al (Lionnie et al., 2011)

Nowadays, the challenge of implementing another effective form of deep learning detection. There are two styles: 1) One-Stage style and 2) Two-Stage style. 1) The one-stage model is a single-stage object detection model which bypasses the regional proposal step of the two-stage model and runs detection directly over a dense sampling of locations. For example, Single Shot Detector (SSD) is suitable for detecting

small objects in images (Liu et al., 2016), which has been applied to robots to control and interact with humans (Gao et al., 2018). YOLO has been developed to increase the speed and accuracy of detection which YOLO v2. For example, Xianlei and Shuying (Qiu & Zhang, 2017) used YOLO v2 to run Grab-and-Go Groceries with a high mAP of 90.53%. Of course, in detecting hand images from normal photos, the hand detection area is smaller but still faster than YOLO v3, optimized to emphasize more accuracy (Redmon & Farhadi, 2018). Additionally, YOLO v4 is faster and more accurate in the dataset (MS COCO AP50 . . . 95 and AP50) than all existing detection models (Bochkovskiy et al., 2020). Moreover, the ultralytics-designed YOLOv5 is characterized by high detection accuracy and good object detection in complex environments (Z. Jin et al., 2021) Trung-Hieu Le et al (Le et al., 2018) used YOLO Architecture to detect hands by optimizing spatial data transfer connections. It provides spatial-transfer connection (STC) between high-level layers and low-level layers to optimize hand detection, which is considered a small object in the picture. 2) Two-Stage Algorithms in this group are R-CNN (Girshick et al., 2014), Fast-RCNN (Girshick, 2015), Faster-RCNN (Shaoqing Ren, Kaiming He Ross Girshick Jian, 2015), and RatinaNet (Lin et al., 2017) has been recognized for its fast and accurate object detection performance. The usage of a process based on Selective Search creates a set of visual proposals scattered about how many objects in the image should be present while examining and managing most of the space and then classifying objects and backgrounds.

1.5.4 You Only Look Once (YOLO) Detectors:

You Only Look Once: Unified, Real-Time Object Detection and released in 2016 (Redmon et al., 2016). YOLO features a single neural network detection to predict bounding boxes from grid lines created with a sliding window to find all objects present in the class. YOLO will divide the input image into a line or grid size $S \times S$ cells. Each cell is responsible for predicting its bounding boxes. Then, it finds the midpoint of the objects in that grid's cell and defines a boundary to find the probability of what is being searched from the grid in each cell. In the next step, it creates a bounding box in the probability of different classes coming together. Then, it will be brought together to select a pair of classes and boxes created that will select the highest score as the answer indicating what the object is and where it is in the image. If no object exists in the grid cell, the YOLO loss function will not optimize for invalid class prediction. The network predicted the probability of only one class per cell regardless of the number of B boxes. The YOLO architecture is similar to GoogLeNet which draws on the DarkNet Architecture with Conv. 1×1 and 3×3 Conv layers and adds class prediction to the vector output fully connected. The last sequence will get the shape $S \times S \times (B * 5 + C)$ as the output Figure 7.

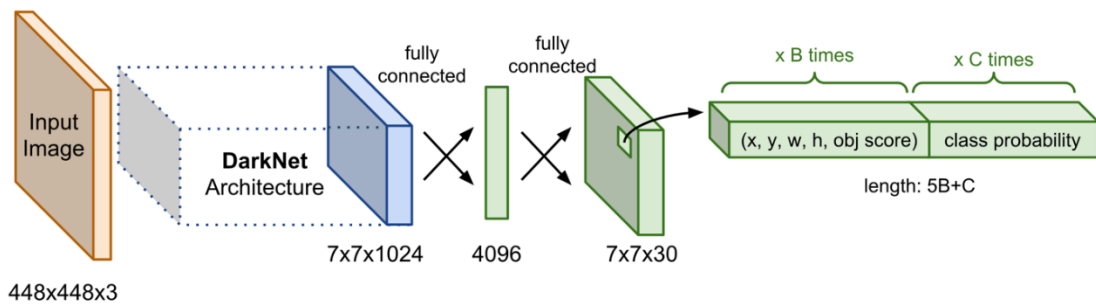


Figure 7 The network architecture of YOLO.

In 2021, YOLO was compared with the research of (Ge et al., 2021) on the MS COCO data set used to train and test the algorithm. It was found that YOLOv5 had better accuracy performance than YOLOv4 and YOLOv3. However, the speed of detection showed that YOLOv3 was faster, so we considered using only these two versions.

In YOLOv3 has been improved to focus more on accuracy and detection of small objects. Improved detection density is achieved by using Darknet 53 as the core to extract features from input images. Afterwards fed to feature pyramid network (FPN) as a neck for feature fusion. It is composed of the several bottom-up and top-down paths and the head is composed of YOLO layer. Finally, YOLO layer generates the results.

In YOLOv5 (Nepal & Eslamiat, 2022) is different from the previous releases. It utilizes PyTorch instead of Darknet and CSPDarknet53 as a backbone. This backbone solves the repetitive gradient information in large backbones and integrates gradient change into the feature map that reduces the inference speed, increases accuracy, and reduces the model size by decreasing the parameters. This backbone solves the repetitive gradient information in large backbones and integrates gradient change into the feature map that reduces the inference speed, increases accuracy, and reduces the model size by decreasing the parameters. It uses the path aggregation network (PANet) as neck to boost the information flow. PANet adopts a new feature pyramid network (FPN) that includes several bottom ups and top down layers. This improves the propagation of low level features in the model.

1.5.5 Long Short-Term Memory (LSTM)

Long Short-Term Memory, developed from a Recurrent Neural Network (RNN), is suitable for use with sequences (Hochreiter & Schmidhuber, 1997), such as video work or extracting words from the text in a book, etc. LSTM was developed to solve the problem of lower gradient values from the back-propagation by storing the state of each node to be used to recognize the true value when a rollback occurs and to decide whether the incoming data should be remembered or discarded with a structure as shown in Figure 8. The internal structure of the LSTM consists of four functions that are used to create functions called gates.

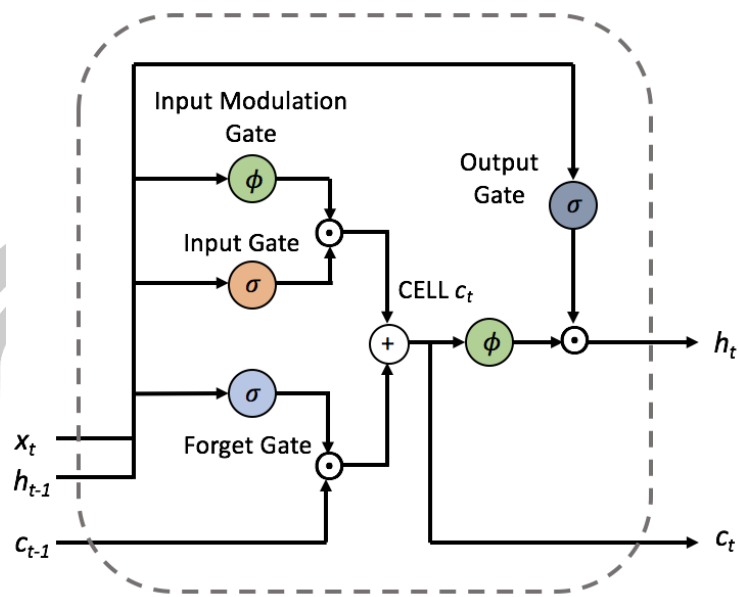
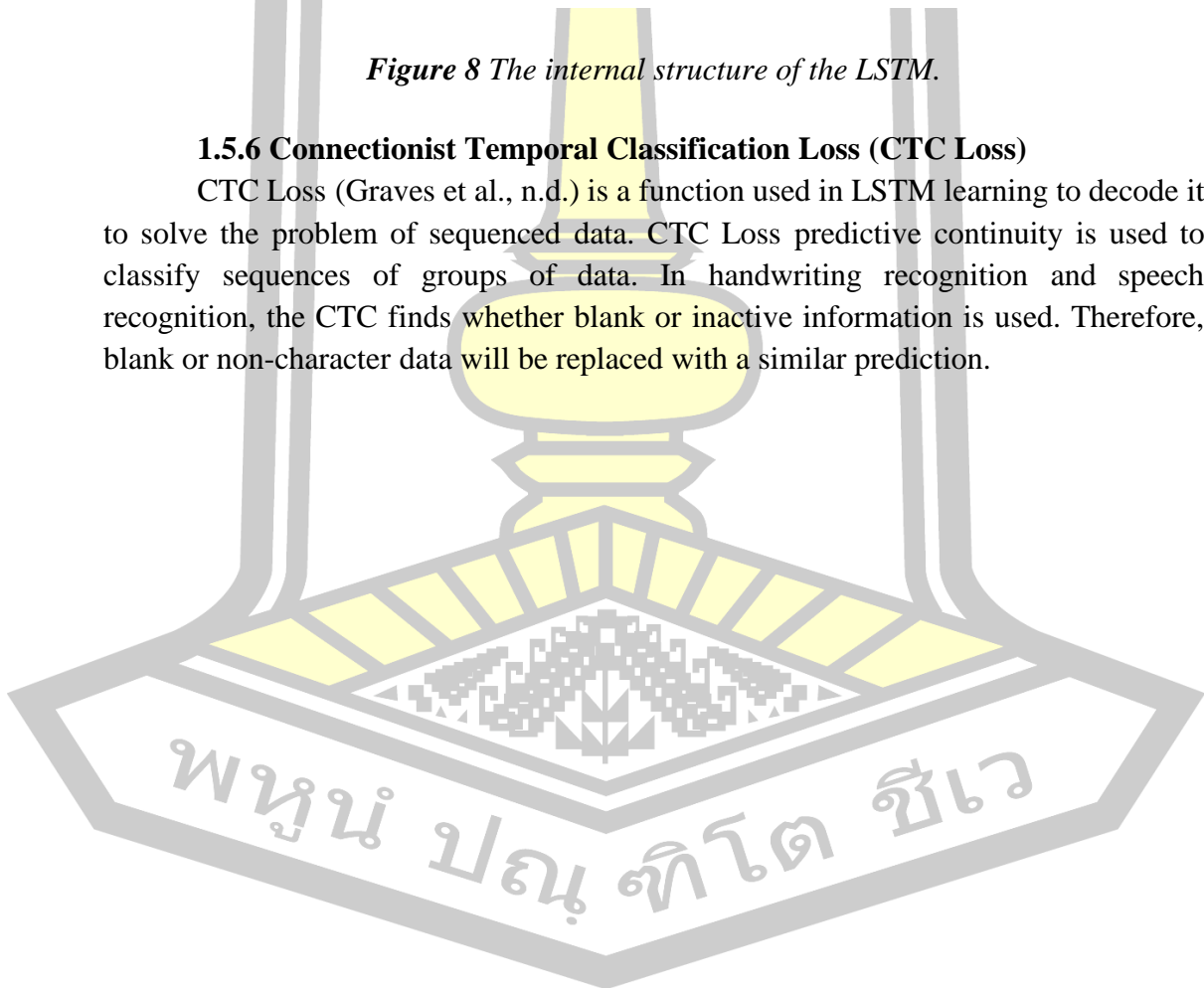


Figure 8 The internal structure of the LSTM.

1.5.6 Connectionist Temporal Classification Loss (CTC Loss)

CTC Loss (Graves et al., n.d.) is a function used in LSTM learning to decode it to solve the problem of sequenced data. CTC Loss predictive continuity is used to classify sequences of groups of data. In handwriting recognition and speech recognition, the CTC finds whether blank or inactive information is used. Therefore, blank or non-character data will be replaced with a similar prediction.



Chapter 2

An End-to-End Thai Fingerspelling Recognition Framework with Deep Convolutional Neural Networks

The WHO reports that approximately 34 million people worldwide are deafness and hearing loss. In 2050, it will increase to 900 million people. It is essential to communicate with the hearing impaired in hand sign language. This paper proposed an end-to-end fingerspelling recognition framework of the Thai sign language based on deep convolutional neural networks (CNNs). Future, we divided our framework into two processes. In the first process, we focus on the detection of hands using the YOLOv3 objection detection framework. In the second process, we proposed using five CNN architectures; MobileNetv2, DenseNet121, InceptionResNetV2, NASNetMobile, and EfficientNetB2, to create the most robust model that provides high recognition performance. Hence, we evaluated the proposed framework to detect and recognize three Thai fingerspelling (TFS) datasets, including TFS, KKU-TFS, and Unseen-TFS. As a result, we found that the YOLOv3 showed a high precision value on the TFS dataset. However, the worst performance presented on KKU-TFS and Unseen-TFS datasets. But, our proposed framework could not detect hands from only one image on the KKU-TFS and Unseen-TFS datasets. Therefore, we also examined the CNN architectures to recognize the 1-stage Thai fingerspelling images. The experimental results showed that the DenseNet121 obtained an accuracy of 93.99% on the TFS dataset and 90.40% on the KKU-TFS dataset.

2.1 Introduction

Humans are considered as a valuable resource. Various factors that affect human beings are imperfect in their physical condition causing disabilities, such as congenital disabilities and accidents. World Health Organization (WHO) reported that over 1 billion people, or approximately 15% of the world's population who were disabled. Globally, more than 1.5 billion people experience some decline in their hearing capacity during their life course, of whom at least 430 million will require care. It is estimated that by 2050 over 700 million people will have disabling hearing loss (World Health Organization., 2021). Therefore, it is assumed that these groups need to communicate with each other using sign language, which is likely to cause communication problems with ordinary people.

Many researchers had developed technologies facilities for the disabled, (Pornthep Sarakon et al., 2021) especially the hearing loss in order to improve the efficiency of sign language communication, such as sensor technology and depth camera. (Hoang et al., 2020) Because of the expensive technology, researchers focus on improving image and video recognition using artificial intelligence techniques. Zhao et al. (Yang et al., 2016) proposed using only a webcam to communicate with disabled

people. These techniques have detected hand movement and interpret hand gestures that disabled people express to ordinary people.

Due to the complexity of sign language, researchers designed and developed processes for static-sign language recognition, which means one gesture has only one meaning. Pariwat & Seresangtakul (Pariwat & Seresangtakul, 2017) and Nakjai & Katanyukul (Nakjai & Katanyukul, 2019) created a static sign language image dataset. The image collections are created by setting the scene and assigning the background color to contrast with human skin color. However, the black suit is also worn to be able to detect the hand area.

In actual situations, sign language interpretation may not be in the setup area. It harmful impacts the algorithm that unable to detect and recognize hand correctly. However, deep learning algorithms, such as single shot detector (SSD) (Liu & Kehtarnavaz, 2016) and YOLO (Redmon et al., 2016), can be proposed to detect the hand area without a black background, making it possible to detect hands in any environment.

2.2 Related work

2.2.1 Hand detection. In 2018, the improved single shot multi box detector (SSD) with the VGG16 architecture as a backbone module was proposed by Gao et al. (Gao et al., 2018) The improved SSD method was invented to detect small target hands for space human-robot interaction in a real-time environment. It increased the detection of the small target hands by adding the feature fusion module.

Le et al. (Le et al., 2018) proposed the YOLO architecture using the spatial transfer connection (STC) for detecting the hands from the complex environment. With the STC operation, the small convolution operation with the size of 1x1 was applied to the input features in each convolution layer to reduce the output feature. These two YOLO architectures were evaluated based on the intersection over union (IOU) value on the hand dataset. The result showed that the YOLO with STC operation outperformed the original YOLO architecture.

In 2019, Bose & Kumar (Rubin Bose & Sathiesh Kumar, 2019) used Faster R-CNN with the InceptionV2 architecture as a backbone module. The Faster R-CNN method was proposed to find out the region proposals and classify the hands from the whole image. This method was trained using three gradient descent optimization techniques (Momentum, RMSprop, and Adam) and train with 35,000 epochs. The result showed that the Adam optimization algorithm performed better than momentum and RMSprop optimizers.

2.2.2 Hand gesture recognition using convolutional neural network. In 2017, Yasir et al. (Yasir et al., 2017) proposed the convolutional neural network to

recognition of the Bangla sign language. First, the Leaf motion device was used to track the hand motion. With the Leaf motion, continuous frames of hand were extracted using the Leaf motion device. Second, the continuous frames were then segmented using the hidden Markov model into the specific state. The time-series data were generated at this state. Finally, the time-series data was sent to train with the CNN method. The experimental result showed that the error rate was decreased from 5% to around 1.5% in only three epochs.

In 2018, Rao et al. (Rao et al., 2018) presented a multi-stage CNN for Indian sign language gesture recognition. In the first stage, the feature extraction module consisted of four convolution layers, four rectified linear unit (ReLU) activation functions, and two stochastic pooling layers. In the second stage, the classification module comprised one dense layer followed by the ReLU layer and a softmax function which was the last layer of the network. The output of the proposed CNN model had 200 units. The multi-stage CNN model obtained a recognition accuracy of 92.88%.

Bhuiyan et al. (Islam et al., 2018) the robust deep features were extracted from human hand gesture images using a deep CNN method. The proposed deep CNN method included five convolution layers and two fully connected layers. The last fully connected created the feature vector of size 4,096 units. Subsequently, the support vector machine (SVM) with a one-against-all strategy was offered for training the deep feature. The result showed that the proposed method provided a recognition accuracy of 94.57% on the American sign language dataset.

2.3 End-to-End Thai Fingerspelling Recognition Framework

This section introduces the end-to-end Thai fingerspelling recognition framework that mainly focuses on 1-stage Thai fingerspelling images, as shown in Figure 9.

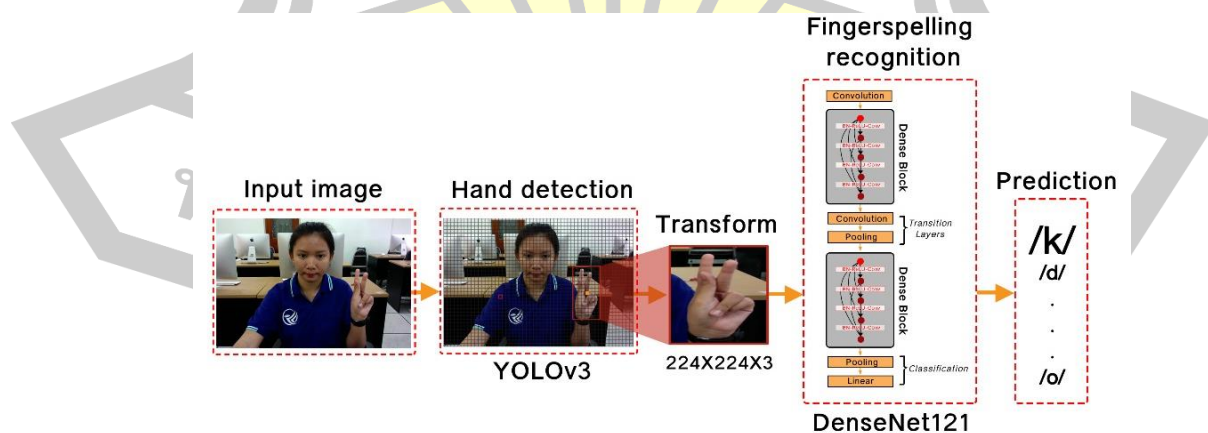


Figure 9 The proposed framework of the 1-stage Thai fingerspelling recognition

2.3.1 Hand Detection using YOLOv3. Redmon et al. (Redmon et al., 2016) proposed YOLO (you only look once) for real-time object detection and could process video in real-time at 45 frames per second, which is implemented and improved the object detection and tracking faster. YOLOv3 consists of two parts; feature extraction and bounding box prediction. In the feature extraction part, the 53 convolutional layers are followed by two fully connected layers for extracting the feature from the grid cell. In the bounding box prediction part, the YOLOv3 algorithm starts with dividing the image into a small grid cell (Dang et al., 2020). Then, the YOLOv3 algorithm will predict the bounding boxes in three different scales from each grid cell and return the class probabilities for those boxes. Finally, these probabilities are used to predict if the grid cell contains target objects, as shown in Figure 10.

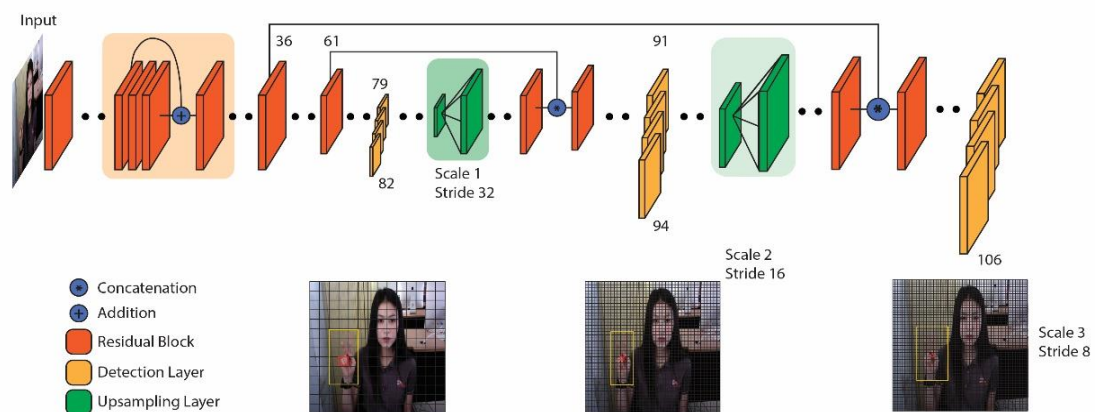


Figure 10 Hand detection with Yolov3 Network Architecture.

2.3.2 1-stage Thai Fingerspelling Recognition with Deep Convolutional Neural Networks. This section described two deep CNNs, including DenseNet121 and MobileNetV2, according to the high performance obtained while evaluating these two CNNs.

DenseNet121. Huang et al. (Huang et al., 2017) proposed a dense convolutional network, called DenseNet. The DenseNet architecture was designed to connect a current layer to other layers in a feed-forward layer. The concept of the DenseNet was that the current layer obtained additional feature maps from all earlier layers. Thus, the feature maps from previous layers were then concatenated with the current feature maps. Further, they connected until the last layer. The advantage of the DenseNet was that it collected all the knowledge of earlier layers. This study used the DenseNet121 architecture consisting of five convolution layers; convolution layer, max-pooling layer, dense block, transition layer, and classification layer. In addition, the classification layer was set as 15 dimensions fully connected and classified with the softmax function.

MobileNetV2. Sandler et al. (Sandler et al., 2018) invented a new lightweight architecture that the extended version of the MobileNet, called MobileNetV2. This architecture was designed based on inverted residual and linear bottlenecks. The depthwise convolution operations were also proposed to create a lightweight architecture because it allowed factorizing the convolution layer into two separate layers; depthwise and pointwise convolution layers. Furthermore, it decreased the parameters to calculate. As a result, the model was relatively small and reduced the chance of overfitting.

2.4 Fingerspelling Datasets

2.4.1 Thai Fingerspelling Dataset (TFS). This research focused only on the one-stage fingerspelling of Thai sign language that contained 15 signs (/k/, /d/, /t/, /n/, /b/, /p^h/, /f/, /m/, /y/, /r/, /l/, /w/, /s/, /h/, and /o/).

The TFS dataset used in the experiments included 7,200 images, 480 images in each class, and 1280*720 pixels resolution. We collected the Thai fingerspelling consonants with the support of 14 undergraduate students in the special education department who have experience using Thai sign language and 50 undergraduate students with no experience in using Thai sign language. We recorded the images in complex and non-complex backgrounds. Examples of the TFS dataset are shown in Figure 11.



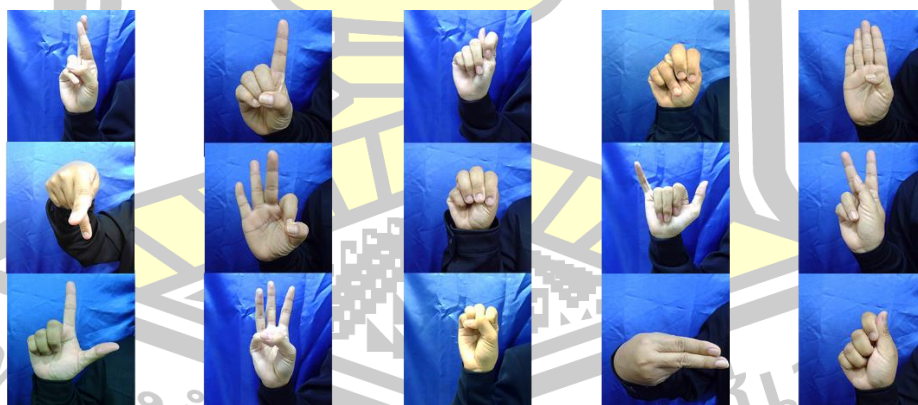
Figure 11 Examples of the 1-stage Thai fingerspelling consonants that recorded in (A) non-complex and (B) complex backgrounds.

2.4.2 Unseen Thai Fingerspelling Dataset (Unseen-TFS). We also proposed the unseen Thai fingerspelling dataset, called Unseen-TFS, to evaluate Thai fingerspelling detection and recognition algorithms. Remarkably, four volunteer undergraduate students with no experience using Thai sign language support us in completing this dataset. The Unseen-TFS dataset consists of 300 images of 15 consonants, as shown in Figure 12(A).

2.4.3 KKU Thai Fingerspelling Dataset (KKU-TFS). The KKU-TFS was proposed by Pariwat and Seresangtakul (Pariwat & Seresangtakul, 2017) in 2017. It contains 15 signs of 1-stage Thai fingerspelling and has 375 images, five images in each sign. Each image has a size of 250*288 pixels resolution recorded from five volunteers. The samples of the KKU-TFS dataset are shown in Figure 12(B).



(A)



(B)

Figure 12 Examples of the TFS datasets. (A) Unseen-TFS and (B) KKU-TFS datasets

2.5 Experimental Results

We evaluated experiments on the TFS datasets, including Thai fingerspelling (TFS), Unseen-TFS, and KKU-TFS. All experiments were performed using TensorFlow deep learning framework that was running on Intel(R) Core-i5 9400F CPU @ 2.90GHz, 32GB RAM, and GPU NVIDIA GeForce GTX 1080. The experiment results are explained as follows.

2.5.1 Experiments on hand detection using YOLOv3. This experiment mainly focused on using the YOLOv3 real-time object detection framework for hand detection. We concentrated on training the network to identify the location of an object which is a hand in an image. We used the mean average precision (mAP) that compares the output of the network and the ground-truth bounding box to evaluate a model. A higher mAP value is presented the better detection performance.

The trained the YOLOv3 model on the TFS dataset that included 4,320 training images and evaluated on 2,880 test images. The sizes of the input images were 1280*720 and 1080*720 pixels resolution. The parameter settings used to train the YOLOv3 model are as follows: down sample and the batch size are 32 and 4, intersection over union (IOU) and non-maximum suppression are 0.5 and 0.5, and the number of training iterations is 100 epochs.

Table 1 Performance evaluation of the YOLOv3 on the TFS dataset.

Classes	mAP	Classes	mAP
/k/	0.9808	/y/	0.9819
/d/	0.9688	/r/	0.9635
/t/	0.9312	/l/	0.9845
/n/	0.9583	/w/	0.9844
/b/	0.9952	/s/	0.9948
/p ^h /	0.9844	/h/	1.0000
/f/	0.9792	/o/	0.9887
/m/	0.9851	Overall	0.9787

Table 1 showed the high average precision value achieved from the YOLOv3 model with mAP = 0.9787. Hence, the highest mAP value appeared in class /h/ with the mAP value of one and also the lowest value appeared in class /t/ with the mAP value of 0.9312. However, most of the out results could not detect the hand from the complex background, as shown in Figure 4b. Moreover, the most expensive computation time spent while training the YOLOv3 was 77 hours.

Therefore, if the increase of the mAP value is needed, thus, it still has an opportunity to improve the detection performance by training more example images, increase more training iteration, and propose novel data augmentation techniques.



(A)



(B)

Figure 13 Illustration of the output that was detected using the YOLOv3.

(A) The correct hand detection and (B) not detecting hand.

2.5.2 Experiments of the CNN architectures on 1-stage fingerspelling.

We performed the transfer learning technique for the CNN experiments to train on the 1-stage fingerspelling images using five CNN models, including MobileNetv2, DenseNet121, InceptionResNetV2, NASNetMobile, and EfficientNetB2. We also concentrated on performing with a small training set. Hence, we selected the training set with 60%, 50%, 40%, and 30%, which was the 4,320, 3,600, 2,880, and 2,160 images, respectively. For the validation, we tested the YOLOv3 five times and reported accuracy and standard deviation. The validation and test set used in the experiments were 720 and 2,880 images.

Table 2 Performance evaluation of the CNN architectures on the validation and test sets when training with the different numbers of training set

CNNs	Validation and test accuracy (%) with different numbers of training set							
	60%		50%		40%		30%	
	Valid	Test	Valid	Test	Valid	Test	Valid	Test
MobileNetv2	99.47 ± 0.003	98.02	97.77 ± 0.002	96.91	97.38 ± 0.004	95.71	96.17 ± 0.006	93.86
DenseNet121	99.27 ± 0.009	98.04	98.55 ± 0.004	97.96	98.16 ± 0.008	96.57	96.90 ± 0.007	92.35
InceptionResNet V2	97.29 ± 0.004	96.62	96.51 ± 0.005	96.19	97.00 ± 0.006	95.21	93.46 ± 0.002	92.80
NASNetMobile	97.09 ± 0.007	95.23	94.72 ± 0.009	94.69	94.82 ± 0.001	92.81	89.78 ± 0.010	89.00
EfficientNetB2	93.38 ± 0.005	93.09	92.30 ± 0.004	92.11	91.68 ± 0.001	91.96	90.15 ± 0.004	90.06

In Table 2, we presented the experimental results with five CNN models. The results showed that all CNN architectures obtained accuracy above 90%, except the NASNetMobile that gave 89% when training with 30% of the training set. At 60% of the training set, MobileNetv2 and DenseNet121 were performed with the highest accuracy above 99% on the validation set.

We found that both MobileNetv2 and DenseNet121 architectures presented significant results when training with small training data. These two models obtained an accuracy of 93.86 and 92.35%.

2.5.3 Experiments of the End-to-End Thai fingerspelling framework.

We considered the results as showed in Table 2, which showed that the DenseNet121 and MobileNetV2 outperformed other CNN architectures. The detection and recognition results are presented in Table 3.

Table 3 The YOLOv3 detection value (mAP) and the CNNs recognition accuracies (%) of 1-stage Thai fingerspelling experiments on the TFS datasets

Processes	Methods	TFS Datasets		
		TFS	KKU-TFS	Unseen-TFS
Detection	YOLOv3	0.9787	0.6553	0.7944
Recognition	DenseNet121	93.99%	90.40%	82.00%
	MobileNetV2	92.81%	65.33%	74.67%

For hand detection, we trained the YOLOv3 model with 1-stage Thai fingerspelling images and achieved the mAP values of 0.9787, 0.6553, and 0.7944 on the TFS, KKU-TFS, and Unseen-TFS, respectively. However, the mAP value of the

KKU-TFS data was relatively low, but it was detected hand in all 375 images. Consequently, the YOLOv3 was not detected in only one image in the Unseen-TFS dataset from 300 images and was not detected in six images from the 2,880 test images of the TFS dataset.

For 1-stage Thai fingerspelling recognition, we received the detected result of the hand image from the YOLOv3 directly then sent it to the CNN models for recognition. In this case, the detected hand images may localize at not the exact location as in the ground-truth bounding box. The result showed that the recognition accuracy of the TFS dataset was decreased to 93.99% and 92.81% when using DenseNet121 and MobileNetV2. As for the Unseen-TFS dataset, these CNN models (DenseNet121 and MobileNetV2) obtained low accuracy performance with only 82% and 74.67%. Therefore, we conclude that the DenseNet121 can cope well with all TFS datasets, especially with the Unseen-TFS dataset.

In comparison, we compared our result with result in Pariwat & Seresangtakul (Pariwat & Seresangtakul, 2017) reported that their method obtained an accuracy of 91.20% on the KKU-TFS dataset, but it tested on only 75 images. Therefore, our experimental results showed an accuracy of 90.40% when testing 375 images of the KKU-TFS dataset, so we experimented with five times larger images than Pariwat & Seresangtakul.

2.6 Conclusions

This paper aims to introduce an end-to-end Thai fingerspelling framework that performs hand detection and 1-stage fingerspelling of Thai sign language using deep convolutional networks. We first proposed to use state-of-the-art YOLOv3 for training on the Thai fingerspelling (TFS) datasets, including Thai fingerspelling (TFS), KKU-TFS, and Unseen-TFS, to detect and localize the hand. Second, we trained five convolutional neural networks (CNNs), consisting of MobileNetv2, DenseNet121, InceptionResNetV2, NASNetMobile, and EfficientNetB2. The main purpose was to decrease the training examples while training the CNN models. So, we experimented with a small training set (60%, 50%, 40%, and 30%). The results showed that DenseNet121 and MobileNetv2 outperformed other CNN models. MobileNetv2 and DenseNet models, when training with only 30% of the training set, obtained an accuracy above 92%. However, when training with 60%, we obtained an accuracy of 98% from these two CNN models. Finally, we evaluated our end-to-end Thai fingerspelling framework on three TFS datasets. We found that our framework could detect hands from the image quite well and give the high mAP value on the TFS dataset. However, the mAP value significantly decreased when evaluated on the KKU-TFS and Unseen-TFS datasets. But, our framework still detects hands at the correct location. Moreover, our framework showed an accuracy above 90% on TFS and KKU-TFS datasets and slightly reduced accuracy to 82% on the Unseen-TFS dataset.

In the future direction, it is important to increase the performance of our end-to-end Thai fingerspelling recognition framework, including recognition of multistage Thai fingerspelling, or recognition hand sign from the video, called action recognition. Another direction for future work is considering train the CNN models on the limited size of the training set and still provide high recognition accuracy.

Acknowledgment. We would like to thank all participants from Rajabhat Mahasarakham University for their time and interest in contributing to this research. Without engagement from participants in collecting the dataset process, this research would not be possible.



Chapter 3

Dynamic Fingerspelling Recognition from Video using Deep Learning

Approach: From Detection to Recognition

The world health organization found that more than 34 million people suffer from hearing loss and these people need to use sign language to communicate. Hence, the sign language recognition system is proposed to communicate with hearing loss people and others. In this paper, we aim to propose an end-to-end system to recognize the dynamic Thai fingerspelling from video. The proposed system includes two main processes. First, we use the YOLOv5 algorithm for the human detection task. Subsequently, a uniform distribution method is proposed to select the robust frames before applying robust frames to the detection algorithm. Second, we propose dynamic fingerspelling recognition that consists of two deep learning architectures: convolutional neural network (CNN) and long short-term memory (LSTM). We then combine CNN and LSTM, called CNN-LSTM architecture, followed by the recognition block. The recognition block comprises dropout, global average pooling, and SoftMax layers. For the CNN architectures, we evaluated three CNNs: MobileNetV2, ResNet50, and DenseNet201. We found that the proposed ResNet50-LSTM architecture achieved an accuracy of 88.42% on the test set of the dynamic Thai fingerspelling dataset and also prevented the overfitting problem.

3.1 Introduction

Language is essential for the communication of people worldwide. According to a World Health Organization survey in 2021 (World Health Organization., 2021), over 34 million people worldwide suffer from hearing loss, and the number of people with hearing loss is increasing. These people need to use sign language for primary communication, thus resulting in the development of technology to learn sign language from the movements of the hands and arms to convey meaning. The sign language of each country around the world has a different identity.

Deep learning research proposes a communication tool for effective communication between persons with disabilities and people without disabilities. Islam et al. (Islam et al., 2018) proposed deep learning to recognize English sign language. Their research used the convolutional neural network (CNN) to extract features of hand images and learn them with the support vector machine (SVM) method. Phong and Ribeiro (Phong & Ribeiro, 2019) used a capsule network to detect hands and recognize them at the same time. However, there are different methods to find the region of interest (ROI), such as Faster R-CNN and YOLO (Mujahid et al., 2021) (Yu, 2019). These methods use CNN as the backbone, giving the algorithm to find ROI and recognize that particular ROI. In (Phong & Ribeiro, 2019), their method is a static recognition of hand signs, which means that their proposed method can recognize from only one image. It is not possible to recognize a dynamic hand sign.

3.2 Related work

In 2016, Chansri and Srinonchat (Chansri & Srinonchat, 2016) presented a method for detecting hands from complex backgrounds based on the depth image captured by the Kinect sensor for high brightness value objects close to the Kinect sensor. It can detect the hand area because hands are closest to the Kinect sensor. The hand area was then extracted using a histogram of oriented gradients (HOG) and recognized by a neural network.

In 2017, Pariwat and Seresngtakul (Pariwat & Seresngtakul, 2017) proposed a method to recognize Thai sign language by performing only on the static hand sign language with a total of 15 Thai symbols. The Thai sign language data used in the test were hand-only images taken against a blue background, allowing segment hands by transforming the color space from RGB to HSV and then transforming HSV to grayscale. Hence, images were transformed from grayscale into a binary images by selecting a threshold value of 0.45 to extract the hand area. Then, global and local features were extracted. Finally, all the features were learned using the SVM method with the RBF kernel and achieved with 91.20% accuracy.

In 2019, Nakjai and Katanyukul (Nakjai & Katanyukul, 2019) presented a method of hand sign recognition for Thai fingerspelling. Their research used a hand extraction method by transforming RGB color images to YCbCr. Then, the skin region was examined by the brightness values from the chroma channels. Then, extract the hand area and learn with CNN, which can recognize 25 classes. The results showed a recognition accuracy of 91.26%.

The limitation of Thai fingerspelling sign language is static hand sign which is only 15 Thai alphabets (Nakjai & Katanyukul, 2019) and in a recognition process designed to recognize only from one image so that it is unable to recognize dynamic hand signs as it requires more than one input image.

In 2020, Sugandi et al. (Sugandi et al., 2020) proposed hand gesture recognition using a back-propagation neural network (BPNN) based on visual tracking in a real-time environment. They used skin color detection based on a YCbCr color filter to extract the hand region from the background and then converted it to grayscale and binary images to speed up the processing time. The hand feature from the binary image of the hand region was divided into six regions. Finally, six regions of the hand feature of each gesture became the input data of the BPNN. The results showed that the BPNN achieved an accuracy of 86.67%.

In 2021, Pariwat and Seresngtakul (Pariwat & Seresngtakul, 2021) designed a Thai sign language recognition system that could recognize dynamic hand signs by receiving video inputs, which can recognize Thai hand sign in up to 42 Thai symbols. Moreover, Nakjai and Katanyukul (Nakjai & Katanyukul, 2021) designed an

automatic Thai finger spelling transcription system that can transcribe Thai Sign Language from the video by being able to classify Thai sign language with 20 Thai characters and 20 vowels. Their system consists of three steps: alphabet separation, sign recognition, sign-sequence classification.

In this paper, an automated end-to-end fingerspelling recognition system is proposed to solve human detection and dynamic fingerspelling recognition tasks. First, we select robust frames from the video using the uniform distribution method and send those frames to detect humans using the YOLOv5 algorithm automatically. The sequence frames are given to extract the sequence pattern and temporal features using CNN and LSTM architectures. For that, we combine state-of-the-art CNN with an LSTM network, called CNN-LSTM, including MobileNetV2-LSTM, ResNet50-LSTM, and DenseNet201-LSTM. Further, we add a recognition block containing; 1) a dropout layer to prevent the overfitting, 2) a global average pooling layer to average output and reduce the dimension of feature maps, and 3) the softmax activation function to recognize dynamic fingerspelling from video. The proposed system also enhances the accuracy and computation time. In addition, we collected 3,025 videos of dynamic Thai fingerspelling, consisting of 42 Thai alphabets, that were taken from 14 volunteers who are proficient in Thai sign language.

The rest of this paper is organized as follows; In Section 2 presents the end-to-end system of dynamic fingerspelling recognition. The Thai fingerspelling dataset is described in Section 3. The experimental results are presented in Section 4. The conclusion and future direction are concluded in the last section.

3.3 Dynamic Fingerspelling Recognition System Architecture

We divided the proposed architecture into three main parts: frame selection, human detection, and dynamic fingerspelling recognition. The proposed architecture is shown in Figure 16 and the details of the three parts are in the following sections.

3.2.1 Frame Selection. Generally, videos were recorded with unequal length. In this paper, we first cut frames at the start and end of the video with 10% and 5%, respectively, because these parts contain unnecessary information. Second, we computed the jump ratio over the video frame using the following equation: $jump\ ratio = \text{floor}(No. of\ all\ frames / No. of\ selected\ frames)$, where $No. of\ selected\ frames$ in our experiment is 32. For example, if the jump ratio was 5, we used frames no. 1, 6, 11, ..., N , where N is the last frame. Finally, we randomly selected only 32 frames using the uniform distribution method which the probability of each frame is equal chances to select, as shown in Figure 14.

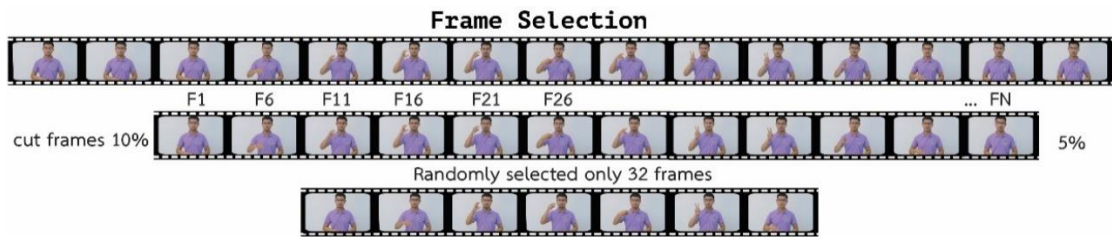


Figure 14 Frame Selection method.

3.2.2 Human Detection. To build an end-to-end fingerspelling recognition system, this paper first aims to detect humans from the videos using a deep learning approach. We proposed to use the fast and accurate YOLOv5 (You Only Look Once) algorithm because we required an algorithm that detects humans in a real-time condition. YOLOv5 [12], the latest update algorithm, the pre-trained model that trained on the COCO dataset was proposed to detect the human from the video, as shown in Figure 15. In this paper, we did not experiment on human detection. We only applied the YOLOv5 algorithm.



Figure 15 Human Detection with YOLOv5.

As shown in Figure 16, after we extracted images from the video, we applied the YOLOv5 algorithm to detect humans as the region of interest (ROI) and send that ROI to the CNN model.

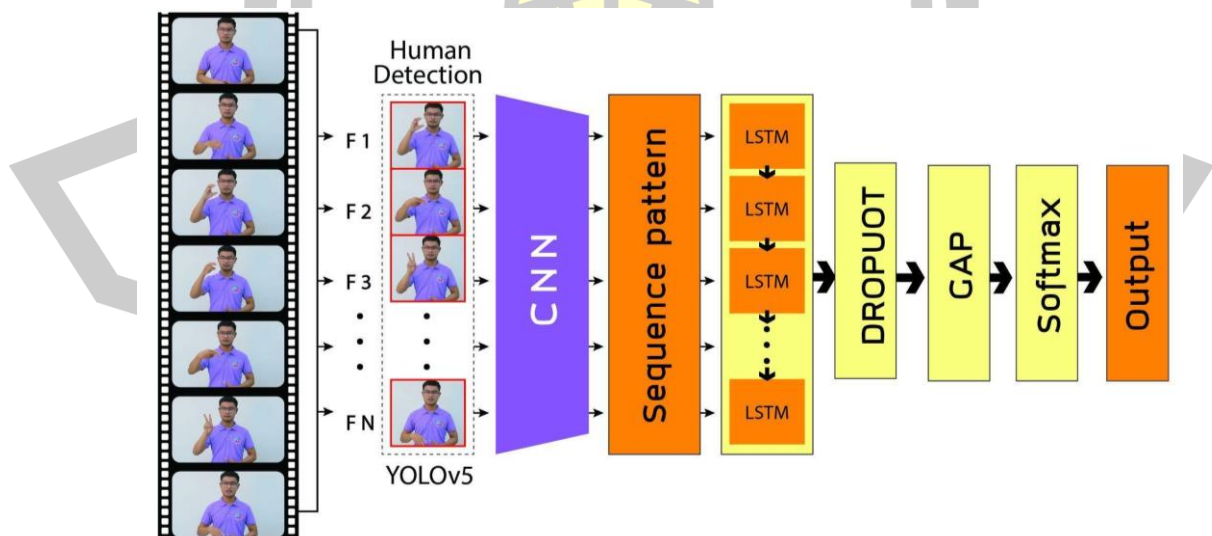


Figure 16 Illustration of the dynamic fingerspelling recognition system

3.2.3 Dynamic Fingerspelling Recognition. Dynamic fingerspelling recognition combines two deep learning approaches: CNN and RNN. The details of the deep learning approaches are described as follows.

CNN. We used different architectures of CNN: MobileNetV2, ResNet50, and DenseNet201, to extract robust deep features from a short video. In this research, the video was first extracted into a single frame (image) and then given to the CNNs to extract the spatial features from the image. We briefly described three CNN architectures in the following section.

MobileNetV2. We used the lightweight MobileNetV2 architecture in our experiments. MobileNet was invented for mobile visual applications (Sandler et al., 2018) because it used depth wise separable convolution to reduce the number of CNN parameters. In addition, MobileNetV2 was designed based on an inverted residual structure and linear bottleneck. The bottleneck block contains three layers (1x1 conv2D with ReLU6, depthwise convolution with ReLU6, and linear 1x1 conv2D) and each layer has the expansion factor that expands the convolutional layers. Further, the residual was proposed to connect two bottleneck layers, like ResNet architecture.

ResNet50. Plain architectures, such as VGGNet and Alexnet, could have problems of vanished and exploding gradients when using very deep layers. The residual network (ResNet) was proposed to solve these problems by introducing the shortcut connection that does not transform the input from the previous to the following building blocks (called bottleneck design)(Kaiming He et al., 2016). In the bottleneck, the 1x1 conv2D is added at the beginning and end of the block to reduce the number of parameters in the network. In our experiments, we used ResNet50 which contains only 50 layers.

DenseNet201. The idea of connections between layers was invented in DenseNet201 and was the same as the ResNet. In the DenseNet architecture (Huang et al., 2017), instead of shortcut connections, dense connections were used to connect between the current and following layers. Hence, the feature maps from the current layer were concatenated and passed to the next layers during forwarding propagation. At the same time, the backpropagation process sent back the error signal from the final classification layer to the earlier layers to adjust the parameters. In this paper, we used the DenseNet201 architecture that contained convolution, pooling, four dense blocks, three transition layers, and a classification layer.

In our proposed network, the last layer of CNN architectures was the global average pooling (GAP) because the fully connected layer (FC) and softmax function were removed from the CNN architectures. After applying the GAP layer, the size of the feature map in each CNN was as follows: MobileNetV2=1x1280, ResNet50=1x2048, and DenseNet201=1x1920. Furthermore, we extracted 32 images from the video and then computed sequence patterns from sequence images. Hence, the

sequence patterns given to the recurrent neural network (RNN) of each CNNs were as follows: MobileNetV2=32x1280, ResNet50=32x2048, and DenseNet201=32x1920.

RNN. The RNN architecture was proposed to learn the sequence pattern of the spatial features extracted by CNN architectures. In the RNN architecture, we added dropout and GAP layers to avoid the overfitting problem and to decrease the size of the feature map before giving them to the softmax layer. We experimented with two RNN architectures: LSTM and GRU. The details are presented as follows.

Long short-term memory. LSTM is a sequential network proposed to learn sequence patterns and address the vanishing gradient problem (Soliman et al., 2019). It has memory blocks that contain three gates: input, output, and forget gates. The LSTM also has a memory cell that can be recurrently self-connected. The memory blocks are designed to control the information inside the LSTM unit that has an advantage in remembering past information and forgetting unnecessary information from the network.

Gated recurrent unit. The GRU was designed to be similar to the LSTM network, but simpler than it (Chen et al., 2021). The GRU network could also solve the vanishing gradient problem that constantly occurs when using the RNN architecture. It contains only two gates: reset (short-term memory) and update gates (long-term memory). The GRU can store sequence patterns from long ago and release irrelevant patterns to the output. Further, the GRU trains faster than the LSTM network. In this paper, we compared the performance of two hidden state sizes (32 and 64 hidden states) of LSTM and GRU.

Recognition block. After extracting the temporal features using RNN architecture, we decided to include two extra layers followed by the softmax layer. 1) The dropout layer to prevent the model from overfitting problems, and 2) A layer to calculate the average output of each feature map using the GAP operation. It also could reduce the dimension of the feature map. The softmax layer is the last layer that computes the final probabilities and recognizes the output.

3.4 Dynamic Thai Fingerspelling Dataset.

Dynamic Thai fingerspelling is a short video database with 3,025 videos and 42 classes. The videos of dynamic Thai fingerspelling were recorded using a DSLR camera using a frame rate of 50 frames per second. We recorded the videos with 14 volunteers who can use Thai sign language fluently. The recorded length of the videos was around 1-5 seconds. Examples of the DTF dataset are shown in Figure 17.

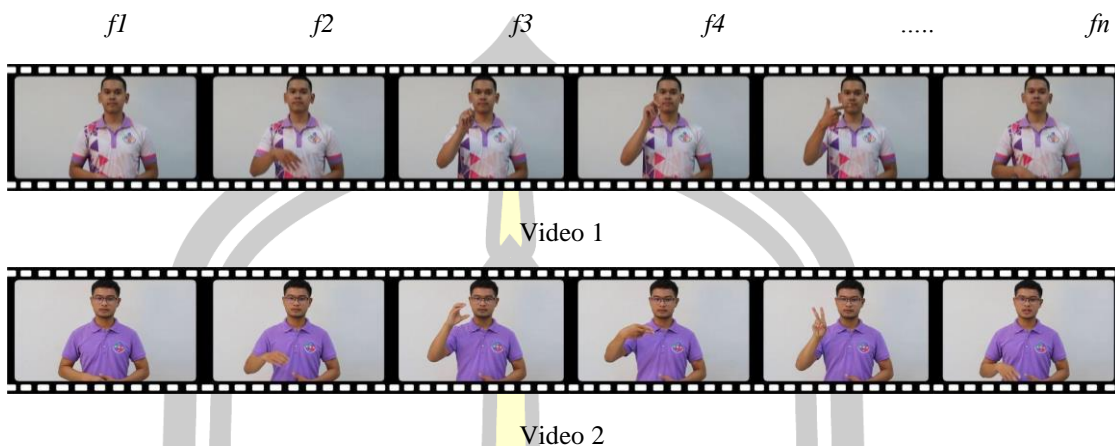


Figure 17 Examples of dynamic Thai fingerspelling dataset.

Note that, $f1, f2, \dots, fn$ means frame number 1, 2, ..., n.

3.5 Experimental Results and Discussion.

The proposed framework was implemented based on the TensorFlow platform running on Intel(R) Core (TM) i7-4790 Processor 3.6GHz, 16GB DDR4 RAM. We trained all deep learning models with these parameters: stochastic gradient descent (SGD) optimizer, the momentum of 0.9, rectified linear unit (ReLU) activation function, and 500 epochs training. Furthermore, we performed the 5-fold cross-validation (5-cv), test accuracy, and testing time as for the evaluation metrics.

3.4.1 Experiments with CNN models using different learning rates. We experimented by extracting the deep sequence features using three CNN models: MobileNetV2, ResNet50, and DenseNet201. Then, all features were given to the LSTM architecture using 32 units to extract temporal features and classify by using the SoftMax function. In this experiment, the CNN parameters were adjusted according to the previous section. Further, we mainly study the impact of the learning rate used while training the CNN-LSTM model using 0.001, 0.0001, and adaptive learning rate that reduces the value from 0.01 to 0.0001. We trained the CNN models on the training set and used the validation set for evaluation. The experimental results are shown in Table 4.

Table 4 Experiment with different Learning rates.

CNN-LSTM Model	Test Accuracy (%) with Different Learning Rates		
	0.001	0.0001	Adaptive
MobileNetV2-LSTM	81.98	80.98	79.01
DenseNet201-LSTM	82.97	72.14	77.76
ResNet50-LSTM	87.93	82.14	81.81

In Table 4 shows the experimental results from three CNN models, when using a learning rate of 0.001 which achieved the highest accuracy on the validation set. As a result, the best CNN model was the ResNet50 which achieved 87.93% accuracy. Note that we subsequently used the learning rate of 0.001 while training the CNN-LSTM models in the following experiments.

3.4.2 Experiments with CNNs and RNNs using different RNN sizes. In this section, we evaluated the performance of the CNNs and RNNs (LSTM and GRU) using two different numbers of units: 32 and 64 units.

Table 5 Comparison between CNN-LSTM and CNN-GRU with different RNN units.

CNN Models	No. of Units in RNNs			
	LSTM		GRU	
	32	64	32	64
MobileNetV2	81.98	83.96	71.23	72.23
DenseNet201	82.97	84.62	72.14	74.24
ResNet50	87.93	89.16	77.52	77.68

As shown in Table 5, it undoubtedly shows that the LSTM, when using both 32 and 64 units, outperformed the GRU. The maximum accuracy of the ResNet50-GRU with 64 units was only 77.68%. In contrast, the ResNet50-LSTM with 64 units had the best performance and achieved 89.16% accuracy on the validation set. In addition, the accuracy obtained above 81% when the CNN model combines with the LSTM. Note that we chose the number of units in the LSTM as 64 units in the following experiments.

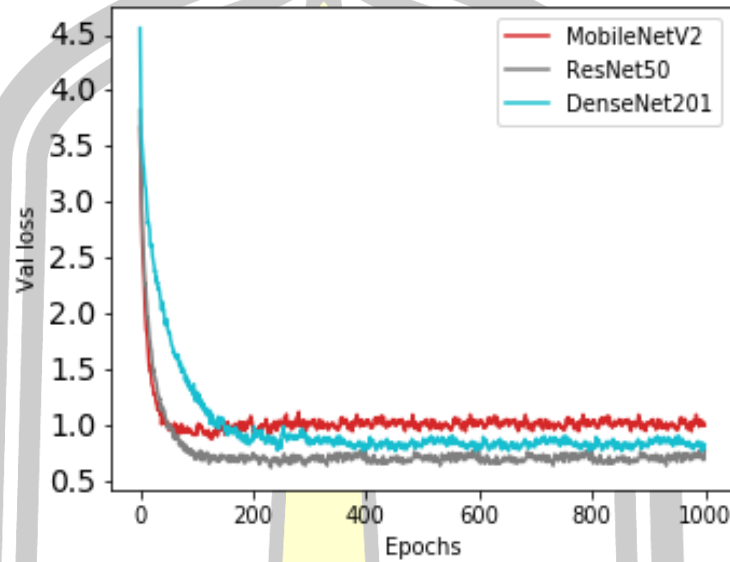
3.4.3 Experiments with CNN-LSTM. In this experiment, as shown in Figure 16, we decided to add the extra densely connected layer with 64 units between the GAP layer and softmax activation function. In the densely connected layer, the dot product was applied as a non-linear transformation between the weighted parameters and the output of the GAP layer.

Table 6 Comparison between CNN-LSTM and CNN-LSTM with a densely connected layer.

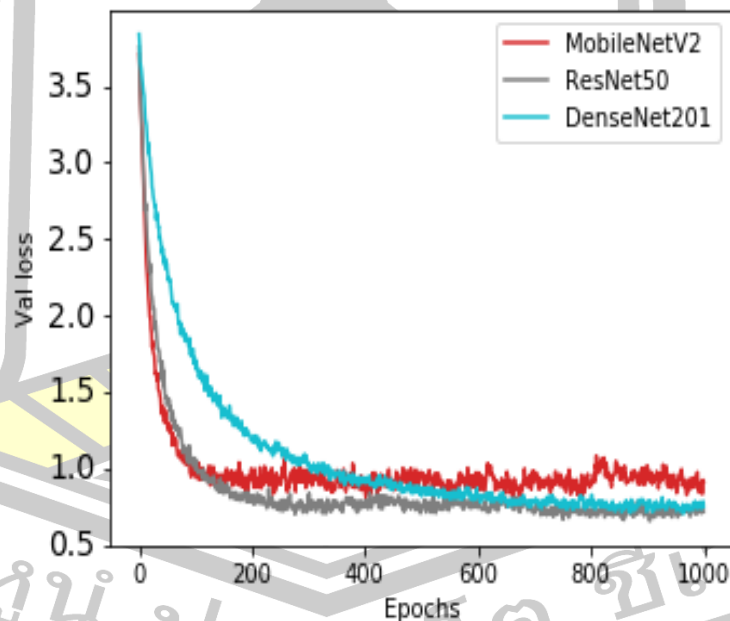
CNN Models	LSTM			LSTM with a densely connected layer		
	5-CV	Test Accuracy (%)	Testing time (seconds per video)	5-CV	Test Accuracy (%)	Testing time (seconds per video)
MobileNetV2	83.95±2.525	83.41	6.70s	82.54±3.006	82.14	6.73s
DenseNet201	84.64±2.462	82.97	7.04s	79.45±1.426	76.85	7.07s
ResNet50	89.15±2.571	88.42	9.50s	86.45±2.146	85.45	9.53s

The experimental results comparing CNN-LSTM and CNN-LSTM with one densely connected layer are shown in Table 6. The results showed that the LSTM

without a densely connected layer outperformed LSTM with one densely connected layer. As a result, the ResNet50-LSTM achieved the best accuracy on both the 5-CV and test set with 89.15% and 88.42%. We confirmed that the proposed network with a combination between CNN and LSTM prevented the overfitting problem.



(A)



(B)

Figure 18 Illustrated the validation loss values of CNN-LSTM models. The loss values of CNN-LSTM (A) without and (B) with a densely connected layer.

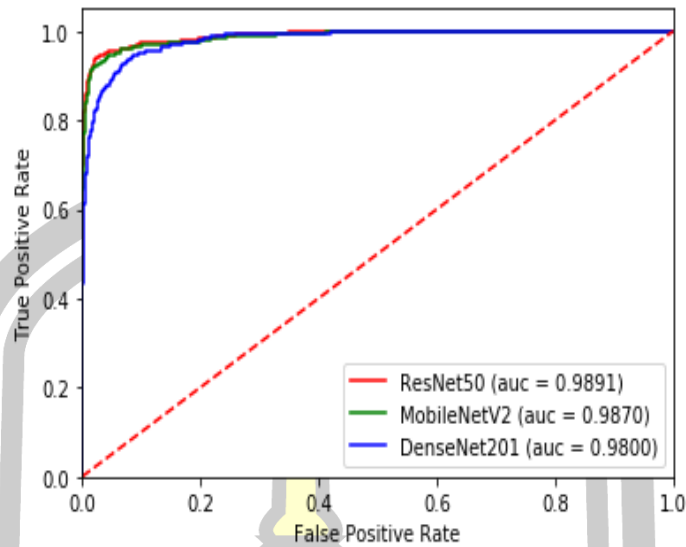


Figure 19 Illustration of the ROC curves for dynamic fingerspelling recognition using different CNNs.

The validation loss values of CNN-LSTM architecture are shown in Figure 18. Figure 18(A) shows that the loss values decreased quickly at approximately epoch 50. Also, the loss values of Figure 18(A) are smoother than Figure 18(B). Consequently, the loss values of Figure 18(A) indicated that the CNN-LSTM without a densely connected layer was trained with the optimal architectures and parameters.

Additionally, the receiver operating characteristic (ROC) curves for CNN models are shown in Figure 19. The ResNet50 had the greatest AUC (0.9891) in dynamic fingerspelling recognition.

3.4.4 Accuracy of Fusion CNNs-LSTM architecture. In other papers, the fusion architecture between two CNNs and LSTM outperformed the single CNN and LSTM. We demonstrate the impact of the fusion CNN-LSTM architectures, as shown in Table 7.

Table 7 Evaluation of Fusion CNNs-LSTM.

Fusion CNNs	LSTM		
	5-CV	Test Accuracy (%)	Testing time (seconds per video)
MobileNetV2+DenseNet201	85.29±3.395	84.29	13.74s
MobileNetV2+ResNet50	89.42±3.670	88.62	16.20s
ResNet50+DenseNet201	88.36±2.917	87.76	16.54s

Table 7 shows the results of experiments with combinations of CNNs: MobileNetV2+DenseNet201, MobileNetV2+ResNet50, and ResNet50+DenseNet201, using the concatenation operation before sending the sequence patterns to the LSTM

network. The combination of MobileNetV2 and ResNet50 outperformed other combination CNNs. The fusion CNNs (MobileNetV2+ResNet50)-LSTM achieved an accuracy of 89.42% and 88.62% on the 5-CV and test sets, respectively.

Consequently, the Fusion CNN-LSTM showed slightly higher accuracy compared with the single CNN-LSTM, but the result was not significant at $p < 0.05$. However, the Fusion CNN-LSTM architecture spent much more time on recognition than using only a single CNN-LSTM architecture.

3.5 Conclusions

We presented an automated end-to-end dynamic fingerspelling recognition system that contains frame selection, human detection, and fingerspelling recognition. We focused on the deep learning approaches; the YOLOv5 algorithm for human detection and CNN-LSTM architecture for recognizing the dynamic fingerspelling from the video. First, 32 video frames were selected using the uniform distribution method and sent to extract the sequence patterns using CNN architecture. Second, the sequence patterns were given to the LSTM architecture, followed by three layers: dropout, GAP, and SoftMax activation function. Moreover, we mainly evaluated the recognition performance of CNN-LSTM architectures. The results obtained from the ResNet50-LSTM architecture achieved the highest accuracy on the validation set. We experimented with the 5-CV and test set to confirm that the ResNet50-LSTM architecture outperformed MobileNetV2-LSTM and DenseNet201-LSTM architectures. Additionally, we compared the MobileNetV2-LSTM with the fusion CNN-LSTM architecture. The results showed that the fusion CNN-LSTM slightly outperformed MobileNetV2-LSTM. The fusion CNN-LSTM spends more computational time on recognition. As a result, we are not recommending the use of fusion CNN-LSTM architecture for real-time video fingerspelling recognition.

In the future work, we plan to apply the proposed method to recognize multiple words and sentences of the Thai sign language. The lightweight 3D-CNN might be included in future experiments to enhance accuracy and computation time.

Acknowledgments. This research project was financially supported by Mahasarakham University. We would like to thank all participants from Rajabhat Mahasarakham University for their interest in contributing to this research.

Chapter 4

Word Recognition in Thai Sign Language Sentences From Video with Deep Learning

Sign language is a common language used for communication among deaf people and is important for the development of deaf people's knowledge and abilities. The use of hand and finger gestures in spelling and gestures to convey the meaning of words and sentences has a complexity of gestures to interpret that is difficult to understand. Hence, the sign language recognition system is proposed to communicate with hearing loss people and others. In this paper, our goals are to recognize words and sentences in Thai Sign Language from the video as an end-to-end system to recognize. The proposed system included two main processes. Firstly, data preparation consists of The YOLOv5 algorithm for the human detection task. It extended data to increase the amount of data and optical flow to optimize the rendering of motion of the optical flow. Secondly, we proposed word recognition in Thai sign language that consists of two deep learning architectures, including four layers of 1D convolutional neural network (CNN) and two layers of long short-term memory (LSTM). After that, we combined CNN and LSTM, called CNN-LSTM architecture, decoded algorithms by the recognition block with Soft-max layers, followed with Connection temporal classification (CTC) to predict continuous sequence characteristics. We tested performance with words error rate (WER) value. For the CNN architectures and evaluated three CNNs; VGG16, ResNet50, and DenseNet121. We found that the proposed DenseNet121-LSTM architecture achieved a WER of 0.4286 on the test set of the Thai sign language sentence (TSLs) dataset.

4.1 Introduction

Sign language is a common language used for communication among deaf people and is important for the development of deaf people's knowledge and abilities. The use of hand and finger gestures in spelling and gestures to convey the meaning of words and sentences has a complexity of gestures to interpret that is difficult to understand. Therefore, many technologies have been developed. For example, sensor technologies to aid in communication, such as Accelerometers, Gyroscope, Flex Sensor (Shukor et al., 2015) and Depth Camera (Huang et al., 2018), (Bantupalli & Xie, 2018). To improve the communication of this hearing impaired. Without help, it may have serious and negative effects on language development. Of course, the use of sign language is necessary for these people, especially, the need to communicate with normal people. In general, sign languages are diverse and unique in different countries. Likewise, Thai sign language has integrated a variety of gestures from hand movements which involve pattern matching, computer vision, natural language processing, and linguistics. To identify pre-generated signals and recognize their meaning, we implemented a sign language gesture recognition system and interpret results with high

speed and accuracy. The important and unique role of technology in image gesture recognition is accepted in the development of systems with a vision-based approach to preprocessing. It can detect and recognize visual sign language using the unique architecture of the convolutional neural network (CNN).

However, we developed powerful hand-detection and feature extraction models and, used CNNs to have impressive capabilities for manipulating still images. However, it didn't cover sequences effectively. Therefore, CNNs were integrated with other deep learning models. It includes recurring neural networks (RNNs), long short-term memory (LSTM), and gated iterative units (GRUs) (Wang et al., 2016; Cheok et al., 2019; Escobedo Cardenas & Chavez, 2020; Lata, et al., 2022). Sign language recognition for use in recognition models has two forms of data input, including static and dynamic gestures. It extracts the necessary properties by offering many in-depth models to use still image inputs (Nakjai & Katanyukul, 2019; Pariwat & Seresangtakul, 2017; Salian, 2017) or sequences (Bantupalli & Xie, 2018). Although, dynamic inputs include sequential information that may be useful in improving the accuracy of sign language recognition. Besides, there is still the challenge in using this information, such as the computational complexity of the input sequence. In addition, dynamic inputs can be divided into dynamic inputs used in word-level recognition and continuous dynamic inputs used at the sentence level to further challenge continuous dynamic input as well as tokens of sentences with separate words. In detecting the beginning and end of a sentence, and handling abbreviations and synonyms of sentences, all sign language recognition that aforementioned will generate a comprehensive sign language recognition system.

The significant contributions of this research are summarized in the following: recognition of words and sentences in Thai Sign Language from the video as an end-to-end system to recognize. We mainly evaluated the recognition performance of CNN-LSTM architectures, which are applied to recognize a Thai sign language sentences dataset. We focused on the deep learning approaches of the YOLOv5 algorithm for human detection. Besides, we used optical flow to enhance image motion learning and CNN-LSTM architecture for recognizing Thai sign language sentence recognition. We extend a data frame of 100 to increase the learning curve due to the small number of STSL datasets. Moreover, the sequence patterns were given to the four 1D convolutions, followed by two LSTM architectures, and the SoftMax activation function. We experimented with connectionist temporal classification (CTC) decoding to help answer sentences before making predictions. The performance was evaluated by determining the character error rate (WER).

Paper Outline: This paper has been organized as follows. Section 2, Related work in techniques of sign language recognition with deep learning. Section 3, Proposed Method. Section 4, experimental setup. Section 5, Discussion. The last section Conclusion the significant findings from this study and describes future work.

4.2 Related work

This section presents a brief introduction to deep learning in recent years regarding the recognition of various sign languages using Deep Learning methods, especially, Computer Vision and Natural Language Processing. Recognition of sign language or hand gestures has still limitations on use in live situations where hand gestures variation, illumination change, or background complexity, and is a large area that does not affect the consideration data issue. Therefore, there is a process to solve that problem of means in finding areas of interest (ROI) or using them to isolate those areas.

In 2016, Chen et al., (2016) (Chen et al., 2016) indicated that techniques of skin models and background subtraction are used to obtain training and test data for CNNs by camera-taken datasets. They adopted a simple Gaussian skin color model to filter out non-skin colors of an image robustly. The accurate result was 93.80%, which assumes recognition with a fixed hand gesture data set.

In addition, Lata & Surinta, (2022) (Lata & Surinta, 2022) developed an end-to-end Thai sign language fingerspelling recognition by using the deep convolutional neural network architecture. The first process was the detection of hand regions “You only live once version 3 (YOLOv3)” to define regions of interest (ROI) and continue with the second process was the usage of DenseNet121 architecture. The result of the ability to recognize sign language characters was an accuracy of 93.99%.

Dadashzadeh et al., (2018) (Dadashzadeh et al., 2018) proposed a two-step in-depth model for hands, gesture segmentation, and recognition. A two-step CNN model was used for pixel-accurate semantic segmentation as well as final hand gesture recognition. Residual deep neural network integration and atrous spatial pyramid integration were used for the segmentation step. Evaluation results on the OUHANDS dataset showed a state-of-the-art recognition accuracy improvement for static hand gestures of 1.6%.

Rastgoo et al., (2018) (Rastgoo et al., 2018) proposed a hand signal recognition model using a Restricted Boltzmann machine (RBM) from two image data formats consisting of RGB and depth. They tested the model in three formats, including original images, cropped images, and cropped images with noise. In the first step, the hand of each crop was detected using a CNN. Subsequently, the gesture would have three input images to the RBM. The output of the RBM two modalities was fused into the other RBM to recognize the output sign label. Afterward, training was performed on four publicly available datasets, including Massey University Gesture Dataset 2012, the American Sign Language (ASL) fingerspelling dataset from the University of Surrey's Center for Vision, Speech and Signal Processing, NYU, and ASL Fingerspelling. The results showed that the model achieved state-of-the-art with a relative accuracy improvement of 27.31%, 28.56%, 2.9%, and 11.13%, respectively.

Bantupalli and Xie (2018) (Bantupalli & Xie, 2018) used deep learning in combination with computer vision to solve the problem of communicating with hearing-impaired people in American Sign Language by creating a vision-based application that offers sign language translation to text thus aiding communication between signers and non-signers. The next step was the use of inception which is CNN and the RNN model training on temporal features. The dataset of 600 training samples of each 300 frames was shuffled with 80-20 splits into test and validation data.

Wu (Wu, 2019) performed hand gesture detection using the edge detection algorithm of Canny to extract border features and perform double-channel mutual recognition using a convolutional neural network, called a double convolutional neural network (DC-CNN algorithm), resulting in higher recognition results than single-channel CNN. Meanwhile, it was adopted to light and dark background patterns as well as simple and complex backgrounds using the two-hand gesture database, the JTD, and the NCD datasets.

Huang et al., (2018) (Huang et al., 2018) proposed attention-based 3D-CNNs for SLR. The framework had 3D convolutional networks that learn spatiotemporal features, and the attention mechanism helps to select the clue. In training 3D-CNN for capturing spatiotemporal features, spatial attention was incorporated into the network to focus on the areas of interest. After feature extraction, temporal attention was utilized to select the significant motions for classification. The proposed method was evaluated on two large-scale Chinese sign language (CSL) datasets. The experiment result demonstrated the effectiveness of their approach compared with state-of-the-art algorithms.

Chaikaew et al., (2021) (Chaikaew et al., 2021) created a Thai Sign Language Recognition Application and developed it for real-time sign language translation with a MediaPipe framework that helps to extract the hand. The recognition model of hand gestures with various Recurrent neural networks (RNN) that were built by LSTM, BLSTM, and GRU had an accuracy greater than 90 percent.

Wang et al., (2022) (Wang et al., 2022) proposed a (2+1)D-SLR network based on (2+1)D convolution. Unlike other methods, the proposed network achieved higher accuracy at faster speeds by using a series. The large Chinese Sign Language video data known as NCSL, including 300 different Sign Language words displayed by 30 volunteers. The results on NCSL and other large Sign Language datasets found that LSA64 has an accuracy of 96.4% and 98.7% respectively. The results indicated that This method can not only achieve accuracy in the competition but faster than currently known sign language recognition methods.

Sanalohit and Katanyukul., (2022) (Sanalohit & Katanyukul, 2022) proposed building upon the recently launched MediaPipe Hands (MPH). MPH was a high-precision well-trained model for hand-keypoint detection. The data investigated three

Thai Finger Spell (TFS) schemes: static-single-hand (S1), simplified dynamic-single-hand (S2), and static-point-on-hand (P1) schemes. The results showed that MPH can satisfactorily address single-hand schemes with an accuracy of 84.57% on both S1 and S2. However, their finding revealed a shortcoming of MPH in addressing a point-on-hand scheme which is an accuracy of 23.66% on P1 conferring to 69.19% obtained from conventional classification trained from scratch. This shortcoming was investigated and attributed to self-occlusion and handedness.

Li et al., (2020) (Li et al., 2020) proposed a deep learning method for word-level sign recognition on large-scale Word-Level American Sign language (WLASL). They compared two different models between a holistic visual appearance-based approach and 2D human gesture-based methods with multiple deep learning methods, both CNN and RNN. Moreover, it is the proposition of a novel pose-based temporal graph convolution network (Pose-TGCN) that models spatial and temporal dependencies in human pose trajectories simultaneously that has further boosted the performance of the pose-based method. The results showed that pose-based and appearance-based models achieve comparable performances up to 62.63% at top-10 accuracy on 2,000 words.

4.3 Proposed Method

This section explains the framework of word and sentence Thai sign language recognition video. There are two main processes. In the first process, data preparation was proposed for the selection of 100 frames in the video recorded at different lengths. We used YOLOv5 to detect humans and determine the ROI of the desired image. Finally, 32 frames according to Lata et al. (Lata, Gonwirat, et al., 2022) from 100 frames were selected forward in the Thai sign language sentence recognition process. To increase the number of datasets, we selected 32 and extends data frames 100 times to obtain different datasets. Therefore, there were 100 datasets in 1 video, and 1 set contains 32 frames. The second process, Thai sign language sentence recognition was used for predicting words and sentences of the Thai sign language. The prepared frames extracted the robust deep features from state-of-the-art CNNs (VGG16, ResNet50, and DenseNet121). These features were fed into conv1D and LSTM to classify words and sentences. Finally, CTC was used to predict a continuous sequence of the sentence.

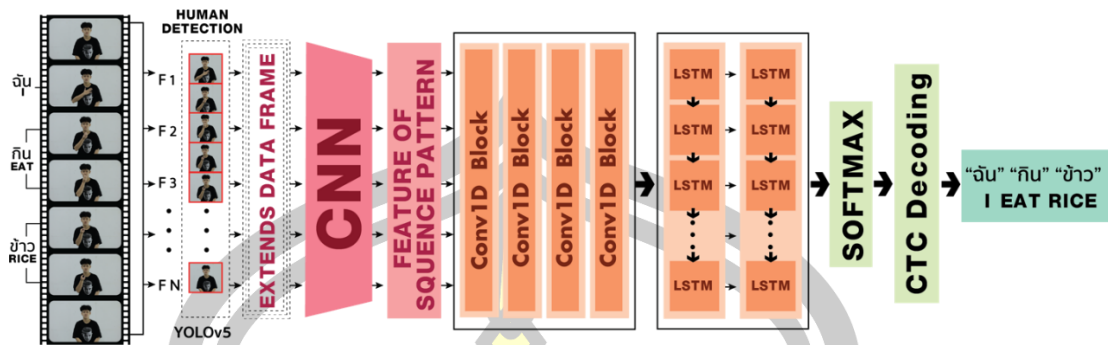


Figure 20 The Proposed Framework for Thai sign language recognition.

1) **VGG16** was proposed by Karen Simonyan and Zisserman (Simonyan & Zisserman, 2014). The architecture of VGGNet had an input of 244x244 RGB images and five convolution layers. Each layer performed 2 to 3 rounds of convolutions using 3x3 filters and strides by 1. Each convolutional layer was followed by Max pooling with a 2x2 filter and stride by 2 using the activation function in each convolution layer as ReLu. Then, it would be sent to the fully connected layer and calculates the classification result by Softmax Activation Function

2) **ResNet50** (Kaiming He et al., 2016) is a 50-layer model designed to solve problems of vanished and exploding gradients when using very deep layers that bottleneck designs. The bottleneck design consists of 4 blocks, each of which has convolutional layers 3, 4, 6, and 3 respectively, excluding the convolutional layer and the input and prediction layers. The experimental part in the bottleneck, the 1x1 conv2D was added at the beginning and the end of the block to reduce the number of parameters in the network.

3) **DenseNet121**: This architecture was conceptualized by solving inter-layer connectivity problems instead of using shortcut connections. After that, we got the feature of the sequence pattern and add 3 additional Conv1D layers before applying 2 LSTM layers. The LSTM was proposed by Hoch Reiter and Schmid Huber in 1997 (Hochreiter & Schmidhuber, 1997) as a solution to the problem of a smaller gradient. LSTM is suitable for sequence data and can update and forget the hidden states by using a set of forget gates, memory units, input, and output gate networks. LSTM maintains the data sequentially from the previous input. The memory blocks were designed to control the internal data which has the advantage of remembering previous data and forgetting unnecessary data before sending it to decryption.

4) **Convolution layer**: The convolutional layer computes the convolutional operation of the input images using kernel filters to extract spatial features. The kernel filters are of the same dimension but with smaller constant parameters as compared to the input images. In a convolutional layer, a neuron is only connected to a local area of input neurons instead of full connection so that the number of parameters to be learned

is reduced significantly and a network can grow deeper with fewer parameters. In this research, we presented three convolution 1D layer to extract spatial features, followed by LSTM network. The kernel size of convolution 1D is $m \times F$, where F is the depth of a filter and m is the size of the convolutional kernel.

5) Long Short-Term Memory (LSTM), invented by (Hochreiter & Schmidhuber, 1997) to present the novel gradient-based method and developed the network based on a recurrent neural network (RNN). It proposed to address the computational complexity, error flow, constraints of the feedforward neural network, and sequence problems of time series data (Kang et al., 2016)(Yan et al., 2018). The LSTM network comprised special units that connect to other units and are designed to cope with the sequence of data; video and speech data, called memory blocks. Each memory block contained the various functions consisting of the forget gate, input gate, update cell state, and the output gate. The memory block is calculated as follows;

$$f_t = \sigma(w_f \cdot [h_{t-1}, x_t] + b_f) \quad (1)$$

$$i_t = \sigma(w_i \cdot [h_{t-1}, x_t] + b_i)$$

$$C_t = \tanh(w_c \cdot [h_{t-1}, x_t] + b_c)$$

$$O_t = \sigma(w_o \cdot [h_{t-1}, x_t] + b_o)$$

$$h_t = o_t \cdot \tanh(C_t)$$

where, x_t is input vector, W is weight, b is bias, h_{t-1} is previous cell output, C_{t-1} is previous cell memory, h_t is current cell output, C_t is current cell memory, σ is sigmoid function, and ' \cdot ' is the Hadamard product.

4) Decoding algorithm: In the experiment, we proposed an additional decoding algorithm to propose to transcribe the sequence data output probability distributions of 23 words which are the output of the SoftMax activation function to use as input of the decoding, and then to send it to be decrypted with Connectionist Temporal Classification (CTC) (Graves et al., 2006). CTC was solved as a loss function used for training the LSTM networks to address sequence problems. There is the ability to predict in a continuous sequence. This research uses Greedy search, a well known algorithm in the language generation tasks of NLP (Natural Language Processing). This method aims to generate the sequence outputs of tokens from a neural network model. Both approaches are focused on sequence-to-sequence models. The model maps an input sequence to a target sequence. In addition, CTC had a function to check the blank or no label, which prevents blank or other label prediction from occurring to predict the sentence.

4.4 Experimental Setup

1) Thai sign language sentences (TSLs) dataset: The data in this research is a dataset in the form of Thai Sign Language sentences. We had allowed from 4 informants who could use Thai Sign Language. The TSLs dataset contained only 23 words. We selected four types of words: Subject, Verb, Object, and Adjective. After that, it was used to compose 125 consecutive sentences. Data collection It does not specify the duration or the rhythm of the gestures of that sentence. Each sentence contains words with complex gestures or multiple strokes shown in figure 19, Finally, we got a total of 500 video data with frequencies ranging from 4 to 132 frequencies as shown in Table 8 and Figure 21-22.

Table 8 Show in Thai sign language sentences (TSLs) Dataset.

type	Used words, In parentheses () is the word frequency in this data set.						
	Subject	You (100)	I (104)	People (100)	They (100)	We (100)	
Verb	Use (132)	Have (96)	Give (100)	Do (104)	See (100)	Request (4)	
Object	Car (92)	Work (28)	Place (96)	Home (88)	Things (84)		
Adjective	Together (72)	Already (92)	Good (20)	All right (60)	Wrong (72)	Many (44)	Damaged (4)

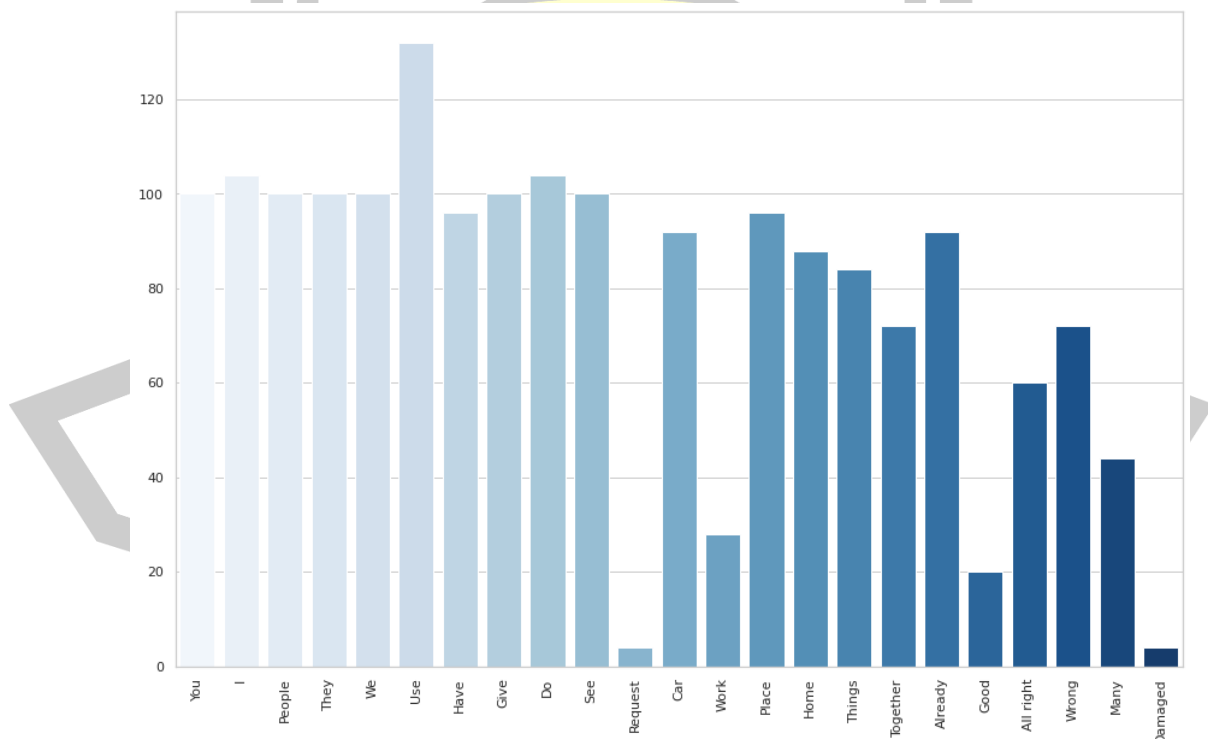


Figure 21 Class distribution of word frequency the TSLs dataset.

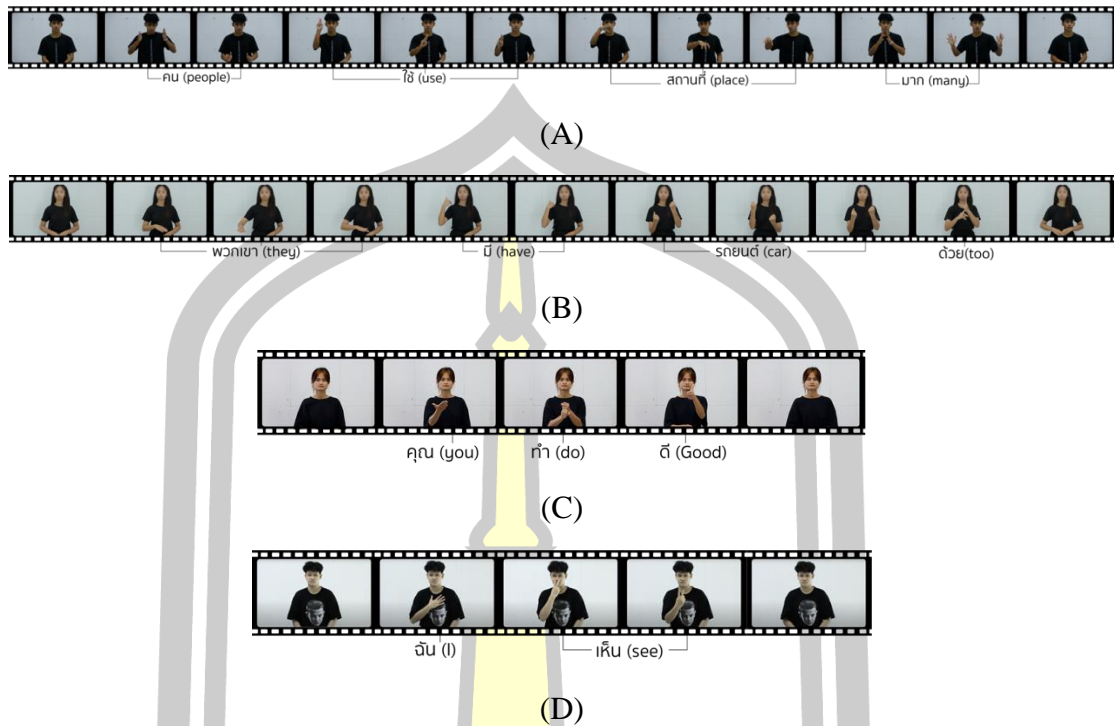


Figure 22 Example Thai sign language sentences Dataset
 (A) People use to place many, (B) They have a car too,
 (C) You do good, and (D) I see.

This dataset was shot with a DSLR camera, video format, 1280x720px image size, and 50p frame rate. The Video files had a different duration depending on each gesture and used a white background that is non-complex.

2) Implementation Detail The proposed framework used the Karas library based on the TensorFlow deep learning framework. All the experiments were trained and evaluated on the Google Colab platform using system specs; GPU Nvidia P100, and 52GB of RAM. We trained and finetuned the models with the following parameters: Adam optimizer, learning rate = 0.0001, moment estimate values = 0.9, batch size = 32, epoch = 100. Second, we divided Training and Testing data in proportions of 80%, and 20%, respectively, performed the 5-fold cross-validation (5-cv) and measured recognition performance with the word error rate (WER).

3) Frame selection & extends data. From a 1280x720 pixel 50p video dataset, we used the pretrained YOLOv5 architecture for human detection. As a result, we got the area of ROI show in figure 23.

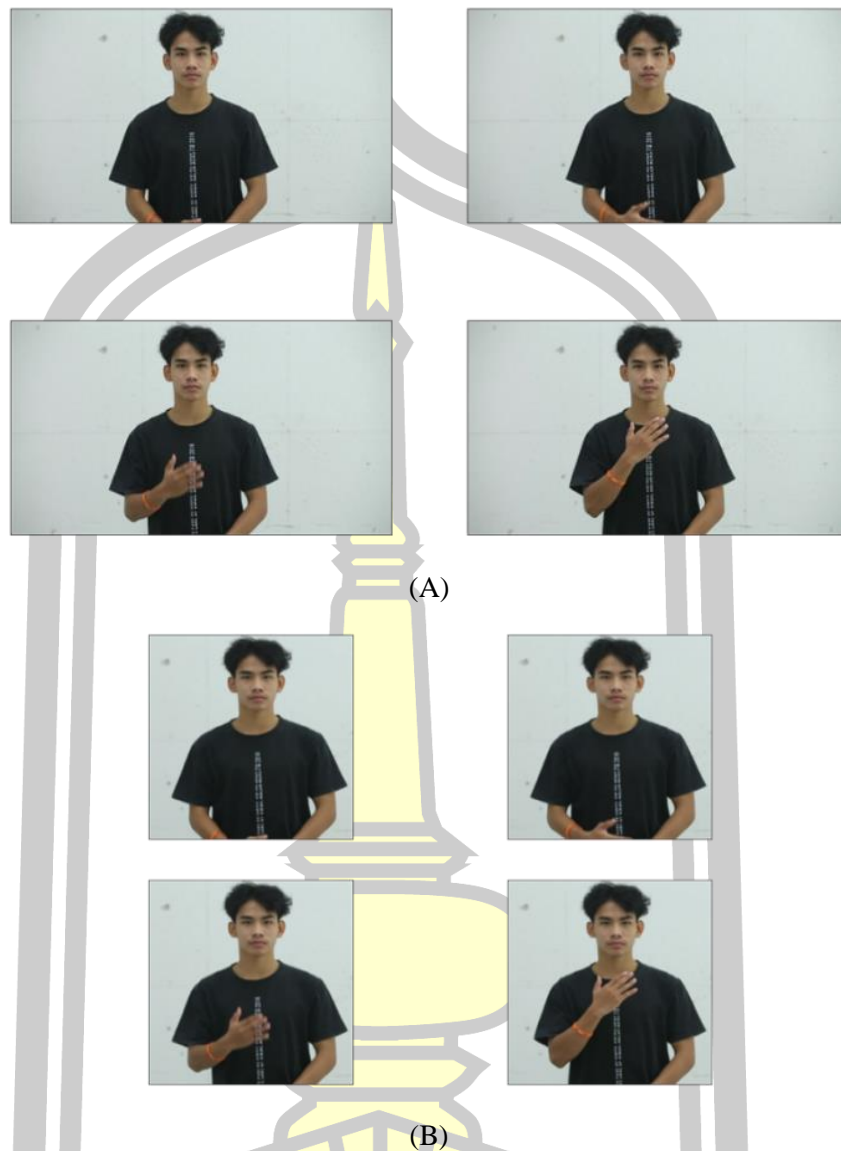


Figure 23 The sample of image frames show (A) Original image file and (B) Human detection image with Yolov5.

Subsequently, we start selected a large number of frames by extracting only 100 frames from figuring out the number of skipped frames with this equation.

$$\text{jump ratio} = \text{floor}(\text{No. of all frames} / 100) \quad (1)$$

In this research, the same number of frames from each source was used for training. We then assign the required frames from the starting frame (S) to the final frame (E). The scope of selection of source frames is limited to only one value of S and E, increasing the variety of data and the possibility of using other frames. Randomly selects a new starting frame and ending frame with the difference S,E between [-5,10] frames , [-10,5] frames respectively, which we call Extends data flames, shown in figure 24.

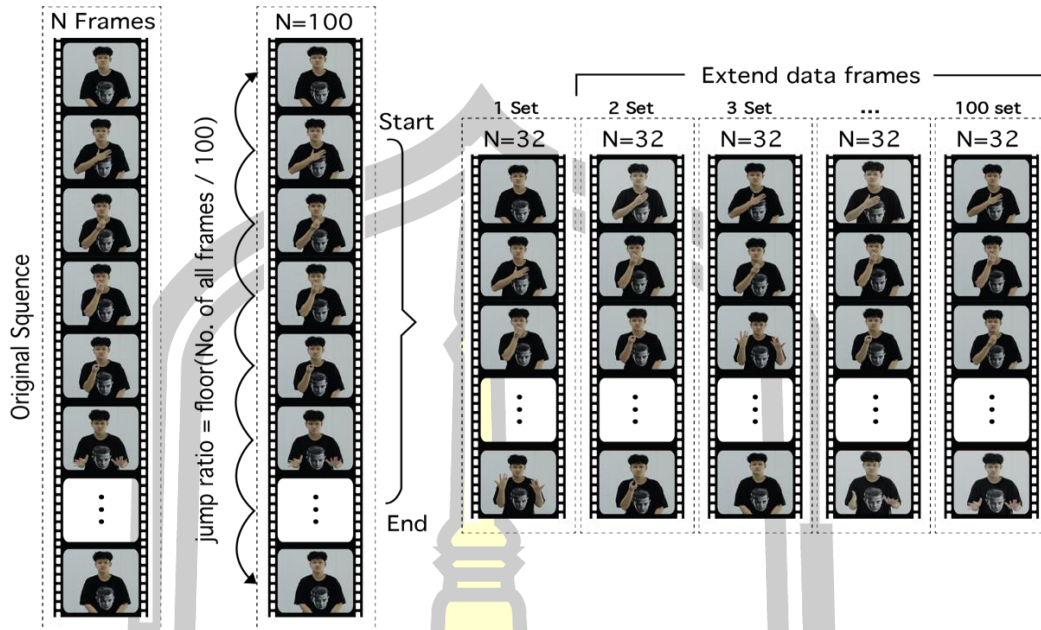


Figure 24 The Process for selecting frames from a video.

we randomly selected the sequence of frames that affects 32 gestures and extend a data frame using a neighboring frame dataset of 20, 100 times. As a result, we obtained an incremental data set before extracting the feature of the sequence pattern. In addition, we further experimented with optical flow recognition techniques and used the three CNNs architectures to compare and find the optimal architecture show in figure 25.

4) Optical flow. The optical flow was used to increase machine learning (Barron et al., 2019). It had one vector representing the motion of a point from the first and second frames. It was determined by the (Consider) pixel $I(x, y, t)$ and will be moved for a distance represented by (d_x, d_y) and so on in the next frame. There is a formula for calculating as follows:

$$I(x, y, t) = I(x + d_x, y + d_y, t + d_t) \quad (2)$$

Consider a pixel $I(x, y, t)$ in the first frame (Check a new dimension, time, is added here. Earlier, we were working with images only, so no need of time. It was moved by a distance (d_x, d_y) in the next frame taken after death time, since those pixels are the same and the intensity does not change. We know the flow of light from the previous frame in relation to the next frame, thus showing the effect of the movement of the image. Of course, the optical flow changes with each move made on the frame show in figure 25 (B).

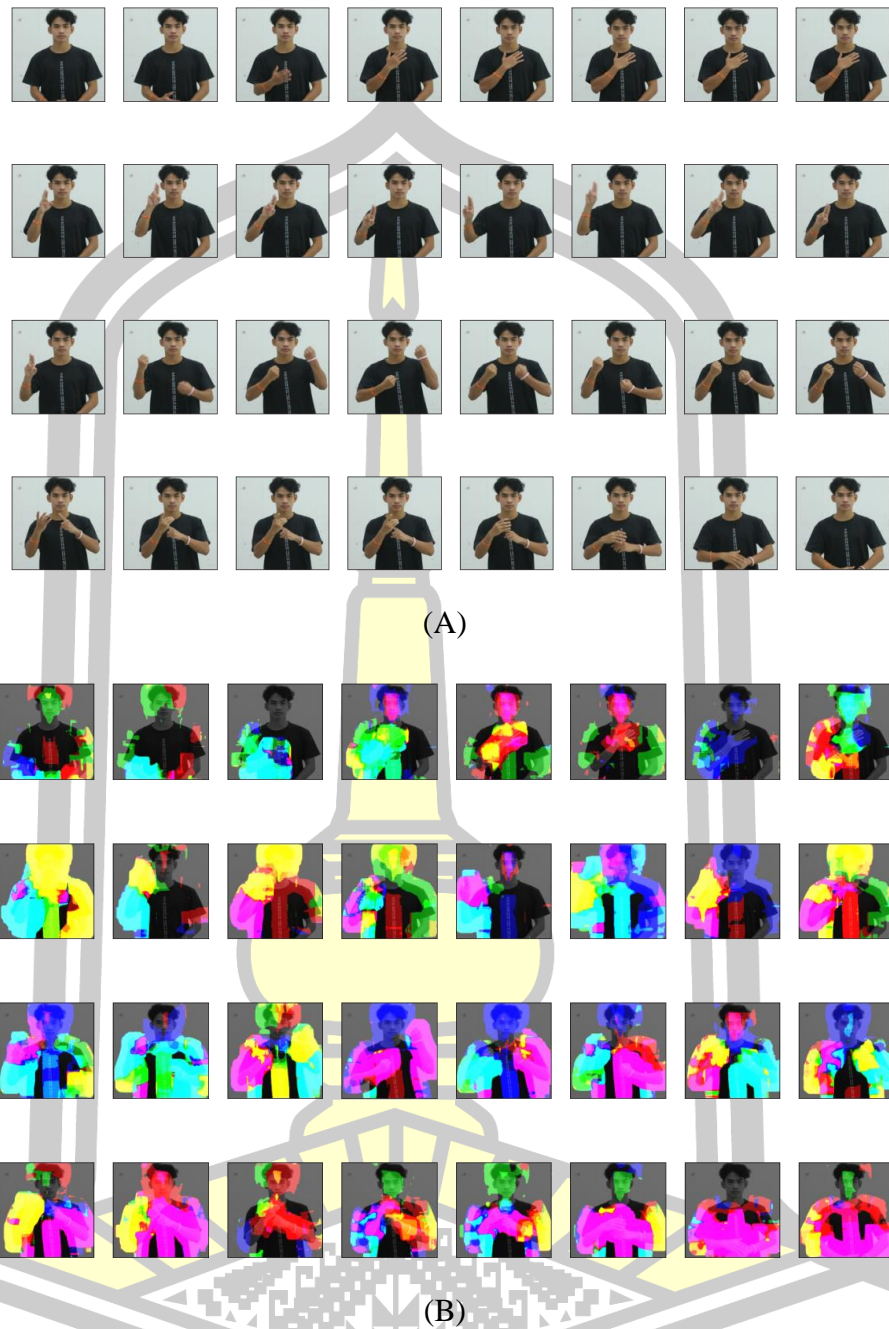


Figure 25 The sample of image frames show (A) Original sequence images, (B) Optical flow on sequence images.

5) Evaluation metrics. The evaluation metrics used for word and sentence Thai Sign language recognition word error rate (WER). The WER was computed by equation 3.

$$WER = \frac{I+S+D}{N} \quad (3)$$

where I is the number of words insertions, S is the number of words substituted, D is the number of words deleted, and N is the total number of words in the ground truth sentence.

4.5 Experimental result and Discussion

1) Experiments with CNN-LSTM comparison and Extend data. In this experiment, we compared the state-of-the-art three CNNs architectures; VGG16, DenseNet121, and ResNet50 to obtain the best performance. We showed the number of extended data 20,100 times with the original 32 frames and feature size to train each CNNs. We used a few convolutional layers Conv1D 3 layers and two layers of LSTM architecture. After that, we classified by using the SoftMax function and performed a final in the CTC decoding algorithms. We determined performance by Words Error Rate (WER) in Table 9.

Table 9 The performance of CNNs architectures and extend data frames on the TSLs dataset.

CNN Feature	Feature sizes	WER		
		Original	Extends data (Times)	
		-	20	100
VGG16	512	0.9412	0.9440	0.4734
DenseNet121	1024	0.9412	0.9384	0.4342
ResNet50	2048	0.9272	0.9380	0.4426

Table 9 showed the experimental results with Three CNN-LSTM architectures and extend data frames, and the number of feature sizes for each architecture in the experiment. The result extends data found original and 20 times the performance is not different. It is different from the 100 Times Extends which are more efficient. As a result, the best CNN model is DenseNet121 on Extends data 100 Times to be the best-performing Words error rate (WER) = 0.4342. Note that, we used to extend data 100 Times in the next section of the experiment.

2) Experiments with Optical flow. After experimenting with extended data frames 100 times (see Table 2), this data set was used for further experimentation. We experimented with adding optical flow to verify frame dynamics. This allows us to compare using and without optical flow with the difference. We still compared all 3 CNNs and showed the values of the parameters shown in Table 10.

Table 10 The Comparison results of the using and not using optical flow.

CNN Feature	Optical flow (WER)		Parameter
	No	Yes	
VGG16	0.4566	0.4986	4,768,664
DenseNet121	0.4342	0.4482	5,751,704
ResNet50	0.4706	0.4622	7,717,784

Table 10 showed the performance demonstrated in the optical flow in use with the ResNet50 architecture with an increment. In addition, the optical flow performance did not improve the Word error rate. Although, the ResNet50 was good but still was less efficient than the DensNet121. Note that, we would not use the optical flow technique in the next experiment because it had more parameter weight than DensNet121.

3) Experiments with Number of Conv1D. After evaluating various CNN-LSTM architectures on the Video Thai sign language sentences, we tried to find the number of layers of Conv1D. Also, we compared the performance of the number of layers of Conv1D consisting of 1-4 layers: input 128,256,512, respectively. We still used three CNN-LSTM architectures. In addition, we determined the effective size of CNN-LSTM and experimented by using three sizes: 128,256,512, respectively. This experiment used LSTM to evaluate performance with WER as shown in Table 11.

Table 11 Evaluation of performance number layers of Conv1D

CNN Feature	CONV1D	(WER)		
		CNN-LSTM size		
		128	256	512
VGG16	1 layer	0.7003	0.8739	0.9020
	2 layers	0.6695	0.6947	0.6947
	3 layers	0.4566	0.4930	0.5126
	4 layers	0.4622	0.4678	0.5098
DenseNet121	1 layer	0.9636	1.0140	1.0672
	2 layers	0.6555	0.6162	0.4510
	3 layers	0.4342	0.4846	0.5210
	4 layers	0.4286	0.4650	0.4762
ResNet50	1 layer	1.0504	0.7815	1.0000
	2 layers	0.6751	0.6555	0.6863
	3 layers	0.4706	0.4538	0.4650
	4 layers	0.4482	0.4342	0.4482

Table 11 shows the performance of Conv1D with different layers. The best results were obtained when using 4 layers of Conv1D across all BiLSTM sizes except DenseNet121 with BiLSTM size=512 where only 2 layers of Conv1D were used, WER = 0.4510. However, the best result of all was the DenseNet121 architecture based

on BiLSTM size =128, using 4 layers of Conv1D, WER = 0.4286. Note that, we also used 4 layers of Conv1D in the next experiment.

4) Comparing the RNNs; LSTM and GRU. It is the comparison of the RNNs Types for word sign language recognition. CNN was previously used to extract features. We selected an interesting comparison of the LSTM and GRU in terms of computation sequence learning called BiLSTM and BiGRU, respectively. In our experiment, we showed the results of using 3 and 4 layers of Conv1D and also compared the DenseNet121 and ResNet50 architectures with the 5-CV, as well as we also showed the magnitude of all 3 RNNs: 128,256,512 in Table 12.

Table 12 The comparison results of different RNN types using BiLSTM, and BiGRU.

CNN Feature	CONV1D	WER (5-cv)					
		BiLSTM size			BiGRU size		
		128	256	512	128	256	512
DenseNet121	3 layers	0.45 ±0.02	0.46 ±0.02	0.49 ±0.03	0.47 ±0.03	0.48 ±0.02	0.48 ±0.02
	4 layers	0.43 ±0.01	0.46 ±0.02	0.48 ±0.01	0.49 ±0.01	0.47 ±0.01	0.48 ±0.03
ResNet50	3 layers	0.47 ±0.01	0.47 ±0.01	0.48 ±0.01	0.51 ±0.01	0.51 ±0.02	0.50 ±0.02
	4 layers	0.45 ±0.02	0.46 ±0.01	0.46 ±0.01	0.52 ±0.01	0.49 ±0.02	0.49 ±0.02

Table 12 shows the results using BiLSTM can learn in a sequence pattern for better word recognition. When used with the DenseNet121 and ResNet50 architectures, all RNNs sizes. Finally, we also found that DenseNet121 retains the best word recognition capability.

To recognize the Word in sign language sentence, we recommend decoding the output using the CTC algorithm. Examples of the correct recognition using DenseNet121-LSTM. Some recognition results and WER values are shown in Table 13.

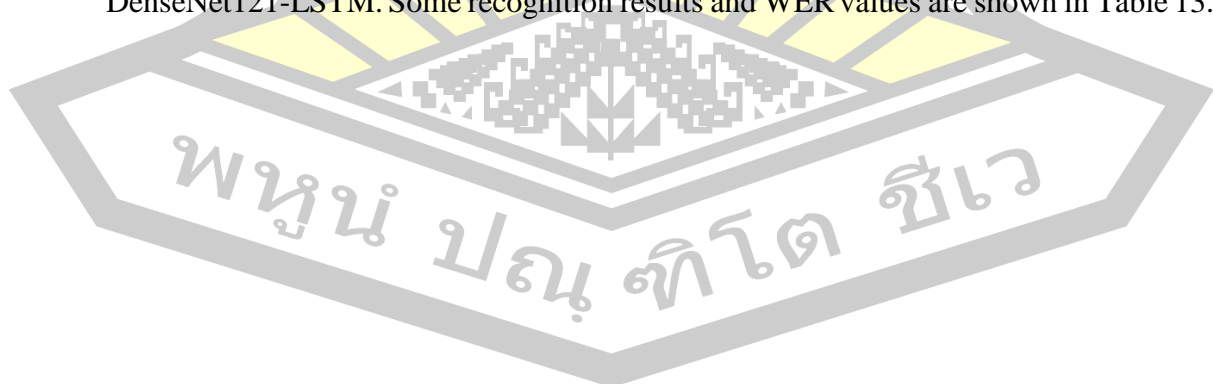










Table 13 The recognition results of the DenseNet121-LSTM architecture using the CTC decoding algorithm.

#	Input Video and Labels	Recognition Results	WER
1	 ['คุณ/You' 'ทำ/Do' 'ผิด/Wrong' 'มาก/Many']	['คุณ/You', 'ทำ/Do', 'ผิด/Wrong', 'มาก/Many']	0
2	 ['คน/People' 'ใช้/Use']	['คน/People', 'ทำ/Do', 'ของ/Things', 'ได้/All right']	1.5
3	 ['พวกเรา/We' 'มี/Have' 'รถยนต์/Car']	['พวกเรา/We', 'ใช้/Use', 'รถยนต์/Car', 'ด้วย/Together']	0.50
4	 ['พวกเขา/They' 'เห็น/See' 'รถยนต์/Car' 'ด้วย/Together']	['พวกเขา/They', 'เห็น/See', 'ของ/Things', 'ด้วย/Together']	0.25
5	 ['พวกเขา/They' 'มี/Have' 'บ้าน/Home']	['พวกเขา/They', 'มี/Have', 'บ้าน/Home', 'ได้/All right']	0.33
6	 ['คุณ/You' 'ให้/Give' 'บ้าน/Home']	['ฉัน/I', 'ให้/Give', 'รถยนต์/Car', 'ได้/All right']	1.0
7	 ['พวกเรา/They' 'เห็น/See' 'สถานที่/Place' 'แล้ว/Already']	['พวกเรา/We', 'มี/Have', 'สถานที่/Place', 'แล้ว/Already']	0.5
8	 Label: ['ฉัน/I' 'มี/Have' 'ของ/Things' 'แล้ว/Already']	['ฉัน/I', 'ให้/Give', 'ได้/All right', 'ด้วย/Together']	0.75

4.6 Discussions

We observed We observed ROI areas with YOLOv5, and the result is shown in Figure 23. We tried to increase the training data by extending the data frames. There was no difference in increments of 20 times but extending data frames by 100 times could cause Efficiency to improve more than + 0.5, as shown in Table 9. In Table 10, we find that the optical flow technique was used to show the results of dynamics

achieved as unsatisfactory. Because there is a continuous movement that causes the image showing use to reduce the details in the image, as shown in Figure 22. The result shows that using the number of Conv1D 4 layers is the most efficient, as in Table 11. It was also found that using the LSTM-based RNN type was more efficient than the GRU, and the RNN's input size of 128 was found to be the most efficient. As shown in the last Table 12, the best-performing architecture was DenseNet121 with 5-cv at WER=0.43.

4.7 Conclusions

We offered a sophisticated automatic gesture recognition system for Thai Sign language, including frame selection & extends data, human detection, optical flow, and words recognition from Thai sign language sentences. We focused on the deep learning approaches; the YOLOv5 algorithm for human detection. Then, we used optical flow to enhance image change learning and CNN-LSTM architecture for word recognition from Thai sign language sentences, the optical flow performance did not improve. First, we selected a frame with the region of the gesture in which 32 frames were selected using the uniform distribution in that region. In addition, we extend data to increase the learning curve due to the small number of STSL datasets. Second, the sequence patterns were given to the Conv1D where we get the best results using 4 layers of Conv1D, after that, followed by two RNNs architectures: LSTM and GRU, CNN-LSTM which was found to be more efficient than CNN-GRU and RNN size equal to 128. It is the best efficiency. Finally, we used the SoftMax activation function and CTC decoding to help answer sentences before making predictions. The performance was evaluated by determining WER. Moreover, we mainly evaluated the recognition performance of CNN-LSTM architectures. We experimented with the 5-CV; the best architecture results were DensNet121-LSTM architectures with the highest accuracy on the test set. Additionally, we compared extends to a data frame. The results showed that the extension of data 100 times will improve its efficiency. We also compared using and without optical flow which indicates that optical flow is not suitable for our dataset. In the paper, we measured the WER value to confirm the effectiveness of word recognition from Thai Sign Language sentences. The best result was DensNet121-LSTM, conv1D using 4 layers, RNN size equal to 128.

In the future work, we are able to train and test to optimize our framework against sign language datasets with other countries, such as American Sign Language sentences, MediaPipe Holistic dataset to find a more universal pattern.

Chapter 5 Discussion

The purpose of this research was to develop an automated sign language translation system for the hearing impaired using deep learning that can translate both alphabet and word sign language. Firstly, we tried to recognize sign language generated in real environments. However, most of them were images with complex backgrounds, lots of unrelated areas, and mostly using images taken in laboratories. We attempted to bypass this limitation by using a hand detection approach using the YOLO v3 architecture to detect the region of interest (ROI) in the hand only. After that, we performed feature extraction and enable it to be recognized with efficient CNNs. Secondly, we attempted to recognize data from the dynamic: Thai Sign Language. we designed to propose an end-to-end system to recognize the dynamic Thai fingerspelling from video. Although, in the case of using gestures with movement and increasing the ROI area, we applied the YOLOv5 algorithm for the human detection task. In addition, we proposed dynamic fingerspelling recognition that consists of two deep learning architectures: convolutional neural network (CNN) and long short-term memory (LSTM) to aid sequence-based image recognition which is called the CNN-LSTM architecture. It was considered to be a combination of two deep learning methods of Convolutional Neural Network and Recurrent Neural Network. This process improved recognition gestures based on Dynamic more efficiently. Thirdly, we proposed the recognition of words from sign language gestures in sentences to comprehend the sign language recognition component. In addition to implementing the YOLOv5 architecture person detection method, we attempted to add learning by extending the low data frame together with using optical flow to display the movement of light from the previous frame. After that, the combination of CNNs and LSTMs was used to enhance recognition. Finally, we tested using Connectionist Temporal Classification (CTC) to decode the output to learn the words in the syntactic form to find the correct sentence.

We briefly presented an automatic sign language recognition system based on the above-mentioned process, as follows:

In Chapter 2, the problem with machine learning in sign language often revolves around complex environments. Also, most of the image area is an area that does not affect learning, and sign language alphabets are still the basis for learning sign language. Therefore, we presented an end-to-end fingerspelling recognition framework of the Thai sign language based on deep convolutional neural networks (CNNs) on the static sign language dataset 15 class. First, we focused on the detection of hands using the YOLOv3 objection detection framework to find areas of ROI. We examined it by using the mean average precision (mAP) that compares the output of the network and the ground truth bounding box to evaluate a model. In the second process, the robust

CNN models were created based on state-of-the-art using five CNN architectures, including MobileNetV2, DenseNet121, InceptionResNetV2, NASNetMobile, and EfficientNetB2, to create the most robust model that provides high recognition performance. Hence, we evaluated the proposed framework to detect and recognize three Thai fingerspellings (TFS) datasets: TFS, KKU-TFS, and Unseen-TFS. We found that YOLOv3 showed a high precision value on the TFS dataset. However, the worst performance was found with KKU-TFS and Unseen-TFS datasets. Also, our proposed framework could not detect hands from only one image on the KKU-TFS and Unseen-TFS datasets. Therefore, we also examined the CNN architectures to recognize the 1-stage Thai fingerspelling images. The experimental results showed that DenseNet121 obtained an accuracy of 93.99% on the TFS dataset and 90.40% on the KKU-TFS dataset. Hence, we evaluated the proposed framework to detect and recognize three Thai fingerspellings (TFS) datasets: TFS, KKU-TFS, and Unseen-TFS. We found that YOLOv3 showed a high precision value on the TFS dataset. However, the worst performance was found with KKU-TFS and Unseen-TFS datasets. Also, our proposed framework could not detect hands from only one image on the KKU-TFS and Unseen-TFS datasets. Therefore, we also examined the CNN architectures to recognize the 1-stage Thai fingerspelling images. The experimental results showed that DenseNet121 obtained an accuracy of 93.99% on the TFS dataset and 90.40% on the KKU-TFS dataset.

In Chapter 3, we introduced Video Dynamic Sign Language Gesture Recognition. The videos were based on the Thai sign language Alphabet 42 class. We focused on the deep learning approaches: the YOLOv5 algorithm for human detection and CNN-LSTM architecture for recognizing the dynamic fingerspelling from the video. First, 32 video frames were selected using the uniform distribution method and sent to extract the sequence patterns using CNN architecture. Second, the sequence patterns were given to the LSTM architecture, followed by three layers: dropout, GAP, and softmax activation function. The results obtained from the ResNet50-LSTM architecture achieved the highest accuracy on the validation set. We experimented with the 5-CV and test set to confirm the ResNet50-LSTM architecture. Moreover, we mainly evaluated the recognition performance of CNN-LSTM architectures. The results obtained from the ResNet50-LSTM architecture achieved the highest accuracy on the validation set. We experimented with the 5-CV and test set to confirm the ResNet50-LSTM architecture.

In Chapter 4, we proposed a recognition system capable of recognizing sign language gestures in words and sentences with gestures connected by a variety of words in it to comprehend the sign language recognition component using the Thai sign language Sentence Dataset (TSLs). The proposed system included three main processes. First, Data preparation consisted of The YOLOv5 algorithm for the human

detection task. It extended a data frames to increase the amount of data and Optical flow to optimize the rendering of motion of the optical flow. Second, we proposed dynamic Thai Sign Language Recognition that consists of two deep learning architectures: three layers of 1D convolutional neural network (CNN) and two layers of long short-term memory (LSTM). Then, We combined CNN and LSTM, called CNN-LSTM architecture, followed by a third process. Third, Decoding algorithms by the recognition block whit soft-max layers, followed whit Connection Temporal Classification (CTC) to predict continuous sequence characteristics. We tested performance with character error rate (CER) value. For the CNN architectures, we evaluated three CNNs; VGG16, ResNet50, and DenseNet201. We found that the proposed VGG16-LSTM architecture achieved a CER of 0.498 on the test set.

5.1 Answers to the Research Questions

This section answered three research questions (RQ) related to improving the system of Thai sign language recognition in detail, according to the research question in Section 1.

In RQ1, we focused on the recognition of Thai Sign Language from Thai Finger Spelling on Static (1-State). We proposed an end-to-end Thai fingerspelling recognition with deep learning. Thai Sign Language was found to be seen through a camera in a typical environment where the background is complex, and the camera is recorded as a landscape image. As a result, many irrelevant areas of interpretation were found. The following questions come up ; The deep learning method for the detection and recognition improve on Static Sign Language recognition efficiency ? Is it possible to find ROI with hand detections? Can the usage of deep convolutional neural networks (CNNs) recognize Thai Sign Language? Which do the best CNNs recognize Thai Sign Language when the amount of data is reduced and unseen data?

To answer RQ1, we focused on one-stage sign language recognition with the TFS dataset is one-stage fingerspelling of Thai sign language that contained 15 signs. We recorded the images with both complex and non-complex backgrounds. In the first process, we proposed the detection of hands using the YOLOv3 objection detection framework to find the area of ROI. The result showed the high average precision value achieved with $mAP = 0.9787$. For the deep learning technique, we first used CNNs to create a robust model. Five CNN models: MobileNetV2 , DenseNet121 , InceptionResNetV2, NASNetMobile, and EfficientNetB2 were trained. The results showed that DenseNet121 and MobileNetV2 outperformed other CNN models. When training with 60%, we obtained an accuracy of 98% from these two CNN models. Finally, we evaluated our end-to-end Thai fingerspelling framework on three TFS datasets. The result showed that the recognition accuracy of the TFS dataset was decreased to 93.99% and 92.81% when using DenseNet121 and MobileNetV2, respectively. The results our framework still detected hands at the correct location and

our framework showed an accuracy above 90% on TFS and KKU-TFS datasets and slightly reduced accuracy to 82% on the Unseen-TFS dataset. We could guarantee from our experimental results we evaluated our end-to-end Thai fingerspelling framework can effectively recognize Thai sign language.

In RQ2, since the learning illustrations in the TFS datasets contain only static datasets, it is not possible to learn the dynamics of Thai Fingerspelling. Can we use Deep learning neural networks in combination with LSTM learning to increase the efficiency of dynamic Sign Language recognition efficiency? How do the end-to-end systems recognize the dynamic Thai fingerspelling from the video? How is the efficiency of the dynamic Thai fingerspelling from video that use CNN and LSTM?

To answer RQ2, we focused on the recognition of Dynamic Thai Sign Language, whose data uses multiple consecutive gestures. Therefore, we created the Dynamic Thai Finger Spelling Dataset (DTFS) as a video of all Thai Sign Language alphabet in 42 classes. Our method started by selecting frames by computing the jump ratio over the video frame using the following equation: $\text{jump ratio} = \text{floor}(\text{No. of all frames} / \text{No. of selected frames})$, where No. of selected frames in our experiment is 32. However, some alphabet movements involved hand and arm movements, so the use of the YOLOv5 algorithm for human detection to find the ROI area. Afterward, our Dynamic fingerspelling recognition combined two deep learning approaches: CNN and RNN. For the CNN architectures, we combined CNN and LSTM, called CNN-LSTM architecture, followed by the recognition block. we evaluated three CNNs: MobileNetV2, ResNet50, and DenseNet201. We also found learning rates suitable for CNNs. As a result, when using a learning rate of 0.001 which achieved the highest accuracy, the best CNN model was the ResNet50 which achieved 87.93% accuracy. We evaluated the performance of the CNNs and RNNs (LSTM and GRU) using two different numbers of units: 32 and 64 units. The result of ResNet50-LSTM with 64 units had the best performance and achieved 89.16% accuracy on the validation set. In this experiment, we decided to add the extra densely connected layer with 64 units between the GAP layer and softmax activation function. As a result, the ResNet50-LSTM achieved the best accuracy on both the 5-CV and test set with 88.42%.

In RQ3, we focused on sign language word recognition in the form of consecutive multi-word sentences to complete the elements of sign language recognition. How do the usages of extend data flame, Optical flow, Conv1D, LSTM, and Connectionist Temporal Classification (CTC) recognize the words and sentences of the Thai sign language? How is the word error rate of frameworks?

To answer RQ3, we utilized the Thai sign language Sentence Dataset (TSLs) with 23 classes of 500 sentences. We continued to use data preparation methods from selecting frames in the video dataset in the affected areas for learning to come out at 32 images and extending the data flame by 100 sets to increase the amount of small data.

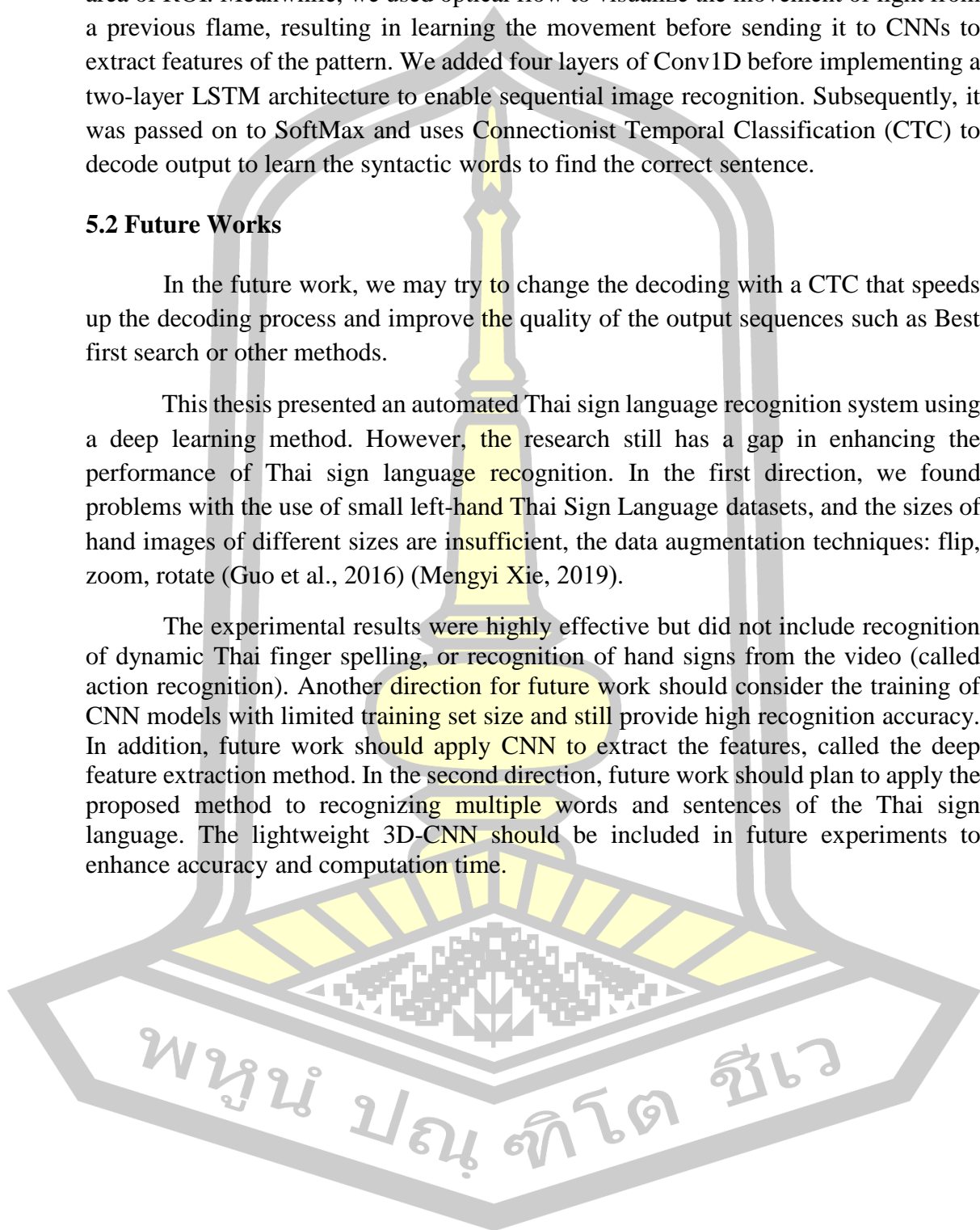
Of course, we examined human detection with the YOLOv5 architecture to find the area of ROI. Meanwhile, we used optical flow to visualize the movement of light from a previous frame, resulting in learning the movement before sending it to CNNs to extract features of the pattern. We added four layers of Conv1D before implementing a two-layer LSTM architecture to enable sequential image recognition. Subsequently, it was passed on to SoftMax and uses Connectionist Temporal Classification (CTC) to decode output to learn the syntactic words to find the correct sentence.

5.2 Future Works

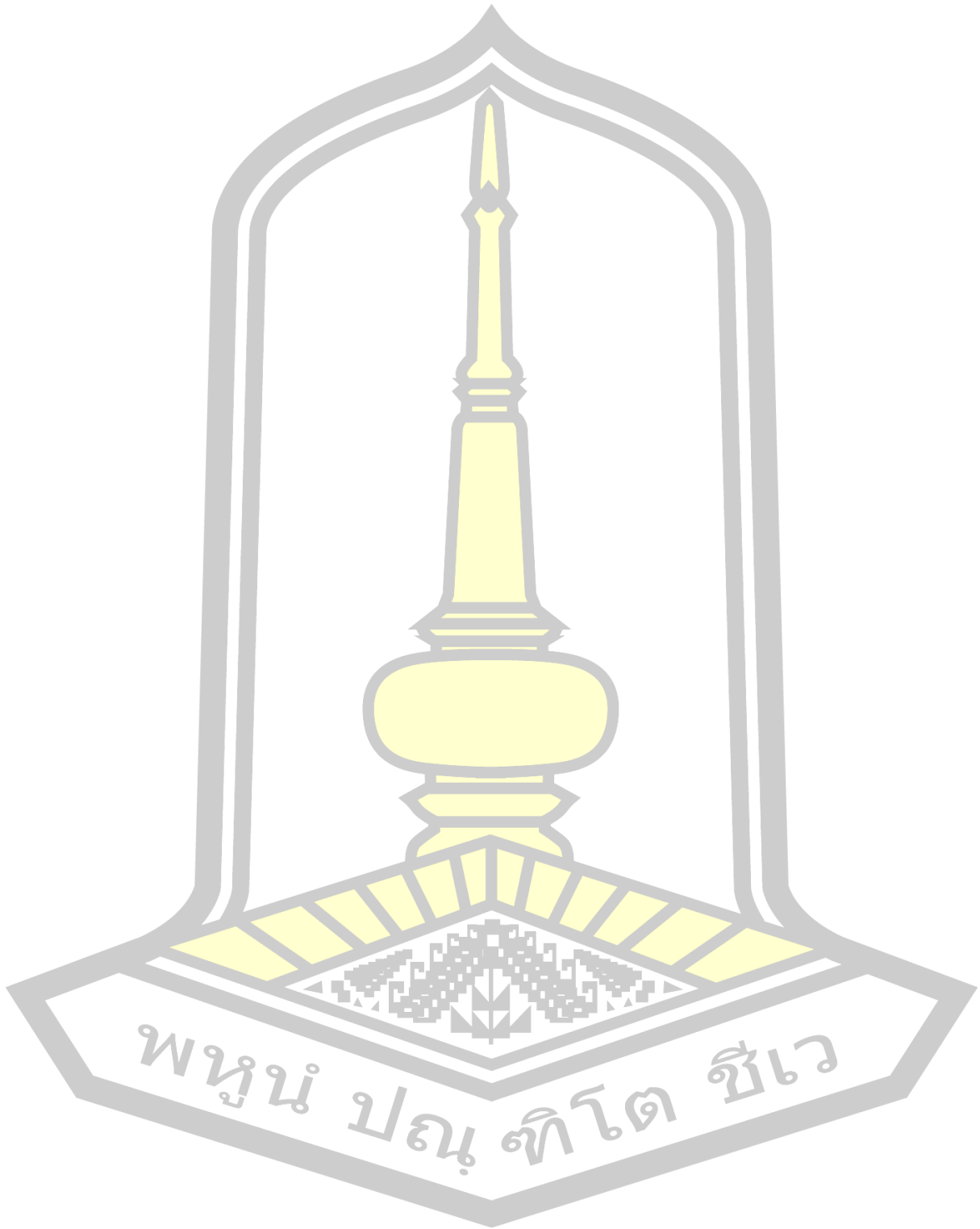
In the future work, we may try to change the decoding with a CTC that speeds up the decoding process and improve the quality of the output sequences such as Best first search or other methods.

This thesis presented an automated Thai sign language recognition system using a deep learning method. However, the research still has a gap in enhancing the performance of Thai sign language recognition. In the first direction, we found problems with the use of small left-hand Thai Sign Language datasets, and the sizes of hand images of different sizes are insufficient, the data augmentation techniques: flip, zoom, rotate (Guo et al., 2016) (Mengyi Xie, 2019).

The experimental results were highly effective but did not include recognition of dynamic Thai finger spelling, or recognition of hand signs from the video (called action recognition). Another direction for future work should consider the training of CNN models with limited training set size and still provide high recognition accuracy. In addition, future work should apply CNN to extract the features, called the deep feature extraction method. In the second direction, future work should plan to apply the proposed method to recognizing multiple words and sentences of the Thai sign language. The lightweight 3D-CNN should be included in future experiments to enhance accuracy and computation time.



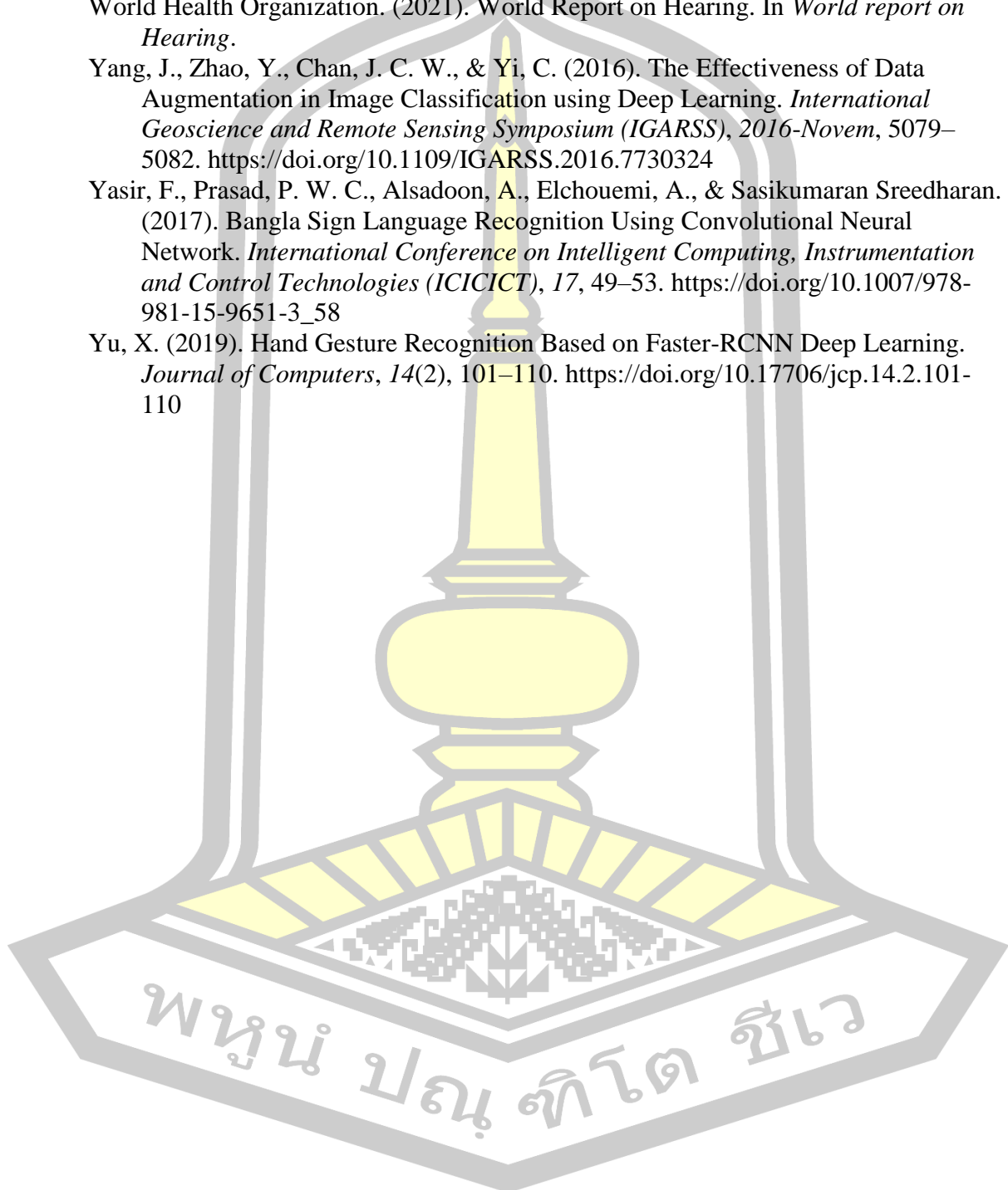
REFERENCES



- Bajpai, D. (2015). Two Way Wireless Data Communication and American Sign Language Translator Glove for Images Text and Speech Display on Mobile Phone. *International Conference on Communication Systems and Network Technologies*, 578–585. <https://doi.org/10.1109/CSNT.2015.121>
- Chansri, C., & Srinonchat, J. (2016). Reliability and Accuracy of Thai Sign Language Recognition with Kinect Sensor. *13th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology, (ECTI-CON)*, 1–4. <https://doi.org/10.1109/ECTICon.2016.7561403>
- Chen, H., Hu, C., Lee, F., Lin, C., Yao, W., Chen, L., & Chen, Q. (2021). A Supervised Video Hashing Method Based on a Deep 3D Convolutional Neural Network for Large-Scale Video Retrieval. *Sensors*, 21(9), 1–15. <https://doi.org/10.3390/s21093094>
- Dang, T. L., Nguyen, G. T., & Cao, T. (2020). Object Tracking Using Improved Deep Sort Yolov3 Architecture. *ICIC Express Letters*, 14(10), 961–969. <https://doi.org/10.24507/icicel.14.10.961>
- Gao, Q., Liu, J., Ju, Z., Zhang, L., Li, Y., & Liu, Y. (2018). Hand Detection and Location Based on Improved SSD for Space Human-Robot Interaction. *International Conference on Intelligent Robotics and Applications, Cham*, 164–175. https://doi.org/10.1007/978-3-319-97586-3_15
- Ge, Z., Liu, S., Wang, F., Li, Z., & Sun, J. (2021). *YOLOX: Exceeding YOLO Series in 2021*. 5, 12. <https://github.com/ultralytics/yolov3>
- Hoang, Q. V., Le, T. H., & Huang, S. C. (2020). An Improvement of RetinaNet for Hand Detection in Intelligent Homecare Systems. *2020 IEEE International Conference on Consumer Electronics - Taiwan, ICCE-Taiwan 2020, 2020–2021*. <https://doi.org/10.1109/ICCE-Taiwan49838.2020.9258335>
- Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. (2017). Densely Connected Convolutional Networks. *30th IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017-Janua*, 2261–2269. <https://doi.org/10.1109/CVPR.2017.243>
- Islam, M. R., Mitu, U. K., Bhuiyan, R. A., & Shin, J. (2018). Hand Gesture Feature Extraction Using Deep Convolutional Neural Network for Recognizing American Sign Language. *4th International Conference on Frontiers of Signal Processing (ICFSP)*, 115–119. <https://doi.org/10.1109/ICFSP.2018.8552044>
- Kaiming He, Zhang, X., Ren, S., & Sun, J. (2016). Deep Residual Learning for Image Recognition. *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 770–778. <https://doi.org/10.1002/chin.200650130>
- Le, T. H., Jaw, D. W., Lin, I. C., Liu, H. Bin, & Huang, S. C. (2018). An Efficient Hand Detection Method based on Convolutional Neural Network. *7th International Symposium on Next-Generation Electronics (ISNE)*, 1–2. <https://doi.org/10.1109/ISNE.2018.8394651>
- Liu, K., & Kehtarnavaz, N. (2016). Real-Time Robust Vision-Based Hand Gesture Recognition Using Stereo Images. *Journal of Real-Time Image Processing*, 201–209. <https://doi.org/10.1007/s11554-013-0333-6>
- Mujahid, A., Awan, M. J., Yasin, A., Mohammed, M. A., Damaševičius, R., Maskeliūnas, R., & Abdulkareem, K. H. (2021). Real-Time Hand Gesture Recognition Based on Deep Learning YOLOv3 Model. *Applied Sciences (Switzerland)*, 11(9), 4164. <https://doi.org/10.3390/app11094164>

- Nakjai, P., & Katanyukul, T. (2019). Hand Sign Recognition for Thai Finger Spelling: an Application of Convolution Neural Network. *Journal of Signal Processing Systems*, 91(2), 131–146. <https://doi.org/10.1007/s11265-018-1375-6>
- Nakjai, P., & Katanyukul, T. (2021). Automatic Thai Finger Spelling Transcription. *Walailak Journal of Science and Technology*, 18(13), 11233–19. <https://doi.org/10.48048/wjst.2021.11233>
- Nepal, U., & Eslamiat, H. (2022). Comparing YOLOv3, YOLOv4 and YOLOv5 for Autonomous Landing Spot Detection in Faulty UAVs. *Sensors*, 22(2). <https://doi.org/10.3390/s22020464>
- Pariwat, T., & Seresangtakul, P. (2021). Multi-Stroke Thai Finger-Spelling Sign Language Recognition System with Deep Learning. *Symmetry* 2021, 3(2), 262.
- Pariwat, T., & Seresangtakul, P. (2017). Thai Finger-Spelling Sign Language Recognition Using Global and Local Features with SVM. *9th International Conference on Knowledge and Smart Technology (KST)*, 116–120. <https://doi.org/10.1109/KST.2017.7886111>
- Phong, N. H., & Ribeiro, B. (2019). Advanced Capsule Networks via Context Awareness. *International Conference on Artificial Neural Networks*. Springer, Cham, April, 166–177. https://doi.org/10.1007/978-3-030-30487-4_14
- Pornthep Sarakon, Hideaki Kawano, Kazuhiro Shimonomura, S. S. (2021). Improvement of Shrinking CNN Architecture Using Weight Sharing and Knowledge Distillation for Tactile Object Recognition. *ICIC International 2021*, 12(7), 627–633. <https://doi.org/10.24507/icicelb.12.07.627>
- Rao, G. A., Syamala, K., Kishore, P. V. V., & Sastry, A. S. C. S. (2018). Deep Convolutional Neural Networks for Sign Language Recognition. *Conference on Signal Processing And Communication Engineering Systems (SPACES)*, 194–197. <https://doi.org/10.1109/SPACES.2018.8316344>
- Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You Only Look Once: Unified, Real-Time Object Detection. *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 779–788.
- Rubin Bose, S., & Sathiesh Kumar, V. (2019). Hand Gesture Recognition Using Faster R-CNN Inception V2 Model. *ACM International Conference Proceeding Series*. <https://doi.org/10.1145/3352593.3352613>
- Salian, S., Dokare, I., & Serai, D. (2017). Proposed System for Sign Language Recognition. *International Conference on Computation of Power, Energy Information and Commuincation (ICCPEIC)*, 58–62.
- Sanalohit, J., & Katanyukul, T. (2022). *Thai Finger Spelling Recognition: Investigating MediaPipe Hands Potentials*. 1–19. <http://arxiv.org/abs/2201.03170>
- Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L. C. (2018). MobileNetV2: Inverted Residuals and Linear Bottlenecks. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 4510–4520. <https://doi.org/10.1109/CVPR.2018.00474>
- Soliman, M. M., Kamal, M. H., El-Massih Nashed, M. A., Mostafa, Y. M., Chawky, B. S., & Khattab, D. (2019). Violence Recognition from Videos Using Deep Learning Techniques. *2019 IEEE Ninth International Conference on Intelligent Computing and Information Systems, (ICICIS)*, 80–85. <https://doi.org/10.1109/ICICIS46948.2019.9014714>

- Sugandi, B., Octaviani, S. E., & Pebrianto, N. F. (2020). Visual Tracking-Based Hand Gesture Recognition Using Backpropagation Neural Network. *International Journal of Innovative Computing, Information and Control*, 16(1), 301–313. <https://doi.org/10.24507/ijicic.16.01.301>
- World Health Organization. (2021). World Report on Hearing. In *World report on Hearing*.
- Yang, J., Zhao, Y., Chan, J. C. W., & Yi, C. (2016). The Effectiveness of Data Augmentation in Image Classification using Deep Learning. *International Geoscience and Remote Sensing Symposium (IGARSS), 2016-Novem*, 5079–5082. <https://doi.org/10.1109/IGARSS.2016.7730324>
- Yasir, F., Prasad, P. W. C., Alsadoon, A., Elchouemi, A., & Sasikumaran Sreedharan. (2017). Bangla Sign Language Recognition Using Convolutional Neural Network. *International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICICT)*, 17, 49–53. https://doi.org/10.1007/978-981-15-9651-3_58
- Yu, X. (2019). Hand Gesture Recognition Based on Faster-RCNN Deep Learning. *Journal of Computers*, 14(2), 101–110. <https://doi.org/10.17706/jcp.14.2.101-110>



APPENDIX



คณะกรรมการจริยธรรมการวิจัยในคน มหาวิทยาลัยมหาสารคาม

เอกสารรับรองโครงการวิจัย

เลขที่การรับรอง : 360-355/2564

ชื่อโครงการวิจัย (ภาษาไทย) ระบบแปลภาษามืออัตโนมัติโดยใช้การเรียนรู้เชิงลึก

ชื่อโครงการวิจัย (ภาษาอังกฤษ) The Automated Sign Language Translation System Using Deep Learning.

ผู้วิจัย : นายสิริวิวัฒน์ ละตา

หน่วยงานที่รับผิดชอบ : คณะวิทยาการสารสนเทศ

สถานที่ทำการวิจัย : คณะวิทยาการสารสนเทศ

ประเภทการพิจารณาแบบ : แบบเร่งรัด

วันที่รับรอง : 17 พฤศจิกายน 2564

วันหมดอายุ : 16 พฤศจิกายน 2565

ข้อเสนอการวิจัยนี้ ได้รับการพิจารณาและให้ความเห็นชอบจากคณะกรรมการจริยธรรมการวิจัยในคน มหาวิทยาลัยมหาสารคามแล้ว และอนุมัติในด้านจริยธรรมให้ดำเนินการศึกษาวิจัยเรื่องข้างต้นได้ บนพื้นฐานของโครงการวิจัยที่คณะกรรมการฯ ได้รับและพิจารณา เมื่อเสร็จสิ้นโครงการแล้วให้ผู้วิจัยส่งแบบฟอร์มการปิดโครงการและรายงานผลการดำเนินงานมายังคณะกรรมการจริยธรรมการวิจัยในคน มหาวิทยาลัยมหาสารคาม หรือหากมีการเปลี่ยนแปลงใดๆ ในโครงการวิจัย ผู้วิจัยจักต้องยื่นขอรับการพิจารณาใหม่

ราตรี สว่างจิตร์

(ผู้ช่วยศาสตราจารย์ เกษักรหญิงราตรี สว่างจิตร์)

ประธานคณะกรรมการจริยธรรมการวิจัยในคน

มหาวิทยาลัยมหาสารคาม

ทั้งนี้ การรับรองนี้มีเงื่อนไขดังที่ระบุไว้ด้านหลังทุกข้อ (ดูด้านหลังของเอกสารรับรองโครงการวิจัย)

BIOGRAPHY

NAME	Mr.Siriwiwat Lata
DATE OF BIRTH	15 November 1984
PLACE OF BIRTH	Chiang Khan District, Loei Province
ADDRESS	399/2, Chiang Khan Subdistrict, Chiang Khan District, Loei Province, 42110
POSITION	Lecturer
PLACE OF WORK	80/191 Department of Computer, Faculty of Science and Technology, Rajabhat Maharakham University.
EDUCATION	2006 Bachelor of Science (B.Sc.) New Media, Maharakham University. 2010 Master of Science (M.Sc.) New Media, Maharakham University. 2022 Doctor of Philosophy (Ph.D.) Information Technology, Maharakham University
Research output	[1] Chompookham, T., Gonwirat, S., Lata, S., Phiphatphaisit, S., & Surinta, O. (2020). Plant Leaf Image Recognition Using Multiple-Grid Based Local Descriptor and Dimensionality Reduction Approach. The 3rd International Conference on Information Science and System (ICISS), 72–77. https://doi.org/10.1145/3388176.3388180 [2] Lata, S., & Surinta, O. (2022). An end-to-end Thai fingerspelling recognition framework with deep convolutional neural networks. ICIC Express Letters ICIC International c, 2022(5), 529–536. https://doi.org/10.24507/icicel.16.05.529 [3] Lata, Siriwiwat, et al. (2022). Dynamic Fingerspelling Recognition from Video using Deep Learning Approach: From Detection to Recognition. ICIC Express Letters ICIC International B, 2022, 949–957. 13(9). 10.24507/icicelb.13.09.949

พหุบัณฑิต ชีวะ