Title	マルチモーダル深層学習に基づいたタイ手話における指文字認識：新しいベンチマークとモデル構成
Author(s)	WUTTICHAI, VIJITKUNSAWAT
Citation	
Issue Date	2024-06
Type	Thesis or Dissertation
Text version	ETD
URL	http://hdl.handle.net/10119/19331
Rights	
Description	Supervisor: Nguyen Minh Le, 先端科学技術研究科, 博士

Deep Multimodal-based Finger Spelling Recognition for Thai Sign Language: A New Benchmark and Model Composition

Wuttichai Vijitkunsawat

Japan Advanced Institute of Science and Technology

Doctoral Dissertation

Deep Multimodal-based Finger Spelling Recognition for Thai Sign Language: A New Benchmark and Model Composition

Wuttichai Vijitkunsawat

Supervisor : NGUYEN Le Minh

Graduate School of Advanced Science and Technology
Japan Advanced Institute of Science and Technology
Information Science

June, 2024

Abstract

Video-based sign language recognition is vital for improving communication for the deaf and hard of hearing. However, due to a lack of resources, creating and maintaining the quality of Thai sign language video datasets is challenging. To address this issue, we assess multiple models with a novel dataset of 90 signs, covering the full letters of alphabets, vowels, intonation marks, and numbers, as demonstrated by 43 signers. We investigate seven deep learning models with three distinct modalities for our analysis: video-only methods (including RGB-sequencing-based CNN-LSTM and VGG-LSTM), human body joint coordinate sequences (processed by LSTM, BiLSTM, GRU, and Transformer models), and skeleton analysis (using TGCN with graph-structured skeleton representation). A thorough assessment of these models is conducted across seven circumstances, encompassing single-hand postures, single-hand motions with one, two, and three strokes, and two-hand postures with static and dynamic point-on-hand interactions. The research highlights that the TGCN model is the optimal lightweight model in all scenarios. In single-hand pose cases, a combination of the Transformer and TGCN models of two modalities delivers outstanding performance, excelling in four particular conditions: single-hand poses, single-hand poses requiring one, two, and three strokes. In contrast, two-hand poses with static or dynamic point-on-hand interactions present substantial challenges, as the data from joint coordinates is inadequate due to hand obstructions stemming from insufficient coordinate sequence data and the lack of a detailed skeletal graph structure. The study recommends integrating RGB-sequencing with visual modality to enhance the accuracy of two-handed sign language gestures. Moreover, experimental results on our dataset show that our method outperforms previous state-of-the-art methods significantly in five out of seven conditional hand pose experiments, especially two-hand poses.

Keywords: Thai Finger Spelling, Sign Language Recognition, Deep Learning, Multimodal Learning, Benchmark Dataset.

Acknowledgments

Firstly, I would like to express my sincerest gratitude to my supervisor during my research period, Professor Nguyen Le Minh of the Japan Advanced Institute of Science and Technology (JAIST). He inspired me to find the appropriate research direction as well as taught me how to deal with problems in study.

I am grateful to Prof. Teeradaj Racharak (Prof.X) for his great suggestions and comments on my research. I really learned quite a lot from Prof.X, especially how to do great research. He is a kind professor and does me a great favor in my research studies and presentations.

I deeply thank the committee members: my second supervisor, Associate Professor Inoue Naoya, and Professor Satoshi Tojo at the Asia University for useful suggestions and comments on my study. Through discussions, they helped me recognize the limited points of my research as well as provided useful suggestions for improving the thesis.

I am deeply indebted to the National Science and Technology Development Agency in Thailand for granting me a doctoral program scholarship, which also supported me to attend and present my work at an international conference during the period of my research. Thanks also to the Setsatian Deaf School, Thungmahamek deaf school, and National Association of the Deaf in Thailand for providing the Thai Finger Spelling dataset to be input data for my research.

Finally, I would like to express my sincere thanks to all members of my family who always supported me with love and great patience. Without their support, I might never complete this work.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Introduction of Sign Language

The World Health Organization's 2023 report reveals the extensive prevalence of deafness and hearing loss, highlighting that these conditions are pervasive across all countries and regions. At present, more than 1.5 billion individuals, or nearly 20% of the entire global population, are living with hearing loss; 430 million of whom have disabling hearing loss. By 2050, it is anticipated that the number of people with disabling hearing loss could increase to over 700 million. On a global level, deafness or hearing loss affects 34 million children, 60% of which are due to preventable circumstances. In the older population, around 30% of those aged 60 and above are affected by hearing loss [37].

Sign language plays a critical role as a communication medium, predominantly within the communities of individuals who are deaf or hard of hearing. This form of communication comprises a comprehensive range of hand signs, facial expressions, and body movements, all of which work together to convey meaning independently of spoken languages. Over the years, sign language has gained immense recognition as a valid linguistic system, a development largely attributed to extensive research and strong advocacy. Sign languages are diverse, with various forms developed across different regions and communities. American Sign Language (ASL), British Sign Language (BSL), Thai Sign Language (TSL) and French Sign Language (LSF) are among the many distinct sign languages in

use today, each with its unique grammar, vocabulary, and syntax [47]. These languages are not universally interchangeable, even between countries that share the same spoken language, highlighting their unique evolution and cultural significance [28].

Thai Sign Language (TSL), developed by Khunying Kamala Krairuek in 1953, has been significantly influenced by American Sign Language (ASL). TSL comprises two principal types: Natural Thai Sign Language (NTSL), which is used to convey complete semantic meanings of words and sentences, exemplified by terms such as "house", "table", "chair" and "tree", and Thai Finger Spelling (TFS), which is utilized for the specific spelling of names or signs lacking standard gestures. TFS has 42 alphabets, 32 vowels, and 4 intonation marks.

1.2 Problems of Thai Sign Language

Thai Finger Spelling, Thailand's standard sign language, suffers from a lack of public datasets and proficient users. The field faces numerous technical challenges, including dense occlusions at hand keypoints [16], hindering accurate manual annotations [19] and, consequently, the creation of reliable hand keypoint recognition models. This is critical as hand detection is essential for various applications, from action recognition to sign language translation, and is complicated by the variety in hand shapes, gestures, and issues like occlusion and low resolution [44]. A considerable number of Thai researchers have proactively developed their own datasets to address these challenges in Thai Sign Language (TSL). However, there is a significant concern that these datasets may not be thoroughly validated by experts and could be incomplete, possibly lacking some letters. This situation highlights the complexities of TSL and the critical need for increased resources and advanced technology solutions. (cf. Sect 2.1)

1.3 Overiew of Contributions

The contribution of our work mainly lies in the following three aspects. Firstly, we have developed a comprehensive video database for Thai Finger Spelling (TFS) in sign language, featuring 10,467 videos of 90 unique letters demonstrated in different poses with one or both hands, contributed to by 43 diverse signers, appearances and backgrounds. Furthermore, our dataset comprehensively covers all aspects of TFS and nearly achieves complete balance, with 95% of the dataset obtained through direct video recording and 5% from internet sources. In this research, our video dataset stands as the largest in the TFS domain, marking a significant milestone. It is the first to comprehensively cover primary letter finger spelling, catering specifically to the needs of the Thai sign language research community.

Secondly, we perform comprehensive research on designing and developing a finger spelling recognizer for TFS based on our collected dataset. In particular, our recognizer is analyzed based on extensive experiments in three modalities and different representation learning techniques: RGB-sequencing-based modality on CNN-LSTM and VGG-LSTM models, The coordinate sequence of joint structure modality in the human body with LSTM, BiLSTM, GRU and Transformer models, and the graph structure on the skeleton modality using TGCN. We have designed seven important experiments to meticulously evaluate our framework, focusing on distinct hand poses and gestures. The experiments cover: static single-hand poses with single-stroke, dynamic single-hand poses that require two or three strokes, two-hand poses with a static point-on-hand, and two-hand poses with dynamic point-on-hand, total two-hand poses. To measure the performance across these various scenarios, we use evaluation metrics such as accuracy(Top-1), Top-3, Top-5, recall, precision, and F1-score, testing 29 experimental models that include single-based, dual, and triple modalities. Upon obtaining the optimal model, it will be compared against baseline and state-of-the-art models to benchmark its performance.

Thirdly, we conduct comprehensive statistical tests, including both in-sample and out-of-sample evaluations, to rigorously identify the model that demonstrates the highest

efficiency. This meticulous approach ensures that we are able to recommend a model that is most suitable for practical, real-world applications, guaranteeing its reliability and effectiveness in various situations.

1.4 Thesis Outlines

This research is mainly focused on solving communication and accessibility for the deaf and hard-of-hearing communities. However, developing and maintaining high-quality sign language datasets from video input is difficult, especially in Thai, because of the lack of a standard Thai finger spelling video dataset. To overcome this challenge, this research must focus on accumulating a larger total of 90 primary letters in Thai Finger Spelling which covers alphabets, vowels, intonation marks and numbers from 43 signers with various backgrounds, genders, and appearances. We conduct seven deep learning-based architectures on three modalities: RGB-sequencing-based CNN-LSTM and VGG-LSTM for video-only modality, a sequence of coordinates of joints in human's body modality using LSTM, BiLSTM, GRU and Transformer models, and the structure of human's joints modality using TGCN, as well as their combinations with many modalities. The thesis contains five chapters:

- Chapter 2 provides a comprehensive overview of diverse topics relevant to our thesis. It begins by introducing Thai Sign Language (TSL), the national sign language of the Thai Deaf community, and delves into an important component of TSL, Thai Finger Spelling (TFS). The chapter then transitions to explore the technological aspects of our study, discussing various types of Convolutional Neural Networks (CNNs), including the VGG16 model. This is followed by an in-depth exploration of Recurrent Neural Networks (RNNs), which includes Long Short-Term Memory (LSTM), Bidirectional LSTM (Bi-LSTM), and Gated Recurrent Unit (GRU). We also describe the Transformer model, encompassing the attention mechanism, self-attention, and multi-head attention, before explaining the concept of Graph Neural Network (GNN). Finally, the chapter concludes with a literature review that syn-

thesizes various research studies related to our thesis, bridging the gap between sign language and advanced neural network models.

- Chapter 3 delves into the TFS dataset used in this study, sourced primarily from the Internet and actual recordings from deaf schools managed by experts. This includes videos from the Office of the Royal Society and the National Association of the Deaf in Thailand, as well as footage from the Royal Patronage of His Royal Highness Crown Prince Maha Vajiralongkorn School for the Deaf and Thungmahamek School for the Deaf. The chapter then outlines the pre-processing of this data, which involves three key steps: hand cropping using the YOLO model to isolate relevant gestures, resizing and padding images for uniformity, and balancing frame sequences through oversampling and downsampling. This is crucial for preparing the data for the next stage of analysis. We proceed to discuss the analytical models used. 2D Convolutional Neural Networks (CNNs) extract spatial details from images, while Recurrent Neural Networks (RNNs), including a stacked LSTM, capture the temporal dynamics between frames. A combination of 2D CNNs and RNNs is employed to analyze spatial-temporal characteristics in the video data, utilizing the VGG16 model pre-trained on ImageNet for extracting spatial features. Further, the chapter explores the coordinate sequencing of human joints, using the holistic key points from the MediaPipe library to map out 24 key points for a body and a single hand, and 48 for two hands. This detailed joint mapping facilitates precise human pose estimation and action recognition. Lastly, the chapter describes the innovative use of graph representation for joint structure modality, a deep learning architecture tailored for graph-structured data. This approach is highly effective for tasks involving human body joint positions. The chapter concludes with a comparative analysis of data fusion techniques, evaluating seven baseline models against specific metrics to determine the most effective model for both in-sample and out-of-sample testing.

- Chapter 4 performs seven main experiments: total single-hand pose, total two-

hand pose, static-single-hand with a one-stroke posture, dynamic-single-hand with two-strokes and three-strokes postures, static point-on-hand, and dynamic-point-on-hand postures. These experiments are evaluated using a confusion matrix that includes accuracy, recall, precision, and F1-score. They are applied across three modalities using 29 models ranging from a single-based model to combinations of two and three modalities.

- Finally, Chapter 5 provides conclusions and discussion about the weak and strong points of proposed models in the main problems of seven experiments. Besides, we also discuss some case studies directions for future work to improve our work.

Chapter 2

Background

This chapter covers various fundamental topics which are crucial knowledge for deep learning-based sign language finger spelling recognition. We start with a brief explanation of Thai Sign Langauge (TFS), Artificial Neural Networks (ANN), Convolutional Neural Networks (CNN), the Recurrent Neural Networks (RNN), Long-Short Term Memory (LSTM), Bidirection-Long Short Term Memory (Bi-LSTM), Gate Recurrent Unit (GRU), Transformer, Graph Neural Network (GNN),and Literature Reviews.

2.1 Thai Sign Language

Thai Finger Spelling (TFS), a technique for using hand posture to spell out particular names, places, or technical words using the alphabet, vowels, intonation marks, and numbers, was created in 1956 by Khunying Kamala Krairuek and is influenced by American finger-spelling methodologies, as shown in Figure 2.1

Thai Finger Spelling, there is a phonetic comparison made to American Finger Spelling (AFS), where Thai characters are paired with American letters that have a similar sound. A case in point is the Thai " ก "(Ko kai), which phonetically matches the “ K ” in American finger-spelling, resulting in the use of the “ K ” hand posture for representing " ก "(Ko kai) in Thai, as seen in Figure 2.2.

Nonetheless, to account for all 42 Thai letters, extra finger-spellings have been incor-

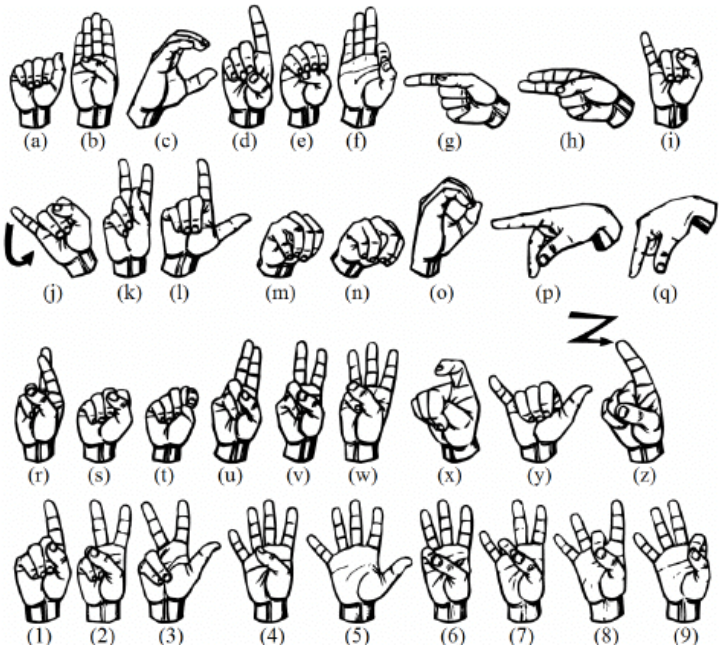


Figure 2.1: The American Finger Spelling hand posture [24]

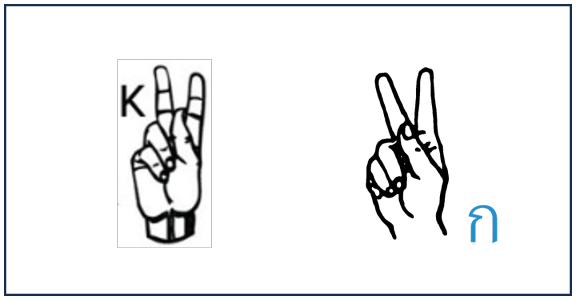


Figure 2.2: Comparison between “K” of AFS and " ก " of TFS

porated through combining various hand postures from American finger-spelling. Some TFS letters need consecutive letters such as " ข "(Kho khai), " ค "(Kho khwai) and " ฆ "(Kho ra-khang) are “K + 1”,“K + 2” and “K + 3”, respectively, as seen in Figure 2.3

In contrast to English, the Thai language has more vowels and intonation marks, requiring complex two-handed finger-spellings for accurate representation. The dominant hand forms the letters, while the subordinate hand selects specific vowels like " -ะ "(Sara a), " -า "(Sara ar), and " เ- "(Sara ae), which are the static point-on-hand with two-hand posture or intonation marks such as " -่ "(Mai ake), " -้ "(Mai tho) and " -๊ "(Mai tee) based on dynamic point-on-hand , as detailed in Figure 2.4 - 2.5.

Figure 2.6 provides a visual representation of all the Thai finger spellings that were formulated, but it does not include values for the obsolete Thai consonants " ฅ " and "

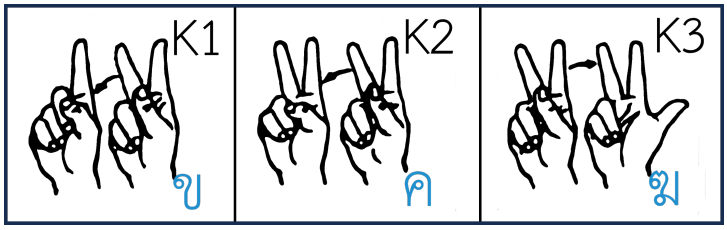


Figure 2.3: Letters " ข ", " ค " and " ฆ " of TFS

Figure 2.4: Example static point-on-hand letters " -ะ ", " -า " and " เ- " of TFS

Figure 2.5: Example dynamic point-on-hand letters " -่ ", " -้ " and " -๊ " of TFS

ฆ ". Finger-spelling for 42 Thai consonants and seven vowel symbols can be performed using just one hand, while other symbols necessitate the use of both hands. In some cases of finger spelling with two-hand poses, they might be required to represent the phoneme of a particular vowel.

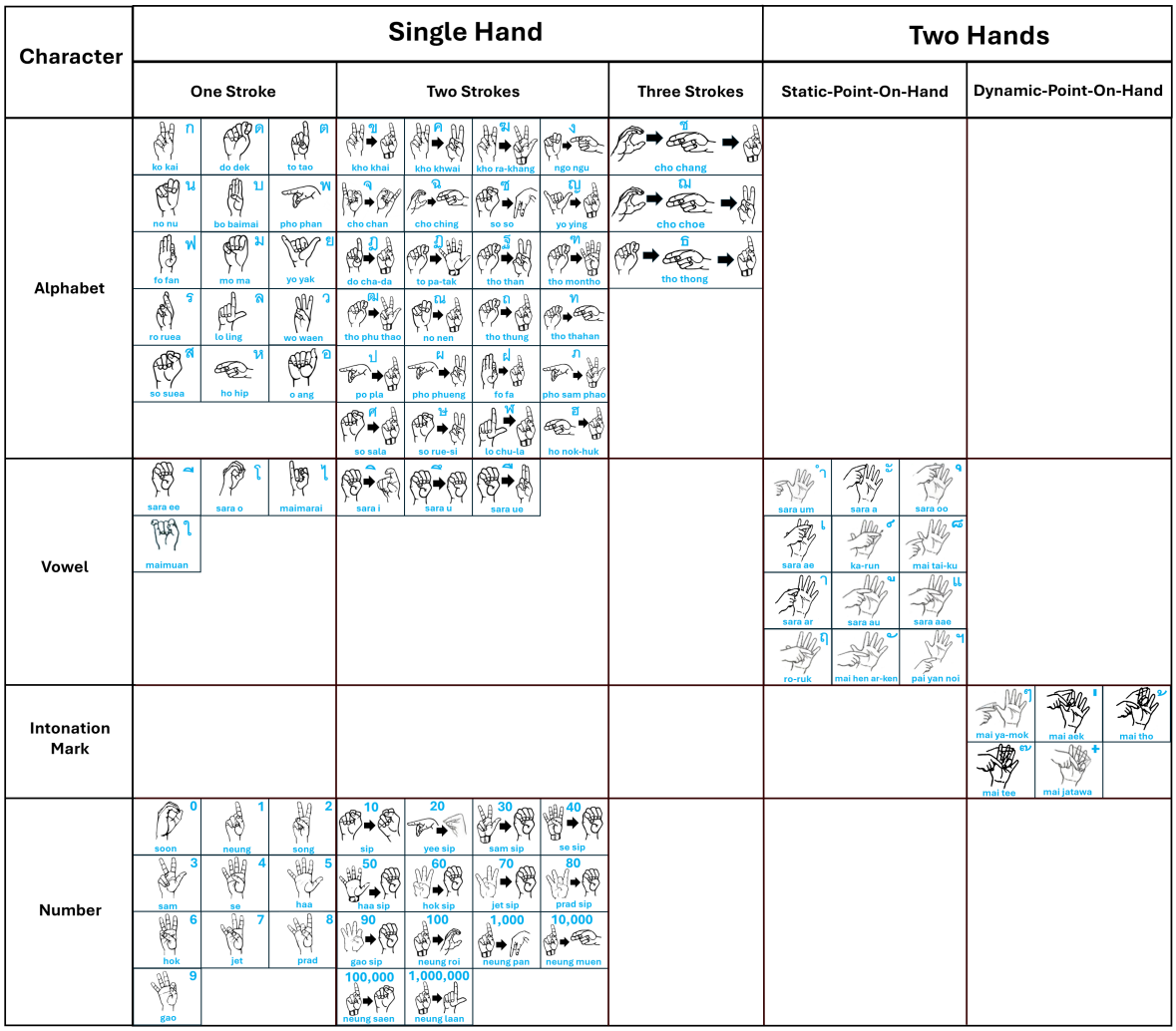


Figure 2.6: Thai Finger Spelling (total of our dataset)

2.2 Artificial Neural Network

The human nervous system served as the inspiration for mathematical models of artificial neural networks. Numerous cells known as neurons are found in the human nervous system. The neurons' axon cell processes the output signals that are sent via it after receiving input signals from their dendritic cells. The human nervous system operates in a complex manner because neurons form networks connecting their axon cells to the dendritic cells of other neurons. An example of a single human neuron is shown in 2.7.

Artificial neural networks, comprising numerous interconnected artificial neurons, emulate human neural processing. Each neuron multiplies various inputs by their weights,

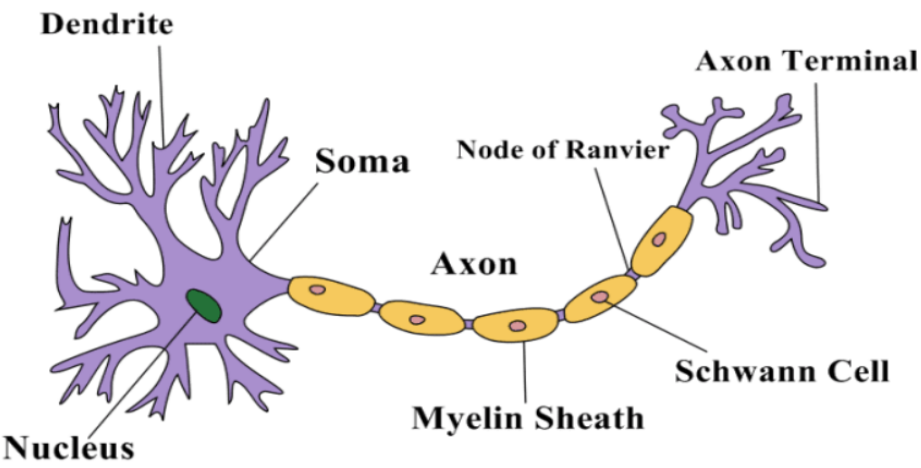


Figure 2.7: An example of a single human neuron [59]

sums them, and then applies an activation function to the result, aiding in decision-making. The simplest activation function used is the binary step function, yielding an output of 0 or 1.

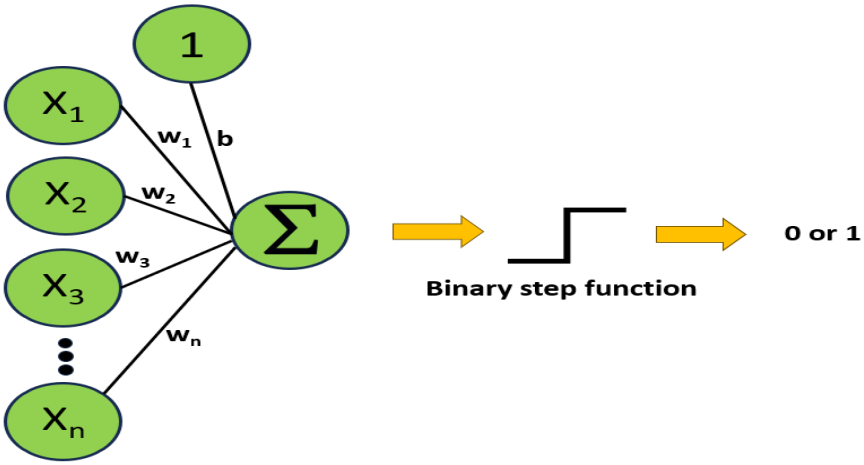


Figure 2.8: An example of a single neuron with binary step function [52]

A single artificial neuron with the inputs $x_1, x_2, x_3, ...x_n$ is shown in Figure 2.8. It uses the binary step function to generate a single binary output. The output also includes a bias, which is a constant value that helps the decision. The output from a single artificial neuron can be calculated by

$$Output = \begin{cases} 0, & \sum_j w_j x_j + b < 0 \\ 1, & \sum_j w_j x_j + b \geq 0 \end{cases} \tag{2.1}$$

where w_j is a weight value of the input x_j and b is bias value.

A single artificial neuron connects to another to form a network, enabling the processing of complex tasks. This implies that the output of one neuron serves as the input for another. Such a network of neurons is termed an Artificial Neural Network. Figure 2.9 displays an artificial neural network example comprising three inputs and two layers. The network's first layer is the Input Layer, the final layer is the Output Layer, and any intervening layers are Hidden Layers. It is noteworthy that the count of layers in a network includes the Input Layer, all Hidden Layers, and the Output Layer.

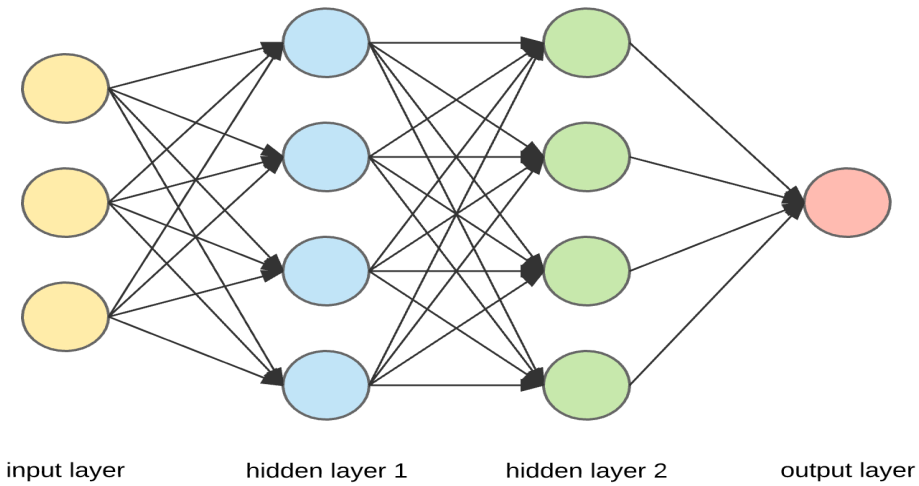


Figure 2.9: An example of an artificial neural network [33]

Although the network is configured as a complex structure, employing the binary step function in the output layer leads to a rather simplistic decision-making process, capable of producing only a 'yes' or 'no' output. Instead, the Sigmoid function is utilized to generate a continuous value within the range [0, 1]. The graphical representation of the Sigmoid function can be seen in Figure 2.10. Utilizing a soft decision value renders artificial neural networks more versatile, applicable to a broader array of tasks beyond mere binary 'yes' or 'no' problems. The output from a neuron utilizing the Sigmoid function can be calculated by Equation 2.2.

$$
\begin{aligned}
z &= \sum_j w_j x_j + b \\
o &= \sigma(z) = \frac{1}{1+e^{-z}}
\end{aligned}
\tag{2.2}
$$

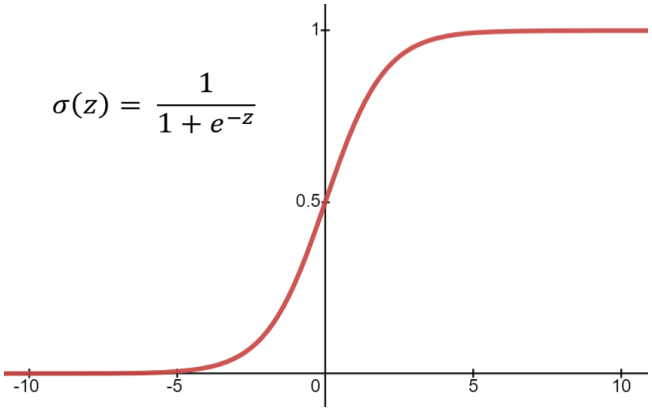


Figure 2.10: Sigmoid function

In addition to the sigmoid function, numerous other functions serve as activation functions within neural networks, one of which is the Tanh function. The Tanh function is quite similar to the sigmoid function; however, they differ in their output ranges. For the sigmoid function, the range is [0, 1], while for the Tanh function, it is [-1, 1]. The graph depicting the Tanh function is displayed in Figure 2.11. Owing to its range, the Tanh function produces values that are zero-centered, which can be more efficient for neural network computations. The output from a neuron employing the Tanh function can be calculated by

$$
\begin{aligned}
z &= \sum_j w_j x_j + b \\
o &= \tanh(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}
\end{aligned}
\tag{2.3}
$$

One of the most popular activation functions used in neural networks is the ReLU, or Rectified Linear Units function, with its graph displayed in Figure 2.12. The sigmoid and tanh functions are commonly utilized as activation functions in the output layer for decision-making purposes. In contrast, the ReLU function is typically employed within hidden layers to introduce a simple yet effective nonlinearity. This nonlinearity is crucial for neural networks, as it allows deep learning models to approximate any continuous function, thereby enabling the network to manage highly complex tasks. The value of a

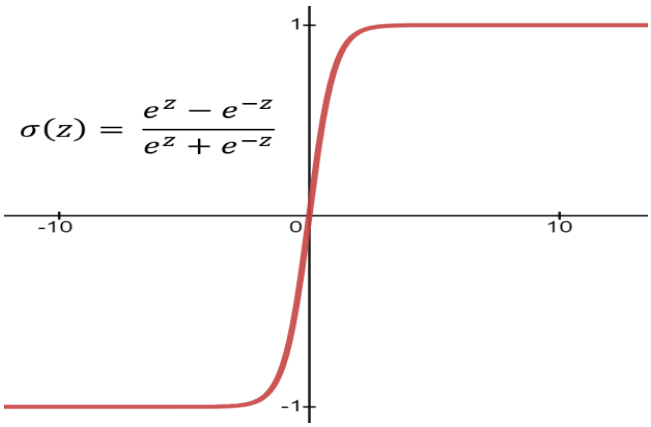


Figure 2.11: Tanh function

neuron employing the ReLU activation can be calculated by

$$ReLU\,(z) = max\,(0, z) \tag{2.4}$$

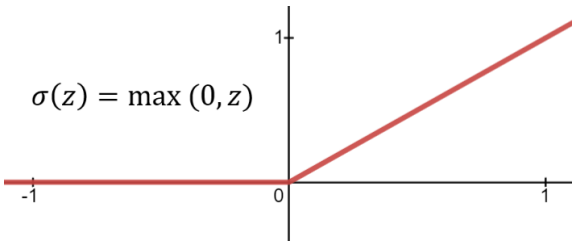


Figure 2.12: ReLU function

2.3 Convolutional Neural Network

Handling more complex data, such as image data, cannot be efficiently processed by traditional neural networks (also known as dense networks) because they are structured to handle 1D data, whereas image data is inherently 2D spatial data, containing both width and height information. Therefore, to make this data compatible with dense networks, the image data must first be flattened, transforming it from 2D to 1D data, enabling computation within these networks.

According to Figure 2.13, an image measuring 28x28 is flattened into a 1D array comprising 784 elements, which corresponds to the total number of pixels in the image. This

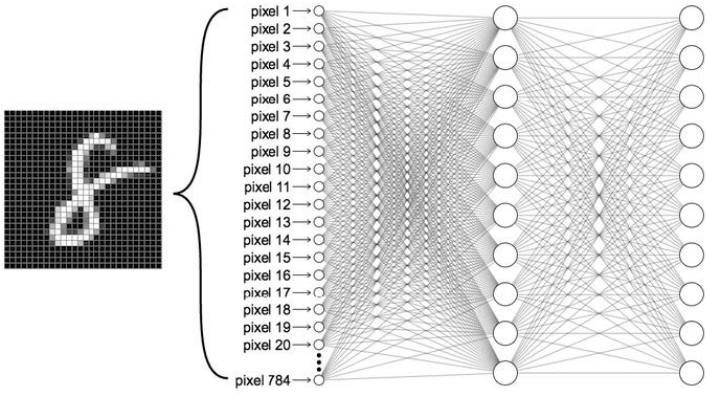


Figure 2.13: An example of flattening an image [58]

example highlights that dense networks are not ideally suited for spatial data computation. Dense networks process each input independently with its respective weights, whereas spatial data typically feature related neighboring elements. Just as with pixels in an image, neighboring pixels tend to be part of the same object and hold some relation to each other. In addition to the loss of spatial relationships, dense networks pose computational complexities. When a dense network uses an image of 28x28 pixels for input, it requires 784 weight parameters. However, 28x28 is considered very low resolution for practical applications. In my thesis, we use the image size of 224x224 pixels would require over 50,000 parameters. Moreover, if the image is a color image containing three color channels—including red, green, and blue—the number of parameters would exceed 150,000 per image.

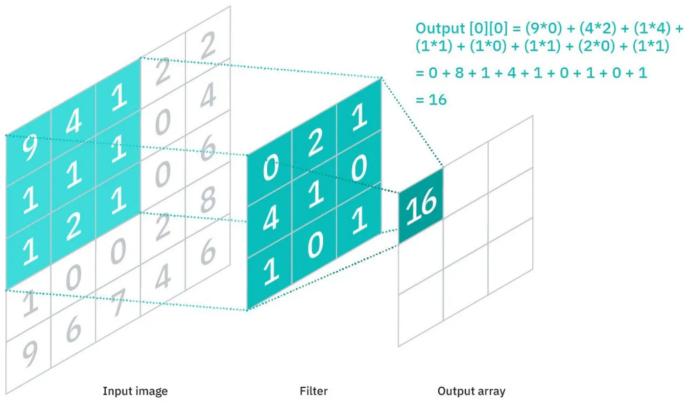


Figure 2.14: An example of convolutional operation [42]

A Convolutional Neural Network (CNN) was developed to address the computational

complexity issue; it is also designed to accommodate the spatial structure of data. A CNN comprises several filters, each acting as a small window that scans across the entire image to extract features. Figure 2.14 illustrates the process of extracting a feature map from an image with a single color channel using a convolutional layer with one filter. As a consequence, utilizing the convolutional layer, as shown in Figure 2.14, with a 28x28 picture reduces the number of parameters from 784 to only nine parameters. This example, on the other hand, depicts a convolutional layer with a single filter. Convolutional layers are often composed of many filters that extract a range of characteristics, allowing the network to accomplish difficult tasks. Figure 2.14 is an example of a convolutional procedure with many filters. Moreover, convolutional layers may be connected to create a deep network, which can greatly simplify computation. Convolutional neural networks are therefore quite efficient for contemporary computer vision tasks.

VGG, an abbreviation for Visual Geometry Group, is a model of a convolutional neural network [51]. A distinguishing feature of VGG16 is its design that replaces a large number of hyperparameters, focusing instead on 3x3 pixel conv2D layers with a stride of 1 and consistent use of "same" padding, as well as 2x2 pixel max pooling with a stride of 2 throughout the architecture. The designation "VGG16" refers to the 16 weighted layers within the network, which is considerable in size and contains approximately 138 million parameters. Also, The default input size for this model is 224x224.

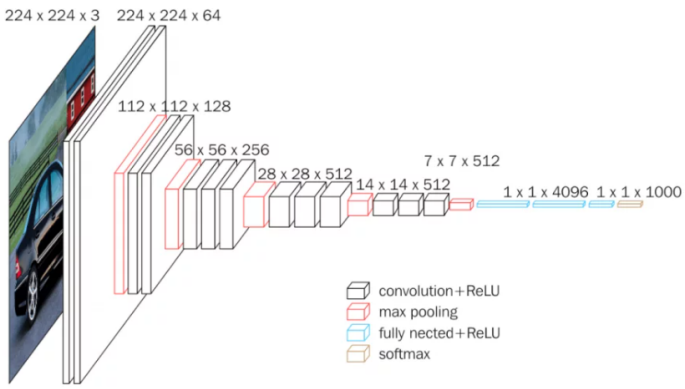


Figure 2.15: VGG16 architecture [2]

2.4 Recurrent Neural Network

Sequence data plays a crucial role in several applications and constitutes a significant portion of the data shared on the internet, including various forms such as videos, sentences, and voice. Nevertheless, it is worth noting that the dense network and the convolutional network are limited in their ability to handle sequence data. This is due to the absence of temporal considerations in their operations, which is a crucial aspect for processing sequence data that is inherently time-dependent. Let W represent words in the sentence "Cat is eating fish," for instance, where $W = \{w_1, w_2, w_3, w_4\} = \{"Cat", "is", "eating", "fish"\}$. Given that w_4 is impacted by the verb at w_3, it is evident from the phrase that it is an edible object.

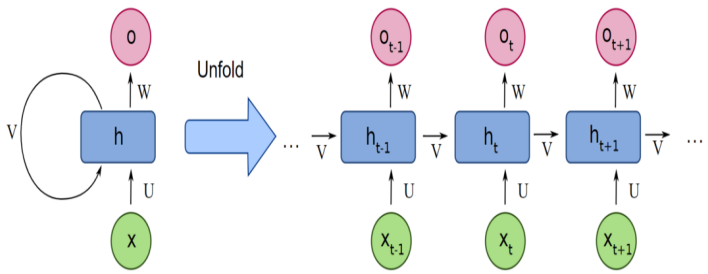


Figure 2.16: Recurrent Neural Network [67]

The Recurrent Neural Network (RNN) has been designed to effectively process temporal data. The recurrent neural network functions by transmitting computed data from the current state t to the subsequent state $t+1$. As a result, the state at time $t+1$ incorporates information from both the preceding state and its own state, which may be explained by equation 2.5.

$$\begin{aligned} h_t &= f\left(Ux_t + Vh_{t-1}\right) \\ o_t &= Wh_t \end{aligned} \tag{2.5}$$

where x_t and h_t represent the input and the hidden state at time t, respectively. The hidden state serves as the 'memory' of the network. The function f denotes a non-linear transformation, such as the hyperbolic tangent (tanh) or the Rectified Linear Unit (ReLU), and U, V, and W are the weight matrices.

Recurrent Neural Networks (RNNs) may be categorized into many varieties according to their input and output formats, each of which is employed for distinct purposes. Figure 2.17 provides an illustration showcasing the diverse range of types and applications of a RNN.

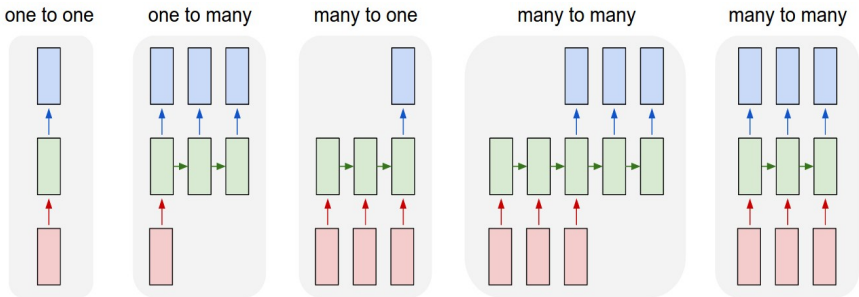


Figure 2.17: Recurrent Neural Network Application [7]

Each rectangular shape in the context refers to a vector, while the presence of arrows signifies the representation of functions, such as matrix multiplication. The input vectors are represented in the color red, while the output vectors are shown in the colors blue. From the left to right images: (1) The vanilla mode of processing, which does not involve Recurrent Neural Networks (RNNs), pertains to a computational approach that operates on fixed-sized input data and produces fixed-sized output data. An example of such a task is image classification. (2) Sequential output refers to the process of taking an input, such as an image, and generating a sequence of words as output, as seen in tasks like image captioning. (3) Sequential input refers to the process of analyzing the sentiment of a particular sentence and classifying it as either conveying positive or negative emotion. (4) sequence input and sequence output in the field of natural language processing is machine translation. In this task, a recurrent neural network (RNN) is employed to process a series of words in English as input and generate a corresponding sequence of words in French as output. (5) The synchronized sequence input and output refers to the process of labeling each frame of a video, such as in the case of video classification. It is important to observe that there are no predetermined limitations on the lengths of sequences in each case, as the recurrent transformation (shown by the green color) remains constant and can be repeatedly performed without restriction.

2.4.1 Long-Short Term Memory (LSTM)

Traditional recurrent neural networks can perform well with sequence data, but they have limitations when dealing with very lengthy sequences. For example, the missing word in the statement "I am Thai, I come from ____" should be "Thailand," as the third word in the sentence suggests. The recurrent neural network performs well in this situation since the text is quite short and the two related terms, "Thai" and "Thailand," are not too far apart, being the third and ninth words, respectively. Consider a longer sentence, for example: "I went to Thailand several years ago; there are beautiful temples, friendly people, and delicious food, which is why I love Thailand.". The words "Thailand" in the 4th and 25th words of the phrase are releated, but their positions are far apart, as opposed to the first example. A recurrent neural network operates by stacking information from a prior state onto the present state, resulting in information from the beginning state having little influence on the final state. This is known as the "long-term dependency" issue. LSTMs are RNNs whose main objective is to overcome the shortcomings of the vanishing gradient and exploding gradient problems. The architecture is built so that they remember data and information for a long period of time.

LSTMs consist of six main operations: (1) Forget gate (f_t), (2) Input gate (i_t), (3) Cell update ($\widetilde{C}_t$), (4) Cell state (C_t), (5) Output gate (o_t) and (6) Output (h_t)

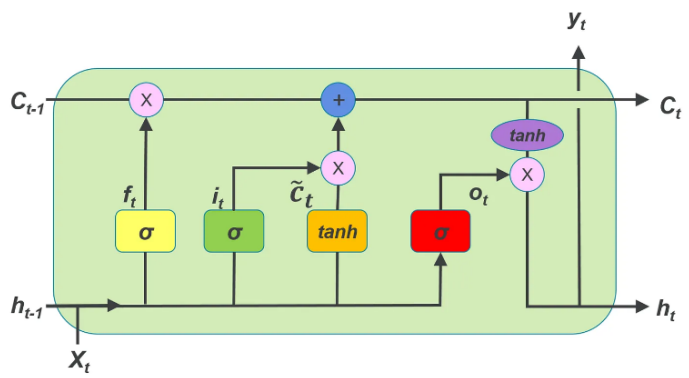


Figure 2.18: Long-Short Term Memory (LSTM) architecture [68]

$$
\begin{aligned}
i_t &= \sigma\left(x_t U_i + h_{t-1} W_i\right) \\
f_t &= \sigma\left(x_t U_f + h_{t-1} W_f\right) \\
o_t &= \sigma\left(x_t U_o + h_{t-1} W_o\right) \\
\widetilde{C} &= \tanh\left(x_t U_g + h_{t-1} W_g\right) \\
C_t &= \sigma\left(f_t * C_{t-1} + i_t * \widetilde{C}_t\right) \\
h_t &= \tanh\left(C_t * o_t\right)
\end{aligned}
\tag{2.6}
$$

2.4.2 Bidirection-Long Short Term Memory (Bi-LSTM)

A Bidirectional-Long Short Term Memory (Bi-LSTM) network is a type of advanced Recurrent Neural Network (RNN) that improves upon the standard LSTM networks by processing data from both forward and backward directions. This dual-direction processing allows the model to have a more complete understanding of the sequence context.

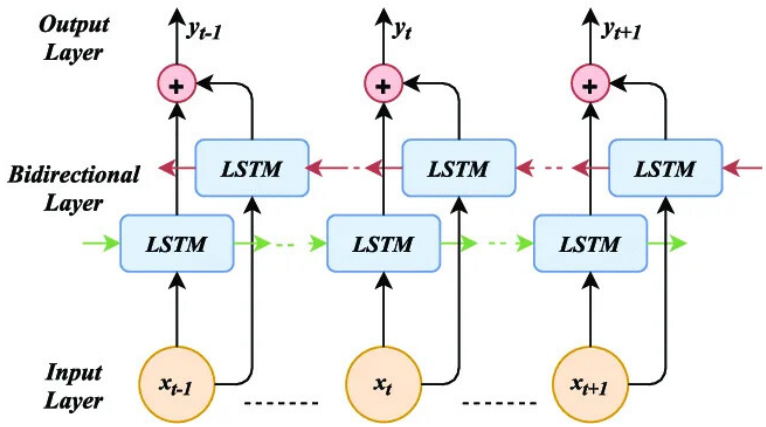


Figure 2.19: Bidirectional LSTM (Bi-LSTM) [22]

For instance, The first statement is “Server can you bring me this dish” and the second statement is “He crashed the server”. In these two sentences, the term “server” carries distinct meanings, which are influenced by the words that come before and after it in each sentence. The Bi-LSTM enhances the machine’s comprehension of this relationship more effectively than a unidirectional LSTM. The Bi-LSTM’s strengths make it an ideal choice for applications in sentiment analysis, text classification, and machine translation.

2.4.3 Gate Recurrent Unit (GRU)

The Gated Recurrent Unit (GRU) is a type of recurrent neural network (RNN) architecture that, like the Long Short-Term Memory (LSTM) network, is designed to efficiently capture dependencies for sequences of data. GRUs were introduced to solve the vanishing gradient problem that can occur in traditional RNNs, which makes it difficult for the RNN to learn and retain long-term dependencies in the data.

GRUs simplify the LSTM model with fewer parameters. They achieve this by combining the forget and input gates into a single "update gate" and by merging the cell state and hidden state. The update gate, which helps the model to decide to what extent the new output will be based on the previous memory. If the update gate is on, the GRU will transfer all the information from the previous time step to the current step without changes. The reset gate, which allows the GRU to decide how much of the past information to forget. This is useful for the model when it needs to remove irrelevant information from the past and helps the GRU to process sequences where the gap between relevant information is large. This can make GRUs faster to compute and easier to train on smaller datasets.

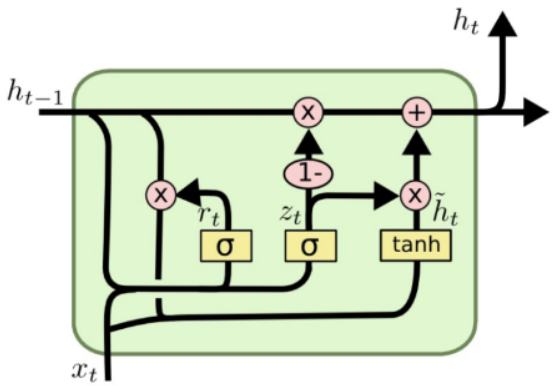


Figure 2.20: Gate Recurrent Unit (GRU) [31]

$$
\begin{aligned}
z_t &= \sigma\left(W_z \cdot [h_{t-1}, x_t]\right) \\
r_t &= \sigma\left(W_r \cdot [h_{t-1}, x_t]\right) \\
\widetilde{h_t} &= \tanh\left(W \cdot [r_t * h_{t-1}, x_t]\right) \\
h_t &= (1 - z_t) * h_{t-1} + z_t * \widetilde{h_t}
\end{aligned}
\tag{2.7}
$$

2.5 Transformer Architecture

2.5.1 Attention Mechanism

In the real-world context, there's an excess of information, yet the amount that is truly meaningful is limited. Information processing involves extracting significant information from original sources, rather than creating new data. This information usually comes with 'noises'—irrelevant environmental signals. The nature of these noises is contingent on the specific task being performed and the aspects of the data we aim to learn. Attention serves as a key tool in models, both human and machine, to zero in on the vital parts of the information.

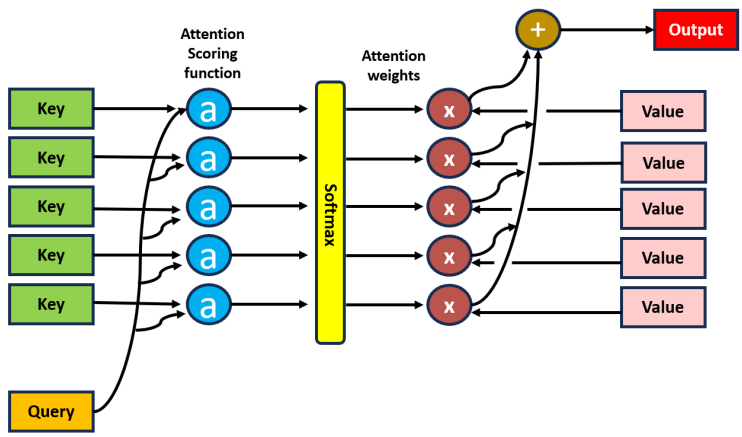


Figure 2.21: Attention Mechanism [54]

Figure 2.21 shows a diagram of a neural network using attention mechanism. There are three essential elements of this paradigm which are *query*, *keys* and *values*. *Query* serves as an indicator, signaling to the model which (*keys*, *values*) pair should be the focus of attention. Let $\mathbf{q}$ be the query vector, (k_i, v_i) be the ith keys, value pair in the candidate

lists, function f to calculate the attentive output follows the Equation 2.8

$$f\left(q,\left(k_1,v_1\right),...,\left(k_n,v_n\right)\right)=\sum_{n}^{i=1}\alpha\left(q,k_i\right)v_i \tag{2.8}$$

In which $\alpha\left(q,k_i\right)$ is determined by applying a softmax function to the outputs from function a which computes the alignment score between a query vector and a key vector, as shown in the Equation 2.9

$$\alpha\left(q,k_i\right)=softmax\left(a\left(q,k_i\right)\right)=\frac{exp(a\left(q,k_i\right)}{\sum_{j=1}^{m}exp(a\left(q,k_i\right)} \tag{2.9}$$

2.5.2 Self-Attention

The Transformer-based architecture prominently utilizes self-attention as its key mechanism. This method enables the model to combine signals from various positions within a sequence, achieving improved representation without relying on a recurrent structure. This architecture forms the foundation for state-of-the-art, pre-trained models in various NLP tasks.

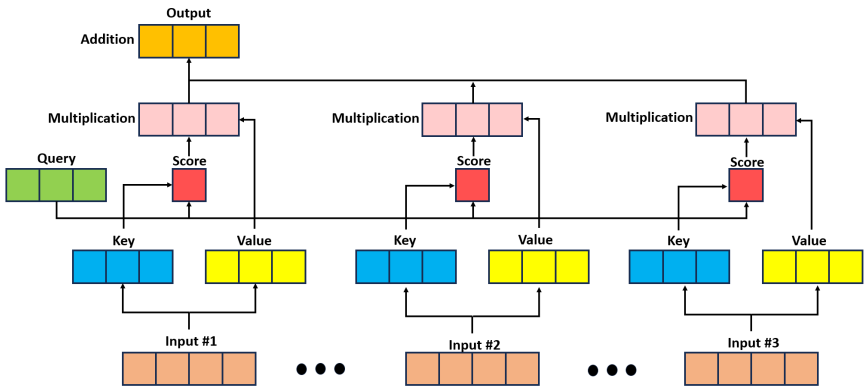


Figure 2.22: Self-Attention [30]

Figure 2.22 illustrates a case of self-attention computation.In this scenario, a new representation for the initial input vector is calculated. First, the query, key, and value vectors are derived from the input vector using relevant weight matrices. Following this, the computation is performed on the query vector of the first input and the (key, value) pairs of all inputs to achieve the result. The process is repeated for the sequence and the new

representations contain signals from all of the different positions in the origin sequence with varying weights.

2.5.3 Multi-Head Attention

Multi-head attention is a key concept in Transformer-based architectures. It enables the model to adopt various perspectives in constructing alignments within an input sequence. As detailed in Figure 2.23, instead of using only one signal attention module, this architecture calculates the attention representation in multiple subspaces (i.e., multiple heads) then concatenate the signal from all heads to a vector. After that, this vector is subsequently modified by fully-connected (FC) layers to achieve the appropriate dimension.

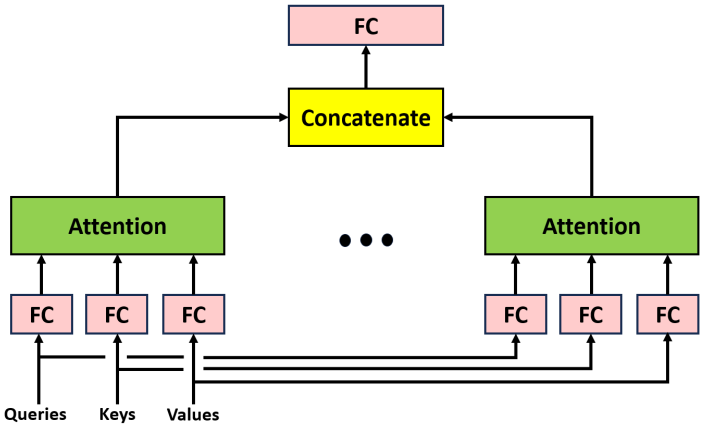


Figure 2.23: Multi-Head Attention [65]

2.6 Graph Neural Network

GNNs, or graph neural networks, are a type of neural network in which the core structure processes input in the form of a graph as in Figure 2.24 [18], [48]. GNNs' applications first shows in supervised learning of molecular characteristics in [17]. GNNs can be constructed and tuned later using a simple Message Passing Neural Network (MPNN) [69]. As a result, larger graph structures have been constructed to study behavior or extract discovery from data, such as social network links or any form of connection data.

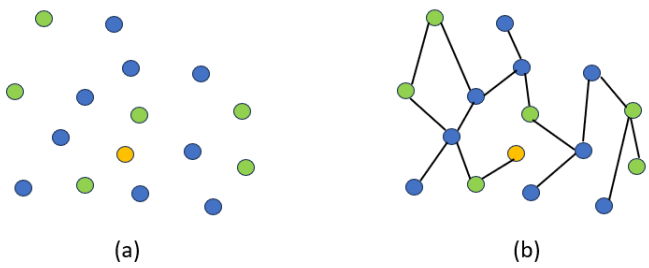


Figure 2.24: Graph structure (a) nodes (b) nodes connected with edges

GNNs are proved as a weak form of the Weisfeiler–Lehman graph isomorphism [14]. The growing interest in merging GNNs has helped in the training of GNNs models in developing "Geometric Deep Learning," which use graph representation to interpret data [32].

To define a graph G, the structures are consisted of nodes v and a set of edges e. The notion of the graph is written here as $G = (v, e)$. The number of nodes are identified ranging from 1 to $|v|$. The relation between node is defined directly as edges from node $i \rightarrow j$ or shown as a pair of nodes $(i, j)\, \epsilon\, v \times v$. In undirected graph case, once the nodes are connected, the direction of the edges are assumed to be in both (i, j) and (j, i) as shown in Figure 2.24.

GNNs pipeline started from taking node with its features x_i and relationship between a pair $x_{(i,j)}$, adjacency matrix $N \epsilon\, 0, 1$ with size $|v| \times |v|$ which is the matrix that declares the relationship of each node in the graph. The adjacency matrix and features pass through small MLPs as in Equation 2.10.

$$h_{(i,j)} = f_{adj}\left(N, x_{(i,j)}\right) \tag{2.10}$$

Each node is an equal representation of MLPs. Later on, this feature will collect to the main node network as in Equation 2.11.

$$y = f_{node}\left(\sum h_{(i,j)},\ x_i,\ x_{(i,j)}\right) \tag{2.11}$$

Both functions f_{adj} and f_{node} are linked and exchange the latent features of the network

structure while improving the weight at each node.

From the base model of GNNs, graph functions can be utilized in many forms. Graph architectures have been developed by many techniques such as graph convolutional network (GCN) [26], graph attention network (GAT) [55], graph transformer network (GTN) [62].

The attention has recently attracted the interest of machine learning researchers working on sequential problems such as natural language processing (NLP) [5]). The highlight of interest is that it can extract relevant areas from input data. According to previous work, self-attention is a more beneficial feature than a single sequence-to-sequence or convolutions in comparison [29].

2.7 Literature Review

In this section, we briefly discuss some research on the TFS that exists at present. TFS recognition has undergone significant advancement, with recent research reflecting the integration of machine learning and sensor data for improved accuracy and speed. The scope of the research ranges from integrating technology in recognition systems to the sociolinguistic aspects of TFS.

One of the sensor data developments in TFS recognition is the incorporation of a sign language-to-alphabet spelling conversion system based on electromyography (EMG) signal recorded from the forearm muscles [4]. The system has two main functions: sign language feedback for guiding the correct gesture and sign language translation for detecting and interpreting the sign language and then converting it into sound or alphabets. The system is able to accurately match the EMG signal for each alphabet gesture with the actual spelling alphabet with a total accuracy of more than 95%. However, the limitations of this research are the need for proper electrode placement and calibration, as well as the need for further testing and validation of the system in real-world scenarios. Chansri et al [11] propose a Kinect sensor for hand detection and recognition that can provide accurate and reliable results for Thai sign language recognition. The proposed

technique uses Histograms of Oriented Gradients (HOG) for feature extraction and does not require a learning phase or training data. The system can provide an accuracy of 94.44%. Nevertheless, the limitations of the research are conducted with a limited number of participants and a small set of signs, which may affect the generalizability of the results. Furthermore, the proposed method may not work effectively in environments with complex backgrounds or poor lighting conditions.

Utilizing machine learning and deep learning, Nakjai and Katanyukul [34] introduced a novel two-stage pipeline for Thai finger spelling (TFS) recognition solely through images. The system combines color and contour-based hand identification with a two-fold approach to image classification, streamlining the TFS recognition process. To ensure accurate and reliable performance, the authors introduced a "confidence ratio" mechanism, targeting and removing invalid TFS signs. This strategic addition propelled their system to achieve a 91.26 mAP, surpassing existing state-of-the-art techniques in automatic visual TFS recognition. The paper also investigates the unique characteristics of TFS and the challenges related to TFS recognition and provides a practical design for TFS sign transcription. Pariwat and Seresangtakul [39] developed an advanced Thai finger-spelling sign language recognition system, integrating both global and local features with four Support Vector Machines (SVMs): linear, polynomial, Radial Basis Function (RBF), and sigmoid kernels. This work utilized a dataset of 375 images, comprising 15 Thai alphabet characters demonstrated by five signers and could achieve a 91.20% accuracy with the RBF kernel. Despite the impressive findings, the study's reliance on a modest dataset of 42 Thai finger-spelling characters and its neglect of sign language's temporal aspects could limit the system's practicality in real-world continuous signing scenarios. These considerations underscore the need for future developments and the incorporation of temporal dynamics to bolster comprehensive sign language recognition. Silanon [50] presented the development of an automatic classification system for recognizing 21 hand postures that represent letters in TFS. The system used the HOG feature and Adaptive Boost learning technique to construct a strong classifier that consists of several weak classifiers to be cascaded in detection architecture. The parameters for the training process

were adjusted in three experiments, false positive rates (FPR), true positive rates (TPR), and a number of training stages (N), to achieve the most suitable training model for each hand posture. The system achieved approximately 78% accuracy on average on all classifier experiments. Sanalohit and Katanyukul [46] explored the capabilities of MediaPipe Hands (MPH) in Thai Finger Spelling (TFS) sign recognition, particularly assessing its accuracy across different TFS schemes such as static-single-hand (S1), dynamic-single-hand (S2), and static-point-on-hand (P1). With an Artificial Neural Network (ANN) as the classifier, MPH demonstrated promising performance, achieving 82.12% accuracy for S1 and 84.57% for S2. Despite achieving promising results, the study admited to certain shortcomings, including a lack of comprehensive analysis on various factors like lighting conditions, camera angles, and hand sizes that could influence the effectiveness of MediaPipe Hands (MPH) in TFS sign recognition. The paper also highlighted potential benefits of investigating other hand-tracking and feature extraction methods in future research.

Pariwat and Seresangtakul [40] presented a deep learning-based recognition system for TFS in video format, enabling to cope with multi-stroke Thai finger-spelling for 42 characters of the Thai alphabet. The system presented impressive results, achieving an average accuracy of 88.00% for single-stroke signs, 85.42% for two-stroke signs, and 75.00% for three-stroke signs. Vijitkunsawat et al. [57] offered preliminary findings on only the Thai sign language digit dataset, including nine signs from 21 signers. They comprehensively analyzed four deep-learning architectures: CNN-Mode, CNN-LSTM, VGG-Mode, and VGG-LSTM. Their evaluation covered two scenarios: whole body poses against various backgrounds, and cropped hand images as a pre-processing method. The study revealed that VGG-LSTM is the most superior, particularly when combined with hand-cropping pre-processing, yielding 81.25% accuracy and an 85.21% F1-score across test datasets. Vijitkunsawat et al. [56] also presented a multimodal-based number finger-spelling recognizer for 24 primary numbers in TFS, achieving an accuracy of 95.0% for in-sample data and 84.1% for out-of-sample data.

Technological advancement has also led to the exploration of multimodal data in sign

Table 2.1: Comparing research of Thai Finger Spelling (TFS) dataset

Researchers	Signs	Signers	Dataset
Adhan and Pintavirooj [4]	42 (A)	unknown	1,050 samples
Saengri et al [45]	16 (A)	unknown	64 samples
Nakjai and Katanyukul [34]	25 (A)	11	1,375 images
Pariwat and Seresangtakul [39]	16 (A)	5	375 images
Chansri and Srinonchat [11]	16 (A)	unknown	320 images
Silanon [50]	16 (A)	unknown	2,100 images
Phothiwetchakun et al. [41]	25 (A)	unknown	15,000 images
Nakjai, Maneerat et al. [35]	25 (A)	12	1,500 images
Chaowanawatee et al. [12]	15 (A)	unknown	1,903 images
Sanalohit and Katanyukul [46]	64 (A,V, I)	unknown	4,319 images
Pariwat and Sereangtakul [40]	25 (A)	4	840 Videos
Vijitkunsawat et al. [57]	9 (N)	21	567 Videos
Vijitkunsawat et al. [56]	24 (N)	43	3,207 Videos
Our work	**90 (A, V, I, N)**	**43**	**10,467 Videos**

where: A = Alphabet, V = Vowel, I = Intonation mark, N = Number

language recognition. Some researchers utilize multimodal data (visual and sensor), processed via LSTM for temporal pattern learning. This LSTM output is then integrated with a CCHMM, enhancing the recognition process's robustness [60]. Zhang et al [64] introduced a multimodal approach to enhances recognition precision, achieving state-of-the-art performance on the datasets of CSL and IsoGD. Additionally, a novel sampling method called ARSS (Aligned Random Sampling in Segments) selected and aligned optimal RGB-D video frames to improve the utilization of multimodal data and reduce redundancy. They also proposed D-shift Net as depth motion feature extraction in the temporal stream using three-dimensional motion information of the sign language.

Bird et al [6] showed a late fusion approach to multimodality in sign language recognition, which improved the overall ability of the model in comparison with singular approaches of image classification and Leap Motion data classification. The approach was tested on a large synchronous dataset of 18 British Sign Language gestures collected from multiple subjects, and the best model achieved 94.44% accuracy. Papadimitriou and Potamianos [38] employed an end-to-end deep learning approach for sign language recognition, combining multiple spatio-temporal feature streams and a fully convolutional

attention-based encoder-decoder with temporal deformable convolutional block structures, and outperformed existing state-of-the-art methods on three sign language datasets (see Table 2.1).

Based on the above literature survey, we propose to extensively investigate a new multimodal method using three input modalities from diverse 29 deep learning architectures. Indeed, our study is based on seven single-based models, 14 combinations of two modalities and eight combinations of three modalities on seven different scenarios. Furthermore, we compare the performances of each model with in-sample and out-of-sample evaluations to find superior models in real world situations.

Chapter 3

Dataset and Methods

This section proposes a novel TFS dataset (Subsection 3.1) and our deep multimodal finger spelling recognizer. Our system uses three primary modalities: RGB-sequencing in the video modality, coordinate-sequencing of joints in the human body modality, and the graph structure of human joints modality.

To deal with video input modality, we introduce a pre-processor (Subsection 3.2) consisting of seven components: the YOLOv5 framework, padder, scaler, image over-sampler, image down-sampler, and background removal. This module is used by the RGB-sequencing modality (Subsection 3.3) for feeding all sequencing frames as inputs to the CNN-LSTM and VGG-LSTM models. The coordinate-sequencing modality (Subsection 3.4) involves modelling of a sequential arrangement of coordinates representing the skeletal structure within a human body obtained through the utilization of MediaPipe APIs. MediaPipe is used to extract the coordinates (x, y, z) of synchronized data from RGB-sequencing frames captured at the palm and arm. These extracted coordinates are then stored in a CSV file. Subsequently, the entire coordinated data is inputted into four separate deep learning models, namely LSTM, BiLSTM, GRU and Transformer, with the objective of evaluating and comparing the efficacy of each model. The modality of the human joint structure utilizes input data derived from the coordinate data of the skeletal system, which is stored in a CSV file. This data is then utilized to classify the quantity of nodes and edges, which are subsequently fed into the graph-structured modality

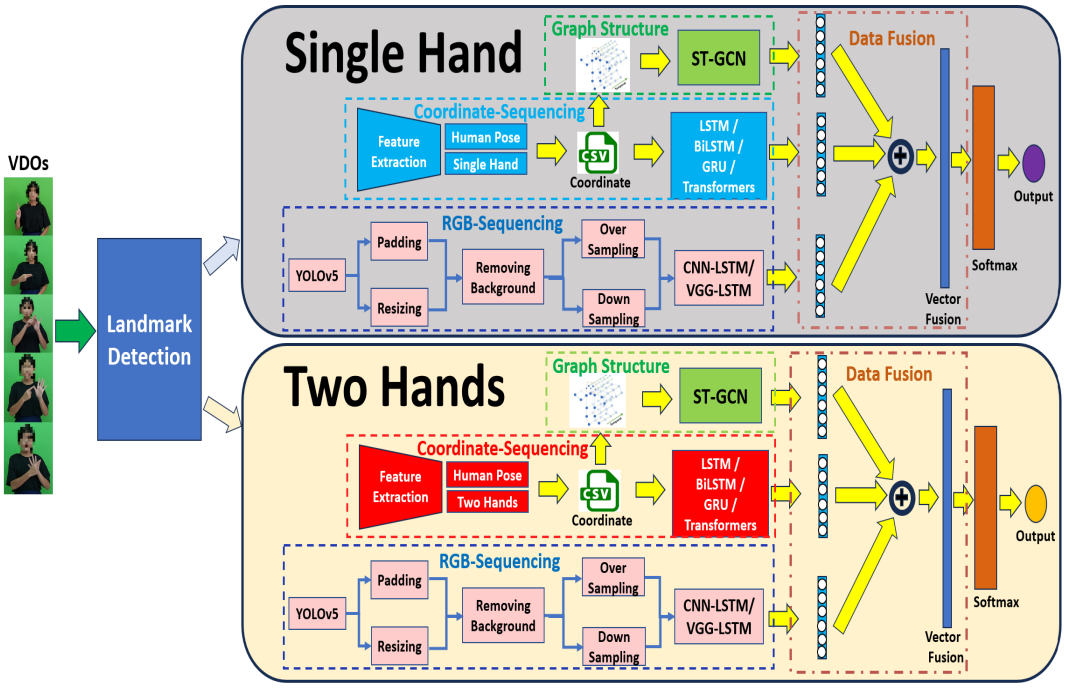


Figure 3.1: Overall architecture

(Subsection 3.5), which is the TGCN model.

3.1 Thai Finger Spelling Dataset

Our Thai Finger Spelling (TFS) dataset is acquired from two sources: online resources and real data recordings performed by signers of deaf schools under the control of professionals in deaf education. For the first source, we download online resources from reliable websites, namely the Royal Society[1], the National Association of the Deaf in Thailand[2] and deaf schools. For the second source, we directly record video files with about 95% of our total dataset from two Thai deaf schools: Setsatian School for the Deaf and Thungmahamek School for the Deaf. Then, the recorded video files are annotated data from annotators who are professional lecturers in a deaf school.

A total of 10,467 videos consists of 90 crucial letters, covering all TFS language and comprising four subcategories: 42 alphabets, 20 vowels, four intonation marks and 24 numbers (see Figure 2.6). The length of each video is in intervals of two to five seconds,

[1]http://164.115.33.116/vocab/index.html
[2]https://www.th-sl.com/search-by-act/

Figure 3.2: Example of Thai Number Finger Spelling datasets

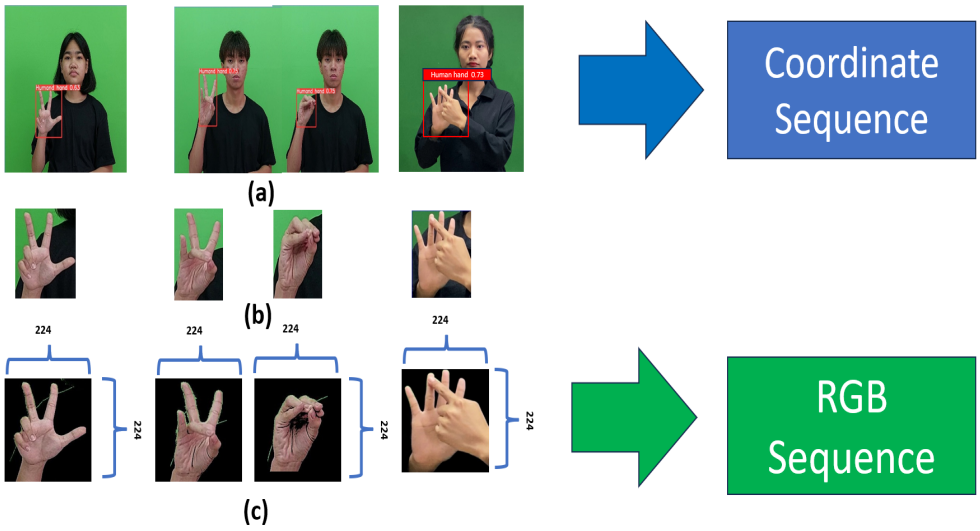


Figure 3.3: (a) whole body pose (b) cropped only hand (c) padding and remove background

depending on the number of strokes of hands. All videos are collected from 43 signers (29 females and 14 males with different ages, backgrounds, and appearances), as shown in Figure 3.2. The videos are categorized into two primary groups for evaluation: in-sample testing and out-of-sample testing. In-sample testing involves the evaluation of a model using the same dataset on which it is trained, whereas out-of-sample testing involves the evaluation of the model using previously unseen data. Consequently, the data is split into in-sample and out-of-sample testing, with a ratio of 15% and 10% correspondingly. After obtaining all the videos, we convert them into individual frames at a rate of 30 frames per second, following the National Television System Committee (NTSC) standard.

3.2 Pre-Processing module

3.2.1 Landmark Detection

After converting videos to frames consisting of the entire body of a signer, the frames are fed into landmark detection to classify the number of hands by using the MediaPipe library to extract the coordinates (x, y) of arm key points (points 13–16) at pose landmark (refer to Figure 3.6). The landmark detection is processed following Algorithm 1.

Algorithm 1 Landmark Detection

Require: $Frame > 0$
  while $Frame \neq 0$ **do**
    if $(Arm_{y=15} > Arm_{y=13})$ and $(Arm_{y=16} > Arm_{y=14})$ **then**
      Two-Hand Pose
    else if $(Arm_{y=15} > Arm_{y=13})$ and $(Arm_{y=16} < Arm_{y=14})$ **then**
      Single-Hand Pose with Left Hand
    else if $(Arm_{y=15} < Arm_{y=13})$ and $(Arm_{y=16} > Arm_{y=14})$ **then**
      Single-Hand Pose with Right Hand
    else
      No Poses
    end if
  end while

3.2.2 Hand Cropping

After converting videos to frames, the total frames are fed into two main processes. Firstly, the entire-body frames serve as input data for the coordinate sequence modality to investigate the joint structure of the hand and arm (see Figure 3.3(a)).

Secondly, hand-cropping frames, the technique uses YOLOv5 for precise cropping of video frames, focusing on the signer's hand to improve sign language recognition. This targeted approach, illustrated in the referenced Figure 3.3(b), is foundational in isolating the hand gestures from the broader video frame, enabling a focused analysis critical for accurate gesture interpretation. The efficacy of YOLOv5 in this context is significantly boosted through fine-tuning with an extensive dataset from the Google Open Images Dataset V6, meticulously annotated to include a diverse array of human hand images.

This dataset, featuring over 22,000 training images and more than 2,000 testing images, serves as a rich resource for training the model across 90 epochs. This comprehensive training regimen culminates in the model achieving a notable F1-score of 79.86%, underscoring the high precision and recall in hand gesture recognition within video sequences.

After recognizing hand gestures, images undergo padding and resizing for size uniformity. This crucial step ensures consistent processing and analysis, enhancing model accuracy by eliminating variations in image sizes.

3.2.3 Image Padding and Scaling

The technique is part of the preprocessing steps necessary for preparing image frames to be used with the VGG16 model, a popular convolutional neural network (CNN) architecture in computer vision. The VGG16 model requires input images to be of a specific size, 224×224 pixels, to maintain consistency and ensure optimal performance across varied inputs.

When an image frame is smaller than the required 224×224 pixels, it is necessary to enlarge the frame to the required size without distorting its content. This is achieved by adding padding around the edges of the image. The padding is typically applied using a black color to minimize distraction and ensure that the added borders do not interfere with the model's ability to learn from the actual content of the image. Additionally, the technique involves the elimination of the background to focus the model's attention on the primary subjects of the image, further enhancing the effectiveness of the learning process, as depicted in Figure 3.3(c).

Conversely, image frames that exceed the required dimensions are scaled down to 224×224 pixels. This resizing is done while preserving the aspect ratio as much as possible to avoid distortion of the image content. Proper resizing ensures that the images retain their original visual context, which is essential for the model to learn and make predictions based on the visual data accurately.

After the images are adjusted to the correct size, either through padding or resizing,

they are then processed further as part of the sampling process (described in Subsection 3.2.4). This additional step is designed to control the number of frames that are fed into the deep multimodal model.

3.2.4 Image Over-sampling and Down-sampling

When the video input has fewer than 30 frames, an over-sampling technique is applied. This method involves duplicating existing frames to increase the total frame count to 30. The duplication process typically involves copying frames from the immediate preceding frame, effectively padding the video sequence to reach the required length. This approach ensures that shorter videos are brought up to a minimal frame count without altering the original content's speed or timing, maintaining the integrity of the sequence's dynamics. Conversely, if a video input contains more than 30 frames, a down-sampling technique is employed. This method aims to reduce the total number of frames to 30 by randomly selecting frames within the video while preserving the order of the sequence of hand posture motions. The selection process is designed to maintain the narrative or instructional integrity of the video, ensuring that the resulting 30-frame sample represents the original motion sequence effectively.

3.3 RGB-Sequencing Module for Visual Modality

Convolutional Neural Networks (CNNs) in the 2D domain are frequently employed to extract spatial information from input images. On the other hand, Recurrent Neural Networks (RNNs) are designed to capture long-term temporal correlations that exist between input data. This module utilizes a 2D CNN-RNN architecture to capture spatial-temporal characteristics from the input video frames. The Long Short-Term Memory (LSTM) is a recurrent neural network (RNN) architecture that is specifically developed to address the problem of vanishing gradients issue in standard RNNs [20, 1, 3, 43]. Therefore, we apply the CNN and LSTM to be CNN-LSTM to extract spatial-temporal information of input images, as seen in Figure. 3.4

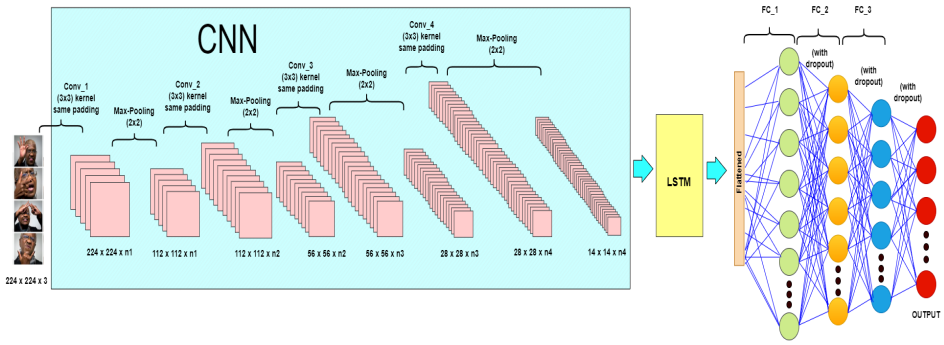


Figure 3.4: CNN-LSTM Model

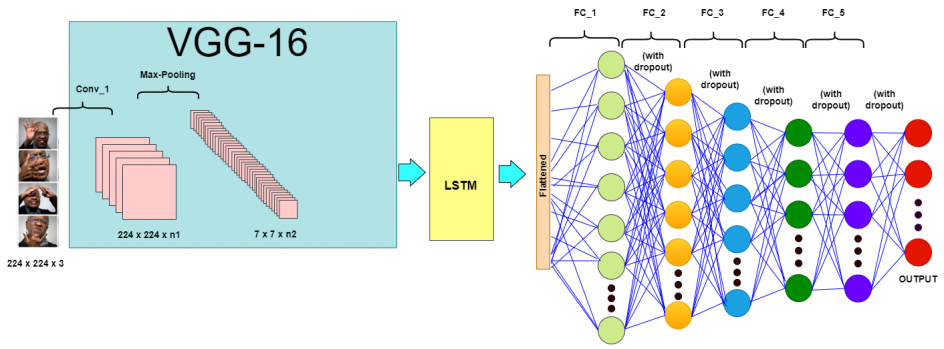
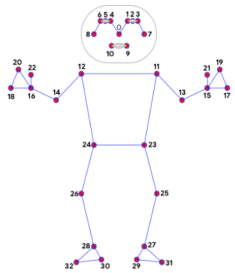


Figure 3.5: CNN-LSTM Model

The VGG16 model, which has been pre-trained on the ImageNet dataset [9], is employed to extract spatial characteristics that are subsequently inputted into a stacked LSTM, as seen in Figure. 3.5. To prevent overfitting in the training set, we set the sizes of hidden units in CNN kernel to be 32, 64, 128, and 256, respectively and the number of stacked recurrent in LSTM architecture is set to 2. In the training phase, we randomly select 30 sequence frames from each video, and the cross-entropy loss is then applied to the output.

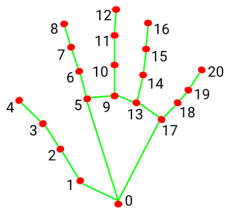
3.4 Coordinate-Sequencing for Human's Joints Modality

Human posture estimation aims to accurately identify the anatomical landmarks or joints of the human body within an image or video. We choose the MediaPipe library to extract the holistic key points because it enables to detect the whole key points of the body and hands [63], as shown in Figure 3.6. In our work, we divide two major categories: single-



Pose Landmarks (6)

0. Nose	7. Left ear	***14. Right elbow***	21. Left thumb	28. Right ankle
1. Left eye inner	8. Right ear	***15. Left wrist***	22. Right thumb	29. Left heel
2. Left eye	9. Mouth left	***16. Right wrist***	23. Left hip	30. Right heel
3. Left eye outer	10. Mouth right	17. Left pinky	24. Right hip	31. Left foot index
4. Right eye inner	***11. Left shoulder***	18. Right pinky	25. Left knee	32. Right foot index
5. Right eye	***12. Right shoulder***	19. Left index	26. Right knee	
6. Right eye outer	***13. Left shoulder***	20. Right index	27. Left ankle	



Hand Landmarks (21)

0. Wrist	***6. Index Finger PIP***	***12. Middle Finger TIP***	***18. Pinky PIP***
1. Thumb CMC	***7. Index Finger DIP***	***13. Ring Finger MCP***	***19. Pinky DIP***
2. Thumb MCP	***8. Index Finger TIP***	***14. Ring Finger PIP***	***20. Pinky TIP***
3. Thumb IP	***9. Middle finger MCP***	***15. Ring Finger DIP***	
4. Thumb TIP	***10. Middle Finger PIP***	***16. Ring Finger TIP***	
5. Index Finger MCP	***11. Middle Finger DIP***	***17. Pinky MCP***	

Figure 3.6: Pose and Hand Landmarks

hand and two-hand poses. In single-hand pose, we totally employ 24 key points from the pose landmarks with 3 key points (points 12, 14, 16 for right arm or 11, 13, 15 for left arm) and hand landmarks with 21 key points. For the two-hand pose, the key points are entirely used 48 landmarks from 6 key points on pose landmarks (points 11-16) and two-hand landmarks with 42 points, as shown in Figures 3.8 and 3.9. Then, the coordinates (x, y, z) of each location are acquired in order to be used as input data for the modalities of coordinates sequencing and human joint structure.

Pose-RNN-based techniques mostly employ to construct sequences of poses in order to evaluate human motion. Inspired by this model, RNN is employed to describe the sequential temporal information of pose movements in our initial posture-based baseline, and the representation created by RNN is used for sign identification. We feed the coordinates as input features into the deep-learning models. The LSTM, BiLSTM, and GRU models are constructed using hidden unit sizes of 128, 256, 256, and 256, which have been determined to be experimentally optimal, as illustrated in Figure 3.7(a)-(c).

The Pose-Transformer model employs self-attention to capture the relationship between sequence segments [54]. Transformer has achieved state-of-the-art results in several domains of artificial intelligence, including computer vision [25] and nature language processing(NLP) [15]. We set a multi-head attention network with embedding size 512, eight

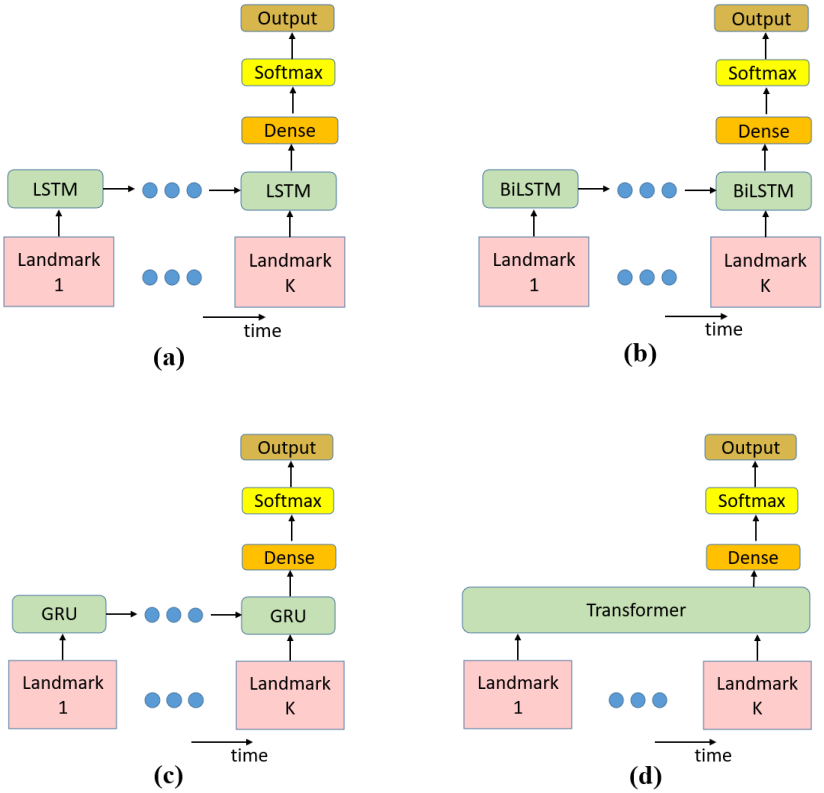


Figure 3.7: (a) LSTM (b) BiLSTM (c) GRU (d) Transformer

heads, and two transformer blocks. Also, we add the 0.1 dropout, layer normalization, and residual connections. Therefore, the final layer can be stacked multiple times, as illustrated in Figure 3.7(d).

3.5 Graph Representation for Joint's Structure Modality

Graph-structured data with spatial and temporal elements may be handled using the advanced deep learning architecture called the Temporal Graph Convolutional Network (TGCN). This neural network performs particularly well on tasks like action identification and posture estimation, where the input data frequently consists of graphs that represent sequences of human body joint locations (skeletons). The TGCN paradigm includes two essential elements: temporal convolutions to simulate temporal dynamics and graph

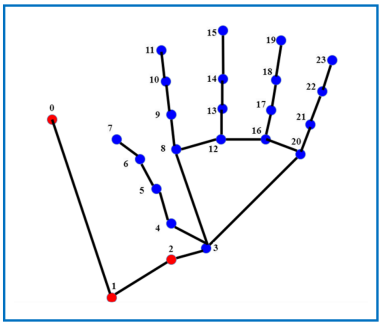


Figure 3.8: Single-Hand Coordination

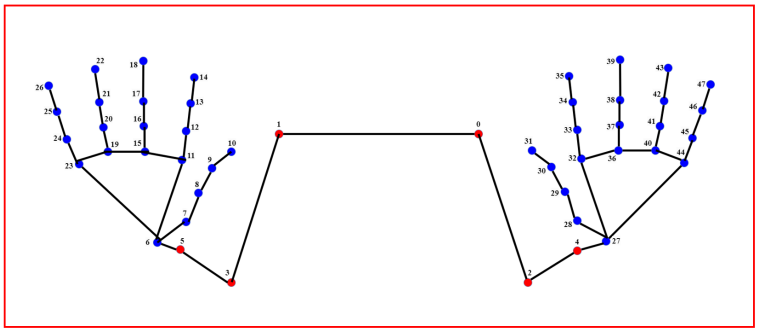


Figure 3.9: Two-Hand Coordination

convolutional networks (GCN) for processing spatial input. While temporal convolutions focus on capturing the temporal patterns throughout a sequence or video frames, the GCN component concentrates on collecting spatial characteristics by aggregating data from nearby nodes in the network. A deep graph convolutional network's nth graph layer is a function called ϑ_n that takes as input features a matrix called $H_n \in \mathbb{R}^{K \times F}$, where F is the feature dimension produced by the layer before it and K is the number of dimensions. The $K \times 2N$ matrix coordinates of body key points in the first layer are fed to the networks, where N is the number of successive frames. Using this formula with a set of trainable weights $W_n \in \mathbb{R}^{F \times F'}$ [61], the following is the expression for a graph convolutional layer:

$$H_{n+1} = \vartheta_n \left(H_n\right) = \sigma \left(A_n H_n W_n\right) \tag{3.1}$$

where A_n is a trainable adjacency matrix for n-th layer and (σ) denotes the activation function $\tanh(\cdot)$ as shown in Figure 3.10. Then, for classification, a softmax layer is used, followed by an average pooling layer.

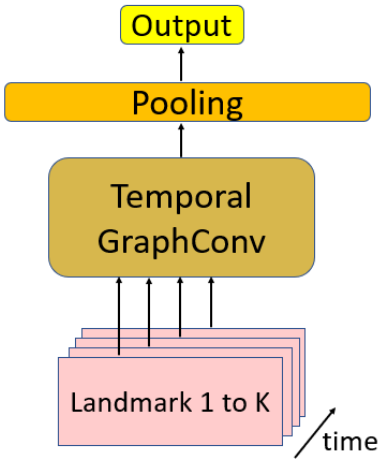


Figure 3.10: Temporal Graph Convolutional Network

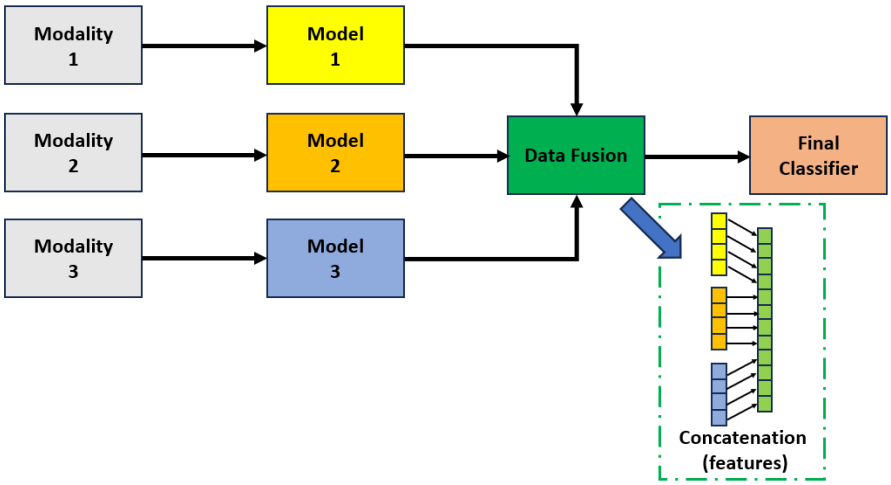


Figure 3.11: Data Fusion

3.6 Data Fusion

Data fusion is a distinctive component of multimodal learning, wherein integrating information from diverse data sources is necessary for building a joint model [53]. This process is employed right before the result classification.

After the data passes each modality, we apply the concatenation technique, which involves combining many features into a single vector. The input data can be raw features when using this technique, as shown in Figure 3.11.

Chapter 4

Experiments and Results

In all experiments, an Intel(R) Core i7 processor with a clock speed of 2.9 GHz and 128 GB of RAM is employed. The models are constructed using PyTorch version 2.0 for the TGCN model, while the RGB-sequencing models and coordinate-sequencing models are produced using TensorFlow version 2.8.0. In addition, the models undergo training across three NVIDIA RTX-3090 GPUs, each equipped with 24 GB of memory. This is achieved by the use of the mirrored technique, wherein all model parameters are replicated across the GPUs, ensuring that identical parameter updates are consistently applied across all devices. The first step in all of our research involves the configuration of each deep learning model. The dataset is partitioned into four distinct groups, namely the training set, validation set, in-sample test set, and out-of-sample test set. These groups are allocated proportions of 60%, 15%, 15%, and 10% of the total data, respectively.

To analyze the results for each scenario, we utilized performance metrics such as accuracy, precision, recall, and the F1 score. Accuracy refers to the proportion of data points that were correctly predicted out of the total data points. Accuracy simply evaluates how frequently the classifier predicts correctly. It is the number of correct predictions divided by the total number of predictions, shown in Equation 4.1. Precision is calculated as the proportion of accurately identified positive classes in relation to the total number of classes predicted as positive, detailed in Equation 4.2. Also, precision is a useful metric in cases where a false positive (FP) is a higher concern than false negatives (FN). Recall

is defined as the ratio of the total number of correctly classified positive instances to the total number of actual positive instances. In other words, it measures the proportion of actual positive classes that have been correctly predicted by the model, as seen in Equation 4.3. Recall is a useful metric in cases where false negative (FN) trumps false positive (FP). F1-Score is the harmonic mean of precision and recall, providing a balance between these two metrics, as shown in Equation 4.4. This score is particularly useful because it considers both false positives and false negatives. In scenarios where both these types of errors have significant consequences, or when there's an uneven class distribution, the F1-score provides a more insightful measure of a model's performance than accuracy alone.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{4.1}$$

$$Precision = \frac{TP}{TP + FP} \tag{4.2}$$

$$Recall = \frac{TP}{TP + FN} \tag{4.3}$$

$$F1 - score = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{4.4}$$

where TP is the number of true positives.

TN is the number of true negatives.

FP is the number of false positives.

FN is the number of false negatives

There are a total of seven experiments in our study. These experiments are categorized into two major experiments: single-hand poses and two-hand poses (see Table 4.1). Furthermore, KerasTuner is utilized to determine the optimal settings for our system in all experimental trials. We considered different combinations of parameter settings: learning rate ranging from 10^{-5} to 5×10^{-4} by in increments of 2×10^{-5}, dropout ranging from 0.1

to 0.5 in steps of 0.1, epoch ranging from 40 to 300 in increments of 20, and memory cell ranging from 30 to 70 in steps 10, optimizer in choice of "adam","rmsprop" and "sgd". Each split is measured using accuracy, Top-3, Top-5, confusion matrix with in-sample and out-of-sample testing on 29 models including single-based modality, the combination of two modalities and the combination of three modalities to obtain the best parameter setting (see Table 4.2).

Table 4.1: The categories of letters in each experiment.

No.	**Scenarios**	**Alphabets**	**Vowels**	**Into**	**Num**
1	Static single hand with single stroke	15	4	-	10
2	Dynamic single hand with two strokes	24	3	-	14
3	Dynamic single hand with three strokes	3	-	-	-
4	Total single-hand poses	42	7	-	24
5	Static point-on-hand with two hands	-	12	-	-
6	Dynamic point-on-hand with two hands	-	1	4	-
7	Total two-hand poses	-	13	4	-

where: Into = Intonation marks and Num = Numbers

Table 4.2: The selected parameters used by each implemented models

Modalities	Models	Parameters				
		Learning rate	Dropout	Optimizer	ES	MC
RGB-Sequencing Based	CNN-LSTM	$2e^{-5}$	0.2	Adam	10	50
	VGG-LSTM	$5e^{-5}$	0.2	Adam	10	50
Coordinate of Skeleton Based	LSTM	$5e^{-5}$	0.1	Adam	10	256
	BiLSTM	$5e^{-5}$	0.1	Adam	10	256
	GRU	$5e^{-5}$	0.1	Adam	10	256
	Transformer	$1e^{-4}$	0.1	Adam	-	-
Graph Based	TGCN	$2e^{-5}$	0.5	tanh	50	256

where: ES = Early stopping, MC = Memory cell

For ChatGPT4, which is currently one of the most popular large language models, we utilize both zero-shot for out-of-sample testing and few-shot prompting for in-sample testing [27] [13]. We input three random sequence images of each label into ChatGPT4 using each approach to evaluate the model's performance across 100 samples for both techniques. Specifically, in the few-shot technique, we enhance the prompt with a demonstration image to guide the model toward improved performance. The example of the prompting template for ChatGPT4 is shown in Figure 4.1, which is separated from the

main seven experiments. Also, the Insertion of images done by the human (it can be automatic by simple classification)

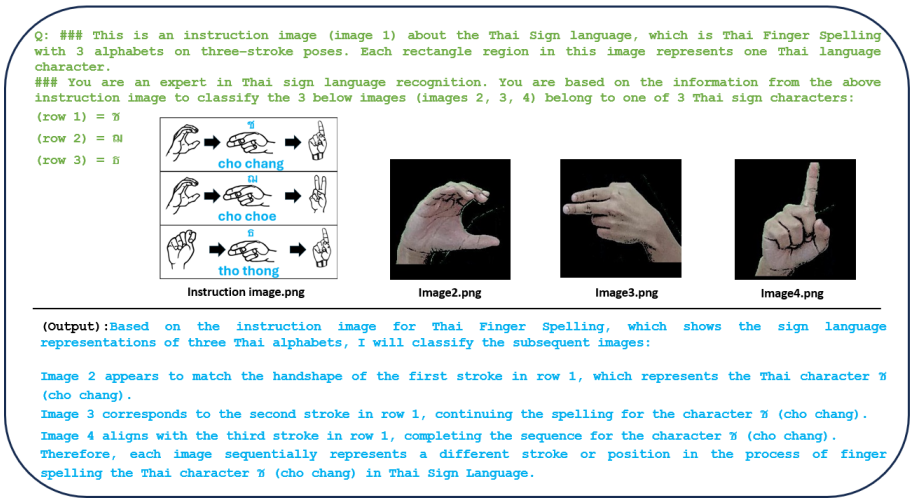


Figure 4.1: Example of prompting template

4.1 Single-Hand Poses

In the single-hand experiments, there are four sub-experiments within the single-hand group: single-stroke pose, two-stroke pose, three-stroke pose and total single-hand pose (see Table 4.1).

According to Table 4.3, it provides a comprehensive analysis of 29 notable models across three different modalities, focusing on a static single-hand pose with a single stroke and 29 letters. The table details total parameter usage and a variety of evaluation metrics such as Top-1 accuracy, Top-3, Top-5, and F1-score, all assessed both in-sample and out-of-sample testing.

Regarding total parameter usage, the TGCN model is considered to be the most effective lightweight model due to its ability to process graph-based data. This model provides a more concise representation for the coordinates of skeleton data compared to other models that rely on RGB-sequencing-based (CNN-LSTM and VGG-LSTM) or coordinates of skeleton-based (LSTM, BiLSTM, GRU, and Transformer) representations. The TGCN model effectively captures spatial relationships in graph-structured data through the uti-

lization of graph convolutional layers. Furthermore, the filters are localized, focussing on a limited neighborhood of nodes and edges, which may result in fewer parameters than RGB-sequencing models and more direct correlations than skeleton models' coordinates.

While the TGCN stands out as the premier lightweight model, it does not maintain this superiority in terms of in-sample evaluation performance. The integration of the coordinate-sequencing modality with the Transformer model and the graph structure modality with the TGCN results in the highest performance levels, achieving over 97% across various evaluation metrics. However, a notable drop is observed in the out-of-sample evaluation, where the accuracy (Top-1), Top-3, Top-5 and F1-score of this combined approach plummet to approximately 79.7%, 94.02%, 95.63% and 77.7% respectively, as shown in Table 4.4. This significant decline in performance is primarily attributable to the high frequency of misclassifications for specific letters, namely " ใ- ", " ด " and " ฟ ", which recorded accuracies of 0%, 13.33%, and 33.33% respectively (detailed in Table 4.5). Also, the column "Pron" in the table means the pronunciation of Thai letters to simplify the understanding.

Table 4.6 shows a performance benchmarking table of a static single-hand pose with single-stroke, demonstrating a comparison across various models using both in-sample and out-of-sample datasets. The models are evaluated based on appearance and pose-based representations, with metrics that include accuracy, precision, recall, and F1 score, essential for understanding model effectiveness and reliability. The appearance representation refers to how the model processes visual input data, while the pose-based representation involves the model's interpretation of the hand's position and movement. The table highlights the superior performance of the proprietary model denoted as the "Our (T+TG)" model represents an innovative combination of Transformer(T) and Temporal Graph Convolutional Networks(TGCN) models, leading to its distinguished performance in single-stroke hand pose estimation tasks. This model's design inherently captures the dynamic spatio-temporal relationships inherent in hand gestures by leveraging the Transformer's capacity for handling sequential data with long-range dependencies and the TGCN's efficacy in modeling time-structured graph data. The synthesis of these techniques is ev-

idenced by its exemplary in-sample accuracy and F1-score of 97.6%, indicating a high degree of model fidelity to the training data. Moreover, the model's out-of-sample accuracy of 79.8% and F1-score of 75.89% show superior generalization capabilities, which are essential for practical deployment. In part of the out-of-sample of ChatGPT4 (zero-shot), it cannot have the capability to interpret or generate pose estimations, making it incompatible with the specific tasks these metrics are intended to measure, or the model is not designed for this type of evaluation. As a result, these metrics are not applicable (N/A) to ChatGPT4, and no data is presented as shown in Figure 4.2.

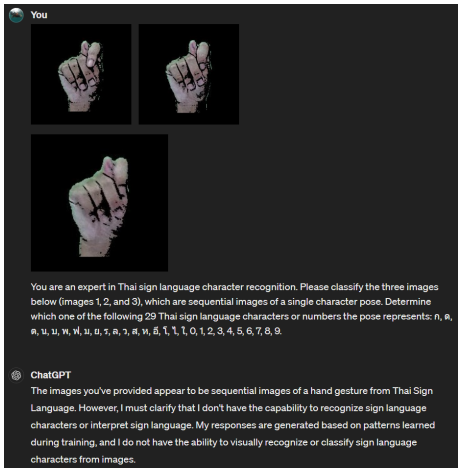


Figure 4.2: Example result of ChatGPT4 in out-of-sample testing

Table 4.7 and 4.8 show evaluation metrics of dynamic single-hand pose with two strokes on 41 letters. The graph structure of single modality in the TGCN model remains the one with the fewest parameters. However, its applicability in real-world scenarios is limited due to its inadequate performance metrics when applied to out-of-sample data. The performance of the combined Transformer and TGCN models on two modalities is demonstrated to be superior through assessments conducted on both in-sample and out-of-sample testing. The in-sample metrics demonstrate high levels of accuracy and F1-score classification, above 99%. However, the metrics evaluation for the two-stroke pose decreases to about 89.3% and 88.6% , respectively, when assessed out-of-sample.

Table 4.9 presents the accuracy results obtained from the integration of Transformer and TGCN models. The experimental findings indicate that the performance of the models is

Table 4.3: In-Sample evaluation metrics for static single-hand poses with single-stroke

No.	Modalities	RGB		Coordinate				Graph	Parameters (M)	In-Sample Evalutation(%)					
		C	V	L	Bi	G	T	TG		Top-3	Top-5	Accuracy	Precision	Recall	F1-Score
1	Single Modality	✓							1.512	96.21	98.62	90.69	91.89	89.81	90.84
2			✓						15.349	93.1	97.59	77.9	80.8	74.64	77.6
3				✓					5.423	97.24	98.28	96.6	97.3	95.91	96.6
4					✓				6.182	99.66	**100**	93.8	95.7	92.17	93.9
5						✓			2.008	97.2	99.1	89.3	97.5	96.11	96.8
6							✓		1.657	78.97	95.17	40.3	43.1	32.57	37.1
7								✓	**0.508**	98.28	100	86.9	89.6	83.98	86.7
8	2 Modalities	✓		✓					10.772	96.21	98.62	86.2	89.4	82.85	86
9		✓			✓				11.617	98.28	99.66	88.6	91.6	86.17	88.8
10		✓				✓			7.444	98.97	**100**	89.7	92.9	86.34	89.5
11		✓					✓		7.009	96.9	98.28	89.3	91.7	87.79	89.7
12		✓						✓	5.827	96.21	98.97	87.6	89.7	85.98	87.8
13			✓	✓					20.549	96.21	98.97	82.4	86.7	78.14	82.2
14			✓		✓				21.394	89.66	94.48	65.9	73	59.89	65.8
15			✓			✓			17.221	98.28	99.31	86.6	90	83.26	86.5
16			✓				✓		16.786	96.9	99.31	83.8	86.3	81.44	83.8
17			✓					✓	15.633	84.83	91.72	66.9	69.3	63.37	66.2
18				✓				✓	6.123	98.62	99.31	91	92.9	89.18	91
19					✓			✓	6.968	96.55	98.97	88.6	90.8	86.31	88.5
20						✓		✓	2.795	**100**	**100**	96.2	96.9	95.51	96.2
21							✓	✓	2.36	**100**	**100**	**97.6**	**98**	**97.2**	**97.6**
22	3 Modalities	✓		✓				✓	11.291	99.31	**100**	90.3	93.7	86.58	90
23		✓			✓			✓	12.137	97.59	99.66	85.52	87.59	83.55	85.52
24		✓				✓		✓	7.963	98.97	**100**	93.4	95.2	91.67	93.4
25		✓					✓	✓	7.528	97.24	98.62	93.1	94.1	91.73	92.9
26			✓	✓				✓	21.068	98.97	99.31	85.9	88.7	83.46	86
27			✓		✓			✓	21.914	87.93	92.07	69	74.1	59.99	66.3
28			✓			✓		✓	17.74	**100**	**100**	89.3	92.5	86.5	89.4
29			✓				✓	✓	17.305	83.1	89.66	57.6	60.7	52.84	56.5

where: C = CNN-LSTM, V = VGG-LSTM, L = LSTM, Bi = BiLSTM, G = GRU, T = Transformer, TG = TGCN

Table 4.4: Out-of-Sample Evaluation metrics for static single-hand poses with single-stroke

No.	Modalities	RGB		Coordinate				Graph	Out-of-Sample Evaluation(%)					
		C	V	L	Bi	G	T	TG	Top-3	Top-5	Accuracy	Precision	Recall	F1-Score
1	Single Modality	✓							42.07	53.56	22.53	34.65	17.27	23.05
2			✓						56.09	66.21	34.9	41.3	26.93	32.6
3				✓					24.37	41.38	9.9	2.4	4.8	3.2
4					✓				57.47	69.43	34	37.6	29.56	33.1
5						✓			80.92	86.9	64.6	68.2	58.54	63
6							✓		61.38	82.76	26.4	25.5	19.5	22.1
7								✓	45.52	62.53	16.3	18.1	10.89	13.6
8	2 Modalities	✓		✓					57.7	68.05	29.7	32.1	24.21	27.6
9		✓			✓				86.9	92.64	64.4	64.6	58.14	61.2
10		✓				✓			81.61	90.8	48.3	54.7	42.29	47.7
11		✓					✓		41.61	53.1	20.2	27.1	15.23	19.5
12		✓						✓	37.93	46.44	20	24	15.72	19
13			✓	✓					81.61	89.43	50.3	53	46.08	49.3
14			✓		✓				57.93	66.44	36.6	35.8	31.48	33.5
15			✓			✓			90.11	94.48	66.9	66.1	62.79	64.4
16			✓				✓		66.44	77.24	36.8	44.3	27.98	34.3
17			✓					✓	41.61	54.48	20.2	24.3	15.73	19.1
18				✓				✓	76.32	88.28	54	56.9	49.25	52.8
19					✓			✓	68.97	79.31	48.3	53.1	41.52	46.6
20						✓		✓	91.49	94.94	72.4	70.9	68.15	69.5
21							✓	✓	**94.02**	**95.63**	**79.8**	**79.6**	**75.89**	**77.7**
22	3 Modalities	✓		✓				✓	69.43	77.93	48	50.2	42.79	46.2
23		✓			✓			✓	60.46	69.2	36.78	35.41	31.84	33.53
24		✓				✓		✓	79.77	87.36	58.6	58.1	54.8	56.4
25		✓					✓	✓	42.3	51.72	24.4	30.3	18.53	23
26			✓	✓				✓	76.09	86.9	45.5	47	38.62	42.4
27			✓		✓			✓	54.71	65.29	31.5	38.7	24.9	30.3
28			✓			✓		✓	88.74	94.48	58.4	61.5	50.57	55.5
29			✓				✓	✓	45.29	59.08	24.6	25.6	20.88	23

where: C = CNN-LSTM, V = VGG-LSTM, L = LSTM, Bi = BiLSTM, G = GRU, T = Transformer, TG = TGCN

Table 4.5: Accuracy for static single-hand poses with single-stroke by each Thai letter

No.	Thai Letters	Pron	In-Sample Test			Out-of-Sample Test		
			Correct	Incorrect	Acc(%)	Correct	Incorrect	Acc(%)
1	0	soon	22	0	100	13	2	86.67
2	1	neung	22	0	100	11	4	73.33
3	2	song	18	0	100	15	0	100
4	3	sam	20	0	100	15	0	100
5	4	se	27	0	100	12	3	80
6	5	haa	29	0	100	15	0	100
7	6	hok	24	0	100	14	1	93.33
8	7	jet	20	0	100	15	0	100
9	8	prad	22	0	100	11	4	73.33
10	9	gao	13	0	100	10	5	66.67
11	ก	ko kai	18	0	100	15	0	100
12	ด	do dek	24	4	85.71	2	13	13.33
13	ต	to tao	27	4	87.09	15	0	100
14	น	no nu	20	0	100	15	0	100
15	บ	bo baimai	16	0	100	15	0	100
16	พ	pho phan	16	0	100	7	8	46.67
17	ฟ	fo fan	16	0	100	5	10	33.33
18	ม	mo ma	22	2	91.66	15	0	100
19	ย	yo yak	36	0	100	14	1	93.33
20	ร	ro ruea	16	0	100	15	0	100
21	ล	lo ling	18	0	100	15	0	100
22	ว	wo waen	24	0	100	11	4	73.33
23	ส	so suea	20	0	100	7	8	46.67
24	ห	ho hip	20	0	100	15	0	100
25	อ	o ang	10	0	100	9	6	60
26	ี-	sara ee	29	0	100	15	0	100
27	โ-	sara o	31	2	88.57	0	15	0
28	ใ-	maimuan	24	0	100	11	4	73.33
29	ไ-	maimarai	27	0	100	15	0	100

Table 4.6: In-Sample and Out-of-Sample performance benchmark for static single-hand poses with single-stroke

No.	Model	App	Pose	In-Sample				Out-of-Sample			
				Acc	Pre	Rec	F1	Acc	Pre	Rec	F1
1	I3D [10]	✓		80.34	88.14	71.88	79.18	23.9	31.98	15.63	21
2	Fusion-3 [21]	✓		93.45	96	92.02	93.97	27.36	38.11	20.57	26.72
3	MEMP [66]	✓		92.41	93.2	91.63	92.41	17.01	28.09	14.64	19.25
4	DeepSign-CNN [49]	✓		91.37	93.02	89.99	91.48	18.85	34.58	12.02	17.84
5	Pose-GRU [28]		✓	95.86	97	94.88	95.93	71.49	71.49	66.86	69.1
6	Pose-TGCN [28]		✓	90.69	91.92	89.24	90.56	16.55	22.48	10.82	14.61
7	SPOTER [8]		✓	47.24	51.13	38	43.6	25.51	26.77	19.35	22.46
8	Bi-RNN [23]		✓	91.72	94.24	89.35	91.73	59.08	61.7	51.72	56.27
9	FNN-LSTM [23]		✓	87.24	89.24	85.73	87.45	44.37	49.58	39.08	43.71
10	ChatGPT4	✓		59	62.86	58.91	56.09	N/A	N/A	N/A	N/A
11	**Our(T+TG)**		✓	**97.6**	**98**	**97.2**	**97.6**	**79.8**	**79.6**	**75.89**	**77.7**

where: App = Appearance Representation, Pose = Pose-based Representation, Acc = Accuracy, Pre = Precision, Rec = Recall and F1 = F1-Score

influenced by the variables represented by the letters "ิ" and " ณ " during the evaluation using unseen data. Specifically, the accuracy rates achieved by "ิ" and " ณ " are 33.3% and 46.67%, respectively.

Table 4.10 presents the performance benchmark for dynamic single-hand pose with two-stroke. The Pose-GRU model achieves unparalleled perfection in in-sample testing, with accuracy, precision, recall, and F1 all reaching the maximum of 100%. This shows an exceptional fit to the training data, indicative of the model's capacity to capture the fine-grained temporal dependencies and complex patterns within the in-sample dataset. However, the integration of the Transformer and Temporal Graph Convolutional model, demonstrates a significant out-of-sample performance edge with an accuracy of 89.3% and an F1-score of 88.6%, outstripping all competitors due to its adept sequential data processing, attention mechanisms, and robust capture of temporal pose dynamics.

The in-sample evaluation of the dynamic single hand with three-stroke postures comprising of 3 letters: " ช ", " ฌ ", " ธ " is shown below Table 4.11. The LSTM and BiLSTM models, which operate on a single modality, demonstrate superior performance across all evaluation measures. However, it is worth noting that the TGCN model remains the least parameter-intensive, with a parameter count of around 500K. In contrast, the LSTM and BiLSTM perform less than combining two modalities, namely Transformer and TGCN.

Table 4.7: In-Sample Evaluation metrics for dynamic single-hand pose with two-stroke

No.	Modalities	RGB		Coordinate				Graph	Parameters (M)	In-Sample Evalutation(%)					
		C	V	L	Bi	G	T	TG		Top-3	Top-5	Accuracy	Precision	Recall	F1-Score
1	Single Modality	✓							4,782,137	93.17	96.1	83.41	85.35	81.25	83.25
2			✓						15,354,217	91.22	95.37	65.4	69.5	61.22	65.1
3				✓					3,422,969	97.8	99.02	92.2	92.9	91.51	92.2
4					✓				6,860,137	93.66	97.32	75.6	78.6	71.53	74.9
5						✓			2,411,097	95.37	97.32	77.3	78.5	73.47	75.9
6							✓		1,005,201	99.76	**100**	95.9	96.3	95.5	95.9
7								✓	**508,489**	94.88	97.32	85.1	86.7	83.17	84.9
8	2 Modalities	✓		✓					21,796,185	94.88	97.32	82.4	85.9	78.81	82.2
9		✓			✓				25,125,913	83.17	89.51	61.5	64	57	60.3
10		✓				✓			20,881,241	88.05	95.12	65.6	68.1	62.35	65.1
11		✓					✓		19,416,618	89.27	94.88	72.4	74.9	68.95	71.8
12		✓						✓	18,882,529	96.59	98.05	89.5	92.1	87.42	89.7
13			✓	✓					18,521,193	92.44	95.85	73.7	77.9	68.33	72.8
14			✓		✓				21,850,921	76.1	84.39	46.8	50.3	40.87	45.1
15			✓			✓			17,606,249	85.61	91.71	58.5	60.1	55.3	57.6
16			✓				✓		16,141,626	94.88	97.8	78	81.1	74.57	77.7
17			✓					✓	15,635,185	70.73	81.46	43.4	45.9	40.98	43.3
18				✓				✓	4,095,537	97.07	98.05	85.6	88.2	82.21	85.1
19					✓			✓	7,425,265	93.9	96.83	74.4	78.4	70.25	74.1
20						✓		✓	3,180,593	99.27	99.76	94.4	95.1	93.51	94.3
21							✓	✓	1,715,970	**100**	**100**	**99.8**	**99.8**	**99.8**	**99.8**
22	3 Modalities	✓		✓				✓	22,315,937	95.61	98.29	76.8	81.3	72.95	76.9
23		✓			✓			✓	25,645,665	92.2	94.88	71.5	74.7	66.75	70.5
24		✓				✓		✓	21,400,993	94.88	98.54	80.49	84.52	76.59	80.36
25		✓					✓	✓	19,936,370	96.1	97.56	85.6	87.6	83.69	85.6
26			✓	✓				✓	19,040,945	91.22	93.9	67.1	71.7	62.7	66.9
27			✓		✓			✓	22,370,673	77.56	85.61	51.5	54.9	47.27	50.8
28			✓			✓		✓	18,126,001	85.85	93.17	61.7	65.7	54.54	59.6
29			✓				✓	✓	16,661,378	78.54	88.05	46.1	48.9	43.07	45.8

where: C = CNN-LSTM, V = VGG-LSTM, L = LSTM, Bi = BiLSTM, G = GRU, T = Transformer, TG = TGCN

Table 4.8: Out-of-Sample Evaluation metrics for dynamic single-hand pose with two-stroke

No.	Modalities	RGB		Coordinate				Graph	Out-of-Sample Evaluation(%)					
		C	V	L	Bi	G	T	TG	Top-3	Top-5	Accuracy	Precision	Recall	F1-Score
1	Single Modality	✓							39.51	46.5	22.6	25.01	16.61	19.96
2			✓						43.09	53.5	23.1	20.3	17.51	18.8
3				✓					80.98	88.29	55.3	59.6	49.53	54.1
4					✓				84.39	92.85	55.4	60.5	48.6	53.9
5						✓			66.18	78.37	38.5	40.1	33.33	36.4
6							✓		89.11	91.87	77.9	81.4	72.87	76.9
7								✓	65.53	75.45	38.4	37.6	31.87	34.5
8	2 Modalities	✓		✓					74.31	82.44	53.7	59.8	47.43	52.9
9		✓			✓				49.59	61.14	25.4	27.6	22.52	24.8
10		✓				✓			56.26	66.83	33.2	35.6	27.61	31.1
11		✓					✓		36.26	50.57	19.5	20.6	14.18	16.8
12		✓						✓	35.61	48.13	17.4	27.5	11.68	16.4
13			✓	✓					71.38	79.67	47.2	51	40.26	45
14			✓		✓				44.88	57.07	22.3	27.7	17.17	21.2
15			✓			✓			56.42	68.94	31.7	29.2	28.61	28.9
16			✓				✓		68.46	80.16	40.2	51.5	30.23	38.1
17			✓					✓	33.66	44.72	15.3	17.2	11.52	13.8
18				✓				✓	69.59	78.05	43.3	47.5	37.64	42
19					✓			✓	47.48	60.65	30.2	29.5	26.11	27.7
20						✓		✓	89.11	92.68	71.9	76.9	64.74	70.3
21							✓	✓	**97.72**	**99.35**	**89.3**	**90.8**	**86.5**	**88.6**
22	3 Modalities	✓		✓				✓	54.96	65.37	32.8	38.4	27.58	32.1
23		✓			✓			✓	43.74	54.63	24.2	22.9	19.91	21.3
24		✓				✓		✓	60.16	74.15	33.82	35.5	29.21	32.05
25		✓					✓	✓	38.05	50.24	19	24.1	13.01	16.9
26			✓	✓				✓	57.24	68.94	32.5	37.9	25.66	30.6
27			✓		✓			✓	55.61	64.72	29.4	32.4	24.5	27.9
28			✓			✓		✓	63.74	75.77	37.4	38.9	30.83	34.4
29			✓				✓	✓	38.37	52.2	17.4	15.4	13.17	14.2

where: C = CNN-LSTM, V = VGG-LSTM, L = LSTM, Bi = BiLSTM, G = GRU, T = Transformer, TG = TGCN

Table 4.9: Accuracy for dynamic single-hand pose with two-stroke by each Thai letter

No.	Thai Letters	Pron	In-Sample Test			Out-of-Sample Test		
			Correct	Incorrect	Acc(%)	Correct	Incorrect	Acc(%)
1	10	sip	36	0	100	14	1	93.33
2	20	yee sip	27	2	93.1	11	4	73.33
3	30	sam sip	27	0	100	15	0	100
4	40	se sip	27	0	100	15	0	100
5	50	haa sip	24	0	100	15	0	100
6	60	hok sip	18	0	100	15	0	100
7	70	jet sip	31	0	100	9	6	60
8	80	pead sip	22	0	100	8	7	53.33
9	90	gao sip	11	0	100	15	0	100
10	100	neung roi	18	0	100	15	0	100
11	1,000	neung pan	18	0	100	15	0	100
12	10,000	neung muen	29	0	100	15	0	100
13	100,000	neung saen	22	0	100	11	4	73.33
14	1,000,000	neung laan	18	0	100	15	0	100
15	ข	kho khai	20	0	100	15	0	100
16	ค	kno khwai	22	0	100	15	0	100
17	ฆ	kho ra-khang	29	0	100	15	0	100
18	ง	ngo ngu	20	0	100	15	0	100
19	จ	cho chan	24	0	100	14	1	93.33
20	ฉ	cho ching	16	0	100	15	0	100
21	ซ	so so	22	0	100	15	0	100
22	ญ	yo ying	16	0	100	15	0	100
23	ฎ	do cha-da	20	0	100	8	7	53.33
24	ฏ	to pa-tak	22	0	100	15	0	100
25	ฐ	tho than	13	0	100	15	0	100
26	ฑ	tho montho	22	0	100	15	0	100
27	ฒ	tho phu-thao	16	0	100	15	0	100
28	ณ	no nen	24	0	100	7	8	46.67
29	ถ	tho thung	18	0	100	15	0	100
30	ท	tho thahan	31	0	100	10	5	66.67
31	ป	po pla	22	0	100	14	1	93.33
32	ผ	pho phueng	29	0	100	15	0	100
33	ฝ	fo fa	33	0	100	15	0	100
34	ภ	pho sam-phao	24	0	100	15	0	100
35	ศ	so sala	20	0	100	12	3	80
36	ษ	so rue-si	27	0	100	11	4	73.33
37	ฬ	lo chu-la	20	0	100	15	0	100
38	ฮ	ho nok-huk	18	0	100	15	0	100
39	ิ	sara i	16	0	100	5	10	33.33
40	ุ	sara u	24	0	100	15	0	100
41	ึ	sara ue	14	0	100	10	5	66.67

Table 4.10: In-Sample and Out-of-Sample performance benchmark for dynamic single-hand pose with two-stroke

No.	Model	App	Pose	In-Sample				Out-of-Sample			
				Acc	Pre	Rec	F1	Acc	Pre	Rec	F1
1	I3D [10]	✓		48.78	58.5	37.93	46.02	15.61	4.97	11.33	6.91
2	Fusion-3 [21]	✓		85.37	86.88	83.58	85.2	40.81	63.72	31.97	42.58
3	MEMP [66]	✓		96.34	96.52	96.14	96.33	33.82	46.26	27.34	34.37
4	DeepSign-CNN [49]	✓		97.8	98.08	97.5	97.79	31.7	40.08	22.98	29.21
5	Pose-GRU [28]		✓	**100**	**100**	**100**	**100**	82.68	83.87	81.5	82.67
6	Pose-TGCN [28]		✓	80.24	82.61	77.42	79.93	37.23	38.01	30.19	33.65
7	SPOTER [8]		✓	69.02	72.03	65.26	68.48	48.61	49.94	42.95	46.18
8	Bi-RNN [23]		✓	98.78	98.93	98.61	98.77	86.18	88.8	81.82	85.17
9	FNN-LSTM [23]		✓	80.24	83.03	77.19	80	38.86	41.12	31.7	35.8
10	ChatGPT4	✓		68	80.32	68.69	69.56	N/A	N/A	N/A	N/A
11	**Our(T+TG)**		✓	99.8	99.8	99.8	99.8	**89.3**	**90.8**	**86.5**	**88.6**

where: App = Appearance Representation, Pose = Pose-based Representation, Acc = Accuracy, Pre = Precision, Rec = Recall and F1 = F1-Score

These models achieve the greatest score of around 90.8% in terms of F1-score for out-of-sample evaluation, as shown in Table 4.12. The " ช " is the only letter for incorrect accuracy both in-sample and out-of-sample test in Table 4.13. When we dive into the full-fledged confusion matrix, we first look at the Figure 4.3 that " ช " is twice incorrect prediction to be " ฌ " and " ธ ". Also, it is wrong predicted to " ฌ " about third times and once at " ธ " of out-of-sample, as shown in Figure 4.4.

The SPOTER model does a great job of three classes for dynamic single-hand pose with three-stroke, as shown in Table 4.14. It gets perfect scores in accuracy, precision, recall, and F1 metrics, both in-sample and out-of-sample. This shows that SPOTER's advanced algorithm handles complex class differences well and stays useful with new data. This shows that it better understands how hand gestures change over time than other complex models, like those that combine Transformers and TGCN.

Table 4.15 and 4.16 presents the assessment metrics for the whole single-hand stance, consisting of 73 distinct letters. The utilization of a combination of GRU and Transformer models on two modalities has demonstrated superior performance in terms of achieving the greatest score during in-sample testing at 98.6% of F1-score and accuracy. However, when considering out-of-sample testing, the combination of Transformer and TGCN models exhibits the highest level of performance at 69.6% of accuracy and 66.8% of F1-score.

Table 4.11: In-Sample Evaluation metrics for dynamic single-hand pose with three-stroke

No.	Modalities	RGB-Sequencing		Coordinate-Sequencing				Graph	Parameters (M)	In-Sample Evalutation(%)			
		C	V	L	Bi	G	T	TG		Accuracy	Precision	Recall	F1-Score
1	Single Modality	✓							1.503	86.7	88.4	85.64	87
2			✓						15.471	66.7	67.3	63.79	65.5
3				✓					5.341	**100**	**100**	**100**	**100**
4					✓				3.714	**100**	**100**	**100**	**100**
5						✓			0.816	93.3	95	92.05	93.5
6							✓		12.026	90	91.9	87.6	89.7
7								✓	**0.507**	93.3	93.3	93.3	93.3
8	2 Modalities	✓		✓					61.893	76.7	84.7	68.27	75.6
9		✓			✓				9.036	83.3	86.5	81.08	83.7
10		✓				✓			6.175	90	90.6	89.61	90.1
11		✓					✓		68.368	90	91.2	89.03	90.1
12		✓						✓	56.860	70	68.6	68.6	68.6
13			✓	✓					20.663	76.7	86.7	67.97	76.2
14			✓		✓				18.813	73.3	88.6	62.36	73.2
15			✓			✓			15.952	76.7	89.2	65.45	75.5
16			✓				✓		27.138	70	72.7	65.66	69
17			✓					✓	15.630	60	55.1	55.3	55.2
18				✓				✓	6.238	86.7	86.9	85.51	86.2
19					✓			✓	4.202	83.3	83.8	81.24	82.5
20						✓		✓	1.526	86.7	89.1	82.55	85.7
21							✓	✓	12.712	93.3	93.9	92.31	93.1
22	3 Modalities	✓		✓				✓	62.413	83.3	87.2	76.85	81.7
23		✓			✓			✓	60.562	90	92.1	87.42	89.7
24		✓				✓		✓	57.702	76.7	76.9	76.5	76.7
25		✓					✓	✓	68.887	73.3	73.4	72.6	73
26			✓	✓				✓	21.183	66.7	65.6	65.2	65.4
27			✓		✓			✓	19.332	83.3	83.8	81.24	82.5
28			✓			✓		✓	16.472	63.3	76.2	41.46	53.7
29			✓				✓	✓	27.657	53.3	62.4	44.13	51.7

where: C = CNN-LSTM, V = VGG-LSTM, L = LSTM, Bi = BiLSTM, G = GRU, T = Transformer, TG = TGCN

Table 4.12: Out-of-Sample Evaluation metrics for dynamic single-hand pose with three-stroke

No.	Modalities	RGB-Sequencing		Coordinate-Sequencing				Graph	Out-of-Sample Evaluation(%)			
		C	V	L	Bi	G	T	TG	Accuracy	Precision	Recall	F1-Score
1	Single Modality	✓							60	48.5	52.67	50.5
2			✓						37.8	52.8	17.25	26
3				✓					71.1	81.5	62.9	71
4					✓				82.2	84.9	79.48	82.1
5						✓			80	83.8	74.36	78.8
6							✓		86.7	87.7	84.75	86.2
7								✓	71.1	72.9	67.69	70.2
8	2 Modalities	✓		✓					40	38.8	36.85	37.8
9		✓			✓				60	48.5	52.67	50.5
10		✓				✓			55.6	47.6	45.83	46.7
11		✓					✓		88.89	91.7	86.83	89.2
12		✓						✓	64.4	50	59.65	54.4
13			✓	✓					44.4	44.4	42.64	43.5
14			✓		✓				55.6	54.6	50.74	52.6
15			✓			✓			73.3	73.5	72.9	73.2
16			✓				✓		57.8	70.6	46.13	55.8
17			✓					✓	40	37.2	22.97	28.4
18				✓				✓	60	61.3	58.75	60
19					✓			✓	53.3	43.6	51.45	47.2
20						✓		✓	66.7	65.2	57.13	60.9
21							✓	✓	**91.1**	**92.4**	**89.25**	**90.8**
22	3 Modalities	✓		✓				✓	55.6	58	55.08	56.5
23		✓			✓			✓	55.6	58.4	54.91	56.6
24		✓				✓		✓	60	59.5	59.1	59.3
25		✓					✓	✓	75.5	85.9	63.77	73.2
26			✓	✓				✓	46.7	36.1	45.61	40.3
27			✓		✓			✓	53.3	54.5	50.64	52.5
28			✓			✓		✓	64.4	49.4	60.77	54.5
29			✓				✓	✓	33.3	11.1	33.7	16.7

where: C = CNN-LSTM, V = VGG-LSTM, L = LSTM, Bi = BiLSTM, G = GRU, T = Transformer, TG = TGCN

Table 4.13: In-sample and Out-of-sample Evaluation for dynamic single-hand pose with three-stroke

No.	Thai Letters	Pron	In-Sample Test			Out-of-Sample Test		
			Correct	Incorrect	Acc(%)	Correct	Incorrect	Acc(%)
1	ช	cho chang	18	4	81.81	11	4	73.33
2	ฌ	cho choe	20	0	100	15	0	100
3	ธ	tho thong	24	0	100	15	0	100

Table 4.14: In-Sample and Out-of-Sample performance benchmark for dynamic single-hand pose with three-stroke

No.	**Model**	**App**	**Pose**	**In-Sample**				**Out-of-Sample**			
				Acc	**Pre**	**Rec**	**F1**	**Acc**	**Pre**	**Rec**	**F1**
1	I3D [10]	✓		90	91.9	87.68	89.74	55.55	54.82	87.68	55.15
2	Fusion-3 [21]	✓		83.33	83.89	81.39	82.62	51.11	61.17	40.88	49.01
3	MEMP [66]	✓		56.67	55.14	56.4	55.76	37.78	44.92	33.99	38.7
4	DeepSign-CNN [49]	✓		96.66	97.14	96.28	96.71	57.77	48.03	49.2	48.61
5	Pose-GRU [28]		✓	90	91.9	87.68	89.74	55.55	54.82	87.68	55.15
6	Pose-TGCN [28]		✓	96.66	96.94	96.36	96.65	73.33	77.5	69.5	73.28
7	SPOTER [8]		✓	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**
8	Bi-RNN [23]		✓	93.33	94.05	92.35	93.19	84.44	86.9	81.97	84.36
9	FNN-LSTM [23]		✓	76.67	78.52	75.72	77.1	40	43.86	36.28	39.71
10	ChatGPT4	✓		98	98	98	98	N/A	N/A	N/A	N/A
11	**Our (T+TG)**		✓	93.3	93.9	92.31	93.1	91.1	92.4	89.25	90.8

where: App = Appearance Representation, Pose = Pose-based Representation, Acc = Accuracy, Pre = Precision, Rec = Recall and F1 = F1-Score

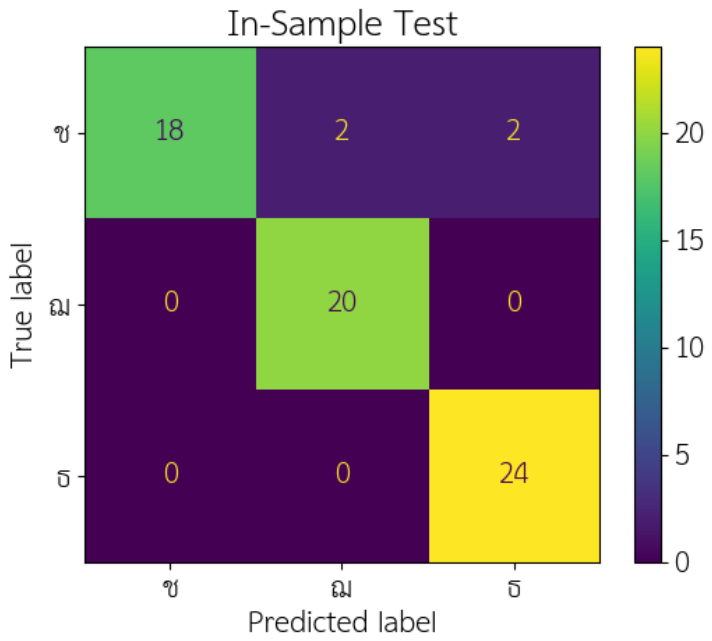


Figure 4.3: Confusion matrix of dynamic single-hand pose with three-stroke on in-sample test

Table 4.17-4.18 compare the in-sample and out-of-sample accuracy. Almost the accuracy of the in-sample test is higher than the out-of-sample test.

The table 4.19 illustrates the performance of various models on a comprehensive single-hand pose classification task encompassing 73 distinct classes. The Pose-GRU model stands out with exceptionally high in-sample accuracy, precision, recall, and F1 score at 98.9%, 99.14%, 98.72%, and 98.93% respectively, and it maintains these great out-of-sample scores of 85.38% in accuracy and 84.37% in F1-Score. The Pose-GRU model demonstrates high proficiency in recognizing various hand poses and maintains strong performance on unseen data, indicating its capacity for learning generalizable features.

Table 4.15: In-Sample evaluation metrics for total single-hand poses

No.	Modalities	RGB		Coordinate				Graph	Parameters (M)	In-Sample Evalutation(%)					
		C	V	L	Bi	G	T	TG		Top-3	Top-5	Accuracy	Precision	Recall	F1-Score
1	Single Modality	✓							1.528	90.41	95.62	75.5	78.6	73.19	75.8
2			✓						15.365	91.23	95.75	68.8	70.9	65.88	68.3
3				✓					3.354	99.59	**99.86**	94.4	95.6	93.03	94.3
4					✓				3.713	98.36	99.59	95.1	95.8	94.41	95.1
5						✓			1.463	97.81	99.32	90.7	92.6	87.35	89.9
6							✓		1.007	97.81	99.45	76.2	77.6	72.76	75.1
7								✓	**0.509**	97.12	98.22	89.7	90.8	88.63	89.7
8	2 Modalities	✓		✓					8.680	96.44	97.95	85.2	88.5	82.32	85.3
9		✓			✓				8.953	88.08	92.74	69.3	73.8	63.91	68.5
10		✓				✓			6.584	94.52	97.95	75.2	79	71.57	75.1
11		✓					✓		6.387	91.1	94.79	75.2	79.2	72.5	75.7
12		✓						✓	5.843	90.27	93.42	75.9	80	71.3	75.4
13			✓	✓					18.457	98.49	99.18	86.4	88.9	84.61	86.7
14			✓		✓				18.730	65.62	75.34	41.6	45.9	37.37	41.2
15			✓			✓			16.361	90	95.21	67.7	73.6	62.16	67.4
16			✓				✓		16.164	96.58	98.9	84.2	86.5	81.83	84.1
17			✓					✓	15.639	66.71	79.59	40.4	44.6	36.26	40
18				✓				✓	2.671	97.12	97.95	85.8	87.2	84.44	85.8
19					✓			✓	4.304	95.21	96.85	84.9	87.2	82.34	84.7
20						✓		✓	1.935	**99.73**	**99.86**	**98.6**	**98.7**	**98.5**	**98.6**
21							✓	✓	1.738	98.36	98.9	91.4	92.6	90.04	91.3
22	3 Modalities	✓		✓				✓	9.2	95.62	98.22	80.6	82.49	77.6	79.97
23		✓			✓			✓	9.473	91.51	95.62	69	74.51	64.27	69.01
24		✓				✓		✓	7.103	98.22	99.32	87	88.99	85	86.95
25		✓					✓	✓	6.907	93.84	95.62	78.5	81.7	75.91	78.7
26			✓	✓				✓	18.977	87.95	93.56	67.3	71.5	63.21	67.1
27			✓		✓			✓	19.250	70.14	79.32	45.8	50.8	39.91	44.7
28			✓			✓		✓	16.730	94.11	96.44	73.3	75.85	69.67	72.63
29			✓				✓	✓	16.683	97.4	99.04	84.4	85.7	82.56	84.1

where: C = CNN-LSTM, V = VGG-LSTM, L = LSTM, Bi = BiLSTM, G = GRU, T = Transformer, TG = TGCN

Table 4.16: Out-of-Sample evaluation metrics for total single-hand poses

No.	Modalities	RGB		Coordinate				Graph	Out-of-Sample Evaluation(%)					
		C	V	L	Bi	G	T	TG	Top-3	Top-5	Accuracy	Precision	Recall	F1-Score
1	Single Modality	✓							30.41	40.37	15.1	20.1	10.86	14.1
2			✓						48.95	60.91	26.5	32.3	20.93	25.4
3				✓					82.1	90.41	53.7	53.1	48.88	50.9
4					✓				78.72	87.12	48.5	49.6	44.48	46.9
5						✓			76.71	84.75	52.3	58.2	46.67	51.8
6							✓		**86.3**	89.59	66.3	64.9	62.54	63.7
7								✓	27.67	36.35	12.5	15.2	7.45	10
8	2 Modalities	✓		✓					71.69	79.18	49.2	55.5	42.13	47.9
9		✓			✓				55.53	65.75	32.3	33.8	28.97	31.2
10		✓				✓			58.81	71.23	33.8	41.2	27.94	33.3
11		✓					✓		27.67	35.71	14.5	20.6	10.26	13.7
12		✓						✓	29.22	37.81	16	18.3	12.42	14.8
13			✓	✓					74.52	83.2	51.1	55.2	44.38	49.2
14			✓		✓				32.33	41.83	13.9	15.8	11.78	13.5
15			✓			✓			63.29	76.16	32.2	34	27.98	30.7
16			✓				✓		72.05	81.83	44.7	49.1	36.85	42.1
17			✓					✓	25.84	36.99	12.1	12.3	9.16	10.5
18				✓				✓	49.77	60.27	28.3	29.4	24.79	26.9
19					✓			✓	48.58	57.35	28.1	29.6	24.65	26.9
20						✓		✓	84.47	89.59	67.3	**72.5**	61.25	66.4
21							✓	✓	85.48	**91.05**	**69.6**	68.6	**65.09**	**66.8**
22	3 Modalities	✓		✓				✓	48.68	59	25.2	30.62	19.68	23.96
23		✓			✓			✓	33.15	43.11	17.9	17.56	13.67	15.37
24		✓				✓		✓	54.16	64.75	30.2	32.76	24.22	27.85
25		✓					✓	✓	40	51.32	21.2	29.2	14.75	19.6
26			✓	✓				✓	48.13	58.54	26.2	27.9	22.16	24.7
27			✓		✓			✓	32.05	41.55	15.3	15.5	12.6	13.9
28			✓			✓		✓	64.47	74.98	35.5	36.06	31.17	33.44
29			✓				✓	✓	69.22	78.54	38.7	43.7	29.47	35.2

where: C = CNN-LSTM, V = VGG-LSTM, L = LSTM, Bi = BiLSTM, G = GRU, T = Transformer, TG = TGCN

Table 4.17: Accuracy for total single-hand poses by each Thai letter

No.	Thai Letters	Pron	In-Sample Test			Out-of-Sample Test		
			Correct	Incorrect	Acc(%)	Correct	Incorrect	Acc(%)
1	0	soon	27	4	85.71	15	0	100
2	1	neung	16	0	100	10	5	66.67
3	2	song	22	0	100	10	5	66.67
4	3	sam	27	0	100	15	0	100
5	4	se	20	0	100	6	9	40
6	5	haa	11	13	45.83	10	5	66.67
7	6	hok	16	0	100	10	5	66.67
8	7	jet	11	11	50	11	4	73.33
9	8	pead	29	0	100	10	5	66.67
10	9	gao	24	0	100	7	8	46.67
11	10	sip	24	0	100	10	5	66.67
12	20	yee sip	27	0	100	9	6	60
13	30	sam sip	24	0	100	12	3	80
14	40	se sip	16	0	100	15	0	100
15	50	haa sip	13	16	44.82	8	7	53.33
16	60	hok sip	27	0	100	14	1	93.33
17	70	jet sip	22	0	100	15	0	100
18	80	pead sip	24	4	85.71	14	1	93.33
19	90	gao sip	20	2	90.9	10	5	66.67
20	100	neung roi	31	9	77.5	14	1	93.33
21	1,000	neung pan	18	0	100	8	7	53.33
22	10,000	neung muen	25	0	100	15	0	100
23	100,000	neung saen	27	0	100	8	7	53.33
24	1,000,000	neung laan	25	0	100	15	0	100
25	ก	ko kai	22	0	100	11	4	73.33
26	ข	kho khai	18	11	62.06	11	4	73.33
27	ค	kno khwai	16	2	88.9	6	9	40
28	ฆ	kho ra-khang	13	2	86.6	15	0	100
29	ง	ngo ngu	24	0	100	6	9	40
30	จ	cho chan	16	2	88.9	15	0	100
31	ฉ	cho ching	20	0	100	14	1	93.33
32	ช	cho chang	20	2	90	14	1	93.33
33	ซ	so so	27	0	100	12	3	80
34	ฌ	cho choe	24	9	72.72	15	0	100
35	ญ	yo ying	22	0	100	15	0	100
36	ฎ	do cha-da	13	4	76.74	5	10	33.33
37	ฏ	to pa-tak	24	2	92.3	14	1	93.33
38	ฐ	tho than	20	0	100	15	0	100
39	ฑ	tho montho	20	0	100	15	0	100
40	ฒ	tho phu-thao	18	0	100	15	0	100
41	ณ	no nen	18	0	100	9	6	60
42	ด	do dek	18	0	100	8	7	53.33
43	ต	to tao	20	0	100	15	0	100
44	ถ	tho thung	16	0	100	15	0	100

Table 4.18: Accuracy for total single-hand poses by each Thai letter (Continue.)

No.	Thai Letters	Pron	In-Sample Test			Out-of-Sample Test		
			Correct	Incorrect	Acc(%)	Correct	Incorrect	Acc(%)
45	ท	tho thahan	22	0	100	10	5	66.67
46	ธ	tho thong	18	9	66.67	15	0	100
47	น	no nu	27	0	100	7	8	46.67
48	บ	bo baimai	13	0	100	14	1	93.33
49	ป	po pla	18	2	90	13	2	86.67
50	ผ	pho phueng	16	2	88.8	15	0	100
51	ฝ	fo fa	18	7	72	15	0	100
52	พ	pho phan	20	2	90.9	5	10	33.33
53	ฟ	fo fan	27	0	100	5	10	33.33
54	ภ	pho sam-phao	27	2	93.1	15	0	100
55	ม	mo ma	11	4	73.3	8	7	53.33
56	ย	yo yak	16	0	100	14	1	93.33
57	ร	ro ruea	18	0	100	14	1	93.33
58	ล	lo ling	29	2	93.54	14	1	93.33
59	ว	wo waen	13	0	100	10	5	66.67
60	ศ	so sala	24	2	92.3	15	0	100
61	ษ	so rue-si	13	0	100	15	0	100
62	ส	so suea	17	0	100	15	0	100
63	ห	ho hip	22	0	100	15	0	100
64	ฬ	lo chu-la	16	9	63	14	1	93.33
65	อ	o ang	22	0	100	14	1	93.33
70	ฮ	ho nok-huk	27	9	75	14	1	93.33
66	ิ-	sara i	13	0	100	15	0	100
67	ี-	sara ee	13	0	100	15	0	100
68	ึ-	sara u	11	4	71.43	3	12	20
69	ื-	sara ue	18	2	88.89	15	0	100
71	โ-	sara o	31	2	93.93	15	0	100
72	ใ-	maimuan	11	0	100	9	6	60
73	ไ-	maimarai	18	9	66.67	3	12	20

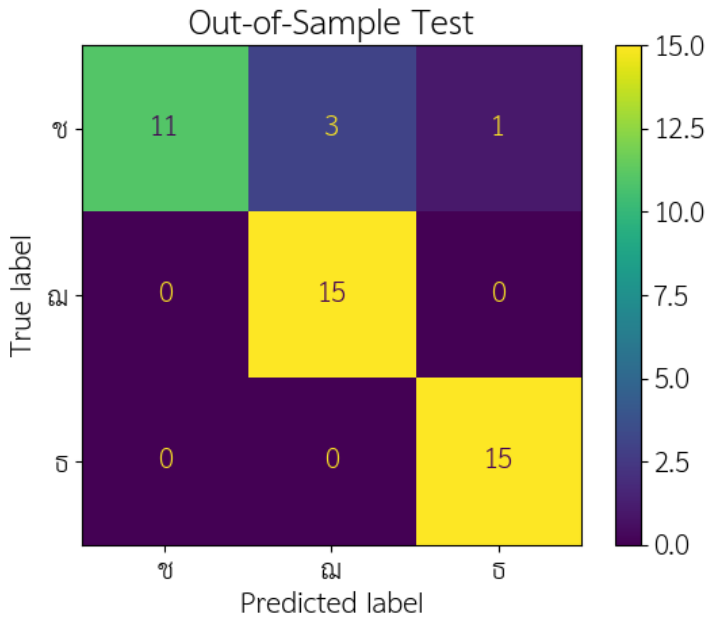


Figure 4.4: Confusion matrix of dynamic single-hand pose with three-stroke on out-of-sample test

4.2 Two-Hand Poses

The two-hand experiments are divided into three sub-experiments consisting of static-point-on-hand posesd, dynamic-point-on-hand poses, and total two-hand poses, (see Table 4.1).

Table 4.20 shows the in-sample evaluation metrics for static point-on-hand poses with two hands. There are twelve important letters in our experiment. Due to the graph-structured data, the TGCN of a single modality is the lightweight model in the total parameters. The result of the evaluation metrics, the GRU of a single modality is about 94.2% accuracy and 94.3% F1-score, but the combination of three modalities, which are CNN-LSTM of RGB sequences modality, BiLSTM of coordinate sequencing modality and TGCN of graph structure modality, is the best in the domain of out-of-sample evaluation both accuracy at nearly 41.7% and F1-score at 41.7%, as shown in Table 4.21. The primary factor contributing to the suboptimal performance of out-of-sample evaluation is the inaccurate classification of several letters. Table 4.22 shows the accuracy rates obtained for the characters " ั ", " ฤ ", and " ุ " are 0%, 13.33%, and 33.33%, respectively. The confusion matrices for both the in-sample and out-of-sample tests are depicted in

Table 4.19: In-Sample and Out-of-Sample performance benchmark for total single-hand poses

No.	Model	App	Pose	In-Sample				Out-of-Sample			
				Acc	**Pre**	**Rec**	**F1**	**Acc**	**Pre**	**Rec**	**F1**
1	I3D [10]	✓		23.83	26.15	13.05	17.41	6.75	4.96	2.69	3.49
2	Fusion-3 [21]	✓		87.67	89.3	86.19	87.72	35.43	50.68	28.91	36.82
3	MEMP [66]	✓		89.73	91.36	88.36	89.83	14.89	23.95	12.25	16.21
4	DeepSign-CNN [49]	✓		93.56	94.49	92.73	93.6	21.09	28.04	15.08	19.61
5	Pose-GRU [28]		✓	**98.9**	**99.14**	**98.72**	**98.93**	**85.38**	**87.36**	**81.58**	**84.37**
6	Pose-TGCN [28]		✓	80.54	82.66	78.66	80.61	22.64	21.62	17.98	19.63
7	SPOTER [8]		✓	58.08	60.03	52.14	55.81	26.02	27.7	20.5	23.56
8	Bi-RNN [23]		✓	97.4	97.84	97.05	97.44	74.16	75.53	69.66	72.48
9	FNN-LSTM [23]		✓	83.01	86.72	79.24	82.81	35.62	39.18	29.95	33.95
10	ChatGPT4	✓		59	62.86	58.9	56.09	N/A	N/A	N/A	N/A
11	**Our (T+TG)**		✓	91.4	92.6	90.04	91.3	69.6	68.6	65.09	66.8

where: App = Appearance Representation, Pose = Pose-based Representation, Acc = Accuracy, Pre = Precision, Rec = Recall and F1 = F1-Score

Figure 4.5 and Figure 4.6, respectively. The confusion matrix of out-of-sample shows that the letter " ั " is a completely wrong classification because the model classifies to be " ั " and " -า ". If we carefully consider in Figure 2.6, we will find that the posture of " ั ", " ั " and " -า " are similar poses.

Table 4.23 displays the efficacy of various models on a 12-class static point-on-hand poses with two hands, with the Multiple Extraction and Multiple Prediction model (MEMP) achieving the highest in-sample scores, suggesting its architecture is highly effective in capturing and learning from the provided data. With in-sample precision, recall, and F1 all at 95%, it indicates MEMP's superior ability to learn discriminative features and correctly label them within the training set. On the other hand, the "Our (C+Bi+TG)" model, integrating CNN-LSTM, Bi-LSTM, and Temporal Graph Convolution Networks, which use input data both appearance and pose-based representation, shows remarkable out-of-sample performance, boasting the highest accuracy and F1-Score at 41.7%. By integrating these models, the model benefits from the complementary strengths of each. CNN-LSTM and pose-based models offer rich, detailed feature extraction from both the visual and structural perspectives. Bi-LSTMs contribute a robust mechanism for understanding temporal sequences and predicting future states based on both past and anticipated information. Finally, TGCNs add a layer of sophistication by

modeling complex interdependencies over time, offering insights into the structural and temporal evolution of the data. This holistic approach ensures that every aspect of the data is thoroughly analyzed, leading to higher accuracy and F1-scores.

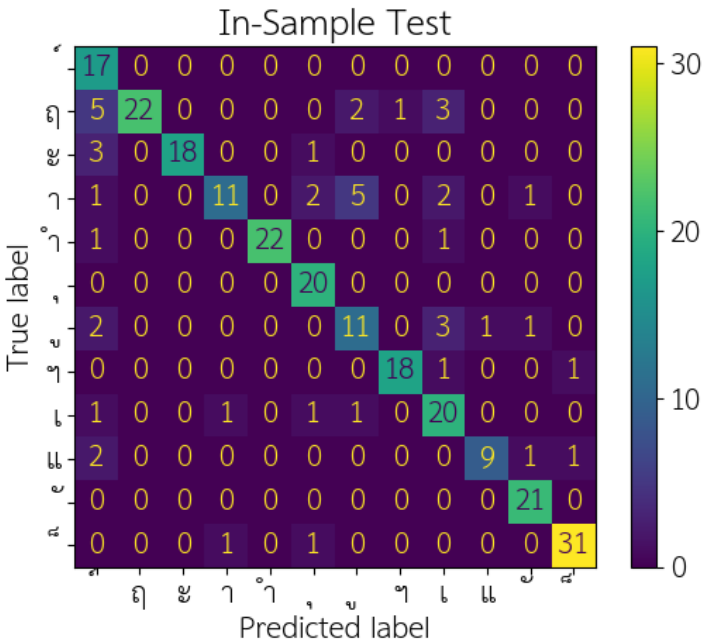


Figure 4.5: Confusion matrix of the static-point-on-hand poses with two hands on in-sample test

Table 4.24 presents the evaluation measure for the dynamic point-on-hand with two-hand posture at in-sample evaluation, which includes five letters in our experiment. The Long Short-Term Memory (LSTM) model, when applied to a single modality, as well as its combination with the Convolutional Neural Network-LSTM (CNN-LSTM), Transformer, and Spatio-Temporal Graph Convolutional Network (TGCN) models across three modalities, have achieved the highest scores across all metrics, reaching a perfect accuracy of 100%. As previously discussed, the integration of three modalities yields optimal results for selecting an appropriate model in out-of-sample scenarios. This is due to the influence of the blocked hand on the input data of each modality. For example, the RGB-sequencing modality is unable to capture significant features of sequence frames. Similarly, the MediaPipe API of coordinate-sequencing modality fails to extract the sequence coordinates (x, y, z) of the arm and hands due to obstruction caused by a blocked hand. Undoubtedly, the out-of-sample metrics exhibit lesser values compared to the in-sample test. Specifically,

Table 4.20: In-Sample Evaluation metrics for static-point-on-hand poses with two hands

No.	Modalities	RGB		Coordinate				Graph	Parameters (M)	In-Sample Evalutation(%)					
		C	V	L	Bi	G	T	TG		Top-3	Top-5	Accuracy	Precision	Recall	F1-Score
1	Single Modality	✓							4.779	95	96.67	85.8	87.5	84.55	86
2			✓						15.343	88.33	95.83	64.2	64.6	61.86	63.2
3				✓					4.630	**99.17**	99.17	80	84	76.73	80.2
4					✓				2.819	95	97.5	67.5	72.7	63.87	68
5						✓			2.745	**99.17**	**100**	**94.2**	**95.1**	**93.51**	**94.3**
6							✓		2.781	84.17	95	46.7	48.2	36.28	41.4
7								✓	**0.535**	97.5	98.33	79.2	79.7	78.31	79
8	2 Modalities	✓		✓					23.897	95	97.5	67.5	73.2	62.8	67.6
9		✓			✓				21.919	91.67	99.17	62.5	65.8	57.73	61.5
10		✓				✓			21.961	97.5	99.17	74.2	74.3	72.13	73.2
11		✓					✓		22.083	46.67	82.5	11.7	11.6	11.8	11.7
12		✓						✓	19.855	96.67	99.17	85.8	86.8	85.21	86
13			✓	✓					19.714	90	95.83	66.7	70.8	62.69	66.5
14			✓		✓				17.736	74.17	86.67	55.8	57	50.4	53.5
15			✓			✓			17.777	93.33	99.17	69.2	72.5	64.74	68.4
16			✓				✓		17.899	79.17	91.67	51.7	53.3	50.19	51.7
17			✓					✓	15.671	80.83	91.67	48.3	54.3	41.9	47.3
18				✓				✓	5.328	90.83	99.17	68.3	71.6	64.02	67.6
19					✓			✓	3.350	91.67	97.5	61.7	60.1	57.75	58.9
20						✓		✓	3.391	97.5	99.17	81.7	82.2	79.83	81
21							✓	✓	3.513	94.17	95.83	65	69.3	59.97	64.3
22	3 Modalities	✓		✓				✓	24.457	95.83	99.17	78.3	80.8	76.33	78.5
23		✓			✓			✓	22.479	97.5	99.17	80.8	81.7	79.72	80.7
24		✓				✓		✓	22.520	97.5	99.17	85	85.8	83.43	84.6
25		✓					✓	✓	22.642	**99.17**	**100**	88.3	90.2	86.29	88.2
26			✓	✓				✓	20.273	88.33	98.33	61.7	60.8	55.26	57.9
27			✓		✓			✓	18.295	83.33	92.5	50.8	54.6	45.95	49.9
28			✓			✓		✓	18.336	94.17	98.33	70.8	71.6	64.02	67.6
29			✓				✓	✓	18.458	82.5	91.67	50.8	54.5	47.22	50.6

where: C = CNN-LSTM, V = VGG-LSTM, L = LSTM, Bi = BiLSTM, G = GRU, T = Transformer, TG = TGCN

Table 4.21: Out-of-Sample Evaluation metrics for static-point-on-hand poses with two hands

No.	Modalities	RGB		Coordinate				Graph	Out-of-Sample Evaluation(%)					
		C	V	L	Bi	G	T	TG	Top-3	Top-5	Accuracy	Precision	Recall	F1-Score
1	Single Modality	✓							**61.11**	72.78	32.2	33.6	28.95	31.1
2			✓						59.44	**78.89**	23.9	20.5	19.91	20.2
3				✓					54.44	68.89	32.2	32.1	28.69	30.3
4					✓				49.44	64.44	16.7	19.4	15.45	17.2
5						✓			63.33	76.11	23.9	27.6	21.23	24
6							✓		42.2	58.89	13.3	8.1	11.49	9.5
7								✓	38.33	62.78	12.8	8.4	7.64	8
8	2 Modalities	✓		✓					52.78	73.89	27.2	35.9	24.32	29
9		✓			✓				53.89	72.22	22.8	27.5	15.59	19.9
10		✓				✓			53.33	70.56	33.3	36.5	27.4	31.3
11		✓					✓		43.89	65	12.8	6.1	7.43	6.7
12		✓						✓	57.78	68.89	36.7	46.2	30.58	36.8
13			✓	✓					53.89	70	26.1	25.4	25.2	25.3
14			✓		✓				48.33	60	21.1	17.1	19.22	18.1
15			✓			✓			62.78	73.33	26.1	25.9	23.61	24.7
16			✓				✓		51.67	70.56	23.9	21.3	22.96	22.1
17			✓					✓	47.78	69.44	20.6	28.2	14.33	19
18				✓				✓	49.44	71.67	25	28.6	22.2	25
19					✓			✓	50	62.78	27.2	34.4	23.47	27.9
20						✓		✓	53.89	71.11	24.4	29.7	20.85	24.5
21							✓	✓	47.78	67.78	18.9	23.2	14.57	17.9
22	3 Modalities	✓		✓				✓	59.44	75.56	32.2	34.6	26.32	29.9
23		✓			✓			✓	**61.11**	71.67	**41.7**	**48.8**	**36.4**	**41.7**
24		✓				✓		✓	57.78	67.78	33.3	37.3	28.48	32.3
25		✓					✓	✓	60.56	70	38.9	50.6	32.39	39.5
26			✓	✓				✓	47.78	68.89	23.3	18.4	22.39	20.2
27			✓		✓			✓	44.44	63.33	20	31.8	15.79	21.1
28			✓			✓		✓	54.44	67.78	27.8	27.6	22.35	24.7
29			✓				✓	✓	46.11	62.78	18.9	21.7	15.38	18

where: C = CNN-LSTM, V = VGG-LSTM, L = LSTM, Bi = BiLSTM, G = GRU, T = Transformer, TG = TGCN

Table 4.22: Accuracy for static-point-on-hand poses with two hands by each Thai letter

No.	Thai Letters	Pron	In-Sample Test			Out-of-Sample Test		
			Correct	Incorrect	Acc(%)	Correct	Incorrect	Acc(%)
1	◌์	ka-run	17	0	100	11	4	73.33
2	ฤ	ro-ruk	22	11	66.67	2	13	13.33
3	-ะ	sara a	18	4	81.81	10	5	66.67
4	-า	sara ar	11	11	50	6	9	40
5	-ำ	sara um	22	2	90.91	10	5	66.67
6	-ู	sara oo	20	0	100	5	10	33.33
7	-ุ	sara au	11	7	61.1	6	9	40
8	ไ	pai yan noi	18	2	90	4	11	26.67
9	เ-	sara ae	20	4	83.33	7	8	46.67
10	แ-	sara aae	9	4	69.23	6	9	40
11	-ิ	mai hen ar-karn	21	0	100	0	15	0
12	-ิ	mai tai-ku	31	2	93.93	8	7	53.33

Table 4.23: In-Sample and Out-of-Sample performance benchmark for static-point-on-hand poses with two hands

No.	**Model**	**App**	**Pose**	**In-Sample**				**Out-of-Sample**			
				Acc	**Pre**	**Rec**	**F1**	**Acc**	**Pre**	**Rec**	**F1**
1	I3D [10]	✓		63.33	69.47	57.27	62.78	12.22	13.93	6.74	9.08
2	Fusion-3 [21]	✓		81.67	84.05	79.81	81.88	26.11	41.6	19.3	26.37
3	MEMP [66]	✓		**95**	**95.45**	**94.55**	**95**	40	46.51	35.95	40.55
4	DeepSign-CNN [49]	✓		85	86.39	82.88	84.6	26.66	32.44	22.38	26.49
5	Pose-GRU [28]		✓	93.33	93.82	92.88	93.35	15.55	22.6	10.53	14.37
6	Pose-TGCN [28]		✓	76.66	76.56	75.31	75.93	8.88	10.42	2.76	4.37
7	SPOTER [8]		✓	46.66	46.19	39.87	42.8	22.77	26.68	19.74	22.69
8	Bi-RNN [23]		✓	74.17	77.22	70.48	73.69	16.11	13.13	13.41	13.26
9	FNN-LSTM [23]		✓	66.67	71.38	62.87	66.85	26.11	28.22	20.93	24.03
10	ChatGPT4	✓		94	95.32	93.86	93.67	N/A	N/A	N/A	N/A
11	**Our(C+Bi+TG)**	✓	✓	80.8	81.7	79.72	80.7	**41.7**	**48.8**	**36.4**	**41.7**

where: App = Appearance Representation, Pose = Pose-based Representation, Acc = Accuracy, Pre = Precision, Rec = Recall and F1 = F1-Score

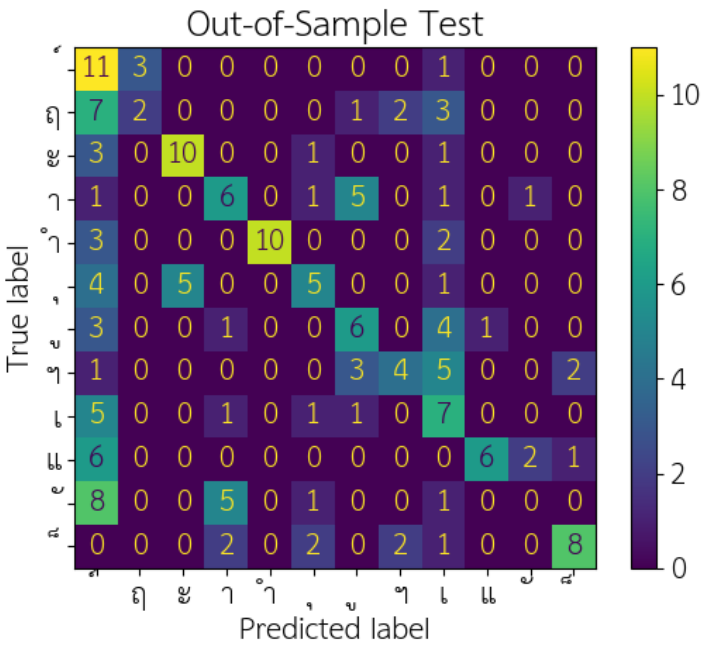


Figure 4.6: Confusion matrix of the static-point-on-hand poses with two hands on out-of-sample test

the accuracy stands at 81.3%, precision at 85%, recall at 79.02%, and F1-score at 81.9% , as seen in Table 4.25. The accuracy for dynamic point-on-hand is presented in Table 4.26 and more information can be found in Figure 4.7 for the in-sample test and Figure 4.8 for the out-of-sample test.

In table 4.27, the "Our (C+T+TG)" model, which utilizes a combination of CNN-LSTM (C), Transformer (T), and Temporal Graph Convolution Networks (TG) and incorporates both appearance and pose-based data representations, emerges as the most proficient model for dynamic point-on-hand pose estimation with two hands over five classes. It achieves in-sample perfection with 100% across all metrics and leads the out-of-sample evaluation with 81.3% accuracy and 81.9% F1-Score. This demonstrates the model's sophisticated ability to analyze and synthesize complex data, attributing to its CNN-LSTM structure's efficiency in feature extraction from visual data, the Transformer's global contextual capabilities, and the TGCN's dynamic temporal relationship interpretation. The integrated model can learn more complex representations by combining the strengths of CNN-LSTM, Transformers, and TGCNs. It benefits from the robust spatial feature and sequential data extraction of CNN-LSTM, the advanced contextual and tem-

poral insights provided by Transformers, and the structural analysis strengths of TGCN.

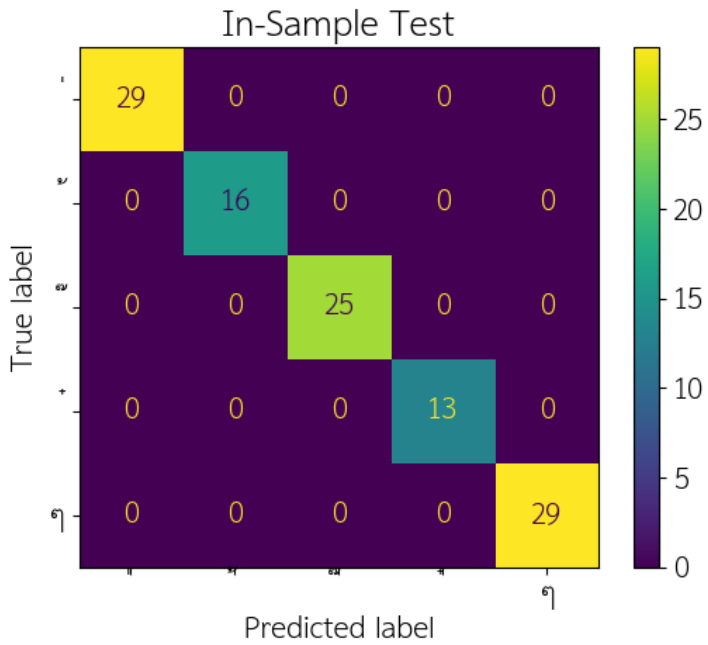


Figure 4.7: Confusion matrix of dynamic-point-on-hand poses with two hands on in-sample test

Table 4.28 illustrates the in-sample outcome of total two-hand poses. Integrating the Gated Recurrent Unit (GRU) with the Spatial-Temporal Graph Convolutional Network (TGCN) demonstrates exceptional performance, surpassing 93% in all evaluation parameters. On the contrary, the out-of-sample test reveals that the categorization percentage decreases to around 47% - 58% in various metrics, as seen in Table 4.29. The reason for the relatively low scores may be attributed to the inappropriate usage of many letters, including " -า ", " ู " , " ั " and " ็ ". These letters have an accuracy rate of less than 50% in the in-sample dataset, as shown in Table 4.30. Additionally, Figures 4.9 and 4.10 provide a detailed analysis of the total two-hand pose.

The table 4.31 illustrates the efficacy of various models in a 17-class total two-hand poses recognition task, with the Pose-GRU model exhibiting superior in-sample performance, reflected by high accuracy (92.94%), precision (93.62%), recall (92.03%), and F1 score (92.82%). This suggests that the Pose-GRU effectively learns and captures the nuances of total two-hand poses within the training dataset. On the other hand, the "Our(C+G)" model, which is an integration of CNN-LSTM (C) and GRU (G) models and utilizes both

Table 4.24: In-Sample Evaluation metrics for dynamic-point-on-hand poses with two hands

No.	Modalities	RGB		Coordinate				Graph	Parameters (M)	In-Sample Evalutation(%)				
		C	V	L	Bi	G	T	TG		Top-3	Accuracy	Precision	Recall	F1-Score
1	Single Modality	✓							16.298	**100**	84	85.4	83.42	84.4
2			✓						15.102	88	40	48.5	36.88	41.9
3				✓					3.478	**100**	**100**	**100**	**100**	**100**
4					✓				5.528	**100**	96	96	96	96
5						✓			4.286	**100**	78	78.9	77.12	78
6							✓		3.271	**100**	80	83	76.65	79.7
7								✓	**0.535**	96	78	80.4	76.31	78.3
8	2 Modalities	✓		✓					64.996	**100**	92	93.9	89.41	91.6
9		✓			✓				66.852	**100**	88	92.5	83.74	87.9
10		✓				✓			65.612	**100**	94	95.2	92.64	93.9
11		✓					✓		64.796	**100**	82	82.9	79.95	81.4
12		✓						✓	19.854	98	90	90.4	89.6	90
13			✓	✓					18.585	**100**	90	91.9	88.37	90.1
14			✓		✓				20.441	**100**	82	86.9	78.34	82.4
15			✓			✓			19.201	**100**	96	96	96	96
16			✓				✓		18.385	88	48	52.4	44.97	48.4
17			✓					✓	15.670	86	38	40.3	36.67	38.4
18				✓				✓	4.199	**100**	96	96.3	95.7	96
19					✓			✓	6.055	**100**	90	91.1	89.12	90.1
20						✓		✓	4.815	**100**	98	98.2	97.8	98
21							✓	✓	3.999	**100**	32	10.9	10.9	10.9
22	3 Modalities	✓		✓				✓	65.556	**100**	88	90.7	86.02	88.3
23		✓			✓			✓	67.411	**100**	86	86	85.8	85.9
24		✓				✓		✓	66.171	**100**	98	98.2	97.8	98
25		✓					✓	✓	65.355	**100**	**100**	**100**	**100**	**100**
26			✓	✓				✓	19.145	**100**	96	96.3	95.7	96
27			✓		✓			✓	21	94	78	78.7	75.37	77
28			✓			✓		✓	19.750	**100**	96	96.4	95.6	96
29			✓				✓	✓	18.944	96	54	58.3	49.6	53.6

where: C = CNN-LSTM, V = VGG-LSTM, L = LSTM, Bi = BiLSTM, G = GRU, T = Transformer, TG = TGCN

Table 4.25: Out-of-Sample Evaluation metrics for dynamic-point-on-hand poses with two hands

No.	Modalities	RGB		Coordinate				Graph	Out-of-Sample Evaluation(%)				
		C	V	L	Bi	G	T	TG	Top-3	Accuracy	Precision	Recall	F1-Score
1	Single Modality	✓							90.67	57.3	58	52.11	54.9
2			✓						76	25.3	19.1	21.89	20.4
3				✓					**100**	66.7	68.2	64.69	66.4
4					✓				97.33	70.7	67.7	69.32	68.5
5						✓			82.67	50.7	42.4	48.63	45.3
6							✓		88	56	57.3	52.88	55
7								✓	85.33	44	30.5	40.24	34.7
8	2 Modalities	✓		✓					93.33	65.3	64.3	63.9	64.1
9		✓			✓				97.33	61.3	60.4	60.4	60.4
10		✓				✓			**100**	76	76.5	75.51	76
11		✓					✓		93.33	78.7	80.7	76.23	78.4
12		✓						✓	86.67	64	65.7	61.07	63.3
13			✓	✓					89.33	60	51.6	57.75	54.5
14			✓		✓				**100**	73.3	76.3	64.32	69.8
15			✓			✓			**100**	76	74.4	74.4	74.4
16			✓				✓		77.33	30.7	28.7	26.21	27.4
17			✓					✓	80	34.7	45.5	26.89	33.8
18				✓				✓	93.33	68	74.2	62.59	67.9
19					✓			✓	93.33	58.7	59.7	57.93	58.8
20						✓		✓	98.67	80	79.8	79.2	79.5
21							✓	✓	86.67	26.7	10.5	22.41	14.3
22	3 Modalities	✓		✓				✓	90.67	46.7	48.3	39.4	43.4
23		✓			✓			✓	89.33	66.7	66.9	64.16	65.5
24		✓				✓		✓	93.33	80	83.3	78.45	80.8
25		✓					✓	✓	**100**	**81.3**	**85**	**79.02**	**81.9**
26			✓	✓				✓	93.33	80	83.7	76.98	80.2
27			✓		✓			✓	84	42.7	43.2	38.47	40.7
28			✓			✓		✓	98.67	74.7	71.9	72.91	72.4
29			✓				✓	✓	80	44	49.2	34.85	40.8

where: C = CNN-LSTM, V = VGG-LSTM, L = LSTM, Bi = BiLSTM, G = GRU, T = Transformer, TG = TGCN

Table 4.26: Accuracy for dynamic-point-on-hand poses with two hands by each Thai letter

No.	Thai Letters	Pron	In-Sample Test			Out-of-Sample Test		
			Correct	Incorrect	Acc(%)	Correct	Incorrect	Acc(%)
1	◌่	mai aek	29	0	100	15	0	100
2	◌้	mai tho	16	0	100	14	1	93.33
3	◌๊	mai tee	25	0	100	10	5	66.67
4	◌๋	mai jatawa	13	0	100	10	5	66.67
5	ๆ	mai ya-mok	29	0	100	12	3	80

Table 4.27: In-Sample and Out-of-Sample performance benchmark for dynamic-point-on-hand poses with two hands

No.	**Model**	**App**	**Pose**	**In-Sample**				**Out-of-Sample**			
				Acc	**Pre**	**Rec**	**F1**	**Acc**	**Pre**	**Rec**	**F1**
1	I3D [10]	✓		54	59.41	46.85	52.39	38.66	31.66	23.17	26.76
2	Fusion-3 [21]	✓		74	81.24	66.33	73.03	48	59.02	38.68	46.73
3	MEMP [66]	✓		90	91.3	88.76	90.01	69.33	73.24	66.58	69.75
4	DeepSign-CNN [49]	✓		94	94.88	93.45	94.16	53.33	43.02	50.9	46.63
5	Pose-GRU [28]		✓	**100**	**100**	**100**	**100**	74.66	78.78	70.79	74.57
6	Pose-TGCN [28]		✓	68	72.28	66.35	69.19	33.33	23.49	27.23	25.22
7	SPOTER [8]		✓	74	80.95	67.37	73.54	69.33	72.68	67.53	70.01
8	Bi-RNN [23]		✓	100	100	100	100	81.33	82.66	77.97	80.25
9	FNN-LSTM [23]		✓	86	86.43	85.5	85.96	58.67	58.13	56.51	57.31
10	ChatGPT4	✓		70	75.36	70	71.07	N/A	N/A	N/A	N/A
11	**Our(C+T+TG)**	✓	✓	**100**	**100**	**100**	**100**	**81.3**	**85**	**79.02**	**81.9**

where: App = Appearance Representation, Pose = Pose-based Representation, Acc = Accuracy, Pre = Precision, Rec = Recall and F1 = F1-Score

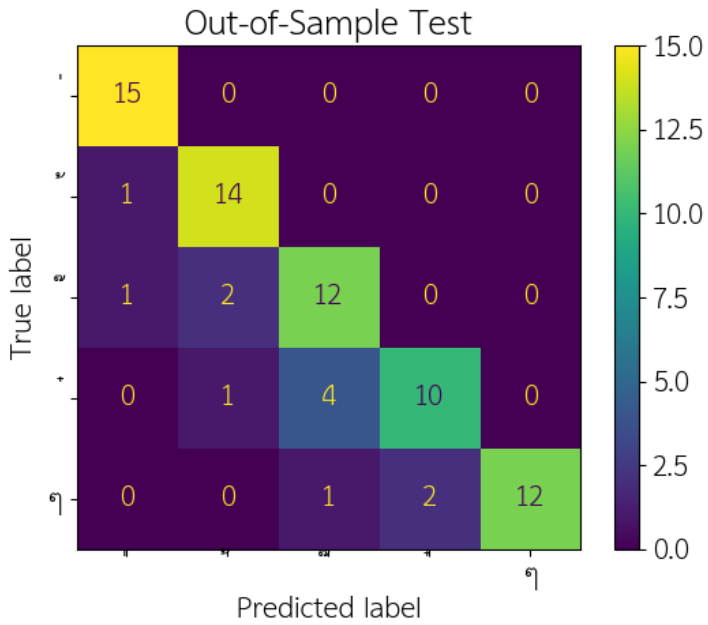


Figure 4.8: Confusion matrix of dynamic-point-on-hand poses with two hands on out-of-sample test

appearance and pose-based representations, outstrips its counterparts in out-of-sample performance with an accuracy of 53.3% and an F1-Score of 52.6%.

Table 4.28: In-Sample evaluation metrics for total two-hand poses

No.	Modalities	Vision		Skeleton				Graph	Parameters (M)	In-Sample Evalutation(%)					
		C	V	L	Bi	G	T	TG		Top-3	Top-5	Accuracy	Precision	Recall	F1-Score
1	Single Modality	✓							15.797	**100**	**100**	92.4	93.6	90.45	92
2			✓						15.209	70.59	80	34.1	38.6	29.75	33.6
3				✓					2.029	95.88	97.65	88.8	90.4	85.72	88
4					✓				1.711	95.29	99.41	78.8	80.3	75.45	77.8
5						✓			3.376	97.06	**100**	86.5	87.7	84.75	86.2
6							✓		3.274	98.24	99.41	84.7	87.1	82.05	84.5
7								✓	**0.535**	90.59	95.29	68.8	73	62.94	67.6
8	2 Modalities	✓		✓					58.581	97.06	99.41	81.2	82.1	79.35	80.7
9		✓			✓				58.096	95.29	97.06	82.9	85.4	79.98	82.6
10		✓				✓			59.901	94.12	98.24	77.6	78.2	74.11	76.1
11		✓					✓		59.831	95.29	98.82	63.5	65.3	59.38	62.2
12		✓						✓	57.077	88.24	95.88	70	74.9	64.48	69.3
13			✓	✓					16.994	88.24	95.88	61.2	62.3	59.37	60.8
14			✓		✓				16.509	72.35	84.71	48.2	51.9	41.79	46.3
15			✓			✓			18.314	92.35	95.29	68.8	70.7	65.68	68.1
16			✓				✓		18.244	84.71	96.47	40.6	47.7	34.74	40.2
17			✓					✓	15.490	95.29	96.47	72.9	74.8	70.71	72.7
18				✓				✓	2.759	96.47	98.24	90.6	91.1	89.32	90.2
19					✓			✓	2.273	97.65	99.41	84.7	85	83.02	84
20						✓		✓	4.078	98.82	98.82	**93.5**	**93.4**	**93**	**93.2**
21							✓	✓	4.009	93.53	95.88	82.4	86.4	79.49	82.8
22	3 Modalities	✓		✓				✓	59.141	93.53	95.88	67.6	71.1	64.98	67.9
23		✓			✓			✓	58.655	85.29	91.76	55.9	60.3	50.9	55.2
24		✓				✓		✓	60.460	94.71	97.06	81.2	80.9	80.5	80.7
25		✓					✓	✓	60.391	26.47	42.94	10	1.8	7.46	2.9
26			✓	✓				✓	17.554	98.24	**100**	80	81.2	78.64	79.9
27			✓		✓			✓	17.068	88.82	93.53	57.6	59.1	53.94	56.4
28			✓			✓		✓	18.564	94.71	98.24	84.7	85.4	83.23	84.3
29			✓				✓	✓	18.804	96.47	99.41	70.6	71.9	67.63	69.7

where: C = CNN-LSTM, V = VGG-LSTM, L = LSTM, Bi = BiLSTM, G = GRU, T = Transformer, TG = TGCN

Table 4.29: Out-of-Sample evaluation metrics for total two-hand poses

No.	Modalities	RGB		Coordinate				Graph	Out-of-Sample Evaluation(%)					
		C	V	L	Bi	G	T	TG	Top-3	Top-5	Accuracy	Precision	Recall	F1-Score
1	Single Modality	✓							59.61	67.06	44.7	53.6	35.62	42.8
2			✓						45.88	60	22.7	21.5	19.05	20.2
3				✓					57.25	64.71	29	34.3	24.09	28.3
4					✓				63.92	76.47	32.5	45.8	27.16	34.1
5						✓			65.88	**79.61**	34.9	36.7	30.81	33.5
6							✓		60.78	70.98	32.5	33.8	29.32	31.4
7								✓	48.24	60.78	19.6	21	12.54	15.7
8	2 Modalities	✓		✓					66.67	73.73	45.5	49.1	39.7	43.9
9		✓			✓				62.35	70.2	46.7	54.4	42.31	47.6
10		✓				✓			**71.37**	76.08	**53.3**	**58.3**	**47.92**	**52.6**
11		✓					✓		67.84	78.82	36.5	46.3	29.32	35.9
12		✓						✓	48.24	64.31	21.2	18.3	17.33	17.8
13			✓	✓					49.02	59.61	24.3	27	21.28	23.8
14			✓		✓				37.25	52.94	18.4	24.8	15.4	19
15			✓			✓			69.41	74.12	45.9	45.8	41.97	43.8
16			✓				✓		47.84	63.53	23.1	23.8	20.11	21.8
17			✓					✓	61.57	70.59	43.1	52.9	33.07	40.7
18				✓				✓	63.53	73.73	33.3	41.2	30.57	35.1
19					✓			✓	57.25	69.02	31.8	34.2	27.69	30.6
20						✓		✓	68.24	75.69	40	44.1	35.44	39.3
21							✓	✓	54.51	67.45	37.6	38.4	27.43	32
22	3 Modalities	✓		✓				✓	60.39	72.94	41.6	52	35.65	42.3
23		✓			✓			✓	55.29	64.31	29.8	29.5	22.77	25.7
24		✓				✓		✓	67.84	79.61	45.1	52.3	40.74	45.8
25		✓					✓	✓	26.67	42.75	7.8	1	5.67	1.7
26			✓	✓				✓	65.88	75.29	39.2	43.5	35.18	38.9
27			✓		✓			✓	51.37	65.49	29.8	39.1	24.2	29.9
28			✓			✓		✓	64.31	78.04	38.8	40.6	33.65	36.8
29			✓				✓	✓	70.98	81.18	33.3	43.4	26.36	32.8

where: C = CNN-LSTM, V = VGG-LSTM, L = LSTM, Bi = BiLSTM, G = GRU, T = Transformer, TG = TGCN

Table 4.30: Accuracy for total two-hand poses by each Thai letter

No.	Thai Letters	Pron	In-Sample Test			Out-of-Sample Test		
			Correct	Incorrect	Acc(%)	Correct	Incorrect	Acc(%)
1	◌์	ka-run	22	0	100	11	4	73.33
2	ฤ	ro-ruk	18	16	52.94	3	12	20
3	-ะ	sara a	27	2	93.1	10	5	66.67
4	-า	sara ar	4	13	30.76	0	15	0
5	-ำ	sara um	18	2	90	10	5	66.67
6	-◌ุ	sara oo	27	0	100	9	6	60
7	-◌ู	sara au	7	13	35	9	6	60
8	-◌่	mai aek	20	0	100	15	0	100
9	-◌้	mai tho	27	0	100	10	5	66.67
10	-◌๊	mai tee	25	0	100	15	0	100
11	-◌๋	mai jatawa	20	0	100	10	5	66.67
12	ๆ	pai yan noi	16	2	88.8	8	7	53.33
13	เ-	sara ae	16	4	80	3	12	20
14	แ-	sara aae	16	4	80	4	11	26.67
15	ๆ	mai ya-mok	16	2	88.8	7	8	46.67
16	-◌็	mai hen ar-karn	11	13	45.83	4	11	26.67
17	-◌็	mai tai-ku	7	11	38.88	8	7	53.33

Table 4.31: In-Sample and Out-of-Sample performance benchmark for total two-hand poses

No.	**Model**	**App**	**Pose**	**In-Sample**				**Out-of-Sample**			
				Acc	**Pre**	**Rec**	**F1**	**Acc**	**Pre**	**Rec**	**F1**
1	I3D [10]	✓		43.52	46.34	33.13	38.64	9.41	10.5	3.65	5.42
2	Fusion-3 [21]	✓		72.94	75.49	68.76	71.97	34.12	40.68	29.23	34.02
3	MEMP [66]	✓		85.88	87.73	84.22	85.94	32.55	38.05	24.45	29.77
4	DeepSign-CNN [49]	✓		83.52	85.84	81.47	83.6	45.88	60.67	40.18	48.34
5	Pose-GRU [28]		✓	**92.94**	**93.62**	**92.03**	**92.82**	47.45	53.62	43.38	47.96
6	Pose-TGCN [28]		✓	80.58	81.06	79.01	80.02	22.74	24.04	17.02	19.93
7	SPOTER [8]		✓	61.17	67.49	51	58.1	32.54	35.55	27.81	31.21
8	Bi-RNN [23]		✓	85.29	86.46	83.09	84.74	32.94	36.68	28.41	32.02
9	FNN-LSTM [23]		✓	79.41	84.16	73.78	78.63	29.8	34.12	22.67	27.24
10	ChatGPT4	✓		86	82.14	86.07	83.39	N/A	N/A	N/A	N/A
11	**Our(C+G)**	✓	✓	77.6	78.2	74.11	76.1	**53.3**	**58.3**	**47.92**	**52.6**

where: App = Appearance Representation, Pose = Pose-based Representation, Acc = Accuracy, Pre = Precision, Rec = Recall and F1 = F1-Score

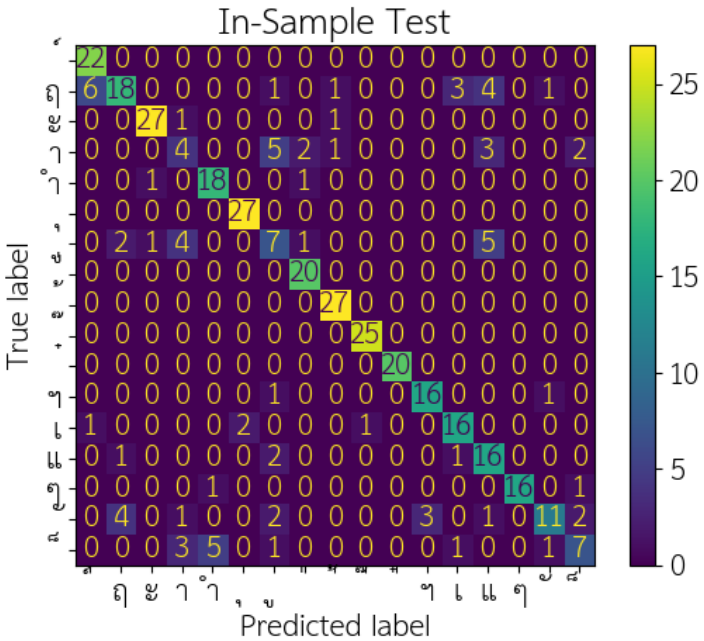


Figure 4.9: Confusion matrix of total two-hand poses on in-sample test

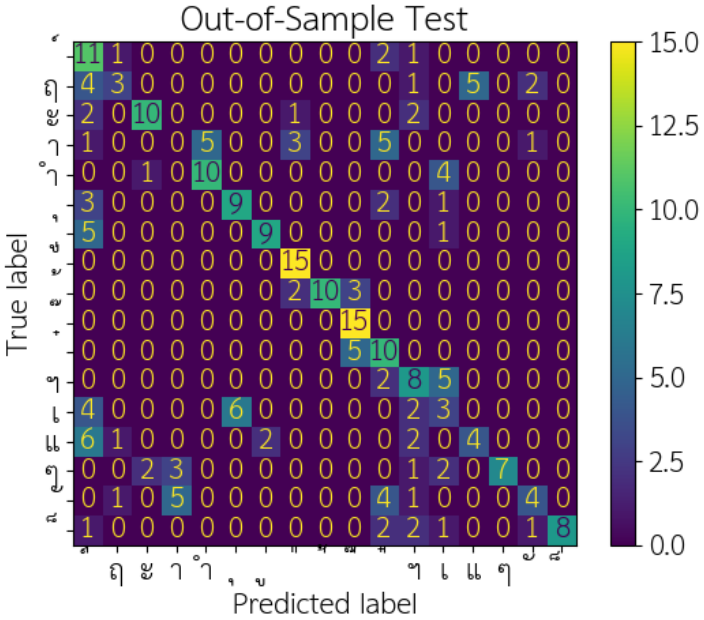


Figure 4.10: Confusion matrix of total two-hand poses on out-of-sample test

Chapter 5

Conclusion and Discussion

This research endeavors to enhance the communication capabilities of deaf and hard-of-hearing individuals by introducing the Thai Finger Spelling (TFS) dataset, which covers all 90 key elements representing TFS letters. Comprehensive experiments were conducted using a variety of deep learning-based architectures categorized based on their application modality.

For visual modality, CNN-LSTM and VGG-LSTM were employed, analyzing RGB sequences. The second modality involved human joint skeletons, utilizing LSTM, BiLSTM, GRU, and Transformer to process coordinates. The third modality used TGCN for graph representations of joint structure. Among these, TGCN stood out as the most efficient and lightweight option across various scenarios.

Notably, in real-life single-hand pose scenarios, the coordinate-sequencing and graph structure-based modalities (pose-based modality) rely on landmark data, it is inherently more robust against variations in lighting, background, and clothing, which can hinder the performance of RGB-based models. This resilience makes pose-based systems particularly effective in diverse and dynamic environments, enhancing their applicability in real-world settings where such variations are common. Consequently, the pose-based modality not only improves the precision of sign language interpretation but also extends the usability and accessibility of TFS technologies across different scenarios. Therefore, a combination of Transformer and TGCN delivered unparalleled performance.

However, the two-hand scenarios, whether analyzed through RGB-based or pose-based modalities, present specific challenges that complicate their interpretation and analysis. One of the primary drawbacks is the increased complexity of tracking and distinguishing the interactions between two hands, especially when gestures involve rapid or overlapping movements. In RGB-based systems, this complexity can lead to occlusions where one hand may obscure the other, making it difficult to detect and classify gestures accurately. Similarly, in pose-based systems, accurately capturing the spatial relationship between two interacting hands can be challenging if the pose estimation algorithms are not finely tuned to recognize close or overlapping hand positions. Additionally, both modalities may struggle with the higher computational demands required to process the additional data from two-hand gestures, potentially leading to slower response times and decreased system efficiency. This increased complexity and computational requirement can hinder the practical deployment of TFS systems in real-time applications where quick and accurate gesture recognition is crucial. Therefore, combining only the skeleton coordinate-based and graph structure-based modalities resulted in suboptimal outcomes. This was due to insufficient data from the obscured hand's joints and inadequate graph-structure data in two-hand pose scenarios; therefore, integrating three modalities is suitable for some scenarios.

For optimal performance in real-world applications, it's essential to combine all three analysis modalities and incorporate a hand pose classification step. Although many models achieve high accuracy with familiar data (in-sample), it's important to note that such success might not extend to real-world scenarios (out-of-sample) or with new, unseen data. Additionally, our benchmark experiments reveal that our methodology outperforms existing state-of-the-art techniques in five out of seven conditional hand pose evaluations, especially in complex scenarios involving two-hand poses. This research aims to improve evaluation methods, ensuring more accurate and precise results that can meaningfully benefit the deaf and hard-of-hearing community.

In future work, we aim to create an extensive and complete dataset of Natural Thai Sign Language (NTSL), which will cover all aspects of Thai sign language [36]. Our objective

is to streamline and optimize our model by reducing its overall parameter count, thereby enhancing the communication experience between people with normal hearing abilities and those who are deaf. Ultimately, we are working towards implementing this advanced technology on AI-embedded boards, making it more accessible and user-friendly in the future.

Bibliography

[1] Wadood Abdul, Mansour Alsulaiman, Syed Umar Amin, Mohammed Faisal, Ghulam Muhammad, Fahad R Albogamy, Mohamed A Bencherif, and Hamid Ghaleb. Intelligent real-time arabic sign language classification using attention-based inception and bilstm. *Computers and Electrical Engineering*, 95:107395, 2021.

[2] Abulikemu Abuduweili, Xin Wu, and Xingchen Tao. Efficient method for categorize animals in the wild. *arXiv preprint arXiv:1907.13037*, 2019.

[3] Sarah Alyami, Hamzah Luqman, and Mohammad Hammoudeh. Isolated arabic sign language recognition using a transformer-based model and landmark keypoints. *ACM Transactions on Asian and Low-Resource Language Information Processing*, 2023.

[4] Varadach Amatanon, Suwatchai Chanhang, Phornphop Naiyanetr, and Sanitta Thongpang. Sign language-thai alphabet conversion based on electromyogram (emg). In *The 7th 2014 Biomedical Engineering International Conference*, pages 1–4, 2014.

[5] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[6] Jordan J Bird, Anikó Ekárt, and Diego R Faria. British sign language recognition via late fusion of computer vision and leap motion with transfer learning to american sign language. *Sensors*, 20(18):5151, 2020.

[7] Andrej Karpathy Blog. The unreasonable effectiveness of recurrent neural networks.

URL: http://karpathy. github. io/2015/05/21/rnn-effectiveness/dated May, 21:31, 2015.

[8] Matyáš Boháček and Marek Hrúz. Sign pose-based transformer for word-level sign language recognition. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 182–191, 2022.

[9] Pranali Bora, Tulika Awalgaonkar, Himanshu Palve, Raviraj Joshi, and Purvi Goel. Icodenet-a hierarchical neural network approach for source code author identification. In *2021 13th International Conference on Machine Learning and Computing*, pages 180–185, 2021.

[10] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6299–6308, 2017.

[11] Chana Chansri and Jakkree Srinonchat. Reliability and accuracy of thai sign language recognition with kinect sensor. In *2016 13th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON)*, pages 1–4. IEEE, 2016.

[12] Kullawat Chaowanawatee, Kittasil Silanon, and Thitinan Kliangsuwan. Thai fingerspelling using vision transformer. *International Journal of Advanced Computer Science and Applications*, 14(11), 2023.

[13] Wenhu Chen. Large language models are few (1)-shot table reasoners. *arXiv preprint arXiv:2210.06710*, 2022.

[14] Harm Derksen. The graph isomorphism problem and approximate categories. *Journal of Symbolic Computation*, 59:81–112, 2013.

[15] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[16] Srujana Gattupalli, Amir Ghaderi, and Vassilis Athitsos. Evaluation of deep learning based pose estimation for sign language recognition. In *Proceedings of the 9th ACM international conference on PErvasive technologies related to assistive environments*, pages 1–7, 2016.

[17] Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. Neural message passing for quantum chemistry. In *International conference on machine learning*, pages 1263–1272. PMLR, 2017.

[18] Palash Goyal and Emilio Ferrara. Graph embedding techniques, applications, and performance: A survey. *Knowledge-Based Systems*, 151:78–94, 2018.

[19] Albert Haque, Boya Peng, Zelun Luo, Alexandre Alahi, Serena Yeung, and Li Fei-Fei. Towards viewpoint invariant 3d human pose estimation. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14*, pages 160–177. Springer, 2016.

[20] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[21] Al Amin Hosain, Panneer Selvam Santhalingam, Parth Pathak, Huzefa Rangwala, and Jana Kosecka. Hand pose guided 3d pooling for word-level sign language recognition. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 3429–3439, 2021.

[22] Isibor Kennedy Ihianle, Augustine O Nwajana, Solomon Henry Ebenuwa, Richard I Otuka, Kayode Owa, and Mobolaji O Orisatoki. A deep learning approach for human activities recognition from multimodal sensing devices. *IEEE Access*, 8:179028–179038, 2020.

[23] Werapat Jintanachaiwat, Kritsana Jongsathitphaibul, Nopparoek Pimsan, Mintra Sojiphan, Amorn Tayakee, Traithep Junthep, and Thitirat Siriborvornratanakul.

Using lstm to translate thai sign language to text in real time. *Discover Artificial Intelligence*, 4(1):17, 2024.

[24] Byeongkeun Kang, Subarna Tripathi, and Truong Q Nguyen. Real-time sign language fingerspelling recognition using convolutional neural networks from depth map. In *2015 3rd IAPR Asian Conference on Pattern Recognition (ACPR)*, pages 136–140. IEEE, 2015.

[25] Salman Khan, Muzammal Naseer, Munawar Hayat, and Syed Waqas Zamir. Transformers in vision: A survey. *ACMComputing Surveys*, 54, 2021.

[26] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[27] Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35:22199–22213, 2022.

[28] Dongxu Li, Cristian Rodriguez, Xin Yu, and Hongdong Li. Word-level deep sign language recognition from video: A new large-scale dataset and methods comparison. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 1459–1469, 2020.

[29] Zhouhan Lin, Minwei Feng, Cicero Nogueira dos Santos, Mo Yu, Bing Xiang, Bowen Zhou, and Yoshua Bengio. A structured self-attentive sentence embedding. *arXiv preprint arXiv:1703.03130*, 2017.

[30] Jun Liu, Haoyu Jin, Guangxia Xu, Mingwei Lin, Tao Wu, Majid Nour, Fayadh Alenezi, Adi Alhudhaif, and Kemal Polat. Aliasing black box adversarial attack with joint self-attention distribution and confidence probability. *Expert Systems with Applications*, 214:119110, 2023.

[31] CJ Masinde, J Gitahi, and M Hahn. Training recurrent neural networks for particu-

late matter concentration prediction. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 43:1575–1582, 2020.

[32] Federico Monti, Davide Boscaini, Jonathan Masci, Emanuele Rodola, Jan Svoboda, and Michael M Bronstein. Geometric deep learning on graphs and manifolds using mixture model cnns. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5115–5124, 2017.

[33] Kapil Kumar Nagwanshi, Ajit Noonia, Shivam Tiwari, Nitika Vats Doohan, Vijeta Kumawat, Tariq Ahamed Ahanger, and Enoch Tetteh Amoatey. Wearable sensors with internet of things (iot) and vocabulary-based acoustic signal processing for monitoring children's health. *Computational Intelligence and Neuroscience*, 2022, 2022.

[34] Pisit Nakjai and Tatpong Katanyukul. Hand sign recognition for thai finger spelling: An application of convolution neural network. *Journal of Signal Processing Systems*, 91:131–146, 2019.

[35] Pisit Nakjai, Patcharee Maneerat, and Tatpong Katanyukul. Thai finger spelling localization and classification under complex background using a yolo-based deep learning. In *Proceedings of the 11th International Conference on Computer Modeling and Simulation*, pages 230–233, 2019.

[36] National Assosiation of the Deaf in Thailand. Thai sign language posture. https://www.th-sl.com/search-by-act/, 2023. Accessed: (Oct 15, 2023).

[37] World Health Organization. Deafness and hearing loss. https://www.who.int/news-room/fact-sheets/detail/deafness-and-hearing-loss, 2023. Accessed: (April 12, 2023).

[38] Katerina Papadimitriou and Gerasimos Potamianos. Multimodal sign language recognition via temporal deformable convolutional sequence learning. In *Interspeech*, pages 2752–2756, 2020.

[39] Thongpan Pariwat and Pusadee Seresangtakul. Thai finger-spelling sign language recognition using global and local features with svm. In *2017 9th international conference on knowledge and smart technology (KST)*, pages 116–120. IEEE, 2017.

[40] Thongpan Pariwat and Pusadee Seresangtakul. Multi-stroke thai finger-spelling sign language recognition system with deep learning. *Symmetry*, 13(2):262, 2021.

[41] Wasupon Phothiwetchakun and Thanawin Rakthanmanon. Thai fingerspelling recognition using hand landmark clustering. In *2021 25th International Computer Science and Engineering Conference (ICSEC)*, pages 256–261. IEEE, 2021.

[42] Ramadam Ramo. Detection and diagnosis of skin diseases by using snake algorithm and neural networks. *Applied Sciences and Technology*, 4, 2022.

[43] Razieh Rastgoo, Kourosh Kiani, and Sergio Escalera. Video-based isolated hand sign language recognition using a deep cascaded model. *Multimedia Tools and Applications*, 79:22965–22987, 2020.

[44] Razieh Rastgoo, Kourosh Kiani, and Sergio Escalera. Sign language recognition: A deep survey. *Expert Systems with Applications*, 164:113794, 2021.

[45] Supawadee Saengsri, Vit Niennattrakul, and Chotirat Ann Ratanamahatana. Tfrs: Thai finger-spelling sign language recognition system. In *2012 second international conference on digital information and communication technology and it's applications (DICTAP)*, pages 457–462. IEEE, 2012.

[46] Jinnavat Sanalohit and Tatpong Katanyukul. Thai finger spelling recognition: Investigating mediapipe hands potentials. *arXiv preprint arXiv:2201.03170*, 2022.

[47] Wendy Sandler and Diane Carolyn Lillo-Martin. *Sign language and linguistic universals*. Cambridge University Press, 2006.

[48] Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. The graph neural network model. *IEEE transactions on neural networks*, 20(1):61–80, 2008.

[49] Jai Amrish Shah et al. *Deepsign: A deep-learning architecture for sign language*. PhD thesis, THE UNIVERSITY OF TEXAS AT ARLINGTON, 2018.

[50] Kittasil Silanon. Thai finger-spelling recognition using a cascaded classifier based on histogram of orientation gradient features. *Computational intelligence and neuroscience*, 2017, 2017.

[51] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[52] Himanshu Singh and Yunis Ahmad Lone. *Artificial Neural Networks*, pages 157–198. Apress, Berkeley, CA, 2020.

[53] William C Sleeman IV, Rishabh Kapoor, and Preetam Ghosh. Multimodal classification: Current landscape, taxonomy and future directions. *ACM Computing Surveys*, 55(7):1–31, 2022.

[54] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[55] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.

[56] Wuttichai Vijitkunsawat, Teeradaj Racharak, and Nguyen Le Minh. Deep multimodal-based number finger spelling recognizer for thai sign language. *International Symposium on Communications and Information Technologies*, pages 99–104, 2023.

[57] Wuttichai Vijitkunsawat, Teeradaj Racharak, Chau Nguyen, and Nguyen Le Minh. Video-based sign language digit recognition for the thai language: A new dataset and method comparisons. *Proceedings of the 12th International Conference on Pattern Recognition Applications and Methods*, pages 775–782, 2023.

[58] Yifan Wang, Fenghou Li, Hai Sun, Wenbo Li, Cheng Zhong, Xuelian Wu, Hailei Wang, and Ping Wang. Improvement of mnist image recognition based on cnn. In *IOP Conference Series: Earth and Environmental Science*, volume 428, page 012097. IOP Publishing, 2020.

[59] Yi Wu and Kymn Kyungsun. Automatic generation of traditional patterns and aesthetic quality evaluation technology. *Information Technology and Management*, pages 1–19, 2022.

[60] Qinkun Xiao, Minying Qin, Peng Guo, and Yidan Zhao. Multimodal fusion based on lstm and a couple conditional hidden markov model for chinese sign language recognition. *IEEE Access*, 7:112258–112268, 2019.

[61] Sijie Yan, Yuanjun Xiong, and Dahua Lin. Spatial temporal graph convolutional networks for skeleton-based action recognition. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

[62] Seongjun Yun, Minbyul Jeong, Raehyun Kim, Jaewoo Kang, and Hyunwoo J Kim. Graph transformer networks. *Advances in neural information processing systems*, 32, 2019.

[63] Fan Zhang, Valentin Bazarevsky, Andrey Vakunov, Andrei Tkachenka, George Sung, Chuo-Ling Chang, and Matthias Grundmann. Mediapipe hands: On-device real-time hand tracking. *arXiv preprint arXiv:2006.10214*, 2020.

[64] Shujun Zhang, Weijia Meng, Hui Li, and Xuehong Cui. Multimodal spatiotemporal networks for sign language recognition. *IEEE Access*, 7:180270–180280, 2019.

[65] Wenting Zhang. Design of news recommendation model based on sub-attention news encoder. *PeerJ Computer Science*, 9:e1246, 2023.

[66] Xinyu Zhang and Xiaoqiang Li. Dynamic gesture recognition based on memp network. *Future Internet*, 11(4):91, 2019.

[67] Yi-Fan Zhang, Peter Fitch, and Peter J Thorburn. Predicting the trend of dissolved oxygen based on the kpca-rnn model. *Water*, 12(2):585, 2020.

[68] Huiting Zheng, Jiabin Yuan, and Long Chen. Short-term load forecasting using emd-lstm neural networks with a xgboost algorithm for feature importance evaluation. *Energies*, 10(8):1168, 2017.

[69] Jie Zhou, Ganqu Cui, Shengding Hu, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. Graph neural networks: A review of methods and applications. *AI open*, 1:57–81, 2020.

Publications

Journal

[1] **Wuttichai Vijitkunsawat**, Teeradaj Racharak, and Nguyen Le Minh. "Deep Multimodal-based Finger Spelling Recognition for Thai Sign Language: A New Benchmark and Model Composition", Machine Vision and Applications, Volume 35, Issue 4, pages 76, 2024.

International Conference papers

[2] **Wuttichai Vijitkunsawat**, Teeradaj Racharak, Chau Nguyen, Nguyen Le Minh, "Video-Based Sign Language Digit Recognition for the Thai Language: A New Dataset and Method Comparisons", 12th International Conference on Pattern Recognition Applications and Methods (ICPRAM2023), pages 775-782, 2023.

[3] **Wuttichai Vijitkunsawat**, Teeradaj Racharak, and Nguyen Le Minh. "Deep Multimodal-based Number Finger Spelling Recognizer for Thai Sign Language", 2023 International Symposium on Communications and Information Technologies (ISCIT2023), pages 105-110, 2023.