How to Cite:
Satiman, S., & Meesad, P. (2022). Deep transfer learning base on sequenced edge grid image technique for sign language recognition. *International Journal of Health Sciences*, *6*(S5), 9982–9997.
https://doi.org/10.53730/ijhs.v6nS5.12112

Deep transfer learning base on sequenced edge grid image technique for sign language recognition

Supathep Satiman
Department of Information Technology, King Mongkut's University of Technology North Bangkok, Thailand
Orcid Id: 0000-0003-2074-0893
Corresponding author email: supathep.s@email.kmutnb.ac.th

Phayung Meesad
Department of Information Technology, King Mongkut's University of Technology North Bangkok, Thailand
Orcid Id: 0000-0002-3742-6457

***Abstract*---**Sign language is a visual-gestural language used by hearing impaired person, they modality the gesture to convey meaning. The main problem with sign language communication is ordinary people do not understand the sign language. Therefore, sign language is one of the challenging problems in machine learning. In this paper, researchers focus on visual-based methods and optimize the data preprocessing apply with existing sign language resources. Researchers propose an innovative technique for video processing called Sequenced Edge Grid Images (SEGI) for sign language recognition to interpret hand gesture, body movement, and facial expression. Researchers collected several of sign language data from the internet, the data including Thai sign language utilize in everyday life. The proposed technique was implemented with a convolutional neural network (CNN). The experiments showed SEGI with CNN has increases test accuracy rate with approximately 11% compared to static hand gesture images. Finally, researchers discovered a CNN structure suitable for dataset and examination data by transferring a pre-trained CNN. The fine-tuning with SEGI technique improved 99.8%, thus highest among all the methods. From the results data-preprocesses technique of dataset generation and deep transfer learning was an effective way to improve the accuracy of sign language recognition.

***Keywords*---**sequenced edge grid image, sign language recognition, convolutional neural network, transfer learning

International Journal of Health Sciences ISSN 2550-6978 E-ISSN 2550-696X © 2022.
Manuscript submitted: 9 April 2022, Manuscript revised: 18 June 2022, Accepted for publication: 27 July 2022

1 Introduction

Sign language is commonly known as a primary language for deaf people with hearing impaired, which use modality of gesture to convey meaning. The main problem of sign language communication is most of ordinary people do not understand a specific language of sign language. Moreover, sign language is one of the most challenging problems in machine learning. In the field of sign language recognition (SLR) systems, there are many techniques and methods that researchers around the world demonstrated including two of the well-known methods which are the data pre-processing methods for SLR visual-based and device-based gesture recognition (Y. Dong, J. Liu, and W. Yan., 2021). On visual-based gesture recognition, many datasets are detecting and cropping specific on the hand movement with a camera to store video clips or static images (I. Makarov, N., et al.,2019), (W. Aly, S. Aly, and S. Almotairi, 2019).

In early periods, most of researchers used static images as datasets to construct the recognition model since a limitation of computer performance and capability limitations of traditional algorithms. And these datasets that use static image to represent the meaning of sign language communication, the category of sign language vocabularies that can be used is restricted and it has its own vocabulary and syntax which is completely different from written languages. As a result, the SLR model had a high accuracy, but researchers cannot use it in real-world applications because a motion of hand movements on complex vocabulary, long syllables words, phrases, or sentences. In addition, the main components of sign language communication include hand position, hand movement, facial expressions, and body movement. Therefore, the static images are not suitable for SLR system. Currently, many research apply dynamic sign language recognition methods but there is some issue on drawbacks with difficulties of recognizing complex hand gestures, low recognition accuracy for most dynamic sign language recognition, and potential problems in larger video sequence data training (Y. Liao, P. Xiong, W. Min, W. Min and J. Lu., 2019).

The devices-based is a popular method for SLR, which a suggested to solve the limitations of static hand gesture image recognition. The sensor devices are work on hand detecting and finger movements for example, gloves type with motion sensors, motion cameras to capture the depths image, Leap Motion Controller sensor for tracking motion of finger bones of human hands, armband wearables sensor for tracking fingerspelling, thermal camera to creating the thermal image of hand detection (D. Avola, M. Bernardi, L. Cinque, G. L. Foresti, and C. Massaroni., 2019), (P. V. V. Kishore, D. A. Kumar, A. S. C. S. Sastry, and E. K. Kumar., 2018).The result of dataset recognition with sensor devices is higher accuracy than static hand gesture image datasets. In addition, device-based can detect hand motion movements very well. However, researchers need to reconstruct all of these datasets with sensor devices which the cost has increases. Thus, existing data sources cannot be used as datasets to perform on the machine because they are visual-based forms such as sign language video clips on websites, sign language learning media in schools, and sign language published on television. Therefore, researchers focus on visual-based methods and optimize the data pre-processing to support these resources for maximum benefit.

This paper aims to study an innovative technique for digital image data preparation called Sequenced Edge Grid Images (SEGI) for SLR and to increase the recognition performance of the convolutional neural network (CNN) based on transfer learning. The CNN structure was optimized for recognition with our datasets generated by the SEGI technique. Additionally, transfer learning is used to solve the problem of insufficient data. Researchers also use fine-tuning when after pre-training a CNN model with our dataset. Then, conducted experiments to evaluate the transfer learning methods after fine-tuning, researchers can finally test the performance of the fine-tuned CNN for SLR based on the SEGI technique.

Dataset and Methods

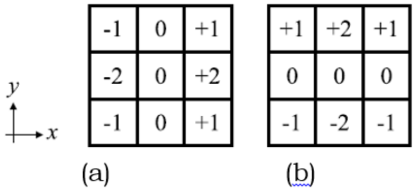
Sign language Datasets

Sequence images

In recent years, researchers proposed new data pre-processing techniques to solve motion in movement problem. Cooper et al. (2017) used features extraction techniques based on 2D and 3D tracking, Zadghorban and Nahvi (2018) used convert sign language video to the feature of hand motion and hand shape. Cui, Liu, and Zhang (2017) used mapping of video segments to glosses by using recurrent convolutional neural network for spatial-temporal feature extraction and sequence learning. Neyra-Gutiérrez and Shiguihara-Juárez [16] developed the continuous SLR with directly transcribes videos of sentences to sequences of ordered classes. Cui, Liu, and Zhang (2017) focus on optimizing the performance of dynamic gestures image by summarizing the hand gesture from sequences of video frames, capturing the key points of the result frames that use deep CNN with stacked temporal fusion layers as the feature extraction module.

Edge detection

Image edge detection is an important basis for image recognition extraction, it can reduce the amount of information from the image to be processed. The procedure of edge detection can compute gradient magnitudes and edge directions in an image and compute the edge strength on the gradient magnitudes of brightness within the image to detect and extract the edges as output (P. Prathusha, S. Jyothi, and D. M. Mamatha., 2018).



-1	0	+1
-2	0	+2
-1	0	+1

+1	+2	+1
0	0	0
-1	-2	-1

Figure 1: Sobel masks of 3 × 3 dimensions: (a) horizontal, (b) vertical.

The Sobel operator is widely used filter to compute gradients, therefore uses a pair of convolution matrices/masks as shown in Figure 1, one for estimating the horizontal gradient and the other for the vertical one. For example, the horizontal gradient mask is constructed by multiplying a horizontal averaging vector with a horizontal differential vector (K. Zhang, Y. Zhang, et al., 2018). Whether researchers let a_{ij} be a brightness value on a cell (i,j) of the source image, and dx_{ij} and dy_{ij} be approximated horizontal and vertical gradients on the cell(i,j), respectively, they are computed by using the Sobel operator as follows:

$$dx_{ij} \equiv [-10+1\ -20+2\ -10+1\,]\cdot$$
$$\left[a_i\ a_{ij+1}\ a_{i+1j+1}\ a_{i-1j}\ a_{ij}\ a_{i+1j}\ a_{i-1j-1}\ a_{ij-1}\ a_{i+1j-1}\,\right],$$

$$dy_{ij} \equiv [+1\ +2\ +1\,0\,0\,0\ -1\ -2\ -1\,]\cdot$$
$$\left[a_{i-1j+1}\ a_{ij+1}\ a_{i+1j+1}\ a_{i-1j}\ a_{ij}\ a_{i+1j}\ a_{i-1j-1}\ a_{ij-1}\ a_{i+1j-1}\,\right] \qquad (1)$$

where (·) means inner product calculation. After calculating the approximated gradients, researchers can calculate a gradient magnitude d_{ij} and its direction θ_{ij} (hereinafter, this is called gradient direction) on the cell (i,j), by the following formula:

$$d_{ij} \equiv \sqrt{dx_{ij}^2 + dy_{ij}^2} \qquad (2)$$
$$\theta_{ij} \equiv tan^{-1}(dy_{ij} + dx_{ij}) \qquad (3)$$

Another algorithm that gained popularity is the canny edge detector, which is an edge detection operator using a multi-stage method to detect a wide range of edges in images. It was developed by John F. Canny in 1986. The Canny Edge Detection Algorithm runs in 5 steps as follow (T. Fujimoto, T. Kawasaki, and K. Kitamura.,2019):

Smoothing: Blurring an image to remove the noise.
Finding gradients: The edges should be marked where the gradient of the image has large magnitudes.
Non-maximum suppression: individual local maxima consider to marked as edges.
Double thresholding: Potential edges are determined by thresholding.
Edge tracking by hysteresis: Final edges are determined by suppressing all edges that are not connected to a very certain (strong) edge.

CNN

In the field of image recognition, researchers have used the sequence image and dynamic image technique for data pre-processing with convolution neural network (CNN) in SLR task (M. A. Bencherif et al., 2021), (R. Rastgoo, K. Kiani, and S. Escalera., 2020). The CNN is a very high-performance machine learning tool for computer vision tasks because CNN has down-sampling layers, reducing the resolution of the feature map (A. TANG, K. LU, et al., 2015). The CNN mainly consists of two parts which are convolutional layers (CONV) and fully-connected layers (FULC) on Figure 2. CONV first extract and combine local features from the input image, and these features are then combined to output feature maps that represent a spatial arrangement of activations. Each unit in a CONV layer receives inputs from a set of units located in a small neighborhood in the previous layer. With a spatial arrangement, neurons can extract primitive visual features such as oriented edges, endpoints, and corners. CONV can support flexible image sizes and generate feature maps up to any size. On the other hand, the FULC needs to fixed-size/length input by their definition. The inputs are processed in FULC and converted into a 1D feature vector (flatten). Then 1D features represented on each neuron in the fully connected layer and multiplied with the weight of the neuron to produce the output (A. Neyra-Gutiérrez and P. Shiguihara-Juárez., 2020), (S. Ji, W. Xu, M. Yang, and K. Yu.,2013). In 2D CNN is performed at the convolutional layers to extract features from local neighborhood on feature maps in the previous layer. Formally, the value of a unit at position (x, y) in the ith feature map in the ith layer, denoted as v_{ij}^{xy}, is given by:

$$v_{ij}^{xy} = tanh\left(b_{ij} + \sum_m \quad \sum_{p=0}^{P_i-1} \quad \sum_{q=0}^{Qi-1} \quad w_{i\,jm}^{pq}\, v_{(i-1)m}^{(x+p)(y+q)}\right) \qquad (4)$$

where $tanh(\cdot)$ is the hyperbolic tangent function, b_{ij} is the bias for this feature map, m indexes over the set of feature maps in the $(i-1)$th layer connected to the current feature map, w_{ijk}^{pq} is the value at the position (p, q) of the kernel connected to the kth feature map, and P_i and Q_i are the height and width of the kernel, respectively .

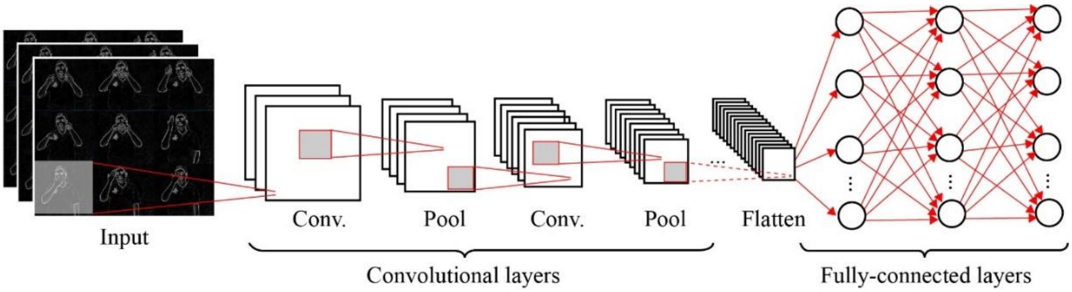


Figure 2: 2D CNN with the dataset was generated by SEGI technique.

Transfer Learning

In machine learning whereas the dataset has small samples, transfer learning method can be the source domain to the target domain, and could help to overcome the difficulty in data insufficient. If a domain is represented by $D = \{\chi, P(X)\}$, where χ is the feature space and $P(X)$ is the edge probability, and a task is represented by $T = \{y, f(x)\}$, where y is the label space and $f(x)$ is the target prediction function, the definition of transfer learning can be formally defined as follows (S. J. Pan and Q. Yang., 2010): given a learning task T_t on domain D_t, researchers can assist from another learning task T_s on domain D_s. Transfer learning aims to improve the performance of predictive function f_t for the task T_t by discovering and transferring knowledge from D_s and T_s, where $D_s \neq D_t$ and/or $T_s \neq T_t$ considering from the relationship between source domain and target domain. Transfer learning methods are divided into four major categories: instance-based, feature-based, parameter-based, and relation-based. Model-based deep transfer learning (H. Chang, J. Han, C. Zhong, et al., 2018) ,(J. Yosinski, J. Clune, Y. Bengio, and H. Lipson.,2014) is the most accepted one, and fine-tuning pre-trained models learned from large benchmark datasets in source domains, has been proven to be more effective than direct transfer learning [29].

The key to the success of model-based deep transfer learning is a low-level and middle-level features represented by a deep CNN is generic for different tasks (M. Oquab, L. Bottou, I. Laptev, and J. Sivic., 2014)

The propose SEGI technique

This research shown an innovative technique for digital image data preparation called Sequenced Edge Grid Images (SEGI) for SLR. The proposed approach for video data pre-processing, frame extraction from video files, data pre-processing and reconstructing new dataset form. SEGI aimed to solve the problem of static hand gesture images, therefore a single image unable to support all hand motion movement details. Moreover, communication requires body movements and facial expressions. As a result, the error rate of machine learning is increased. A SEGI is a technique to generate a set of motion within single images that stores gesture movement details at a different time. Optimal SEGI and input size are used to train CNN to classify Thai SLR. The video pre-processing to SEGI shown in Figure 3 is a semi-automated method with 4 steps as follow:

Image Frames: Image frames are captured from a video clip and the number of frames is set as desired.

Cropped Image: This step is to get an optimal number of image frames. The first step is to cut off an empty area and borders of each image frame.

Edge Detection: In this step converts the image frames (RGB mode) into a grey scale to perform edge detection. This process will reduce the information of images by remove color and texture (high-frequency data).

Concatenate sub-image: This step constructs SEGI by combining sub-image (edge image converted) and order from top-left to top-right.

The sign language translation considers three main parts consist of hand gestures, body movement, and facial expressions. Therefore, SEGI technique collects all spatial features from the sign language video clip then extract to sequences images (sub-image). Next, researchers convert sequenced images by a canny edge detector. Finally, researchers combine sub-images into the SEGI. The role of SEGI is to pre-process hand gestures according to each vocabulary from video clips collected into a single image as input to 2D CNN. In the SEGI, researchers proposed to increase the performance of 2D CNN able to leverage context across the height and width of the slice to make predictions.

Results and Discussion

Dataset based-on SEGI technique

The SEGI technique constructs our dataset (Figure 3), which takes a 5×5 grid image (row × column) from a video clip by sorting the sequenced image (sub-images) from top left to right. Preliminary experiment, a comparison between our dataset (SEGI technique) and sequenced grid image (SGI) in RGB mode (shown in Figure 4). The dataset for measuring the efficiency is counting numbers with sign language from zero to nine to recognition by CNN with 64x64 pixels of input sizes. That total of 9 classes and a total of 450 images, was divided into 360 training sets and 90 test sets, the results as shown in table1.

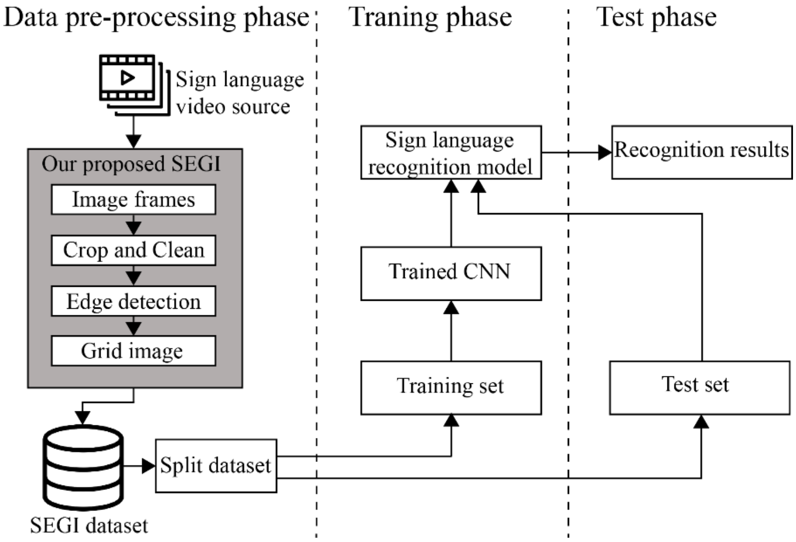


Figure 3: Block diagram representation of the proposed SEGI technique

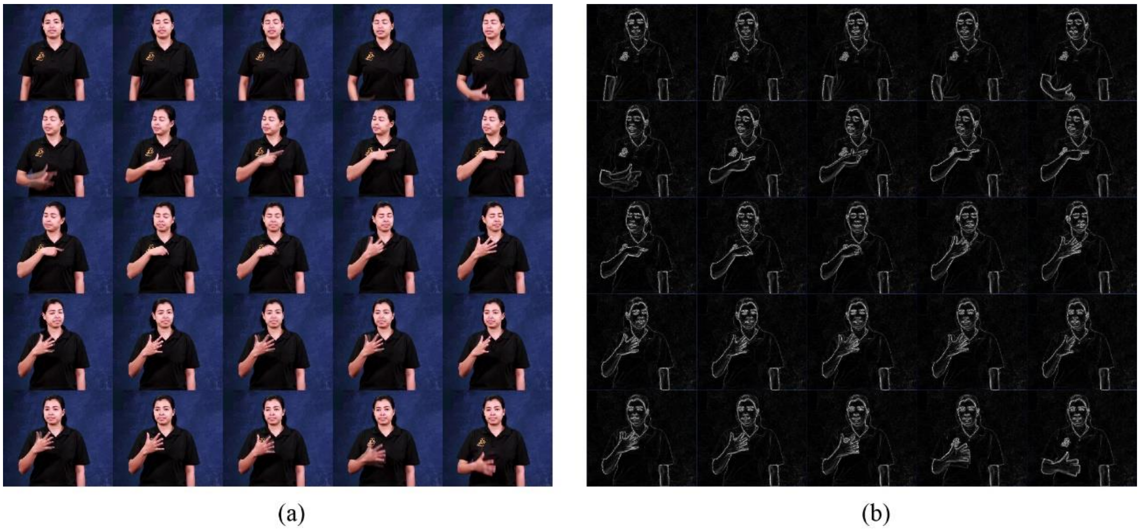


Figure 4: (a) image size 5×5 by SGI (RGB mode) and (b) image size 5×5 by SEGI

Table 1: Performance of sign language recognition compared to static images datasets, SGI technique and SEGI technique

Epochs	*Image dataset*	*Time (sec.)*	*Train Acc.*	*Train Loss*	*Valid Acc.*	*Valid Loss*	*Test Acc.*
30	Static	2132	0.821	0.024	0.613	3.896	0.659
	SGI	9897	0.995	0.017	0.677	2.689	0.673
	SEGI	2341	0.994	0.024	0.727	1.783	0.874
60	Static	6523	0.911	0.072	0.560	4.849	0.680
	SGI	15654	0.995	0.035	0.635	4.237	0.745
	SEGI	7975	0.994	0.051	0.729	3.053	0.882

The results showed that the dataset constructed with the SEGI technique improves the recognition performance for SLR compared to SGI. In addition, it found SEGI technique reduces the cost of processing time with approximately 49% for 60 Epochs of training compared to the SGI technique. In addition, the SEGI technique improves test accuracy with approximately 20% compared to the static image dataset.

Optimization the dataset based-on SEGI technique

The observe of dataset obtained from SEGI technique, found the sub-images were too redundant (as shown in Figure 5) since each of vocabulary has a simple gesture and short movements. Preliminary, one image from SEGI technique took sequence images as 25 frames per second from sign language video clip to transform into 5×5 sub-images. Therefore researchers focused on the optimal images size by rescaled the images size from 5×5 to 4×4 and 3×3. Finally, researchers compared the recognition performance when researchers reduced the grid size by different complexity gesture that is the length of the vocabulary from sign language video clips of 6 signers as follows:

Dataset 1: one-syllable vocabulary with 30 classes consisting of 4,500 images divided into 3,240 training sets and 900 test sets.
Dataset 2: two-syllable vocabularies with 37 classes consisting of 4,950 images divided into 3,960 training sets and 990 test sets.
Dataset 3: three and more syllable vocabularies with 10 classes consisting of 1,500 images divided into 1,200 training sets and 300 test sets.

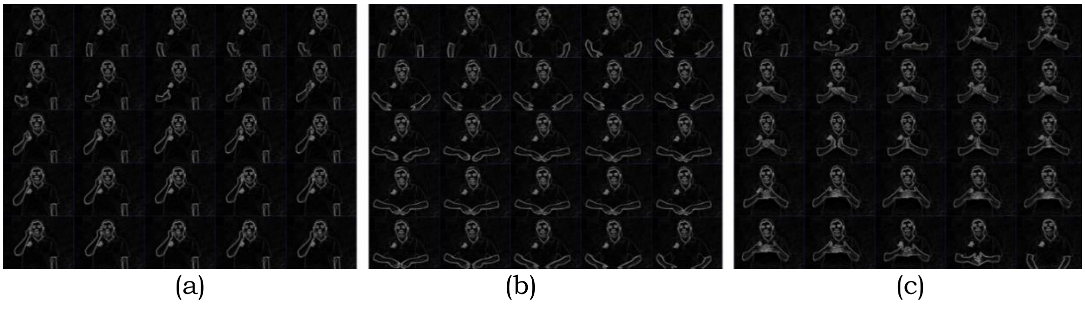

(a) (b) (c)

Figure 5: Example SEGI size 5×5 according to the length of Thai sign language vocabulary

(a) SEGI of one-syllable, (b) SEGI of two-syllable and (c) SEGI of three and more syllable.

Researchers also tested three input sizes: 32×32, 64×64, and 128×128 pixels. The CNN architecture, which uses the basic structure (CNN initial structure) contain Conv2D first layer (filters size 64, kernel size 3 and activation as ReLU) and MaxPooling2D last layer. The second layer is Conv2D (filters size 128, kernel size 3 and activation as ReLU) and the last MaxPooling2D layer. Next layer is Conv2D (filters size 256, kernel size 3 and activation as ReLU) and the last MaxPooling2D layer. And dropout equal to 0.2, flatten layer. The fully-connected layers are 256 nodes with ReLU activation and the last layer as 97 nodes with SoftMax activation. In leaning round is 60 epochs, which used 10-fold cross-validation to estimate the skill of the recognition model to find the optimal input size and SEGI size to obtain the best recognition performance, results are shown in Table 2.

Table 2: The recognition performance of dataset optimization (divided the sub-dataset by the length of the vocabularies)

Vocabulary length	*Input size (px.)*	*SEGI size (row × col)*	*Performance evaluation*					
			Time (sec.)	*Train acc.*	*Train loss*	*Valid acc.*	*Valid loss*	*Test acc.*
	32×32	3×3	893	0.031	3.402	0.033	3.401	0.033
One syllable		4×4	1403	0.030	3.439	0.035	3.562	0.032
		5×5	1994	0.030	3.402	0.031	3.403	0.030
	64×64	3×3	1170	0.986	0.048	0.967	0.120	0.882
		4×4	1723	0.982	0.058	0.957	0.156	0.717
		5×5	2328	0.973	0.091	0.912	0.484	0.216
	128×128	3×3	2183	0.993	0.023	0.989	0.037	**0.983**
		4×4	2934	0.990	0.036	0.974	0.092	0.874
		5×5	3855	0.988	0.041	0.946	0.264	0.557
	32×32	3×3	934	0.026	3.497	0.030	3.497	0.030
Two syllable		4×4	1466	0.028	3.497	0.030	3.497	0.030
		5×5	2119	0.028	3.497	0.030	3.497	0.030
	64×64	3×3	1224	0.988	0.040	0.974	0.087	0.903
		4×4	1771	0.986	0.046	0.955	0.191	0.744
		5×5	2536	0.974	0.053	0.928	0.213	0.715
	128×128	3×3	2353	0.992	0.027	0.989	0.039	**0.985**
		4×4	3017	0.989	0.038	0.977	0.077	0.894
		5×5	4064	0.985	0.052	0.943	0.267	0.558
	32×32	3×3	377	0.985	0.042	0.957	0.159	0.590
Three and more syllable		4×4	582	0.982	0.051	0.936	0.305	0.457
		5×5	868	0.979	0.058	0.911	0.427	0.250
	64×64	3×3	525	0.992	0.023	0.977	0.074	0.913
		4×4	806	0.990	0.030	0.963	0.133	0.750
		5×5	1177	0.984	0.045	0.939	0.263	0.453
	128×128	3×3	988	0.993	0.023	0.990	0.031	**0.997**
		4×4	1417	0.992	0.026	0.984	0.052	0.937
		5×5	1888	0.988	0.035	0.964	0.111	0.737

Table 2 shows the experimental results. The smallest size of SEGI with 3×3 provides the best performance compared to bigger sizes like 4×4 and 5×5, although the input size is scaled to a minimum of 32×32 pixels. When the input size is scaled up, it increases the performance of Thai SLR. When researchers used an input size of 128×128 input images, all sizes of SEGI can approach a higher accuracy performance in training, validation, and test sets because due to the reason that the SEGI is not over-compressed with the large size of the input. However, using larger images will cost higher computing time, a high-performance computing device is needed to process. Nevertheless, taking a long time to process does not mean better performance when the sign language dataset is large, which may be a problem in future experimentation. So, researchers recommend using input 64×64 pixels and SEGI size 3×3 as the default size.

Next, the experimentation of sign language recognition performance with our dataset based on SEGI (size 3x3) to compare the traditional dataset (static hand gesture images). Both datasets are all sign language vocabulary (total of three subsets, one-syllable, two-syllables, and three and more syllables) consisting of 10,950 images from 98,550 sub-images with 73 classes divided into 8,760 training sets and 2,190 test sets. Finally, researchers used 10-fold cross-validation to evaluate the recognition model.

Table 3: The comparison of image recognition performance between static hand gesture images and datasets based on SEGI technique

Sign language datasets		*Performance evaluation*				
	Epochs	*Train acc.*	*Train loss*	*Valid acc.*	*Valid loss*	*Test acc.*
Static hand gesture images	60	0.978	0.098	0.959	0.195	0.733
Dataset based on SEGI technique	60	0.979	0.064	0.819	1.087	**0.846**

From table 3 researchers found that the dataset with SEGI technique by images sizes 3x3 provide higher performance when compared to static hand gesture images. In addition, the results found researcher's dataset test accuracy rate had increase with approximately 11%.

Optimisation of 2D CNN structure for SEGI

The experimented in this section, researchers modifying the structure of a 2D CNN, aiming to obtain a suitable structure to enhance the sign language recognition efficiency in conjunction with the SEGI dataset. Researchers examined with different configurations of convolution function and the number of layers, activation function, pooling function and parameters as follows:

CNN optimize structure 1: first block consist of first two layers are Conv2D (filters size 64, kernel size 3 and activation as ReLU) and the last MaxPooling2D layer. The next block consists of the first two layers are Conv2D (filters size 32, kernel size 3, and activation as ReLU) and the last MaxPooling2D layer of this block. Researchers add dropout equal to 0.2, flatten layer. The fully-connected layers are 256 nodes with ReLU activation and 97 nodes with SoftMax activation.

CNN optimize structure 2: first block consists of the first two layers that are separableConv2D (filters size 64, kernel size 3, and activation as ReLU) and the last MaxPooling2D layer. And next block consists of the first two-layer is separableConv2D (filters size 32, kernel size 3, and activation as ReLU) and the last MaxPooling2D layer of this block. Researchers add dropout equal to 0.2, flatten layer. The fully-connected layers are 256 nodes with ReLU and 97 nodes with SoftMax.

CNN optimize structure 3: first block consists of SeparableConv2D on the first two layers (filters size 64, kernel size 3, and activation as ReLU) and SeparableConv2D (filters size 32, kernel size 3 and activation as ReLU). The next block contains SeparableConv2D (filters size 32, kernel size 3, and activation as ReLU) and convolution layer is SeparableConv2D (filters size 16, kernel size 3 and activation as ReLU). For MaxPooling2D researchers add dropout equal to 0.2, flatten layer. The last two fully-connected layers are 256 nodes with ReLU and 97 nodes with SoftMax.

CNN optimize structure 4: first block consists of SeparableConv2D (filters size 64, kernel size 3, and activation as ReLU) for the first layer then SeparableConv2D layer (filters size 32, kernel size 3, and activation as ReLU) and the last AveragePooling2D layer of this block. The next block contains SeparableConv2D (filters size 32, kernel size 3, and activation as ReLU) as the first layer then SeparableConv2D (filters size 16, kernel size 3, and activation as ReLU) and the last AveragePooling2D layer of this block. Researchers add dropout equal to 0.2, flatten layer. The fully-connected layers are 256 nodes with ReLU, 128 nodes with ReLU and 97 nodes with SoftMax.

CNN optimize structure 5: first block consists of SeparableConv2D (filters size 64, kernel size 3, and activation as ReLU) then SeparableConv2D (filters size 32, kernel size 3, and activation as ReLU), and researchers add dropout equal to 0.2 and flatten layer of this block. The next block consists of SeparableConv2D as a first layer (filters size 32, kernel size 3, and activation as ReLU) then SeparableConv2D (filters size 16, kernel size 3, and activation as ReLU), dropout equal to 0.2 and flatten layer. The last two fully-connected layers are 256 nodes with ReLU and 97 nodes with SoftMax.

CNN + Bi-LSTM and CNN + Bi-GRU: CNN initial structure is base of convolutional layer and add 2 layers of Bi-LSTM / Bi-GRU before the fully-connected layer.

VGG16 and ResNet: researchers load library of the pre-train model and retrain all layers with our dataset.

Table 4: The performance of optimized CNN structures for sign language recognition based on SEGI technique

CNN structure	*Performance evaluation*				
	Train acc.	*Train loss*	*Valid acc.*	*Valid loss*	*Test acc.*
CNN initial structure	0.979	0.064	0.819	1.087	0.846

CNN optimize structure1	0.988	0.156	0.968	0.217	0.874
CNN optimize structure2	0.991	0.036	0.974	0.119	0.874
CNN optimize structure3	0.990	0.036	0.978	0.092	0.906
CNN optimize structure4	0.991	0.037	0.978	0.110	**0.936**
CNN optimize structure5	0.995	0.023	0.975	0.154	0.824
CNN+Bi-LSTM	0.994	0.018	0.997	0.010	0.932
CNN+Bi-GRU	0.965	0.125	0.958	0.142	0.918
CNN+Bi-LSTM+Bi-GRU	0.957	0.152	0.952	0.162	0.911
VGG16	0.921	0.057	0.919	0.085	0.899
ResNet	0.907	0.482	0.898	0.461	0.891

From table 4 shown SeparableConv2D and AveragePooling2D are suitable for researcher's dataset. The filter size should be 64 and 32, 3 of kernel size, and ReLU activation function in the convolutional layers. In fully-connected layer have three layers comprising 256, 128, and 97 nodes respectively. The optimized CNN structures performance, from the experiment of 7 structures, structure 4 has the highest test accuracy at 93.6%. This structure has increase 12.3% compared to the CNN initial structure previously. The CNN optimize structure 4 is a pre-trained model (model-based) that will be used in for transfer learning method.

Transfer Learning

The final experimental of this research, apply model-based to transfer a new sign language datasets with a workplace vocabulary and communication from 4 signers. The SEGI technique is a dataset generator consisting of 2,250 images (20,250 sub-images) with 23 classes, which are divided into 1,800 training sets and 450 test sets. In transfer learning, researchers performed three strategies of fine-tuning as follows:

Strategy 1: Train the entire model. Researchers use the architecture of the pre-trained model and train it according to our dataset, which learns the model from scratch.
Strategy 2: Train some layers and leave the others frozen. The lower layers refer to general features, while the higher layers refer to specific features. Researchers apply and adjust the weights of the network.
Strategy 3: Freeze the convolutional base. Researchers kept the convolutional base in the original form and then use its outputs to feed the classifier by using the pre-trained model as a fixed feature extraction mechanism.

Table 5: The recognition accuracy of 3 fine-tuning strategies with the dataset

Fine-tune strategy	*Performance evaluation*					
	Process time	*Train acc.*	*Train loss*	*Valid acc.*	*Valid loss*	*Test acc.*
Strategy 1	1748.63	0.996	0.019	0.994	0.025	0.998
Strategy 2	1259.80	0.996	0.023	0.994	0.019	**0.998**
Strategy 3	1100.66	0.995	0.034	0.992	0.038	0.991

Table 5 shows the results of the transfer learning experiment, the results found strategy 2 of fine-tuning technique had the highest recognition accuracy as same as strategy 1, at 99.8%. However, strategy 2 had reduce processing time with approximately 28% compared to strategy 1. Therefore, the transfer learning method for the datasets created by SEGI technique (the fine-tuning of training some layers and leaving the others frozen) is optimal for researcher's dataset.

Conclusion

Researchers proposed a new video processing technique called Sequenced Edge Grid Images (SEGI) for sign language recognition applications. A dataset initiated from researcher's technique can be generated as a sequence image with edge detection to use with 2D CNN. Researchers found the recognition performance of 2D CNN with SEGI has higher accuracy than the dataset by SGI and static hand gesture image. Researchers compared dataset and static hand gesture images to recognize the 73 classes of sign language vocabularies from 5 signers. The results showed that SEGI technique enhances performance of 2D CNN for the sign language recognition system. In addition, with the reduced size SEGI, researchers found SEGI size 3×3 image yield higher accuracy than SEGI sizes 4×4 and 5×5 with input image size 32×32 pixels and 64×64 pixels. Moreover, by scaling up input size to 128×128 pixels and all sizes of SEGI can lead to high performance. In addition, the SEGI got a higher test accuracy rate than the static hand gesture images. Finally, researchers discovered a CNN structure suitable for the dataset by the SEGI with high recognition performance. Along with the grid image size 3×3 had increased a new sign language dataset on workplace vocabulary and communication from 4 signers. In addition, researchers applied transfer learning method to incrementally learn the new sign language datasets. For future study, researchers will optimize the SEGI algorithm to generate the dataset faster and reduce processing time for the better support real-time sign language translation applications.

Reference

1. A. Neyra-Gutiérrez and P. Shiguihara-Juárez., (2020). Feature Extraction with Video Summarization of Dynamic Gestures for Peruvian Sign Language Recognition, IEEE XXVII International Conference on Electronics, Electrical Engineering and Computing (INTERCON), Sep. 2020, pp. 1–4. doi: 10.1109/INTERCON50315.2020.9220243.
2. A. TANG, K. LU, Y. WANG, J. HUANG, and H. LI., (2015). A Real-Time Hand Posture Recognition System UsingDeep Neural Networks, ACM Trans. Intell. Syst. Technol., vol. 6, no. 2, pp. 1–23, Mar. 2015.
3. D. Avola, M. Bernardi, L. Cinque, G. L. Foresti, and C. Massaroni., (2019). Exploiting Recurrent Neural Networks and Leap Motion Controller for the Recognition of Sign Language and Semaphoric Hand Gestures, IEEE Trans. Multimed., vol. 21, no. 1, pp. 234–245, Jan. 2019, doi: 10.1109/TMM.2018.2856094.
4. D. S. Breland, S. B. Skriubakken, A. Dayal, A. Jha, P. K. Yalavarthy, and L. R. Cenkeramaddi., (2021). Deep Learning-Based Sign Language Digits Recognition From Thermal Images With Edge Computing System, IEEE Sens.

J., vol. 21, no. 9, pp. 10445–10453, May 2021, doi: 10.1109/JSEN.2021.3061608.

5. H. Bhavsar and J. Trivedi., (2019). Hand Gesture Recognition for Indian Sign Language using Skin Color Detection and Correlation-Coefficient algorithm with Neuro-Fuzzy Approach, International Conference on Advances in Computing, Communication and Control (ICAC3), , pp. 1–5. doi: 10.1109/ICAC347590.2019.9036832.
6. H. Chang, J. Han, C. Zhong, A. M. Snijders, and J.-H. Mao., (2018). Unsupervised transfer learning via multi-scale convolutional sparse coding for biomedical applications, IEEE Trans. Pattern Anal. Mach. Intell., vol. 40, no. 5, pp. 1182–1194.
7. H. Cooper, E.-J. Ong, N. Pugeault, and R. Bowden., (2017). Sign Language Recognition Using Sub-units, in Gesture Recognition, S. Escalera, I. Guyon, and V. Athitsos, Eds. Cham: Springer International Publishing, 2017, pp. 89–118. doi: 10.1007/978-3-319-57021-1_3.
8. I. Makarov, N., et al., (2019). Russian Sign Language Dactyl Recognition, 42nd International Conference on Telecommunications and Signal Processing (TSP), Jul. 2019, pp. 726–729. doi: 10.1109/TSP.2019.8768868.
9. J. Yosinski, J. Clune, Y. Bengio, and H. Lipson., (2014). How transferable are features in deep neural networks? in Proc. Adv. Neural Inf. Process. Syst. (NIPS), 2014, pp. 3320–3328.
10. K. Revanth and N. S. M. Raja., (2019). Comprehensive SVM based Indian Sign Language Recognition, IEEE International Conference on System, Computation, Automation and Networking (ICSCAN), Mar. 2019, pp. 1–4. doi: 10.1109/ICSCAN.2019.8878787.
11. K. Zhang, Y. Zhang, P. Wang, Y. Tian, and J. Yang., (2018). An Improved Sobel Edge Algorithm and FPGA Implementation, in Proceedings of the International Conference of Information and Communication Technology-2018, 2018, vol. 131, pp. 243–248. doi: https://doi.org/10.1016/j.procs.2018.04.209.
12. M. A. Bencherif et al., (2021). Arabic Sign Language Recognition System Using 2D Hands and Body Skeleton Data, IEEE Access, vol. 9, pp. 59612–59627, 2021, doi: 10.1109/ACCESS.2021.3069714.
13. M. Long, H. Zhu, J. Wang, and M. I. Jordan., (2016). Unsupervised domain adaptation with residual transfer networks, in Proc. Adv. Neural Inf. Process. Syst. (NIPS), 2016, pp. 136–144.
14. M. Oquab, L. Bottou, I. Laptev, and J. Sivic., (2014). Learning and transferring mid-level image representations using convolutional neural networks, in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Columbus, OH, USA, pp. 1717–1724.
15. M. Zadghorban and M. Nahvi., (2018). An algorithm on sign words extraction and recognition of continuous Persian sign language based on motion and shape features of hands, Pattern Anal. Appl., vol. 21, no. 2, Art. no. 2, May 2018, doi: 10.1007/s10044-016-0579-2.
16. Mustafa, A. R., Ramadany, S., Sanusi, Y., Made, S., Stang, S., & Syarif, S. (2020). Learning media applications for toddler midwifery care about android-based fine motor development in improving midwifery students skills. International Journal of Health & Medical Sciences, 3(1), 130-135. https://doi.org/10.31295/ijhms.v3n1.290

17. P. PAUDYAL, J. LEE, A. BANERJEE, and S. K. S. GUPTA.,(2019). A Comparison of Techniques for Sign Language Alphabet Recognition Using Armband Wearables, ACM Trans. Interact. Intell. Syst., vol. 9, no. 2–3, pp. 1–26, Apr. 2019.
18. P. Prathusha, S. Jyothi, and D. M. Mamatha., (2018). Enhanced Image Edge Detection Methods for Crab Species Identification, International Conference on Soft-computing and Network Security (ICSNS), Feb. 2018, pp. 1–7. doi: 10.1109/ICSNS.2018.8573629.
19. P. V. V. Kishore, D. A. Kumar, A. S. C. S. Sastry, and E. K. Kumar., (2018). Motionlets Matching With Adaptive Kernels for 3-D Indian Sign Language Recognition, IEEE Sens. J., vol. 18, no. 8, pp. 3327–3337, Apr. 2018, doi: 10.1109/JSEN.2018.2810449.
20. R. Cui, H. Liu, and C. Zhang., (2017). Recurrent Convolutional Neural Networks for Continuous Sign Language Recognition by Staged Optimization, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Jul. 2017, pp. 1610–1618. doi: 10.1109/CVPR.2017.175.
21. R. Cui, H. Liu, and C. Zhang., (2019). A Deep Neural Framework for Continuous Sign Language Recognition by Iterative Training, IEEE Trans. Multimed., vol. 21, no. 7, Art. no. 7, Jul. 2019, doi: 10.1109/TMM.2018.2889563.
22. R. Rastgoo, K. Kiani, and S. Escalera., (2020). Hand sign language recognition using multi-view hand skeleton, Expert Syst. Appl., vol. 150, p. 113336, Jul. 2020, doi: 10.1016/j.eswa.2020.113336.
23. S. J. Pan and Q. Yang., (2010). A survey on transfer learning, IEEE Trans. Knowl. Data Eng., vol. 22, no. 10, pp. 1345– 1359, Oct. 2010
24. S. Ji, W. Xu, M. Yang, and K. Yu., (2013). 3D Convolutional Neural Networks for Human Action Recognition," IEEE Trans. Pattern Anal. Mach. Intell., vol. 35, no. 1, Art. no. 1, doi: 10.1109/TPAMI.2012.59.
25. Suryasa, I. W., Rodríguez-Gámez, M., & Koldoris, T. (2022). Post-pandemic health and its sustainability: Educational situation. International Journal of Health Sciences, 6(1), i-v. https://doi.org/10.53730/ijhs.v6n1.5949
26. T. Fujimoto, T. Kawasaki, and K. Kitamura.,(2019). Canny-Edge-Detection/Rankine-Hugoniot-conditions unified shock sensor for inviscid and viscous flows | Elsevier Enhanced Reader, vol. 396, pp. 264–279, Nov. 2019, doi: 10.1016/j.jcp.2019.06.071.
27. T. Pariwat and P. Seresangtakul., (2017). Thai finger-spelling sign language recognition using global and local features with SVM, 9th International Conference on Knowledge and Smart Technology (KST), Feb. 2017, pp. 116–120. doi: 10.1109/KST.2017.7886111.
28. W. Aly, S. Aly, and S. Almotairi, (2019). User-Independent American Sign Language Alphabet Recognition Based on Depth Image and PCANet Features," IEEE Access, vol. 7, pp. 123138–123150, 2019, doi: 10.1109/ACCESS.2019.2938829.
29. Y. Dong, J. Liu, and W. Yan., (2021). Dynamic Hand Gesture Recognition Based on Signals from Specialized Data Glove and Deep Learning Algorithms, IEEE Trans. Instrum. Meas., vol. 70, pp. 1–14, 2021, doi: 10.1109/TIM.2021.3077967.
30. Y. Liao, P. Xiong, W. Min, W. Min and J. Lu., (2019). Dynamic Sign Language Recognition Based on Video Sequence with BLSTM-3D Residual Networks,

in *IEEE Access*, vol. 7, pp. 38044-38054, 2019, doi: 10.1109/ACCESS.2019.2904749.

31. Y. Wang, C. Wang, H. Zhang, Y. Dong, and S. Wei., (2019). A SAR dataset of ship detection for deep learning under complex backgrounds, Remote Sens., vol. 11, no. 7, p. 765, 2019.