Video-Based Sign Language Digit Recognition for the Thai Language: A New Dataset and Method Comparisons

Wuttichai Vijitkunsawat[a], Teeradaj Racharak[b], Chau Nguyen[c] and Nguyen Le Minh[d]

Japan Advanced Institute of Science and Technology, Nomi, Ishikawa, Japan

Keywords: Thai Sign Language, Sign Language Recognition, Benchmark Dataset.

Abstract: Video-based sign language recognition aims to support deaf people, so they can communicate with others by assisting them to recognise signs from video input. Unfortunately, most existing sign language datasets are limited to a small vocabulary, especially in low-resource languages such as Thai. Recent research in the Thai community has mostly paid attention to building recognisers from static input with limited datasets, making it difficult to train machine learning models for practical applications. To overcome this limitation, this paper originally introduces a new video database for automatic sign language recognition for Thai sign language digits. Our dataset has about 63 videos for each of the nine digits and is performed by 21 signers. Preliminary baseline results for this new dataset are presented under extensive experiments. Indeed, we implement four deep-learning-based architectures: CNN-Mode, CNN-LSTM, VGG-Mode, and VGG-LSTM, and compare their performances under two scenarios: (1) the whole body pose with backgrounds, and (2) hand-cropped images only as pre-processing. The results show that VGG-LSTM with pre-processing has the best accuracy for our in-sample and out-of-sample test datasets.

1 INTRODUCTION

Over 5% of the world's population – or about 450 million people worldwide – require rehabilitation to address their 'disabling' hearing loss, as reported by (World Health Organization, 2021). The use of hearing assistive technologies such as sign language interpretation can further improve access to communication and education for people with hearing loss. However, many people with normal hearing cannot understand sign language. Moreover, most countries have developed their sign languages because they have different cultures, alphabets and vowels. This fact may also create a barrier for promoting the development of assistive sign language interpreters.

According to the current progress of global sign language recognition research, there are five main aspects that must be considered when working with sign language recognition in deep learning: feature fusion, input modality, training dataset, language complexity, and deep models (Rastgoo et al., 2021). Firstly, feature fusions can be organised into three categories: using only hand pose features, using both hands and face pose features, and using the body, hand and face pose features to enhance the accuracy of the sign language system (Chen et al., 2020; Doosti, 2019; Wang et al., 2018). Secondly, the input modality can be divided into gloved-based and vision-based. The glove-based model uses an electronic circuit and sensors attached to a glove to send signal data for hand pose detection. In another way, vision-based modalities like RGB, depth, thermal, and skeleton offer a more realistic and natural system based on data humans can sense from their environment (Zheng et al., 2017; Kim et al., 2017). Thirdly, Sign Language Recognition (SLR) models have various languages in the input data, such as American Sign Language (ASL) (Pugeault and Bowden, 2011), Indian Sign Language (ISL), (Forster et al., 2014), Boston ASL, and so on. They have garnered more attention due to more popularity and usage. However, understanding sign language requires very precise domain knowledge, and it is not feasible to try to label many samples per class (Li et al., 2020). Next, language complexity determines some grammatical rules to connect the movements of the face, hands, and body parts because of several parameters such as eyelashes, eye gaze, eyebrows, orientation, shape, and mouth parameters. Finally, deep models are used for the automatic index-

[a] https://orcid.org/0000-0003-2157-7661
[b] https://orcid.org/0000-0002-8823-2361
[c] https://orcid.org/0000-0003-0068-0387
[d] https://orcid.org/0000-0002-2265-1010

Vijitkunsawat, W., Racharak, T., Nguyen, C. and Minh, N.
Video-Based Sign Language Digit Recognition for the Thai Language: A New Dataset and Method Comparisons.
DOI: 10.5220/0011643700003411
In *Proceedings of the 12th International Conference on Pattern Recognition Applications and Methods (ICPRAM 2023)*, pages 775-782
ISBN: 978-989-758-626-2; ISSN: 2184-4313

ing of signed videos, pose estimation, multi-person, hand detection, and other interactions between humans and computer applications (Newell et al., 2016).

Regarding deep applications to SLR, most works have used Convolution Neural Network (CNN) with other deep learning architectures, such as the Recurrent Neural Network (RNN), to increase performance when dealing with the video input than the only CNN. Although CNN and RNN models as well as their combinations were designed long ago, most researchers, as studied by (Rastgoo et al., 2021), have continued to use them in SLR, but only a few changes in the used modalities and datasets. For example, having achieved high training accuracy on ISL, (Wadhawan and Kumar, 2020) proposed static signs in sign language recognition using CNN on RGB images. In addition, (Ferreira et al., 2019) presented multi-modal learning techniques from three specific modalities for an accurate SLR, using colour, depth on Kinect, and Leap Motion data based on CNN.

Our contributions are twofold. First, we originally introduce a new video database for Thai sign language recognition on digits. Our dataset has about 63 videos for each digit and is performed by 21 signers. To our knowledge, this is the first video dataset for the Thai sign language research community. Second, we conduct a substantive study on the design and development of deep learning systems based on our dataset. Specifically, we implement and investigate four systems: CNN-Mode, CNN-LSTM, VGG-Mode, and VGG-LSTM, and compare their performances under two scenarios: (1) the whole body pose with backgrounds, and (2) hand-cropped images only as pre-processing. The paper is structured as follows. Section II describes related work on the Thai sign language (TSL) datasets that currently exist in the Thai research community. Next, we explain our dataset, methodology, and pre-processing steps in section III. Section IV discusses the steps and results of our experiments. Finally, we conclude the experiments and discuss the direction of future work in section V.

2 RELATED WORK

In this section, we briefly discuss some of the Thai sign language datasets that exist at present. According to the situation of persons with disabilities in Thailand, there were 393,027 people, or 18.69%, with a hearing impairment and interpretive disability in December 2021[1], representing the second leading disability type among all 2,102,384 disabled people. This problem causes difficult communication between those who can hear and the groups of deaf and hard-hearing people who communicate with sign language, a subset of hand gestures. Although Thai Sign Language (TSL) was initially developed from American Sign Language (ASL), it has distinct hand gestures from other countries based on tradition, culture, and geography. The structure of TSL consists of 5 parts: the hand shapes, position of the hands, movement of the hands, orientation of the palms in relationship to the body or each other, and face of the signer. Even though TSL is the only standard sign language in Thailand, it still lacks public sign language datasets and signers. As a result, most Thai researchers have to provide datasets on their own without experts' involvement (see Tables 2 and 3).

Furthermore, TSL can be split into two major directions: fingerspelling and natural sign language. Fingerspelling is used for specific names such as places, people, and objects that cannot be signed using gestures. (Chansri and Srinonchat, 2016) proposed investigating the hand position in real-time situations with Kinect sensors but without the environmental contexts, such as the skin colour and background. (Pariwat and Seresangtakul, 2017) presented an example of a system based on Thai fingerspelling using global and local features with Support-Vector Machine. At the same time, (Nakjai and Katanyukul, 2019) employed a histogram of Oriented Gradients (HOG) with CNN to deal with Thai fingerspelling.

Despite the aforementioned, most deaf and hard-of-hearing people use the natural Thai sign language to communicate with each other because it is easy and fast. However, a significant problem with natural signs is that the number of Thai sign language datasets is very low. For example, (Chaikaew et al., 2021) prepared their dataset by using five gestures and shot 100 videos per word, so the total was 500 videos containing 50 FPS with H.264 format for each video. Then, input datasets were trained with RNN-based models: LSTM, BiLSTM, and GRU. Although their results demonstrated greater than 90% accuracy, they presented only in-sample evaluation. Undoubtedly, the in-sample domain is higher than the out–of-sample evaluation. Next, (Chaikaew, 2022) applied the holistic landmark API of MediaPipe to extract features from live video capture consisting of face, hand and body landmarks. Afterwards, they trained their data on three models to evaluate the performance of each model. However, neither research paper showed the number of signers. Generally, a good sign recognition model should be robust to inter-signer variations in the input data, such as signing paces and signer appearance, to generalise well to real-world scenarios.

[1] https://dep.go.th/th/

Figure 1: Examples of Thai digit number datasets.

3 METHODS AND DATASET

In this section, we explain the dataset and methods for our processes. Firstly, we describe the Thai digit number dataset, including how to calculate the average videos step by step, followed by the model architecture comprised of four crucial deep-learning models, including CNN-Mode, CNN-LSTM, VGG-Mode, and VGG-LSTM to compare the performance of each model. Next, we explain our application on the YOLOv5 model to detect hand only as our preprocessing from video inputs. We discuss our implementation including parameters used by YOLOv5.

3.1 Thai Digit Number Dataset

The digit number (1-9) dataset used in this study was acquired from two main sources: the Internet and persons, by controlling the deaf person experts. First, there are multiple educational sign language websites, including the Office of the Royal Society[2] and the National Association of the Deaf in Thailand[3]. Another main source was videos from the general public. However, experts controlled all the processing of sign poses. Finally, we selected videos whose titles clearly describe the words of the sign.

In total, we acquire 567 videos consisting of 540 videos for the in-sample test set and 27 videos for the out-of-sample test set, and the length of each video is 2-4 seconds varied by sign language gesture. There are 21 signers performed in all the videos, including 15 women and 6 men, as illustrated in Figure 1.

After the collection of in-sample videos, we calculate the length of all videos to be 27.08 minutes (1,628 secs). Hence, the average length per video is

$$\text{average length per video} = \frac{\text{length of all videos}}{\text{number of videos}} \quad (1)$$

$$= \frac{1,628}{540} \approx 3.015 \text{ sec}$$

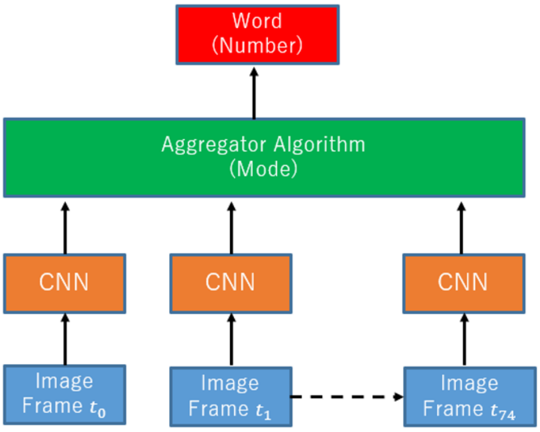

Figure 2: CNN-Mode for digit classification from a video.

Next, we convert all the videos into image frames at 25 fps following the Phase Alternating Line theory (PAL system). Thus, we have $25 \times 3.015 = 75.375 \approx 75$ frames per video or $75 \times 540 = 40,500$ image frames for all videos. However, the sizes of image frames are in different scales, so we resize all image frames to be 96×96 to feed the input of models.

3.2 Model Architecture

We implement four deep learning-based system: CNN-Mode, CNN-LSTM, VGG-Mode and VGG-LSTM, and evaluate their performance based on the collected dataset. Each deep system is investigated under two scenarios, i.e., (1) whole body poses with background and (2) only hand-cropped poses, to find out the best design of deep learning-based systems.

3.2.1 CNN-Mode

2D CNNs are widely used to extract spatial features of input images. Considering that a video input is a sequence of image data, our first implementation uses a CNN model (given in Table 1) to determine the class for each image input. The output represents the posterior probability that the input represents a digit. Each predicted output is aggregated by the statistical mode operator. This deep architecture is referred to as *CNN-Mode* and is illustrated in Figure 2. Note that the mode is the most commonly observed value in a set of data. The outputs of the softmax layer are calculated to determine the most occurred digit in a video input as an output prediction from the video, as shown in Equation (2). In the equation, x_i represents the predicted digit at frame i (there are 75 frames per video).

$$\text{Word} \leftarrow Mode(CNN_{i \in \{1,\ldots,75\}}(x_i)) \quad (2)$$

[2] http://164.115.33.116/vocab/index.html

[3] https://www.th-sl.com/search-by-act/

Table 1: The CNN model used in CNN-Mode and CNN-LSTM architectures.

Layer	**Filters**	**Kernel Size**	**Strides**	**Activate Function**	**Neural Units**
Conv2D	8	3×3		ReLU	
MaxPooling2D			2		
Conv2D	16	3×3		ReLU	
MaxPooling2D			2		
Conv2D	32	3×3		ReLU	
MaxPooling2D			2		
Conv2D	64	3×3		ReLU	
MaxPooling2D			2		
Fully Connected 1	-	-	-	ReLU	256
Fully Connected 2	-	-	-	ReLU	84
Fully Connected 3	-	-	-	softmax	9

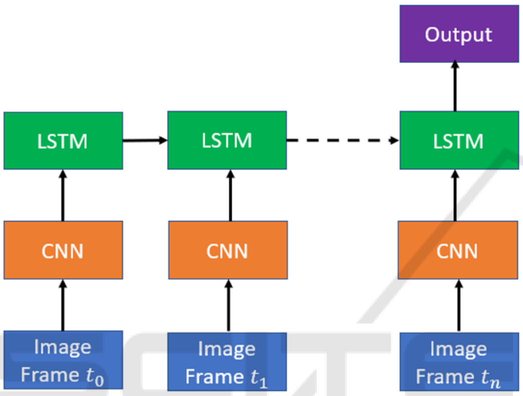

Figure 3: CNN-LSTM fordigit classification from a video.

3.2.2 CNN-LSTM

Recall that Recurrent Neural Networks (RNN) and its variants e.g. LSTM are employed to capture the long-term temporal dependencies among inputs. Thus, our next architecture is constructed by a CNN and a LSTM to capture spatio-temporal features from input video frames. In particular, the CNN extracts features from each frame, and the LSTM aggregates the information over time. Finally, two consecutive fully-connected layers (256 and 84 units with ReLU activations) and a softmax layer are utilized to obtain final classification scores. This architecture is referred to as *CNN-LSTM* as shown in Figure 3. Table 1 details the architecture of the CNN model used in CNN-LSTM. The size of LSTM cell is set to 30 and the number of the stacked recurrent layers in LSTM is set to 1.

It is worth mentioning that CNN-LSTMs are often employed for visual time series prediction and generating textual descriptions from video inputs (Brownlee, 2017). This work also investigates the utilization of this architecture for Thai sign language from video on our collected dataset.

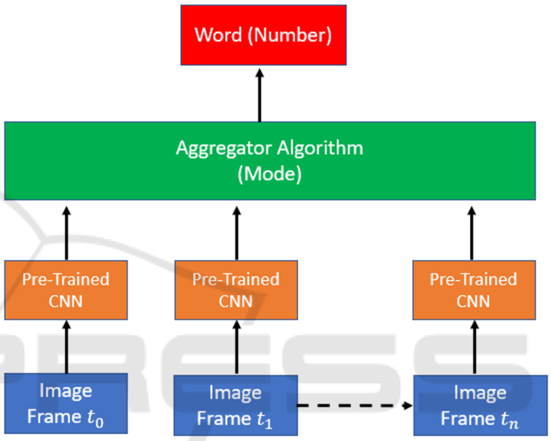

Figure 4: VGG-Mode for digit classification from a video.

3.2.3 VGG-Mode and VGG-LSTM

Both CNN-Mode and CNN-LSTM are trained from scratch. It is natural to further investigate the utilization of state-of-the-art architectures on the collected dataset. Here, we use VGG16 (Simonyan and Zisserman, 2014) pre-trained on ImageNet to extract spatial features and then feed the extracted features to the statistical mode operator, called *VGG-Mode* (cf. Figure 4), and LSTM, called *VGG-LSTM* (cf. Figure 5). The LSTM part here is also set the same as CNN-LSTM. Note that the size of an input image for the pre-trained VGG16 is set to $96 \times 96 \times 3$ – not $224 \times 224 \times 3$ as used in the original VGG16 work.

3.3 Training a YOLOv5 Model for Human Hand Recognition

In the object detection task, YOLO series (Redmon et al., 2016; Redmon and Farhadi, 2017; Redmon and Farhadi, 2018) play an important role in one-stage detectors. YOLO examines an image by dividing it into a grid of smaller parts and then performs

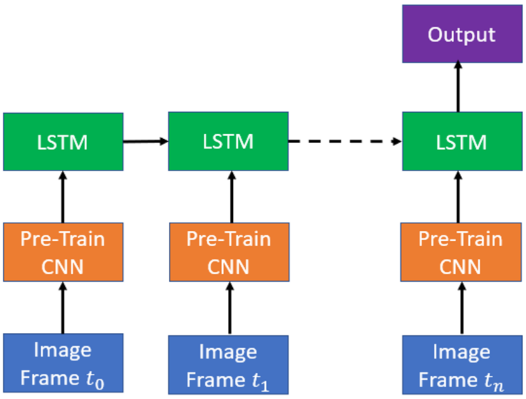

Figure 5: VGG-LSTM for digit classification from a video.

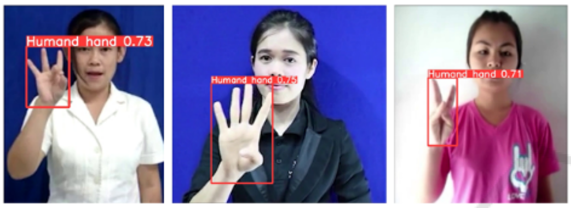

Figure 6: Cropping only hand by using YOLOv5.

object detection on them. By inspecting the image only once, YOLO models enable high-speed application to real-time object detection. YOLOv5 (Jocher et al., 2021) inherits the characteristics of the procedures with more optimised speed and accuracy.

Considering that each video contains the whole body of a signer (see Figure 6a). We design a hand cropper as a pre-processer by implementing a human hand detector using YOLOv5. Here, we train a YOLOv5 model from scratch. First, we obtain images with annotations on human hands from Google Open Images Dataset V6 comprising 22,094 training images and 2,056 validation images. Then, we train the model for 90 epochs to reach a precision of 84.47% and a recall of 75.73% for the validation set. The trained model is then used for recognising the hands of the signers in the videos.

3.3.1 Cropping Hand Method

For cropping only a hand of the signer scenario, we provide the YOLOv5, set the default Intersection over Union (IoU), and the confidence threshold for cropping hand to 0.45 and 0.7, respectively. Examples of hand detection are illustrated in Figure 6. We use a 0.7 confidence threshold because of high fidelity hand motion capture at speed.

After cropping hands, we acquire 16,221 frames, which is a nearly fourfold decrease from the original. Then, we need to calculate the average of cropped-hand frames because the average number of frames for each pose changed.

$$\text{average cropping} = \frac{\text{all cropped-hand frames}}{\text{number of videos}} \qquad (3)$$

$$= \frac{16,221}{540} \approx 30 \text{ frames}$$

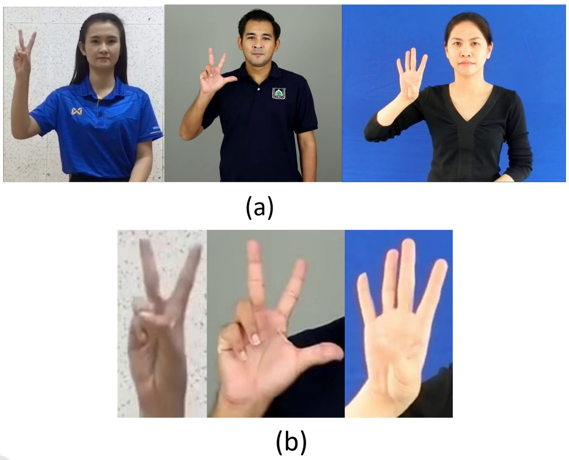

Figure 7: (a) whole body pose (b) cropping only hand.

Next, we continually use a normalisation method to standardise the input frames by creating dummy files and a technique for padding image frames because the sizes of frames are reduced after the only cropping hand process, as illustrated in Figure 7. The conditions for creating dummy files and padding image frames are as follows.

3.3.2 Padding and Resizing Images

The images are normalised to a size of 96 × 96 pixels using padding, resizing, and re-shaping techniques. On the condition that the size of the image frames is less than 96 × 96 pixels, it is necessary to make white padding on the edge of the image, as shown in Figures 8a and 8b. On the other hand, it is necessary to resize the scale to 96 × 96 pixels if the size is greater than 96 × 96 pixels.

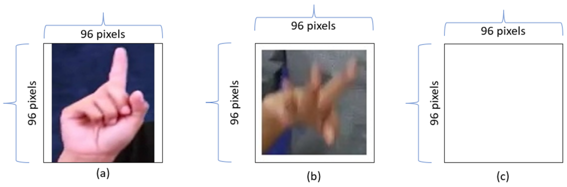

Figure 8: (a) Vertical padding image, (b) Both vertical and horizontal padding images and (c) dummy file.

3.3.3 Dummy Files and Random Images

For the dummy file and randomly selected images condition, we must add the dummy file (white image) to 30 frames provided the number of images is

Table 2: Comparisons of Thai sign language datasets with static images.

References	Words	Images	Mean	Signers
(Chansri and Srinonchat, 2016)	16	320	20	unknown
(Pariwat and Seresangtakul, 2017)	15	75	5	5
(Nakjai and Katanyukul, 2019)	25	125	5	11

lower than 30 frames, as shown in Figure 8c. If the number of images is higher than 30 frames, however, we have to use the frame-down sampling technique by randomly selecting only 30 frames sorted by the sequence of hand pose movement to standardise the quality of diverse frames and decrease the computational requirement.

4 EXPERIMENTS AND RESULTS

In this section, we detail the evaluation of the proposed architectures for Thai sign language on the collected video dataset. We use an Intel(R) Core i7, 2.9 GHz with 64 GB of RAM; all models are created using TensorFlow and Keras version 2.8.0 for all experiments. Furthermore, the models are trained on a GPU NVIDIA RTX-3090 with 24 GB memory.

In the first step of our experiment, we set up each deep-learning model as described in Section 2. Then, the dataset was split 6:2:2 into training, test and validation sets. Subsequently, we set the CNN parameters as described in Table 1 and the training parameters as described in Table 4; other hyper-parameters are set as default in the original models. Table 5 illustrates the evaluation performance of each model under two scenarios: the whole body with background (denoted by $+$), and the only hand-cropped (denoted by $*$). The evaluation table comprises total parameters and training accuracy, as well as in-sample and out-of-sample evaluation. The in-sample evaluation is the data from the test set frames, which split data from the previous process (540 videos from eighteen signers). The out-of-sample evaluation is the other data (27 videos from three signers), not the input dataset.

According to the total parameter data, the models with LSTM use the parameters more than the mode operator due to its algorithm and architecture's complexity. However, it can be seen that while the mode operator uses low parameters, the training accuracy is high at nearly 96%, higher than LSTM on the only hand-cropped scenario. Also, the training accuracy for the mode rises dramatically compared to LSTM on whole body pose conditions.

Evaluation metrics for both in-sample and out-of-sample test sets are accuracy, precision, recall, and F1-score. From the training accuracy on each model and focused scenario, it can be observed that higher training accuracy results in a greater F1-score on the in-sample evaluation. In addition, considering the out-of-sample evaluation, the CNN-Mode* has the lowest number of parameters compared to other models, but it is fairly suitable on the F1-score. However, the VGG-LSTM* is the best model for Thai sign language if we would like to get the highest performance because the accuracy and F1-score are 81.25% and 85.21%, respectively.

5 DISCUSSION AND CONCLUSION

This paper originally introduces a video-based Thai signed digit language dataset and conducts extensive experiments on various deep learning-based architectures, namely CNN-Mode, VGG-Mode, CNN-LSTM and VGG-LSTM under two different scenarios: the whole body and the only hand-cropped. From the experiment, many models may get high percentages for training accuracy and the in-sample evaluation. However, they cannot guarantee the out-of-sample evaluations (cf. CNN-Mode$^+$ and VGG-Mode$^+$ from Table 5). The VGG-LSTM* has the highest efficiency for both in-sample and out-of-sample test sets.

In the future, we plan to collect more Thai sign language words, both Thai fingerspelling and isolated Thai signed language, to cover more fundamental vocabularies sufficient for the communication with deaf people. Moreover, we aim to reduce the total number of parameters in the model for easier installation on AI embedded boards to facilitate communication between normal-hearing people and deaf people.

ACKNOWLEDGEMENTS

This research has been partially supported by the Setsatian School for the Deaf under the Royal Patronage of His Royal Highness Crown Prince Maha Vajiralongkorn and the Thungmahamek School for the Deaf.

Table 3: Comparisons of Thai sign language datasets with real-time videos.

References	Words	Videos	Mean	Signers
(Chaikaew et al., 2021)	5	500	100	unknown
(Chaikaew, 2022)	15	900	60	unknown
Our Dataset	**9**	**567**	**63**	**21**

Table 4: The parameter and hyper-parameters used by each implemented model.

Model	Batch size	Learning rate	Dropout	Epochs	Optimizer	Early Stopping	LSTM Cell
CNN-Mode$^+$	32	0.0001	0.2	70	Adam	5	-
VGG-Mode$^+$	32	0.0001	0.2	70	Adam	5	-
CNN-LSTM$^+$	16	0.00001	0.1	50	Adam	5	30
VGG-LSTM$^+$	16	0.00001	0.1	50	Adam	5	30
CNN-Mode*	32	0.0001	0.2	70	Adam	5	-
VGG-Mode*	32	0.0001	0.2	70	Adam	5	-
CNN-LSTM*	16	0.00001	0.1	50	Adam	5	30
VGG-LSTM*	16	0.00001	0.1	50	Adam	5	30

Table 5: Evaluation metrics for each implemented model.

Model	Total Parameter	Training Accuracy (%)	In-sample Evaluation (%)				Out-sample Evaluation (%)			
			Accuracy	Precision	Recall	F1 Score	Accuracy	Precision	Recall	F1 Score
CNN-Mode$^+$	122,377	96.53	65.22	64.71	63.27	63.98	19.25	17.12	18.5	17.78
VGG-Mode$^+$	15,916,945	97.24	83.59	81.08	79.79	80.43	27.25	25.3	20.2	22.46
CNN-LSTM$^+$	129,249	65.12	23.55	22.86	21.42	22.11	18.5	16.36	15.76	16.05
VGG-LSTM$^+$	15,301,657	74.81	46.48	24.96	38.88	30.4	23.25	18.66	21.42	19.94
CNN-Mode*	122,377	97.59	71.14	68.49	71.11	69.77	64.25	64.66	66.28	65.45
VGG-Mode*	15,916,945	99.83	89.81	87.8	84.81	86.27	66.72	72.22	91.66	80.78
CNN-LSTM*	129,249	98.45	88.58	71.71	79.52	80.59	62.5	59.79	58.57	59.17
VGG-LSTM*	***15,301,657***	***99.93***	***93.51***	***94.06***	***93.51***	***93.78***	***81.25***	***89.58***	***81.25***	***85.21***

REFERENCES

Brownlee, J. (2017). *Long short-term memory networks with python: develop sequence prediction models with deep learning*. Machine Learning Mastery.

Chaikaew, A. (2022). An applied holistic landmark with deep learning for thai sign language recognition. In *2022 37th International Technical Conference on Circuits/Systems, Computers and Communications (ITC-CSCC)*, pages 1046–1049.

Chaikaew, A., Somkuan, K., and Yuyen, T. (2021). Thai sign language recognition: an application of deep neural network. In *2021 Joint International Conference on Digital Arts, Media and Technology with ECTI Northern Section Conference on Electrical, Electronics, Computer and Telecommunication Engineering*, pages 128–131. IEEE.

Chansri, C. and Srinonchat, J. (2016). 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.

Chen, X., Wang, G., Guo, H., and Zhang, C. (2020). Pose guided structured region ensemble network for cascaded hand pose estimation. *Neurocomputing*, 395:138–149.

Doosti, B. (2019). Hand pose estimation: A survey. *arXiv preprint arXiv:1903.01013*.

Ferreira, P. M., Cardoso, J. S., and Rebelo, A. (2019). On the role of multimodal learning in the recognition of sign language. *Multimedia Tools and Applications*, 78(8):10035–10056.

Forster, J., Schmidt, C., Koller, O., Bellgardt, M., and Ney, H. (2014). Extensions of the sign language recognition and translation corpus rwth-phoenix-weather. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 1911–1916.

Jocher, G. et al. (2021). ultralytics/yolov5: v5. 0-yolov5-p6 1280 models, aws, supervise. ly and youtube integrations, april 2021. *DOI: https://doi.org/10.5281/zenodo*, 4679653.

Kim, S., Ban, Y., and Lee, S. (2017). Tracking and classification of in-air hand gesture based on thermal guided joint filter. *Sensors*, 17(1):166.

Li, D., Rodriguez, C., Yu, X., and Li, H. (2020). 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.

Nakjai, P. and Katanyukul, T. (2019). Hand sign recognition for thai finger spelling: An application of convolution neural network. *Journal of Signal Processing Systems*, 91(2):131–146.

Newell, A., Yang, K., and Deng, J. (2016). Stacked hourglass networks for human pose estimation. In *European conference on computer vision*, pages 483–499. Springer.

Pariwat, T. and Seresangtakul, P. (2017). Thai fingerspelling 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.

Pugeault, N. and Bowden, R. (2011). Spelling it out: Real-time asl fingerspelling recognition. In *2011 IEEE International conference on computer vision workshops (ICCV workshops)*, pages 1114–1119. IEEE.

Rastgoo, R., Kiani, K., and Escalera, S. (2021). Sign language recognition: A deep survey. *Expert Systems with Applications*, 164:113794.

Redmon, J., Divvala, S., Girshick, R., and Farhadi, A. (2016). You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788.

Redmon, J. and Farhadi, A. (2017). Yolo9000: better, faster, stronger. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7263–7271.

Redmon, J. and Farhadi, A. (2018). Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*.

Simonyan, K. and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Wadhawan, A. and Kumar, P. (2020). Deep learning-based sign language recognition system for static signs. *Neural computing and applications*, 32(12):7957–7968.

Wang, M., Chen, X., Liu, W., Qian, C., Lin, L., and Ma, L. (2018). Drpose3d: Depth ranking in 3d human pose estimation. *arXiv preprint arXiv:1805.08973*.

World Health Organization (2021). Deafness and hearing loss. https://www.who.int/news-room/fact-sheets/detail/deafness-and-hearing-loss.

Zheng, L., Liang, B., and Jiang, A. (2017). Recent advances of deep learning for sign language recognition. In *2017 International Conference on Digital Image Computing: Techniques and Applications (DICTA)*, pages 1–7. IEEE.