CJBPP
Cambodia Journal for Business and Professional Practice, Vol 1
ISSN: 3078-4131 (Print)
DOI: https://doi.org/10.71215/cjbpp.202412

Feature Extraction and Convolutional Neural Networks for Static Hand Gesture Recognition and Context Sentence Generation

Kimlong Ngin [1], Pakrigna Long [2] & Pichkhemara Morn [2]

CJBPP Vol. 1

Received: Jan 22, 2024
Revised: Feb 13, 2024
Accepted: Jul 09, 2024

Corresponding Author. Email: nginkimlong@gmail.com
1. University of Heng Samrin Thbongkhmum, Thbongkhmum Province, Cambodia
2. ACLEDA University of Business, Phnom Penh, Cambodia

ABSTRACT

Hand gestures based on human-computer interaction are both intuitive and versatile, with multiple and diverse applications including in smart homes, games, operating theaters and vehicle infotainment systems. This research presents a novel architecture by combining a convolutional neural network (CNN) and traditional feature extractors to examine the accuracy of static hand gesture recognition. This research provides three significant contributions. First, we use the Non-Dominated Sorting Genetic Algorithm II (NSGAII), an evolutionary algorithm to classify and select image features across five methods, including the Gabor filter, the Hu-moment, the Zernike moment, the Complex moment, and the Fourier moment. Experimental results demonstrated that the combination of the Gabor filter, the Hu moment, and the Zernike moment achieved the best result with an accuracy of 98.3% to 99.0%. The Zernike moment combined with the Hu-moment output had an accuracy of 95.5% to 98.0%. The second contribution proposes the use of the Multiple Feature Convolutional Neural Network (MFCNN) model to generate better image recognition through the combination of validation techniques and features descriptors. Extensive experimentation was conducted utilizing binary and grayscale, as well as two different validation techniques - the Holdout technique and the Cross-validation of leaving one subject out of the validation. The proposed architecture was evaluated on two dataset types and is compared with the state-of-the art convolutional neural networks (CNN). The Massey's dataset, contained 2,524 images and 36 gestures, and the OUHANDs dataset contained 3,000 images and 10 gestures. Experimental results demonstrated a high recognition rate using descriptors with low computational cost and reduced size. The third contribution is the sequence sentences generation based on the Beam Search (BS) algorithm. The data obtained from CNN/Daily

Mail documents and results of image recognition, i.e., the image's label, were used to test various question size with four different sizes of questions, including 100, 1,000, 10,000, and 40,000. The experimental results showed that our method could achieve high-quality sentence generation.

Keywords: *Feature extraction, convolutional neural networks, static hand gesture recognition, human-computer extraction, context sentence generation*

1. Introduction

Background

Human communication encompasses various elements that act as channels for conveying information. One of the most fundamental modes of communication in human-computer interaction is through hand gestures, which have been extensively explored in recent studies (Islam et al., 2019; Noble et al., 2023). In this context, static hand gestures are considered a basic human acquisition. The hand gestures can be used as an effective and natural human-machine interaction (Sharma et al., 2018). Static hand gestures can also be used in order to reduce work complexity, increase production, and save on cost and time (Pisharady & Saerbeck, 2015; Tairych et al., 2016). There are many studies aimed at identifying solutions to match specific requirements, and as human-computer interactions gain more adoption and become less expensive, there is a demand for a high rate of accuracy (Premaratne et al., 2010). Several studies applied common features of hand image recognition to complement a convolutional neural network (CNN) model (Yu et al., 2018). According to Schmidhuber (2015), Ciresan et al. (2012), and Yu et al. (2017), the CNN model is one of the best neural network models which uses small common features and yields accurate results. However, the segmentation and extraction of a high-quality set of common hand gesture features using effective extraction methods present several challenges. Various studies have employed different feature extraction methods, such as histograms of oriented gradient (HOG) and data augmentation, before feeding these features into CNN models (Eid & Schwenker, 2023; Özerdem & Bamwenda, 2019). The determination of common features and their extraction is a key element that must first be accomplished due to the significance of the CNN model used to recognize the objects without expending expensive computing resources and still achieve good results. In other words, the key point of hand gesture recognition is based on the pre-processing solution to standardize the objects.

Research problem and objectives

Many investigations have been conducted on the techniques of static hand gesture recognition. In addition, the applied models performed well in training and testing, generating an accuracy of over 90 percent. The ability to further increase the performance of these approaches is a significant area of ongoing research (Al Omari, F., Zitar, R. A., & Al-Jarrah, O. 2009).

Our research applies powerful methods to sustain data quality throughout the measuring methods while loading data into the recognition model. High-quality data and robust recognition models can ensure smooth application performance and may reduce costs and execution time. The purpose of this research is to propose a novel model of multiple features convolutional neural network (MFCNN) for fast and accurate static hand gesture recognition. A multi-objective genetic algorithm, the Non-Dominated Sorting Genetic Algorithm II (NSGAII), has been designed to improve accuracy by minimizing the numbers of features and simplifying connections in an artificial neural network (ANN). Extensive experimentation was conducted with three main datasets: the Massey dataset, OUHANDS dataset and CNN/Daily Mail dataset. The features were extracted from the datasets using specialized methods, such as the Hu-moment, the Zernike moment, the Complex moment, the Gabor filter, and the Fourier moment. Additionally, the generation of sequence sentences was studied and presented in the research. To generate the sentences, a beam search algorithm was applied to generate labels on each image which were obtained from the recognition model.

2. Literature Review

While many earlier studies investigated static hand gesture recognition, we focus on more recent papers and journal articles published between 2019 and 2023.

Islam et al. (2019) presented a static hand gesture recognition model utilizing CNN and data augmentation techniques. A dataset of 8,000 images was used for training, and the remaining 1,600 images were used for testing. Ten input variables were selected in the process. The model achieved an accuracy of 97.12%.

Noble et al. (2023) proposed capacitive sensing, a technique to capture hand gestures by using sensors. To facilitate high accuracy, a dataset with five gestures coupled with five

algorithms were used for the training and testing processes. The results showed that multi-layer perceptron neural network (MLP) achieved the best output with an accuracy of 96.87% and an F1 score of 92.16%.

Akintola & Emmanuel (2020) measured static hand gesture recognition, using a three-phase process across datasets consisting of 25 hand gestures (Akintola & Emmanuel, 2020). Twenty types of hand gestures were initially extracted by using segmentation and morphological techniques. Five hundred images of the 20 extracted hand gesture types were then used for training. Lastly, another 500 images were used for testing. In the training and testing processes, multi-layer neural networks were applied. The results demonstrated that the performance of the approach attained an accuracy of 96.4%.

According to Pinto et al. (2019), CNN was applied to detect and recognize static hand gestures. In their study, in addition to the model, several preprocessing operations were performed to select the best features for training and testing. The preprocessing approaches included morphology, contour, polygon, and segmentation of the datasets. Using the proposed algorithm, Pinto et al. (2019) achieved an accuracy of up to 96%.

Özerdem and BAMWENDA (2019) proposed two models for static hand recognition, artificial neural networks (ANN) and support vector machines (SVM). In their study, a dataset of 24 hand gestures was implemented. Before performing the training and testing, histogram of oriented gradient (HOG) was employed. With the SVM and ANN, the system achieved an accuracy of 93.4% and 98.2%, respectively.

Eid and Schwenker (2023) used CNNs to detect and recognize static hand gestures with high performance. Their study employed a dataset containing images with complex backgrounds and varying hand sizes. Preprocessing steps included data augmentation and skin segmentation techniques to enhance model performance. The results demonstrated a testing accuracy of up to 96.5%.

Even though recent research has achieved great progress in the recognition of static hand gestures utilizing a variety of methods, including CNNs, SVMs, and neural networks, further investigation is still required in a number of crucial areas. Numerous current methods concentrate on certain datasets or small subsets of hand gestures, which may make them difficult to apply in a variety of real-world contexts. Furthermore, strong performance remains challenging in contexts with diverse hand sizes, lighting conditions, and complicated

backgrounds. By putting forth an integrated framework that makes use of deep learning models and sophisticated feature extraction techniques, this study seeks to close these gaps. Our specific goals are to increase recognition precision, strengthen model resilience to changes in the environment, and investigate useful applications in domains like assistive technology and human-computer interaction.

3. Research Methodology

Feature classification

In our study, we designed a three-layer perceptron system to classify features of static hand gestures. We integrated the NSGAII algorithm to optimize the classification of image features using multiple inputs. Our model architecture included *l* inputs, with *n* neurons customized in the hidden layer. The hidden layer processes these inputs to produce *m* output results.

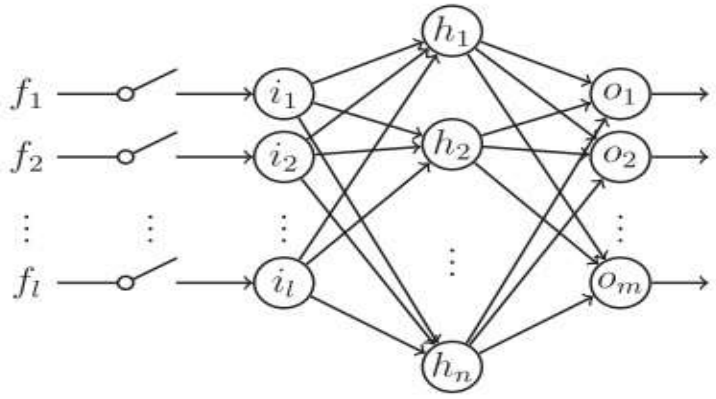


Figure 1: Features configuration showing the input variables in MLP and hidden layer

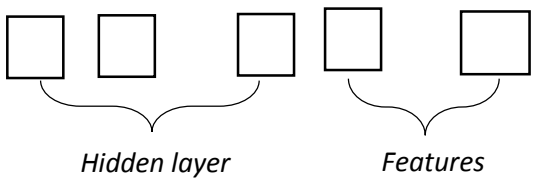


Figure 2: Configuration of NSGAII chromosome binary

To optimize the MFCNN model, evolutionary algorithms utilize "chromosomes" composed of parameters derived from the hidden layer of the neural network along with $l + 8$ inputs from 9 values, as illustrated in Figure 3. The MFCNN model is configured with specific parameters to enhance its performance in solving the given problem.

- Define an array of l switches, l switches value will keep in an array, and connected between the image features vector f_k with the input network's layer i_k. The arrangement of the switches defines the number of active neurons in the input layer.

- The neuron number in the network hidden layer *(n)*, the neuron parameters are applied to treat the number of the connection within the network.

Each chromosome can determine the *(n)* neurons number in the network's hidden layer (9 bits) and the useable features *(l bits)*, that corresponded to the switches array, as illustrated in Figure 2.

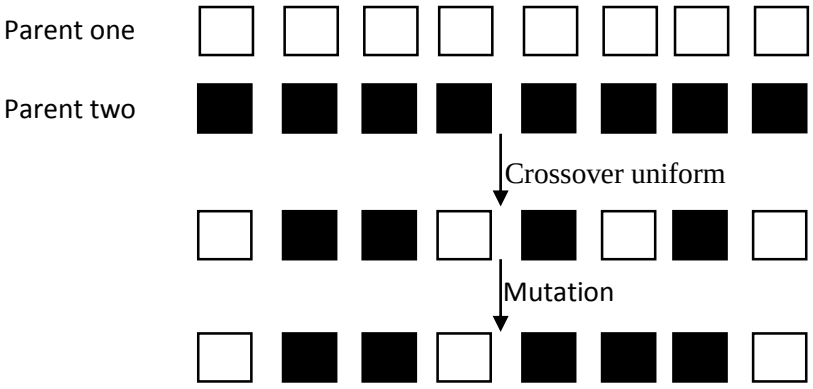


Figure 3: The process of genetic algorithm. Both parents will do the crossover uniform and then mutation processing

Genetic operators are used to define a new population Q_t which is generated from the parent's populations, as demonstrated in Figure 3. In the crossover's platform, use a probability P_c for all population chromosomes p_t. Furthermore, the crossover uniform process involves combining two randomly selected chromosomes from the parent population p_t to create each new chromosome. However, if the operators are not applied, a chromosome will just do a simple processing by copying from population p_t to the new generation population Q_t. A bit-flip mutation

is then applied with probability p_m for each bit. Algorithm 1 demonstrates the processing of a genetic algorithm and dataset using.

Algorithm 1: Non-Dominated Sorting Genetic Algorithm II (NSGAII)

1: Image feature extracted from datasets

2: Randomly separate images from the dataset into both subset training and testing

3: The initialize genetic population P_0 of size N $t = 0$

5: while $t <$ *Maximum iterations* do

6: Q_t is the population of chromosomes with size *N* obtained from P_t through crossover and mutation

7: $R_t = P_t \cup Q_t$

8: for R_t chromosome do

9: Provide MLP and encoded the size of inputs layer and hidden layer

10: Classify the training set by using cross-validation

11: end for

12: Generate new size N of population P_{t+1} from R_t based on non-domination sorting and crowded-distance

13: t = t + 1

14: end while

15: Define and return the latest Pareto-front of the fitness populations.

For the dataset, this solution accuracy is evaluated by 5-fold cross-validation applied on the training set. We used k = 5 for k-fold cross-validation because it provides a good balance between bias and variance. Although k = 10 is often considered optimal, using k = 5 reduces computational cost and runtime while still offering reliable model performance estimation. This choice ensures efficient yet robust validation, allowing for sufficient training and validation data in each fold. We can denote that the cross-validation technique is executed in the optimization algorithm inner loop to evaluate a single chromosome performance. Then the last Pareto-front will be tested on the dataset testing subset after the optimization loop.

This technique is called a wrapper approach for feature selection. Commonly, we can observe that NSGAII constitute each feature is the same as in feature selections of other binary applying, where features *l* has encoded in a string of binary. In a condition, only two bits will show the status of features selected or unselected. If a bit has value "1" then the feature is selected, and a bit that has value "0" is unselected.

Moreover, the feature will not be presented in the initial population for most of the available features, like "111101111", or with very few features with solutions, like "111101111", thus restricting the capability of the algorithm search. To create a population with a uniform distribution while respecting the selected number of features, we need to perform uniform initialization. Algorithm 2 demonstrates the initialization of a population as a set of chromosomes.

Algorithm 2: Initialization of a Population as a Set of Binary Chromosomes

1: Define *n* number of bits' in a chromosome

2: Define *m* is the small number of features of a chromosome

3: for chromosome C_i do

4: C_i equal to binary n "0"

5: $r = max\ (m, i)$

6: Assign an array with generating r-value from 1 to n

7: for array *n* do

8: Give C_i equal to 0

9: end for

10: end for

11: Result

Multiple Features Convolutional Neural Networks (MFCNN)

Generally, a convolutional neural network (CNN) consists of a series of convolutional and subsampling layers, followed by a fully connected layer of a multilayer perceptron. The convolutional and subsampling layers' act as feature extraction methods, while the final fully connected layers are responsible for delivering the final recognition result.

In the context of hand gesture recognition, the combination of CNNs with multiple image features does not appear extensively in the literature. We question whether this specific approach has been studied before. While CNN architectures have been successfully applied to various domain problems, as demonstrated in Figure 4, the input data is typically a binary image or a single grayscale image. Images rescale to 32×32 dimensional pixels, and input gestures work as a channel of convolutional. The first convolution channel contains two layers that are concatenated with max-pooling layers. The number of layers has been adjusted experimentally.

Generally, two convolution layers perform better than three, four, or five for evaluation images. The same gesture is also executed at a Gabor filter and then at a second convolutional channel. The second channel has the same number of layers as the previous channel. The architecture has arranged outputs of the final max-pooling layers of both channels by concatenating into a one-dimensional vector of the feature. Additionally, during the training dataset, some channels' neurons from this single feature vector can be disabled.

In order to further increase the diversity of information, an additional set of fully-connected neurons obtain features and express the shape and contour of hand gestures. This additional feature vector has been obtained by the concatenation of the Gabor filter, the Zernike moment, the Hu moment, the Fourier moment, and the Complex moment. Furthermore, since a set of moments can vary in size, the auxiliary feature vector also varies in size (M). Both feature vectors are then combined and input into a fully connected layer. We note that dropout is applied only to the features extracted by the convolutional layers, as these features are more numerous and more likely to contain redundant information than the manually extracted features.

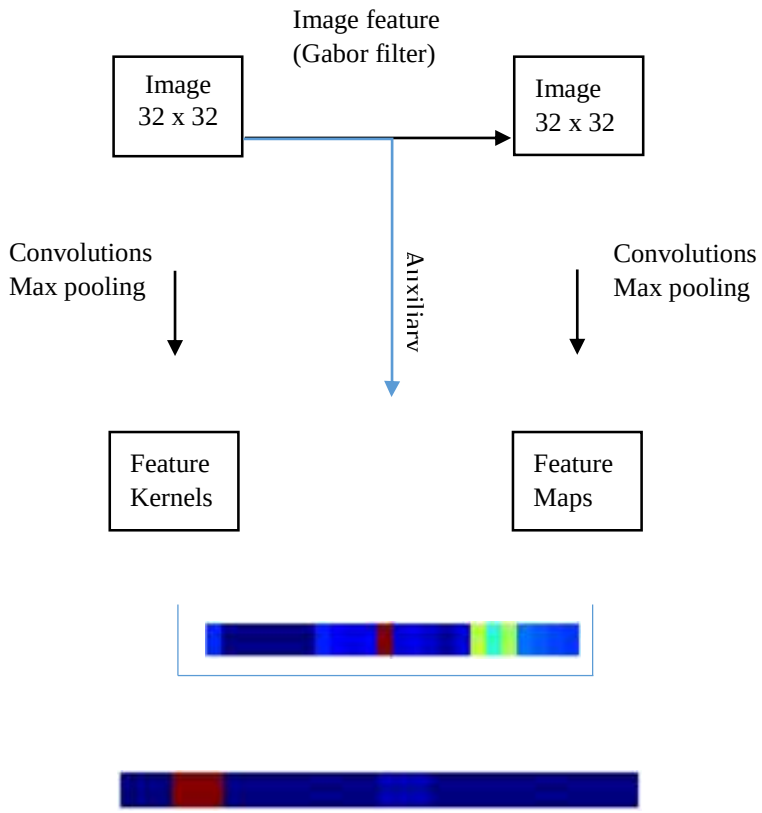


Figure 4: An illustration of the layer activations in the MFCNN

Finally, the output layer contains the same number of neurons as classes in the dataset. The final classification result was obtained by choosing the output neuron with the fittest (highest) activation.

Sequence sentence generating

In this context, we implemented the Beam Search (BS) algorithm in our research. Figure 5 demonstrates the processing diagram of sentences generation. The character level is responsible for training in this network. Meanwhile, the text is represented as a sequence of individual characters.

$$(C_1, \dots, C_t, \dots, C_T)$$

that *X* features presented and provide the label of the images consequently $(C_1, \dots, C_t, \dots, C_T)$. in the English alphabet's version $\sum(\ the\ vocabulary\ size\ V\ is\ equal\ to\ |\Sigma|)$. The two main concatenate text vector are the input layer.

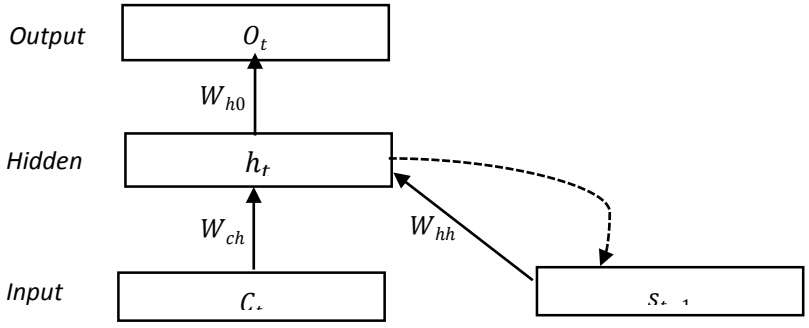


Figure 5: Beam search algorithm process for real sentences generation. C_t is the input character, h_t is the hidden layer execute in s_t time, and O_t is the output

The current C_t is the first character that represents encode as a vector. The second vector is an explanation about the procedure of the hidden layer at the previous time-step, $s_t - 1$. The sigmoid and hidden units are connected to the output layer O_t of size V. Moreover, the probability of the text will share the next character the C_{t+1} and then provide its context P $(C_{t+1} \mid C_t, S_{t-1})$. W_{ch}, W_{hh} and W_{ho} as the matrices of weigh.

Firstly, Beam Search (BS) uses network fragments to evaluate the probabilities of initial words, followed by running an encoding network. Subsequently, the encoded words are decoded to obtain probabilities by considering a set length of possible words to retain in memory. Figure 6 illustrates the encoding and decoding process of the network.

$$\mathrm{P}(y^{<1>}, y^{<2>} \mid x) = \mathrm{P}(y^{<1>} \mid x)\mathrm{P}(y^{<2>} \mid x, y^{<1>})$$

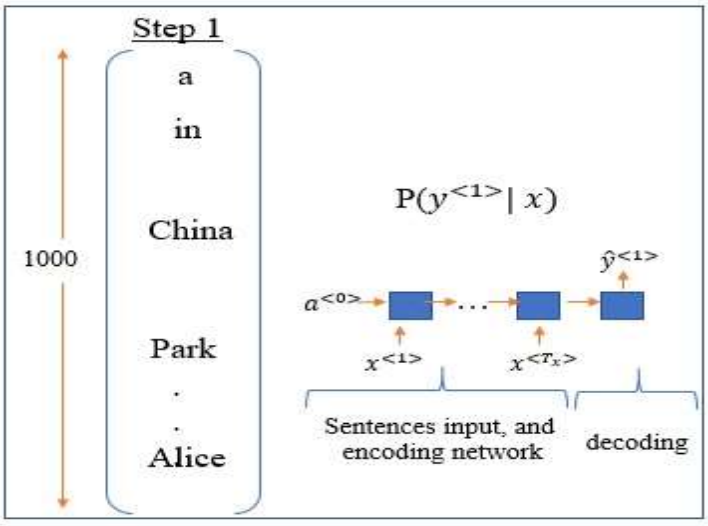


Figure 6: Beam Search encode and decode network

To generate the sentences, BS demand to select beam width size. Because it is important for BS to compare and find the related word in network easily. Figure 7 demonstrated the example of actual sentences by defining BS width that equals 3. There are three words selected (yellow circle) from 1,000 vocabularies of database. The probability of the first word is stored in $\hat{y}^{<1>}$. To find the second word beam-width demand probability of the second word (blue circle) and put together with first probability $P(y^{<2>}|\, x, "at")$. The same meaning of $P(y^{<2>}|\, x, "at")$:

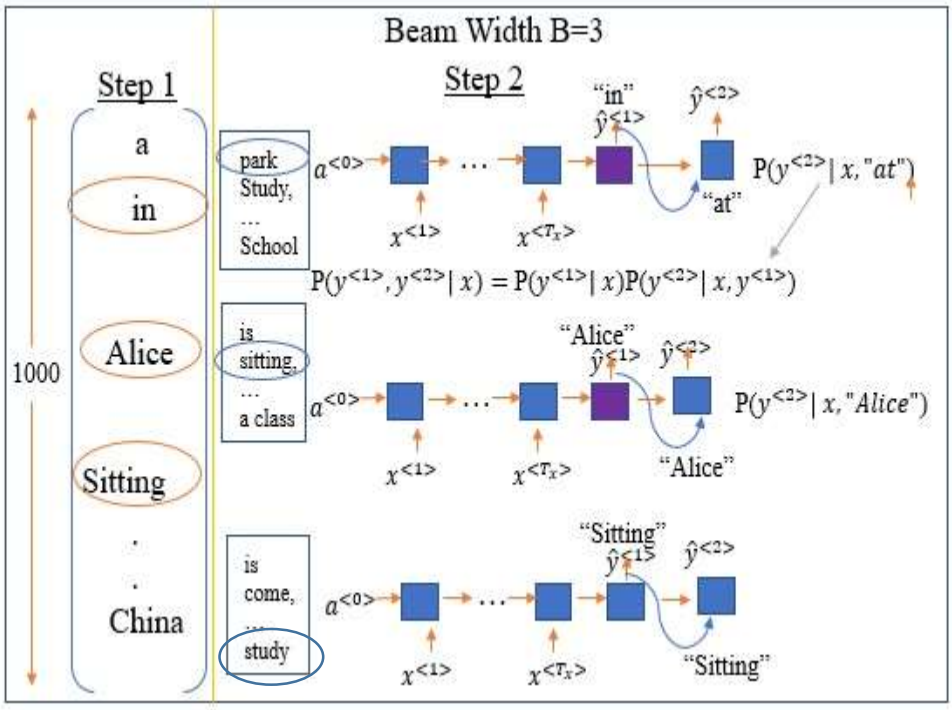
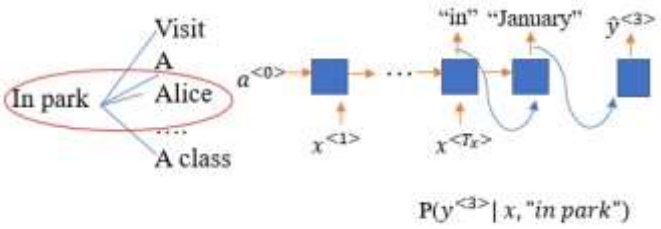


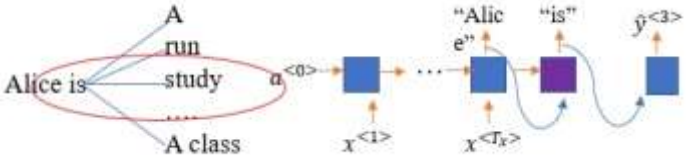
Figure 7: Sample of actual sentences generation in beam search

Based on the rule of BS, the first word will be rejected if the word in second probability has been selected, for example, the word: "Sitting". In this context, there are only two possible candidates: $y^{<1>}$ and $y^{<2>}$. But the BS still considers three possibilities because our beam width is 3; therefore, the probability of $y^{<3>}$ is:

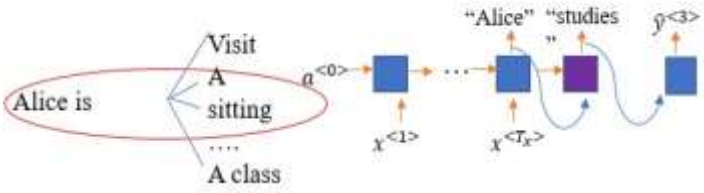
(1) In park Alice,



(2) Alice is study



(3) Alice is sitting,



So, the possible sentences is P($y^{<1>}, y^{<2>}| x$): "Alice is sitting in park."

4. Results and discussion

Results of features classification based on NSGAII

Our research succeeded in extracting and classifying the features from the Massey dataset. Table 1 illustrates the results of multiple-object features classification by using the NSGAII algorithm.

Multi-layer perceptions have been utilized for features extraction and arrangement. The system designed 300 neurons for the Hu-moment, 400 neurons for both the Zernike moment and the Hu-moment, 300 neurons for the Gabor filter, 400 neurons for the combination of the Gabor filter, the Zernike moment, and the Hu-moment, and finally, assigned 400 neurons for the Gabor filter. The results presented in Table 1 demonstrate that the most accurate features are received from the combination of the Gabor filter, the Zernike moment, and the Hu-moment. The accuracy rate varies from 98.3% to 99.0%. Inversely, the Hu-moment accuracy rate is lower than other methods. The rate of accuracy from this method ranges from 80.3% to 83.3%., The system can produce a better result, but the time execution for multiple objects in the CPU is higher and slower.

Table 1: Results of features classification

Type	Features	Hidden Layer	CPU time (mn)	Accuracy (%)
Hu	7	300	≅20	≅ 80.3~83.3
ZM, Hu	32	400	≅30	≅ 95.0~ 98.0
GB	48	300	≅28	≅ 90.0~91.0
GB, ZM, Hu	169	400	≅32	≅ 98.3 ~ 99.0
CM	10	300	≅22	≅ 85.0~86.0
FM	35	300	≅22.5	≅ 89.0~90.0

Results of hand gesture recognition based on the MFCNN Model

The results of static hand gestures tested on the OUHANDs dataset across eight subjects are presented in Table 2. The results demonstrate that this model correctly identifies images with a high accuracy rate and does not encounter confusing cases.

Table 2: Result of image recognition with eight subjects from OUHANDs dataset

Subjects	(Heights, widths)	Confuses	Activations
A	32 x 32		9.1770476e-01
C	297 x 399		8.9346465e-03
E	202 x 456		7.3182590e-02
F	32 x 32		9.1579060e-02
J	191 x 429		8.3959335e-01
H	32 x 32		7.3848947e-03
I	32 x 32		8.0621385e-01
K	32 x 32		7.8502666e-08

Figure 8: Results of image prediction using the OUHANDs dataset. The red rectangle represents the matched image, while the blue rectangles show the shapes of the other nine images. A gesture resembling the character "a" is predicted by MFCNN

Figure 8 shows an example of a hand gesture (character "A") recognition. Based on result, MFCNN model can produce a very accurate result and the high score. Specific to processing time, the execution time is from 0.3 seconds to 0.7 seconds faster than another model in the same case of using CPU. Our model can recognize the numerous images that have different sizes and colors.

Comparison between MFCNN with some models

Table 3 uses the Massey dataset and compares results from the holdout and leave-one-subject-out validations. Table 4 compares the rate of recognition of different methods on binary (B) and grayscale (G) objects. Table 5 contains a similar comparison using leave-one-subject-out validation. The results of MFCNN on the Grayscale subset are significantly better than other results in the same table.

Table 3: Comparison of accuracy on Massey subsets using holdout validation

Model	Grayscale (G)	Binary (B)
O. K. Oyedotun et al. (2020)	92.20	86.62
E.Nasr-Esfahani et al. (2016)	96.97	96.16
Zheng et al. (2019)	96.97	95.86
MFCNN	98.5	96.3

Table 4: Comparison of accuracy on Massey subsets using leave-one-subject-out validation

Model	Grayscale (G)	Binary (B)
O. K. Oyedotun et al. (2020)	73.86	76.61
E.Nasr-Esfahani et al. (2016)	79.04	77.62
Zheng et al. (2019)	78.51	79.88
MFCNN	84.42	82.02

The proposed MFCNN architecture was shown to provide statistically better results compared to the current start-of-the-art analysis models used on most datasets. In this section, the results are examined by considering the influence of color space, rescaling technique and classification architecture.

Results of sequence sentence generation based on the beam search algorithm

The image label is very important for generating sentences. The label of the image can describe the characteristics of the hand gestures. In our experiment, we randomly selected 100 gestures from OUHANDs dataset.

Table 5: Results of sequence sentences generation based on a beam search algorithm

Words	Training loss	Evaluate loss
100	4.0 ~ 5.0	4.0 ~ 4.8
1,000	5.0 ~ 5.3	4.9 ~ 5.0
10,000	5.0 ~ 5.4	5.0 ~ 5.5
40,000	5.5 ~ 6.0	5.5 ~ 6.5

The label sizes for testing in this model were 100, 1,000, 10,000 and 40,000. The label is generated from static hand gestures and contains only 100 labels; thereafter, these labels will be combined with the CNN/Daily Mail dataset. Table 6 shows the results of the sequences of sentences generation. The experimental results indicate that employing the Beam Search (BS) algorithm for sentence generation yielded favorable outcomes.

As stated, our research focused on three types of results. Firstly, the result of features classification from the use of the NSGAII. This algorithm can evaluate and select the image features that have been extracted from five methods as described above. The fittest feature set will be used to feed the recognition model. Secondly, the results of hand gesture recognition from the use of the novel model of MFCNN. This model can smoothly recognize single static hand gestures. Lastly, the result of sequence sentences generation by using the BS algorithm has utilized the image label to generate the sentences. The success of sentences generation was a positive achievement which can facilitate and simplify effective communication.

5. Conclusion and Future Research

Findings

Based on the experiments conducted in this research, several key findings have emerged:

1. Effective feature combinations: The combination of Gabor filter, Hu moment, and Zernike moment features yielded high accuracy rates ranging from 98.3% to 99.0% in

static hand gesture recognition tasks. This indicates that integrating multiple feature descriptors enhances the model's ability to distinguish between different gestures effectively.

2. Robust Performance Across Datasets: The proposed Multiple Feature Convolutional Neural Network (MFCNN) demonstrated consistent and robust performance across different datasets, including Massey's dataset with 2,524 images and 36 gestures, and OUHANDs dataset with 3,000 images and 10 gestures. The model's ability to generalize well across varied datasets underscores its robustness and effectiveness.

3. Optimized Computational Efficiency: Selected feature descriptors were chosen for their computational efficiency, ensuring that the model can operate in real-time scenarios without compromising accuracy. This makes the system suitable for practical applications where efficiency is critical.

Conclusion

A novel architecture of a convolutional neural network combined with multiple features was proposed in this research. Results allow us to state significant findings, including:

- The complementing common features in the recognition model perform well in both single and multiple objects.
- The conversion of object gestures into a standard model, generates greater efficiencies and results to extract the features, boost system speed, and save some resources.

The research works are summarized as follows:

1. The research focused on static hand gestures recognition by using multiple Benchmark datasets, OUHANDs, and Massey dataset**.**
2. For the feature extraction and classification, the object features retrieval is based on five special methods, such as the Zernike moment, the Hu moment, the Gabor filter, the Complex moment, and the Fourier filter. To classify these features, we used an evolutionary algorithm called Non-Dominated Sorting Genetic Algorithm II (NSGAII). The features classification defines the fitness features, reduce the redundant, and minimize features value before feeding in a recognition model.
3. The proposed static hand gestures recognition is based on multiple features convolutional neural network two channels (MFCNN). A novel architecture of this

model has challenged and compared to other methods, tracking the speed of recognition and accuracy.

4. The sequence sentences generation based on the BS algorithm; The sentences can be generated by using the label of the image and CNN/Daily Mail dataset.

Future research

While our research achieved good results, we believe that additional gains and outcomes can be achieved. We provide the following suggestions for future research endeavors:

1. Enhance System Capability for Common and Real-Time Hand Gesture Recognition: Future work should focus on improving the system's ability to recognize a wide range of hand gestures in real-time, making it more versatile and applicable in dynamic environment.

2. Improve Object Gesture Quality with Depth Cameras: Utilizing depth cameras can significantly enhance the quality of object gesture recognition by providing additional spatial information, leading to more accurate and robust recognition.

3. Expand Studies to alternative Applications and Version Environments: Replicating studies across different platforms and environments, such as mobile devices, tablets, and websites, can validate the system's versatility and robustness.

4. Enhance the Beam Search Algorithm: The current implementation of the Beam Search algorithm, while effective, remains slow and is limited by exposure bias and evaluation-training mismatches. Future research should focus on optimizing the algorithm to address these issues, improving both its speed and accuracy.

5. Develop a System for Dynamic Hand Gesture Recognition: Moving from static to dynamic hand gesture recognition will likely present additional research challenges and opportunities. This advancement can broaden the system's applicability and improve interaction in more complex scenarios.

By focusing these avenues, future research can build on our current findings to develop more advanced, accurate, and versatile hand gesture recognition systems.

References

Akintola, K.G., & Emmanuel, J.A. (2020). Static Hand Gesture Recognition Using Multi-Layer Neural Network Classifier on Hybrid of Features. *American Journal of Intelligent Systems*, 10, 1-7. http://article.sapub.org/10.5923.j.ajis.20201001.01.html

Al Omari, F., Zitar, R. A., & Al-Jarrah, O. (2009). A feature selection model for hand gesture recognition using genetic algorithms. *Journal of Intelligent and Robotic Systems*, 56(1), 33-43. doi:10.1007/s10846-009-9337-0

Cireşan, D., Meier, U., & Schmidhuber, J. (2012). Multi-column deep neural networks for image classification. *CVPR 2012*, 3642-3649. https://doi.org/10.48550/ARXIV.1202.2745

Eid, A., & Schwenker, F. (2023). Visual static hand gesture recognition using CNN. In *Preprints*. https://doi.org/10.20944/preprints202306.0662.v1

Eid, A., & Schwenker, F. (2023). Visual Static Hand Gesture Recognition Using CNN. https://doi.org/10.20944/preprints202306.0662.v1

Eid, A., & Schwenker, F. (2023). Visual static hand gesture recognition using CNN. In *Preprints*. https://doi.org/10.20944/preprints202306.0662.v1

Islam, M. Z., Hossain, M. S., ul Islam, R., & Andersson, K. (2019). Static hand gesture recognition using convolutional neural network with data augmentation. *2019 Joint 8th International Conference on Informatics, Electronics & Vision (ICIEV) and 2019 3rd International Conference on Imaging, Vision & Pattern Recognition (icIVPR)*. https://doi.org/10.1109/ICIEV.2019.8858563

Nasr-Esfahani, E., Samavi, S., Karimi, N., Soroushmehr, S. M. R., Jafari, M. H., Ward, K., & Najarian, K. (2016, August). Melanoma detection by analysis of clinical images using convolutional neural network. In 2016 38th annual international conference of the IEEE engineering in medicine and biology society (EMBC) (pp. 1373-1376). IEEE.

Noble, F., Xu, M., & Alam, F. (2023). Static hand gesture recognition using capacitive sensing and machine learning. *Sensors (Basel, Switzerland), 23(7)*, 3419–3434. https://doi.org/10.3390/s23073419

Oyedotun, O. K., Aouada, D., & Ottersten, B. (2020). Improved highway network block for training very deep neural networks. IEEE Access, 8, 176758-176773.

Özerdem, M. S., & Bamwenda, J. (2019). Recognition of static hand gesture with using ANN and SVM. DÜMF Mühendislik Dergisi, 10(2), 561–568. https://doi.org/10.24012/dumf.569357

Özerdem, M. S., & Bamwenda, J. (2019). Recognition of static hand gesture with using ANN

and SVM. *DÜMF Mühendislik Dergisi*, *10*(2), 561–568. https://doi.org/10.24012/dumf.569357

Pinto, R. F., Borges, C. D. B., Almeida, A. M. A., & Paula, I. C. (2019). Static hand gesture recognition based on convolutional neural networks. *Journal of Electrical and Computer Engineering, 2019*, 1–12. https://doi.org/10.1155/2019/4167890

Pisharady, P. K., & Saerbeck, M. (2015). Recent methods and databases in vision-based hand gesture recognition: A review. *Computer Vision and Image Understanding*, *141*, 152-165. https://doi.org/10.1016/j.cviu.2015.08.004

Premaratne, P., Nguyen, Q., & Premaratne, M. (2010). Human computer interaction using hand gestures. In *Communications in Computer and Information Science* (pp. 381–386). Springer Berlin Heidelberg.

Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural Networks: The Official Journal of the International Neural Network Society, 61*, 85–117. https://doi.org/10.1016/j.neunet.2014.09.003

Sharma, A., Yadav, A., Srivastava, S., & Gupta, R. (2018). Analysis of movement and gesture recognition using Leap Motion Controller. *Procedia Computer Science, 132*, 551–556. https://doi.org/10.1016/j.procs.2018.05.008

Tairych, A., Xu, D., O'Brien, B. M., & Anderson, I. A. (2016). Non-verbal communication through sensor fusion. In Y. Bar-Cohen & F. Vidal (Eds.), *SPIE Proceedings*. SPIE.

Yu, Q., Zhou, S., Jiang, Y., Wu, P., & Xu, Y. (2019). High-performance SAR image matching using improved SIFT framework based on rolling guidance filter and ROEWA-powered feature. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, *12*(3), 920–933. https://doi.org/10.1109/jstars.2019.2897171

Yu, Qinghua, Wang, J., Zhang, S., Gong, Y., & Zhao, J. (2017). Combining local and global hypotheses in deep neural network for multi-label image classification. *Neurocomputing*, *235*, 38–45. https://doi.org/10.1016/j.neucom.2016.12.051

Zheng, Y., Ji, P., Chen, S., Hou, L., & Zhao, F. (2019). Reconstruction of full-length circular RNAs enables isoform-level quantification. Genome medicine, 11, 1-20.

Authors' Biography

Kimlong Ngin, born in 1990 in Kandal Province, obtained a Master's degree in Computer Science (Applied Computer Application) from the University of Science and Technology of China in 2019. He earned his bachelor's degree in Computer Science from the Royal University of Phnom Penh in 2013. With extensive experience in software development and team leadership, Mr. Ngin has made significant contributions to the field. In recognition of his expertise, he was awarded the title of Assistant Professor in 2023. Currently, he serves as a lecturer at the Institute of Information Technology at the University of Heng Samrin Thbong Khmum, where he continues to educate and inspire future generations of computer scientists.

Pakrigna Long is a passionate technology lecturer in Phnom Penh. He has a dual bachelor's degree; in Computer Science from National Polytechnic Institute of Cambodia and Informatics Engineering from STMIK IKMI Cirebon, Indonesia. He holds a Master of Engineering in Computing Engineering Systems from King Mongkut's Institute of Technology Ladkrabang, Thailand. Currently, he is a PhD candidate at the Universiti Kuala Lumpur, researching on Data Science and Analytics. In addition to his qualifications and current role, he used to be an IT officer, IT Programmer, IT Supervisor in Phnom Penh, and a data analyst in Kuala Lumpur. His research and development interests are Natural Language Processing, Data Science, Robotics, and Machine Learning.

Pichkhemara Morn was born in Kampong Cham Province in 1979, and holds a master's degree in Information Technology from Norton University of Cambodia in 2011. He has worked as a lecturer in a related field since 2003 in Phnom Penh and provinces. Currently, he is a full-time senior lecturer at ACLEDA University of Business (AUB). Besides teaching, he was a developing software conductor. Furthermore, he contributes research in team. He's interested in desktop and mobile app development the most. He is an outstanding lecturer and developer at AUB.

Authorship Disclaimer

The authors are solely responsible for the content of this article. The views expressed herein are those of the authors and do not necessarily reflect the views of the journal, its editors, or the publisher.