# Self-Learning Model for Pattern Recognition in Vision System Based on Adaptive Kernel

Aradea*, Rianto, Nina Herlina, Irani Hoeronis
Department of Informatics, Faculty of Engineering, Siliwangi University, Kode Pos 46115, Tasikmalaya, Indonesia
E-mail: aradea@unsil.ac.id, rianto@unsil.ac.id, ninaherlina@unsil.ac.id, iranihoeronis@unsil.ac.id
*Coresponding Author

**Keywords:** neural network, pattern recognition, self-adaptation, self-learning, vision system

**Received:** October 4, 2024

*Recently, the solution for recognizing and understanding an object based on visuals is to integrate the adaptation function (continuous machine-driven process) into the system update function involving humans (continuous human-driven process). However, this has created a gap between the adaptation function and the system. This situation requires understanding the system viewed as a dynamic composition of the learning process. This research introduced a self-learning model in the form of an adaptive kernel equipped with the SpinalNet architecture, and the goal of this study is to increase the Convolutional Neural Network (CNN) accuracy. The model consisted of a domain model, contextual knowledge, and adaptive learner developed based on the CNN with SpinalNet. The combination of Adaptive Kernel and SpiralNet in this CNN has a significant impact, allowing the model to adjust the selection of subsequent kernels based on the optimal input from the previous kernel. Moreover, this combination results in lower memory usage during training. The evaluation results show that our proposed model provides better classification accuracy than the SpiralNet model without the Adaptive Kernel. Furthermore, in terms of inference speed, our model outperforms SpiralNet, as evidenced by the use of fewer parameters.*

*Povzetek: Prilagodljiv model samoučenja, ki temelji na jedru, izboljša prepoznavanje vzorcev v sistemih za vid, integracijo CNN s SpinalNet za izboljšanje natančnosti klasifikacije, optimizacijo izbire jedra in zmanjšanje uporabe pomnilnika med usposabljanjem.*

## 1 Introduction

Computer (system) vision is a field of artificial intelligence that trains computer machines to interpret and understand (recognize) the visual world through deep learning models. The goal is that the machine can accurately identify and classify real-world objects and then react to what it sees. At present, the recognition and reaction capabilities of a system will be associated with the complexity of a highly dynamic, unpredictable, and uncertain environment [1]. In addition, the involvement of various elements of the real world interacting with the system will require the adaptability of the system. This ability will determine the success or failure of a system in recognizing and acting on what is occurring in its environmental context [2]. In fact, [3] states that the need to develop a system has entered the wave of learning from experience, namely the deployment of machine learning techniques. It functions to support various system functions to create an adaptive system, including a system capable of operating under conditions of uncertainty. Also, it can guarantee that its main property will function optimally. Therefore, a vision system for the current world requires a pattern recognition model possessing adaptability and a reliable optimization level.

Adaptability in a system aims at realizing the behavior of adapting a system built based on special requirements [4]. This situation, among others, requires a system to recognize changes in its application domain. Additionally, it can change itself to produce alternative behaviors [5]. Further, [3] in his latest review of long-term challenges that could trigger a new wave of scrutiny in the field of self-adaptation, raises an interesting question, namely the extent to which to develop systems to handle conditions that were not (fully) anticipated at the time the system was cultivated. Researchers have proposed various approaches to fostering adaptability in a system based on their respective problem domains. As a result, currently, neither a definition nor a specification for a system's adaptability has been widely agreed upon [6]. Besides, this applies to the specification of adaptability in vision systems. As an example, there is a need for deep meta-learning applied to image recognition problems [7]. This problem can be resolved by understanding the system viewed as a dynamic composition of the learning process, namely how to enhance the system with self-learning abilities [3].

The perspective of growing adaptability is grounded in the self-learning model. The idea is to overcome the gap existing in the traditional perspective. In particular, there is a need to integrate the adaptation function (continuous machine-driven process) into the system update function involving humans (continuous human-driven process). Consequently, the system can only run for a short cycle since it has to wait for updates to deploy. Researchers generally develop adaptability for pattern recognition in vision systems by expanding various features to complement machine learning's ability to recognize visual cues. Some approaches or techniques can be used. Generally, they can be categorized into three categories, namely feature-based, template-matching, and image-

based [8]. Image-based techniques are one of the concerns in this study because they can utilize all parts of an image. As a result, the detection process does not depend on the characteristics of an image or not focus on matching small parts of the image, becoming the model [9], [10]. It is expected that our research can be more flexible in developing a generic model to capture and recognize objects holistically with an optimal level of accuracy with self-learning capabilities.

The existing main problem related to the application of machine learning for vision systems is to determine the most optimal algorithm. In addition, the researchers have conducted miscellaneous empirical investigations on various existing algorithms. One of them is a neural network. Nowadays, neural networks have become a method in machine learning with great success, including in object detection research [11] which initially had difficulties in its development. With various extensions of existing neural network-based methods, the development process has become easier [12]. One of them is the utilization of a deep neural network. It is a neural network architecture delving image data. In the context of a vision system, object detection is performed by training a computer to interpret and understand the visual world through a deep learning model. Hence, the machine can accurately identify and classify objects and react to what it screens. Therefore, the vision system requires a pattern recognition model to reach a reliable level of optimization and adaptation.

There are myriad neural network algorithms. One of the developments (types) is the Convolutional Neural Network (hereafter, CNN) algorithm. CNN is a variation of Multilayer Perceptron (hereafter, MLP) designed to process two-dimensional data. On the one hand, MLP is not suitable for use in the case of image classification since it does not store special information from image data and considers each pixel as an independent feature, resulting in poor results [13]. On the other hand, CNN is also a type of deep neural network designed to process two-dimensional data with a high network depth and is widely applied to image data [14]. Based on research [15], CNN has shortcomings in terms of the old model training process. Therefore, there has been a plethora of studies developing the CNN algorithm to get results or performance, especially regarding the level of accuracy so that it gets better. One of the developments in the use of optimization algorithms. Several optimization algorithms are included in the minibatch-based adaptive algorithm or algorithms included in the gradient descent optimization algorithm.

These works allow us to extend the self-learning model based on neural network theory. A neural network, as a fundamental primitive, can provide flexibility in designing an architecture that focuses on adaptability. However, its impact on computational complexity should also be noted. Furthermore, various existing research results mainly accentuated the level of accuracy in the pattern recognition process. Only a tiny proportion pays attention to the adaptability of the learning process. One of the reasons for this is the lack of a good representation for meta-learning [7]. This study introduced a self-learning model for pattern recognition in the vision system by bringing up the adaptability function in the learning process. The model consisted of an optimized CNN algorithm employing an adaptive kernel. Thus, CNN can adapt to the model parameters in the learning process. The rest of this study consists of the second part discussing relevant studies, the third part describing the proposed model, and the fourth part eliciting the application of the model. In particular, it consists of experiments and a discussion of the evaluation results. Finally, the fifth part concludes all the work results and discusses future job opportunities. The rest of this study consists of the second part discussing relevant studies, the third part describing the proposed model and the fourth part eliciting the application of the model. In particular, it consists of experiments and a discussion of the evaluation results. Finally, the fifth part concludes all the work results and discusses future job opportunities.

## 2 Related work

There have been various empirical results relevant to machine learning for pattern recognition needs in vision systems. [16] compared the results of applying various optimization algorithms in deep learning, namely CNN, with three different CNN architectures. This study deployed two machine learning models, namely supervised and unsupervised learning. There were ten algorithms compared in this study, including the minibatch-based adaptive algorithm or algorithms included in the gradient descent optimization algorithm, namely the Stochastic Gradient Descent (SGD) algorithm, SGD-Momentum, SGD-Nesterov, AdaGrad, AdaDelta, RMSProp, Adaptive Momentum, AdaMax, Nadam, and AMSGrad. Four datasets were utilized: MNIST, CIFAR-10, LFW, and Kaggle Flowers. One of the results of this study was that the Adaptive Momentum optimization algorithm worked optimally. In other words, it reached the highest level of accuracy when applied to the first and third CNN architectures with the dataset applied as LFW. Besides, [17] also compared the performance of CNN. The results indicated that the Adaptive Momentum optimization algorithm had the highest level of accuracy. This study applied the Adaptive Momentum algorithm to three different CNN architectures, namely ShallowNet, LeNet, and AlexNet. The results reported that the best way to increase the accuracy of photosynthetic pigment prediction on plant digital images was to deploy the adaptive momentum algorithm combined with the LeNet architecture.

Currently, the use of CNN architecture has reached a higher level by adding an adaptive scheme to the training process. The research in [18] introduced an adaptive learning rate rule in CNN training by integrating the Egret Swarm Optimization Algorithm (ESOA) and quadratic interpolation (QIESOA) to improve prediction accuracy. Adapting the learning rate improved CNN's weaknesses in multi-domain image classification tasks, achieving the highest accuracy of 97.15% on the test dataset. Luo and Hu [19] developed Adaptive Attention ResNet (AA-ResNet), which addresses overfitting and training errors in CNNs with deeper networks. Feature extraction became a

primary focus of their research, using residual modules and adaptive attention to enhance feature representation. The developed model demonstrated high performance on the Cifar-10, Caltech-101, and Caltech-256 datasets. The research by Jiang et al. [20] discussed the role of activation functions in Convolutional Neural Networks (CNNs). It introduced the Adaptive Offset Activation Function (AOAF) as a solution to improve image classification accuracy. AOAF is a new parametric activation function that connects negative and positive values by adding an adaptive parameter (the average of the input feature tensor) [20]. The results showed that AOAF significantly improved accuracy, especially on datasets with high feature complexity. Wu and Pan [21] introduced an adaptive modular convolutional neural network (CNN) model design to improve efficiency and accuracy in image recognition tasks. Through a gate unit based on attention mechanisms, the model adaptively selects the optimal network structure based on learning. The results showed high accuracy on three Kaggle datasets (Cats-vs.-Dogs, 10-Monkey Species, Birds-400). The research by Guo et al. [22] focused on developing an Adaptive Pooling Network (APN) based on memristor arrays to improve the performance and resilience of CNNs in managing information loss during pooling. The results demonstrated that APN enhanced CNN performance in terms of both accuracy and robustness on the MNIST and CAPTCHA datasets. To clarify the research results and identify gaps in the state-of-the-art concerning adaptability in vision systems, especially CNNs, we have summarized the findings in a table, as shown in Table 1.

Table 1: State-of-the-art

| Research | Proposed Method | Problem | Contribution | Result | Weakness |
|---|---|---|---|---|---|
| Wei dkk. [18] | CNN + QIESOA | Slow convergence of traditional CNNs | Adaptive learning rate update with ESOA and Quadratic Interpolation. | 91.25% (Cifar-10), 88.66% (EMNIST), 95.87% (EuroSAT), 88.66% (Fashion-MNIST), 97.15% (RiceImage). | Adaptation to datasets with high dynamics or specialized domains has not been discussed. |
| Luo dan Hu [19] | AA-ResNet | Overfitting due to network depth. | Adaptive attention, multitask loss function. | 92.43% (Cifar-10), 69.61% (Caltech-101), 52.29% (Caltech-256). | Adaptation to large-scale datasets or new domains has not been tested. |
| Jiang dkk. [20] | AOAF | Low performance of the ReLU function. | Using negative values in feature extraction. | Accuracy increased by 4.8% compared to ReLU | Not tested on datasets with high noise or different distributions. |
| Wu dan Pan [21] | Adaptive Modular CNN Model | Overfitting, large parameters. | Parallel modules and submodules, adaptive reduction of FLOPs. | 99.3% (Cats-vs-Dogs), 99.26% (10-Monkey Species), 99.13% (Birds-400) | Not evaluated on datasets with noise or extreme variations. |
| Guo dkk. [22] | APN (Memristor-based) | Information loss in CNN pooling. | Adaptive pooling without backpropagation. | 99.3% (MNIST), 92.6% (CAPTCHA). | Difficult to adapt to systems without memristors and large datasets. |

Self-learning capabilities for vision systems have also been developed [7] by proposing a framework consisting of three main modules: the concept generator, meta-learners, and concept discriminators. This framework integrated the representational power of deep learning into meta-learning. The results substantially improved vanilla meta-learning, demonstrated in various few-shot image recognition problems. Other researchers, including [23], employed a new structure and concept called SpinalNet. SpinalNet is an amalgamation of DNN and Gradual Input implementations. This study highlighted the shortcomings of DNNs related to computational intensity due to the size of the input network. Therefore, this study applied gradual input, which was the concept of input gradually, to reduce the burden of the calculation process. The results of this

study indicated that SpinalNet was able to increase the accuracy of the usual DNN.

We had studied the model's adaptability before, starting with integrating the self-adaptation approach into requirements modeling [24]. As an illustration, we introduced a self-adaptation approach embedded into the primitive system requirements specification. Furthermore, in the study [25], we added a contextual-requirements approach to the adaptation pattern of the primitive system requirements. The goal was to capture the relevant context attributes so that the adaptive behavior of the system would match the prevailing context. In another study [1], we developed a pattern of adaptation to deal with the variability of system services. In this case, our primitive system requirements map to the various service levels of the system. In this study [2], we introduced the adaptation requirements for the adaptive systems (ARAS) framework, extending the system modeling language with control loop patterns and the context inheritance hierarchies. Technically, both were mapped into a graph network (Bayesian Network). We have defined several formalizations for adaptability in a graph. However, the results specific to the requirements of the vision system have not been attained. More recently, in the paper [26], [27], we merely attempted to apply the adaptability of this graph network to the needs of the Internet of Things (IoT) network system.

We captured research opportunities Based on the related job descriptions and studies conducted previously. In this case, the study can be performed to improve (to enhance) the adaptability of the learning process in pattern recognition for vision systems. One example is the expansion of the CNN model development. More technically, the addition of the SpinalNet architecture to the CNN model that we have developed can have the opportunity to increase the adaptability and optimization of the learning process. Meanwhile, based on studies in related studies, it was explained that CNN fit image data. It has even been widely applied to image data [8]. Consequently, image-based techniques were also our concern when formulating the needs of this research. Additionally, [7], contended that there is a lack of a good representation for meta-learning, where this meta-learning will learn the learning algorithm (meta-learner) of many related tasks. These statements and facts have motivated us to develop a new model with self-learning capabilities for pattern recognition in vision systems.

# 3 Proposed method

The perspective used in developing this proposed model was inspired by [3]. In this sense, [3] notes that the challenge in the long-term triggering a new wave of research in the field of self-adaptation is to understand the system as a dynamic composition of the learning process. The idea is to enhance a system with self-learning capabilities. To illustrate, a system allows it to learn from the variety of data it collects and autonomously develops its learning process under changing and unpredictable conditions. In the context of the vision system, the work of [7] applied this perspective by proposing deep meta-learning. Further, they also demonstrated its usefulness in image recognition problems. This work was extremely inspiring for us to propose a new model of self-learning capability for vision systems. Our model consists of three main components, namely the domain model, contextual knowledge, and adaptive learner as presented in Figure 1:

a. Domain model is a domain modeling in the form of a graph network structure to capture high-level visual signal representations.
b. Contextual knowledge represents the relevant context attributes in the model domain according to the current dynamic visual cue context.
c. Adaptive learner consists of utility (utility function) and learner (learner function) functions that carry out learning and recognize visual cues representations based on the prevailing context.

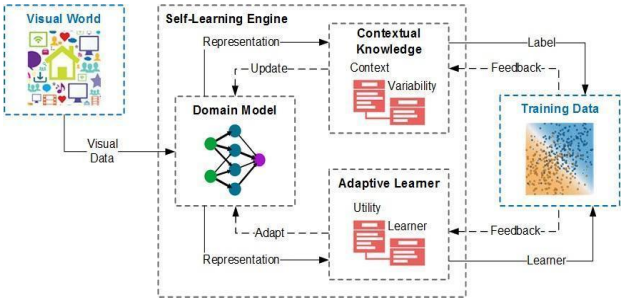



Figure 1: Self-learning model for pattern recognition in the vision system

**Domain model**

In a previous study [1], [2], we defined every element in the model domain indicating a dependency relationship. Furthermore, the model was regarded as a dynamic property in nature to be monitored based on certain parameter values. In this study, we developed it to specific representations for monitoring and capturing high-level visual cues. More specifically, the model deployed the SpinalNet structure developed by [23] taking inspiration from the human somatosensory system as presented in Figure 2. Following the way of how the human spinal network works, Spinal Net utilized gradual input (Gradual Input). All the layers contained in the model contributed to the main output of X in the same way that reflexes worked. Next, the modular input was sent to the main output of X. It was similar to how the brain works.

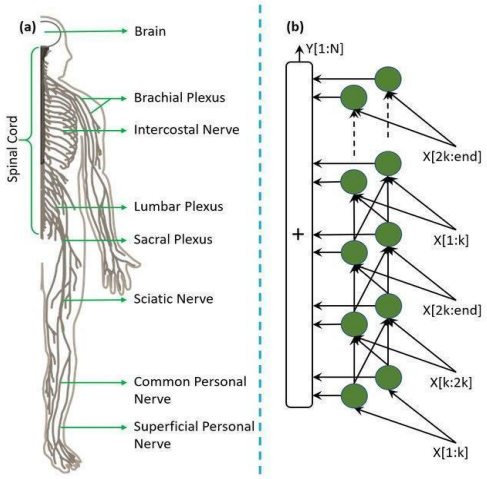



Figure 2: SpinalNet model from (Adapted from [25])

In a model like the one illustrated in figure 3, the first layer utilized a simple linear function and obtained only the sum of weight w from x1-x5. The second layer of the model gained the total weight w of x6-x10 as one input and the result of layer 1 as the other input. Briefly stated, the definition can be formulated as follows:

a. For each layer $x_i \in \{x_1, x_2, \ldots, x_n\}$ will contribute to the main output layer $X$.
b. For each input $N_i \in \{N_1, N_2, \ldots, N_n\}$ can be modularized into each of its $x_i$ layers and become inputs for $x_{i+1}$. layers.

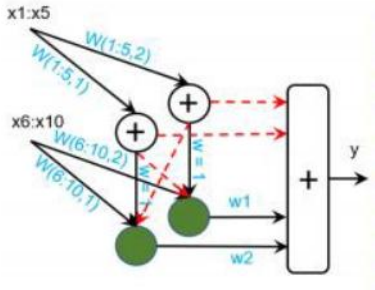



Figure 3: Simplified SpinalNet as a single hidden layer from (Adapted from [23])

**Contextual Knowledge**

Contextual knowledge is a representation of dynamic properties in the model domain [2]. It refers to the abstraction of domain properties relevant to the expected system behavior. Also, it covers the specific context in which this expected behavior applies [28], [29], [30]. In this investigation, contextual knowledge was specified for the needs of context attributes related to visual cues in the model domain. The attribute applied as contextual knowledge in this research was the kernel dimension. It was intended to determine the size of the matrix to perform convolution and input shift. The kernel on convolution is formulated as follows:

a. $F(x) * F(y)$ is the dimension of the kernel matrix.
b. $N(x) * N(y)$ is the dimension of the input matrix.
c. The output dimension of the convolution is $N(x) - F(x) + 1 * N(y) - F(y) + 1$.
d. Convoluting the kernel $Q_{u,v}$ with the activation function tanh will result in weight $K_{u,v}$.

**Adaptive learner**

The adaptive learner is a module that can automatically serve adjustments due to changing and growing needs. The main purpose of this module is to model the system dynamically. In particular, the module learned to recognize every need existing in the model domain and contextual knowledge on a run-time basis. The main problem to be handled was related to variables with varying, different, and flexible properties. This module indicated two functions, namely the utility function in the form of a function to sort or define alternative varieties according to their use for individual visual cues, and the learner function to carry out learning and introduction to obtain the most optimal results. The new kernel function was obtained through the result of the convolution of each input convolution $Q_{(u,v)}$ as in the following equation:

$$\sigma_{u,v} = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} Q_{(u,v)_{i,j}} x_{i,j} \quad ..(1)$$

The new kernel $K_{(u,v)}$ can then be deployed to perform convolution on the input image to produce S. Subsequently, it was applied as the output kernel as in the following calculation:

$$S = \sum_{u,v} x_{u,v} K \left( \sum_{i,j} Q_{u,v_{i,j}} x_{i,j} \right) ..(2)$$

$$f = tanh(S) ..(3)$$

# 4 Experiment

This section describes the evaluation of our proposed model for recognizing visual cue patterns, particularly handwriting patterns. In this experiment, we deployed MNIST datasets sourced from the research of LeCun, et. al. [31]. These datasets refer to a collection of handwritten images of numbers 0-9 consisting of 60,000 training data and 10,000 test data. The images were black and white. Each image was 28x28 pixels. The use of the MNIST dataset on the CNN method was performed by Saqib, et. al. [32]. The study succeeded in building a model recognizing and classifying handwritten figure images. The experimental results showed that the CNN model attained the highest classification of accuracy for a certain number of hidden layer neurons. Another scrutiny was conducted by Anwar, et. al. [33] Involving the MNIST dataset as the classification object of CNN. In addition to using MNIST, we also applied other datasets such as KMNIST, QMNIST, Fashion-MNIST, and EMNIST to strengthen the validation of the model we have developed.

**Preparation of model application**

The network architecture structure developed in this experiment was inspired by the SpinalNet architectural model by carrying out several expansions, namely combining it with the Convolutional Neural Network architecture through an adaptive kernel on the convolution layer. The experimental mechanism was applied to the MNIST dataset with several models. As an example, the conventional CNN model commonly used covers the CNN model combined with the Adaptive Kernel, the SpinalNet model, and the model developed by the authors. Contextual knowledge elicitation was conducted to identify the relevant context attributes in the model domain related to the dynamic context of visual cues. This provides dynamic parameter updates during training on the adaptive kernel. The adaptive kernel parameters are iteratively updated during the backpropagation process. The kernel adapts by minimizing the cross-entropy loss through the gradient descent algorithm. The utility function calculates the optimal kernel value based on the contextual knowledge that has been learned. This mechanism allows the kernel to dynamically shift its focus and optimize the most relevant features for visual signals. Unlike non-adaptive kernels, which rely on static parameters, adaptive kernels dynamically adjust their parameters during training. For instance, after each convolution operation, the kernel dimensions are updated to optimize weight alignment in subsequent layers. This flexibility results in higher accuracy and efficiency, as demonstrated in our model. Figure 4 shows the distinction between the adaptive and non-adaptive kernel processes to clarify the differences.

In this experiment, we identified the data collected from the results of pre-processing and preparation for the application of the model as contextual knowledge, namely the kernel that can change according to the determined input.

**Pattern recognition implementation and operation**

Our proposed model applied three main parts, namely the Adaptive Kernel, Convolutional Layer, and Full Connected Layer as shown in figure 5. First, the Adaptive Kernel was a Convolutional Layer involving an adaptive system in its kernel parameters. The determination of the kernel was based on the optimal input of the previously applied convolution. Second, the Convolutional Layer, both Adaptive Kernel and Convolutional Layer applied Maxpooling and Relu as activating functions. By applying the Spinal Layer to the Full Connected Layer section, the input parameters were smaller. As a result, memory usage can be kept to a minimum in learning the model. Third, Spinal Layer divided the input into several parts and then processed it with a linear function. In our model, the input was divided into two equal sizes and was processed linearly in six layers. In the final stage of the full connected layer, a linear function was utilized to combine the applied Spinal Layers. The utilized Spinal Net structure is shown in Figure 6.

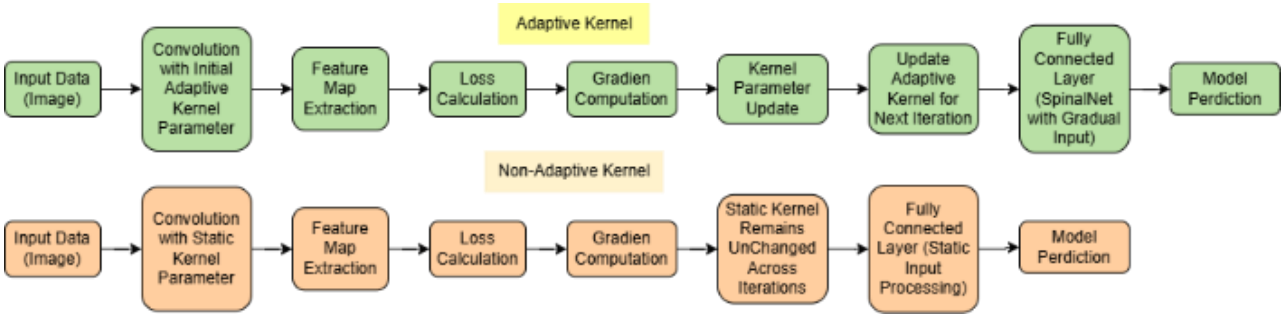



Figure 4: Differences between adaptive kernel approaches compared to non-adaptive methods.

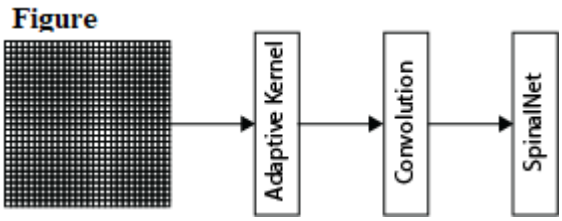



Figure 5. Self-learning model architecture

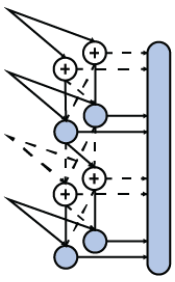


Figure 6: SpinalNet architecture in full-connected layer

More specifically, the implementation of the integration between the adaptive kernel and SpinalNet is shown in Figure 7, which illustrates the workflow of the proposed model.

The model designed in Figure 7 processes a 28x28 grayscale input image through a series of steps, starting from the dynamic kernel to the fully connected layer. In

the dynamic kernel, the kernel weights are adaptively adjusted during training, resulting in 25 feature maps (28x28x25). This output is then passed to the dynamic layer, where the results from multiple kernels are combined, and the channels are reduced to 6 (28x28x6). Next, the Conv2D Layer extracts deeper features, followed by the MaxPooling Layer for downsampling, producing an output of 12x12x20. A Dropout Layer is applied to prevent overfitting without altering the data dimensions. The data is then flattened through the Flatten Layer into a 1D vector (500 elements), which is processed progressively by the SpinalNet Layers by splitting the vector into six segments and generating a combined representation with a total of 1500 elements. Finally, the Fully Connected Layer processes this representation into logits for 10 classes to generate probabilities, determining the final class prediction. Combining the Adaptive Kernel, Convolutional Layer, and SpinalNet ensures computational efficiency and model adaptability in handling visual data.

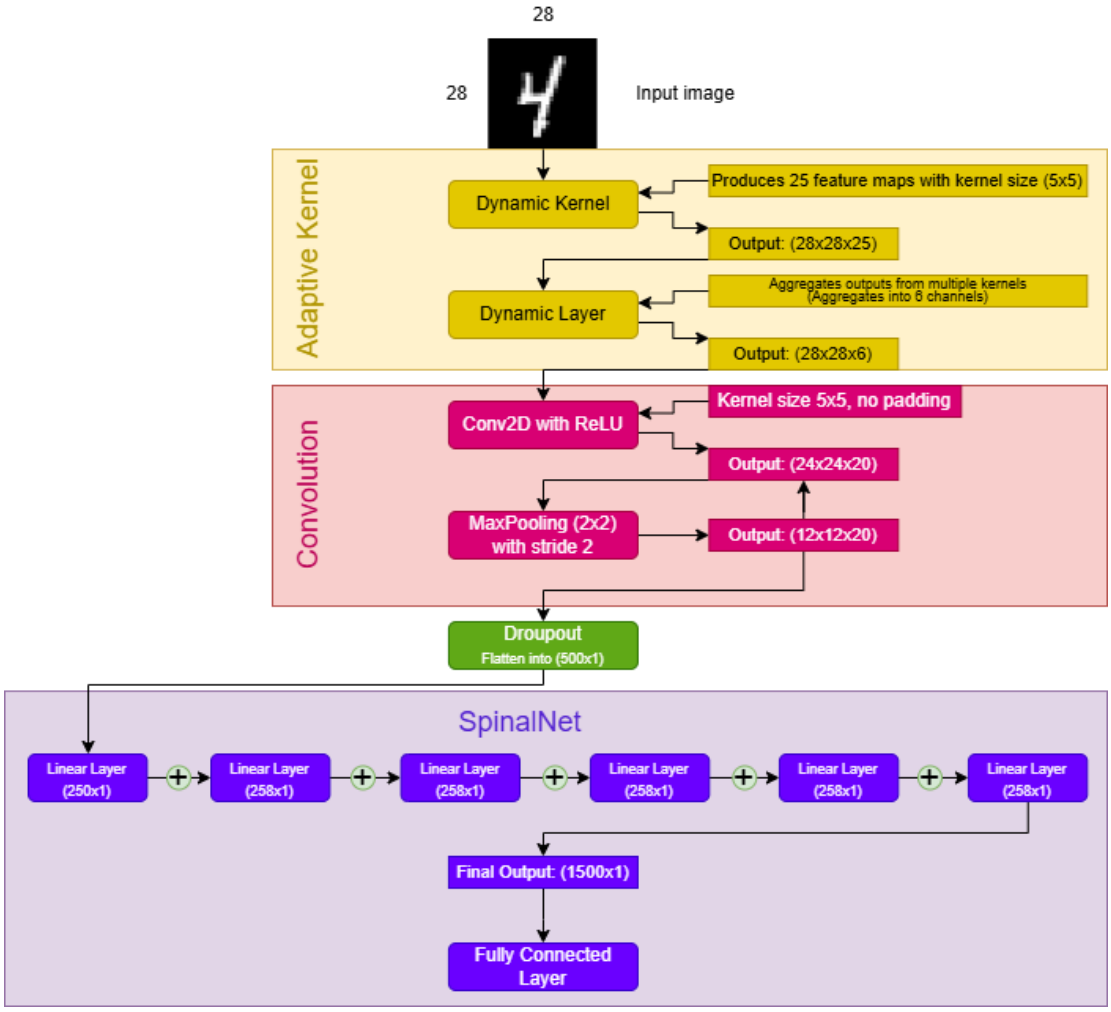


Figure 7: Purposed method framework

The integration results between SpinalNet and the Adaptive Kernel were trained using various datasets for classification tasks. This experiment has two training scenarios: one where the model is trained with the additional VGG-5 network [34] and another where the model is trained without that additional network. The model with the added VGG-5 was trained for 100 epochs, using a batch size of 128 and a learning rate $5\times10^{-3}$. In contrast to the hyperparameters used in the first scenario, the model without the VGG-5 addition was trained for eight epochs, using a batch size of 128 and a learning rate of $1\times10^{-2}$. The difference in hyperparameter usage was made to adjust to the needs of each model being trained to maximize the potential of the training results. Additionally, both training scenarios were optimized using Stochastic Gradient Descent (SGD) with the same momentum value 0.9.

These hyperparameters were determined based on the results of a systematic evaluation of several hyperparameter choices using a grid search approach. The evaluation was based on validation accuracy across various configurations while also monitoring the stability of the loss function and the efficiency of the number of parameters in the model. The evaluation results for each hyperparameter choice are shown in Tables 1 and 2.

Table 1: Hyperparameter testing for the proposed model with the added VGG-5

| Hyperparameter | Range | Optimal Value |
|---|---|---|
| Learning Rate | [0.001, 0.005, 0.01] | 0.005 |
| Batch Size | [32, 64, 128] | 128 |
| Hidden Layer in SpinalNet | [64, 128, 256] | 128 |
| Neuron per Layer in SpinalNet | [64, 128, 256] | 128 |
| Momentum | [0.5, 0.7, 0.9] | 0.9 |

Table 2: Hyperparameter testing for the proposed model without VGG-5

| Hyperparameter | Range | Optimal Value |
|---|---|---|
| Learning Rate | [0.001, 0.005, 0.01] | 0.01 |
| Batch Size | [32, 64, 128] | 128 |
| Hidden Layer in SpinalNet | [4, 6, 8] | 8 |
| Neuron per Layer in SpinalNet | [125. 250, 500] | 250 |
| Momentum | [0.5, 0.7, 0.9] | 0.9 |

From the tests in Table 1, the optimal configuration for the proposed model with the added VGG-5 was obtained, which included a learning rate of 0.005, a batch size of 128, 128 hidden layers in SpinalNet, 128 neurons per layer, and a momentum value of 0.9 for SGD. Meanwhile, the optimal performance for the proposed model without the VGG-5 addition was achieved with a learning rate of 0.01, a batch size of 128, 8 hidden layers in SpinalNet, 250 neurons per layer, and a momentum value of 0.9. This configuration provided the highest validation accuracy, maintained a stable loss curve throughout training, and showed a balance between performance and computational efficiency.

Before the training process, we performed data preprocessing on all datasets used. In this process, we applied the same steps to all datasets, which included converting the images to tensors and normalizing the values. In the tensor conversion process, the pixel values of the images were changed from the original range (0 to 255) to the range [0.0, 1.0] by dividing each pixel value by 255. Afterward, the converted pixel values underwent normalization using the Z-Score normalization method. After passing through this data preprocessing stage, the model training process is expected to be faster and more stable, accelerating convergence and reducing imbalance.

**Model evaluation and comparison**

To validate the proposed model, we compared this research model with the original SpinalNet model. The comparison included accuracy, the number of parameters used, and the inference speed of the model on each test dataset used. Tables 3 and 4 compare the evaluation results between our model and SpinalNet.

Table 3: Comparison of Adaptive-SpinalNet and SpinalNet with the added VGG-5.

| Dataset | Adaptive-SpinalNet | | SpinalNet [23] | |
|---|---|---|---|---|
| | Accuracy | Inference Time | Accuracy | Inference Time |
| MNIST | 99.78% | 5.21s | 99.72% | 5.33s |
| KMNIST | 99.24% | 5.25s | 99.15% | 6.12s |
| QMNIST | 99.54% | 16.77s | 99.68% | 16.92s |
| Fashion-MNIS | 95.21% | 5.70s | 94.68% | 6.43s |
| EMNIST (Digits) | 99.74% | 12.57s | 99.82% | `13.03s |
| EMNIST (Letters) | 94.69% | 8.68s | 95.88% | 9.17s |

Table 3 highlights the comparison between VGG-5 + Adaptive-SpinalNet and VGG-5 + SpinalNet regarding accuracy and inference time. It is evident that the inference speed of our model consistently outperforms across all datasets. Similarly, the Adaptive-SpinalNet model demonstrates a speed advantage compared to the original SpinalNet model. The adaptive kernel dynamically adjusts weights based on the input it receives, enabling a focus on the most relevant features for the classification task and thereby reducing processing time for less significant information. Additionally, parameter efficiency is achieved by minimizing redundancy in kernel weights. This results in optimal representation without excess parameters that could slow the inference process. The comparison of parameter reduction is illustrated in Figure 8.

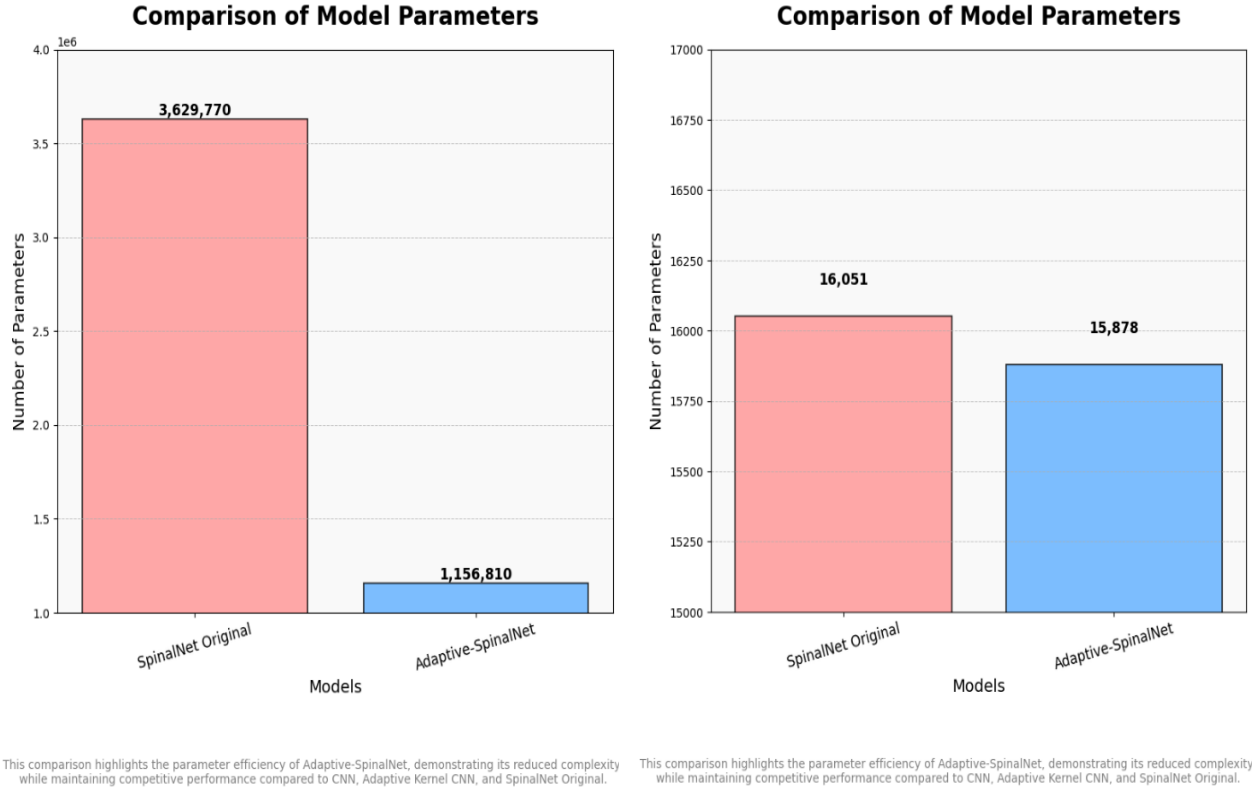
Table 4: Comparison Between Adaptive-SpinalNet and SpinalNet

| Dataset | Adaptive-SpinalNet | | SpinalNet [23] | |
|---|---|---|---|---|
| | Accuracy | Inference Time | Accuracy | Inference Time |
| MNIST | 98.93% | 3.42s | 98.48% | 3.61s |
| KMNIST | 92.52% | 3.81s | 88.25% | 4.08s |
| QMNIST | 98.47% | 12.85s | 98.07% | 13.03s |
| Fashion-MNIS | 87.92% | 3.54s | 86.61% | 3.90s |
| EMNIST (Digits) | 99.35% | 9.09s | 99.16% | 9.29s |
| EMNIST (Letters) | 91.43% | 5.24 | 90.23% | 5.97s |

In addition to its positive impact on parameter reduction, the SpinalNet architecture combined with Adaptive Kernel generally enhances accuracy across all datasets. This is particularly evident in Table 4, demonstrating that directly applying Adaptive Kernel to SpinalNet improves the model's accuracy on all test datasets. This indicates that our model, tested on various datasets (including MNIST, Fashion MNIST, KMNIST, and EMNIST), can generalize across different data distributions. Experimental results reveal that the Adaptive-SpinalNet

model consistently achieves competitive performance, even on datasets with significantly different visual patterns from MNIST. This highlights the model's ability to adapt to diverse data distributions. This adaptability is further reinforced by the Dynamic Kernel mechanism, which dynamically adjusts kernel weights based on input patterns during inference. This allows the model to capture relevant features under varying data conditions. Furthermore, the SpinalNet architecture processes feature independent segments, offering additional flexibility in handling shifts in data distribution.

Furthermore, to provide stronger validation, we compared the model's performance with related studies using the same dataset benchmarks. This comparison is presented in Table 5.




(a) (b)

Figure 8: Comparison of the number of parameters between Adaptive-SpinalNet and SpinalNet: (a) with VGG-5, (b) without VGG-5

Table 5: Comparison of Adaptive-SpinalNet with related studies

| Model | Accuracy | | | | Number of Parameters |
|---|---|---|---|---|---|
| | MNIST | KMNIST | QMNIST | Fashion MNIST | |
| SpinalNet [23] | 98.48% | 88.25% | 98.07% | 86.61% | 16K |
| VGG-5 + SpinalNet [23] | 99.72% | 99.15% | 99.68% | 94.68% | 3.6M |
| CNN + QIESOA [18] | - | - | - | 97.15% | Not Mentioned |
| APN (Memristor-based) [22] | 99.3% | - | - | - | Not Mentioned |
| R-ExplaiNet26-64 [35] | 99.70% | 98.66% | - | 93.03% | 0.89M |
| Improved Efficient Capsnet [36] | - | 98.43% | - | - | 0.58M |

| | | | | | |
|---|---|---|---|---|---|
| PMM [37] | 97.38% | - | - | 88.58% | 4.9K (MNIST), 16.7K (Fashion MNIST) |
| ConvPMM [37] | 99.10% | - | - | 90.94% | 0.13M (MNIST), 0.28M (Fashion MNIST) |
| Adaptive-SpinalNet | 98.93% | 92.52% | 98.47% | 87.92% | 15.9K |
| VGG-5 + Adaptive-SpinalNet | 99.78% | 99.24% | 99.54% | 95.21% | 1.1M |

Table 5 demonstrates that the VGG-5 + Adaptive-SpinalNet model outperforms all other models in terms of accuracy on the MNIST and KMNIST datasets. Although its accuracy on the QMNIST and Fashion MNIST datasets remains slightly below the VGG-5 SpinalNet and CNN + QIESOA models, the differences are insignificant, indicating that our model performs well in handling data variability. The Adaptive-SpinalNet model has the fewest parameters compared to other models, except for the PMM model on the MNIST dataset. This proves the effectiveness of the Adaptive Kernel in reducing computational complexity in the SpinalNet model with minimal accuracy trade-offs. This performance is achieved through the Dynamic Kernel, which dynamically adjusts weights to extract relevant features, while SpinalNet processes features in independent segments to enhance flexibility and computational efficiency. With a low parameter count, Adaptive-SpinalNet demonstrates strong generalization across various datasets, making it suitable for real-world applications involving diverse data. In addition to the appropriate selection of hyperparameters, the performance achieved by Adaptive-SpinalNet is also attributed to the optimal sizing of the Dynamic Kernel. The kernel size significantly affects the model's adaptability. To clarify this, Table 6 presents the model's performance trained on the MNIST dataset using different Dynamic Kernel sizes.

Table 6: Comparison of adaptive-spinalnet model performance on the MNIST dataset based on kernel size

| Ukuran Kernel | Recall | Precision | F1-Score | Accuracy |
|---|---|---|---|---|
| (3x3) | 98.91% | 98.91% | 98.91% | 98.91% |
| (5x5) | 98.93% | 98.93% | 98.93% | 98.93% |
| (7x7) | 98.80% | 98.80% | 98.80% | 98.80% |
| (9x9) | 98.38% | 98.38% | 98.38% | 98.38% |

The results in Table 6 show a performance improvement when the kernel size is increased from (3x3) to (5x5). This suggests that enlarging the kernel size in the Dynamic Kernel can enhance performance. However, when the kernel size is further increased to (9x9), performance decreases. A larger Dynamic Kernel does not necessarily guarantee an improvement in model performance, as a very large kernel tends to aggregate information over a larger area, potentially overlooking important small or local patterns. In addition to this finding, another interesting observation from the comparison in Table 6 is the consistency between precision, recall, F1-Score, and accuracy. Identical values for precision, recall, and F1-score indicate that our model works effectively, achieves an optimal balance, and handles class distribution well. This demonstrates that our model performs well on the MNIST dataset.

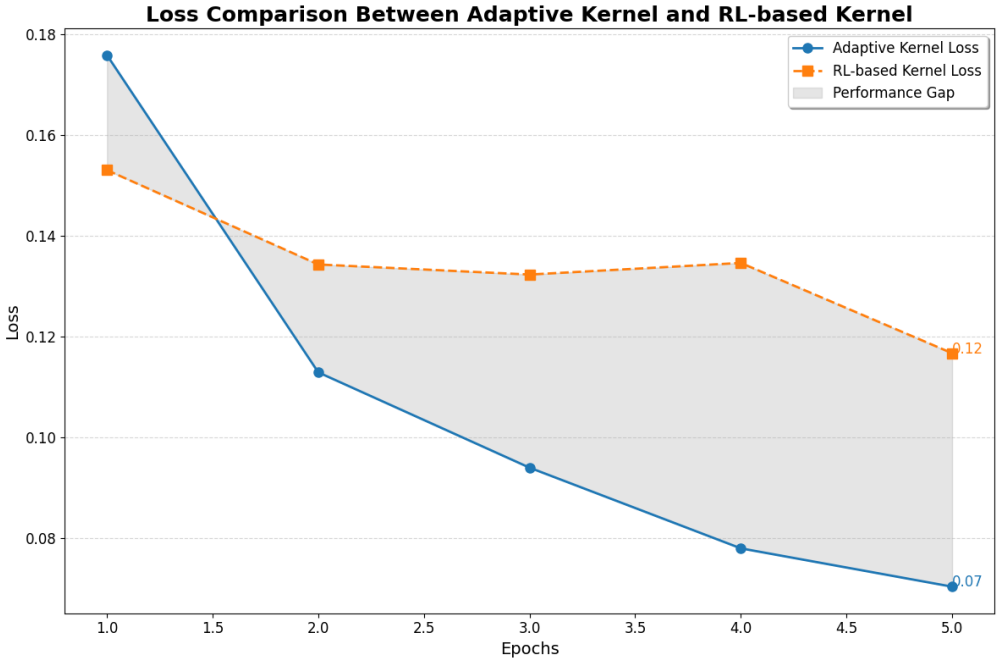
Another option that can be used as an adaptation method for the SpinalNet model is Reinforcement Learning (RL)--based adaptivity, which can be used to select or adjust kernels based on feedback from the environment to optimize performance. While this method may have the potential to adjust kernels based on experience, weaknesses such as computational overhead, dependence on reward design, and stability issues make it less ideal for high-efficiency real-time applications. The performance comparison between the Adaptive Kernel and RL methods in Table 7 demonstrates this.

Table 7: Performance comparison of adaptive kernel and rl methods on the spinalnet model using the MNIST dataset

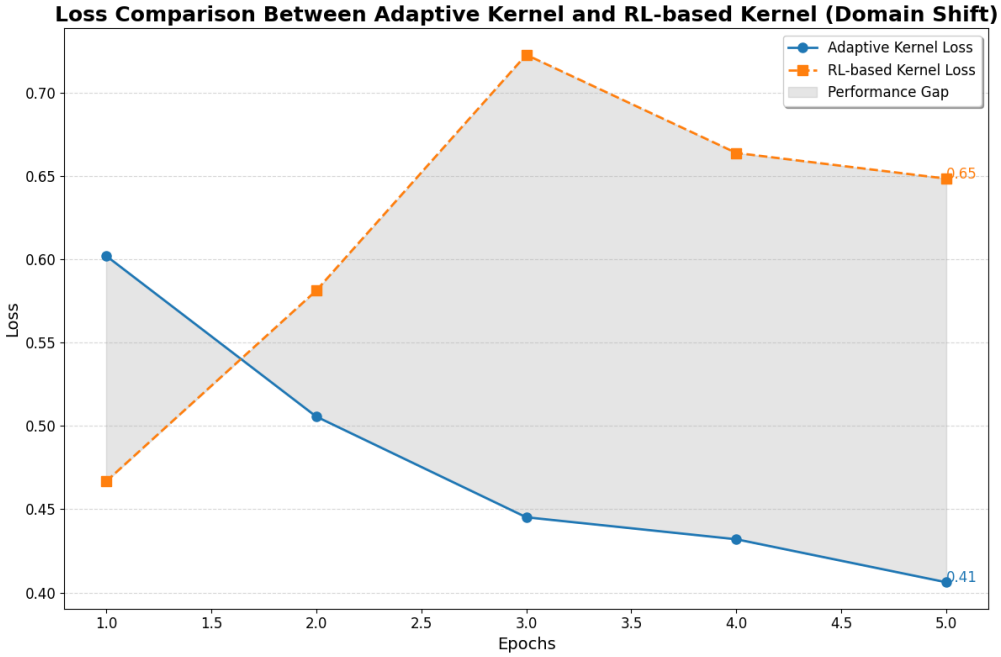
| Method | Epoch | Acc (%) | Inference Time (s) | Domain Shift Acc (%) |
|---|---|---|---|---|
| Adaptive Kernel | 5 | 97.85 | 3.42 | 88.97 |
| Reinforcement Learning-Based Adaptivity | 5 | 96.67 | 5.65 | 85.74 |

The comparison results in Table 7 show that the Adaptive Kernel method has a significant advantage over the Reinforcement Learning-based Adaptivity approach in terms of accuracy, inference time efficiency, and handling domain shift. Both methods were tested with five training epochs, with the Adaptive Kernel method achieving an accuracy of 97.85%, higher than the RL-based method, which only reached 96.67%. Furthermore, the inference time of the Adaptive Kernel is much faster, at 3.42 seconds, compared to 5.65 seconds for the RL method.

This indicates that the Adaptive Kernel method is more efficient for real-time applications than the RL-based adaptivity method. To further test the adaptability, we performed data augmentation for domain shift, which included random image rotation of up to 30°, brightness variation, and contrast changes. The evaluation results showed that the Adaptive Kernel's adaptation to domain shift was also superior, with an accuracy of 88.97% compared to 85.74% for the RL-based method. These results confirm that the direct adaptation mechanism of the Adaptive Kernel is more effective and efficient than the RL-based exploration, making it more suitable for adaptive vision systems that require high performance and resilience to data distribution changes. Another advantage is shown in the loss values generated at each epoch. Although the Adaptive Kernel method has a higher loss value than the RL-based adaptivity method in the first epoch, in subsequent epochs, the loss values for the proposed method consistently stay lower than those of the RL method. The comparison of loss values is shown in Figure 9.




(a)




(b)

Figure 9: Comparison of Loss Values Between Adaptive Kernel and RL-based Adaptivity. (a) Original MNIST Test Data, (b) Augmented MNIST Test Data

**Threats to validity**

a. Pre-Liminaries validity
   Validity in the preliminaries stage is measured by looking at the problem domain, which is understood as the clarity of data rows, datasets, and pre-processing. The transformation from non-linear to linearly separable transforms the data to a higher level by adding features using kernel functions. The raw data consists of various variants of the MNIST dataset, including MNIST itself, KMNIST, QMNIST, and Fashion-MNIST. Data identification is carried out to ensure that the pre-processing process in the model preparation stage is carried out correctly. This is a preparation stage for implementing the model as contextual knowledge. The data used is not too large. This was done to see how the model could be used with a limited amount of data but with high accuracy.

b. Fitting validity
   In the evaluation of our research, the process of determining the model is carried out using cross-validation and a confusion matrix. Validation is done by estimating the error and how our model can accommodate the unseen data. K-fold cross-validation is used to reduce parts that cause underfitting. By reducing training data, it is possible to lose trends in the data set, increasing the error caused by bias. The validation used is cross-validation, which generalizes the independent/unseen data set.At the validation stage, our learning self-learning model ensures that each process is carried out with attention to the evaluation of metrics, how to handle overfitting, and processes to reduce bias.

c. Bias validity

   Measurement of the accuracy of each model is optimized by optimization of Stochastic Gradient Descent (SGD) and Cross Entropy Loss. To eliminate the habit of estimating gradients, SGD is required to reduce the cost of each iteration. The computing cost of each iteration will run linearly from O(n) to O(1). In determining the SGD variable, the learning rate affects the resolution of the conflicting goal by reducing the learning rate dynamically as optimization progresses. Cross entropy is determined to define the loss function in optimization. This is done by minimizing the cross entropy. Defining cross entropy indirectly proves the equivalence of the relationship between objects. Done as long as the entropy data is constant..

# 5 Conclusion dan further studies

This study introduces a self-learning model for pattern recognition in vision systems. The model is developed through a self-adaptation approach where the system is regarded as a dynamic composition of the learning process. The goal is to enhance the system with self-learning capabilities, enable it to learn from collected visual data and develop its learning process autonomously. Our model encompasses three main components, namely (a) domain model to capture high-level representations of visual cues, (b) contextual knowledge representing context attributes relevant to the current dynamic context of visual cues, and (c) adaptive learner performing learning and recognizing visual cue representations based on the prevailing context. This model is prepared with a formulation combining the adaptive kernel method on the CNN architecture and the utilization of SpinalNet in the fully connected layer of the CNN.

The validity of the proposed model was evaluated using cross-validation with several testing schemes. In addition to the evaluation results compared with the original SpinalNet model, we also validated the model by comparing its performance through evaluations with methods used in related studies, varying kernel sizes, and comparisons with other adaptation methods. The evaluation results indicate that the proposed model performs very well regarding accuracy and computational complexity. The results of this work pave the way for future studies. In other words, future studies can include developing and expanding our proposed model for other domain needs (e.g., audio recognition, machine translation, and so on).

# References

[1] A. Aradea, I. Supriana, and K. Surendro, “Self-adaptive model based on goal-oriented requirements engineering for handling service variability,” *Journal of Information and Communication Technology*, vol. 19, no. 2, pp. 225–250, 2020, doi: 10.32890/jict2020.19.2.4.

[2] Aradea, I. Supriana, and K. Surendro, “ARAS: adaptation requirements for adaptive systems,” *Automated Software Engineering*, vol. 30, no. 1, p. 2, 2022, doi: 10.1007/s10515-022-00369-3.

[3] D. Weyns, “Wave VII: Learning from Experience,” in *An Introduction to Self-adaptive Systems: A Contemporary Software Engineering Perspective*, 2020, pp. 201–226. doi: 10.1002/9781119574910.ch10.

[4] A. Mollajan, A. Shahdadi, A. Ashofteh, F. Hamedani-KarAzmoudehFar, and S. H. Iranmanesh, “System Adaptability Enhancement Based on Improving the System Reconfigurability by Modularization of the System Architecture,” 2023. doi: 10.2139/ssrn.4519777.

[5] M. Bhadra, D. S. Lopera, R. Kunzelmann, and W. Ecker, “ A Model-Driven Architecture Approach to Accelerate Software Code Generation ,” in *2024 7th International Conference on Software and System Engineering (ICoSSE)* , Los Alamitos, CA, USA: IEEE Computer Society, Apr. 2024, pp. 23–30. doi: 10.1109/ICoSSE62619.2024.00012.

[6] M. Huisman, J. N. van Rijn, and A. Plaat, “A survey of deep meta-learning,” *Artif Intell Rev*,

vol. 54, no. 6, pp. 4483–4541, Aug. 2021, doi: 10.1007/s10462-021-10004-4.

[7] T. Gong, X. Zheng, and X. Lu, "Meta Self-Supervised Learning for Distribution Shifted Few-Shot Scene Classification," *IEEE Geoscience and Remote Sensing Letters*, vol. 19, pp. 1–5, 2022, doi: 10.1109/LGRS.2022.3174277.

[8] P. Terhörst, M. Huber, N. Damer, F. Kirchbuchner, K. Raja, and A. Kuijper, "Pixel-Level Face Image Quality Assessment for Explainable Face Recognition," *IEEE Trans Biom Behav Identity Sci*, vol. 5, no. 2, pp. 288–297, 2023, doi: 10.1109/TBIOM.2023.3263186.

[9] S. Malakar, W. Chiracharit, and K. Chamnongthai, "Masked Face Recognition With Generated Occluded Part Using Image Augmentation and CNN Maintaining Face Identity," *IEEE Access*, vol. 12, pp. 126356–126375, 2024, doi: 10.1109/ACCESS.2024.3446652.

[10] H.-I. Kim, K. Yun, and Y. M. Ro, "Face Shape-Guided Deep Feature Alignment for Face Recognition Robust to Face Misalignment," *IEEE Trans Biom Behav Identity Sci*, vol. 4, no. 4, pp. 556–569, 2022, doi: 10.1109/TBIOM.2022.3213845.

[11] D. Reis, J. Kupec, J. Hong, and A. Daoudi, "Real-Time Flying Object Detection with YOLOv8," May 2023, doi: https://doi.org/10.48550/arXiv.2305.09972.

[12] I. H. Sarker, "Deep Learning: A Comprehensive Overview on Techniques, Taxonomy, Applications and Research Directions," Nov. 01, 2021, *Springer*. doi: 10.1007/s42979-021-00815-1.

[13] C. Xu, "Applying MLP and CNN on Handwriting Images for Image Classification Task," in *2022 5th International Conference on Advanced Electronic Materials, Computers and Software Engineering (AEMCSE)*, 2022, pp. 830–835. doi: 10.1109/AEMCSE55572.2022.00167.

[14] L. Kurniasari and A. Setyanto, "Sentiment Analysis using Recurrent Neural Network," in *Journal of Physics: Conference Series*, Institute of Physics Publishing, Mar. 2020. doi: 10.1088/1742-6596/1471/1/012018.

[15] G. Priyadharshini and D. R. Judie Dolly, "Comparative Investigations on Tomato Leaf Disease Detection and Classification Using CNN, R-CNN, Fast R-CNN and Faster R-CNN," in *2023 9th International Conference on Advanced Computing and Communication Systems (ICACCS)*, 2023, pp. 1540–1545. doi: 10.1109/ICACCS57279.2023.10112860.

[16] D. Soydaner, "A Comparison of Optimization Algorithms for Deep Learning," *Intern J Pattern Recognit Artif Intell*, vol. 34, no. 13, p. 2052013, Dec. 2020, doi: 10.1142/S0218001420520138.

[17] G. L. Sree and R. Baskar, "Performance Analysis of CNN Algorithm in Comparison with LR algorithm for Face Recognition in Smart-Lock," in *2024 International Conference on Trends in Quantum Computing and Emerging Business Technologies*, 2024, pp. 1–5. doi: 10.1109/TQCEBT59414.2024.10545038.

[18] P. Wei, M. Shang, J. Zhou, and X. Shi, "Efficient adaptive learning rate for convolutional neural network based on quadratic interpolation egret swarm optimization algorithm," *Heliyon*, vol. 10, no. 18, Sep. 2024, doi: 10.1016/j.heliyon.2024.e37814.

[19] J. Luo and D. Hu, "An Image Classification Method Based on Adaptive Attention Mechanism and Feature Extraction Network," *Comput Intell Neurosci*, vol. 2023, no. 1, Jan. 2023, doi: 10.1155/2023/4305594.

[20] Y. Jiang, J. Xie, and D. Zhang, "An Adaptive Offset Activation Function for CNN Image Classification Tasks," *Electronics (Switzerland)*, vol. 11, no. 22, Nov. 2022, doi: 10.3390/electronics11223799.

[21] W. Wu and Y. Pan, "Adaptive Modular Convolutional Neural Network for Image Recognition," *Sensors*, vol. 22, no. 15, Aug. 2022, doi: 10.3390/s22155488.

[22] W. Guo *et al.*, "A Memristor-Based Adaptive Pooling Network for Cnn Optimization," 2023. doi: 10.2139/ssrn.4648000.

[23] H. M. D. Kabir *et al.*, "SpinalNet: Deep Neural Network With Gradual Input," *IEEE Transactions on Artificial Intelligence*, vol. 4, no. 5, pp. 1165–1177, Oct. 2023, doi: 10.1109/TAI.2022.3185179.

[24] Aradea, I. Supriana, K. Surendro, and I. Darmawan, "Integration of Self-adaptation Approach on Requirements Modeling," in *Recent Advances on Soft Computing and Data Mining*, T. Herawan, R. Ghazali, N. M. Nawi, and M. M. Deris, Eds., Cham: Springer International Publishing, 2017, pp. 233–243, doi: 10.1007/978-3-319-51281-5_24.

[25] Aradea, I. Supriana, and K. Surendro, "Self-adaptive software modeling based on contextual requirements," *Telkomnika (Telecommunication Computing Electronics and Control)*, vol. 16, no. 3, pp. 1276–1288, 2018, doi: 10.12928/TELKOMNIKA.v16i3.7032.

[26] A. Aradea, R. Rianto, and H. Mubarok, "Inference Model for Self-Adaptive IoT Service Systems," *International Journal of Intelligent Engineering and Systems*, vol. 14, no. 4, pp. 337–349, 2021, doi: 10.22266/ijies2021.0831.30.

[27] Aradea, Rianto, and H. Mubarok, "Cultivating Service Knowledge Models for IoT-Based Systems Adaptability," *Informatica (Slovenia)*, vol. 46, no. 5, pp. 115–122, 2022, doi: 10.31449/inf.v46i5.3874.

[28] M. Acheli, D. Grigori, and M. Weidlich, "Discovering and Analyzing Contextual Behavioral Patterns From Event Logs," *IEEE Trans Knowl Data Eng*, vol. 34, no. 12, pp. 5708–5721, 2022, doi: 10.1109/TKDE.2021.3077653.

[29] B. Yang, W. Wu, Y. Liu, and H. Liu, "A Novel Sleep Stage Contextual Refinement Algorithm Leveraging Conditional Random Fields," *IEEE Trans Instrum Meas*, vol. 71, pp. 1–13, 2022, doi: 10.1109/TIM.2022.3154838.

[30] W. Zhao, S. Peng, J. Chen, and R. Peng, "Contextual-Aware Land Cover Classification With U-Shaped Object Graph Neural Network," *IEEE Geoscience and Remote Sensing Letters*, vol. 19, pp. 1–5, 2022, doi: 10.1109/LGRS.2022.3177778.

[31] L. Deng, "The MNIST Database of Handwritten Digit Images for Machine Learning Research [Best of the Web]," *IEEE Signal Process Mag*, vol. 29, no. 6, pp. 141–142, 2012, doi: 10.1109/MSP.2012.2211477.

[32] N. Saqib, K. F. Haque, V. P. Yanambaka, and A. Abdelgawad, "Convolutional-Neural-Network-Based Handwritten Character Recognition: An Approach with Massive Multisource Data," *Algorithms*, vol. 15, no. 4, Apr. 2022, doi: 10.3390/a15040129.

[33] M. Anwar, H. M. Ali, M. A. Hossain, and A. Mohon, "Recognition of Handwritten Digit using Convolutional Neural Network (CNN)",Global Journal of Computer Science and Technology, 2019, doi: 10.34257/gjcstdvol19is2pg27.

[34] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," in *3rd International Conference on Learning Representations (ICLR 2015)*, San Diego: Computational and Biological Learning Society, Sep. 2014, pp. 1–14. [Online]. Available: https://doi.org/10.48550/arXiv.1409.1556

[35] P. I. Kaplanoglou and K. Diamantaras, "Learning local discrete features in explainable-by-design convolutional neural networks," *arXiv preprint arXiv:2411.00139*, Oct. 2024, doi: https://doi.org/10.48550/arXiv.2411.00139

[36] M. Bukowski, I. Antoniuk, and J. Kurek, "Improved efficient capsule network for Kuzushiji-MNIST benchmark dataset classification," *Bulletin of the Polish Academy of Sciences: Technical Sciences*, vol. 71, no. 6, 2023, doi: 10.24425/bpasts.2023.147338.

[37] P. Cook, D. Jammooa, M. Hjorth-Jensen, D. D. Lee, and D. Lee, "Parametric Matrix Models," *arXiv preprint arXiv:2401.11694*, Jan. 2024, [Online]. Available: https://doi.org/10.48550/arXiv.2401.11694