A Comparative Study of CNN and Prototypical Networks for MNIST and Thai Handwritten Digit Recognition under Small Training Sample Conditions

Yoshihiro Mitani
KOSEN, King Mongkut's University of Technology Thonburi
Bangkok, Thailand

Kitsada Doungjitjaroen
Engineering-Science College
King Mongkut's University of Technology Thonburi
Bangkok, Thailand

Yusuke Fujita
Graduate School of Sciences and Technology for Innovation
Yamaguchi University
Ube, Japan

***Abstract*—This study compares seven classifier settings for handwritten digit recognition with limited training data: a conventional convolutional neural network (CNN), ordinary 1-shot, 5-shot, and 10-shot Prototypical Networks, and their full-prototype variants. In the full-prototype settings, episodic training is unchanged, while final prototypes use all available training embeddings. MNIST and Thai digits from Burapha-TH are evaluated with 100–400 training images and 400 test images per class over five holdout trials shared by all classifiers. Full prototypes achieve lower mean error rates than their ordinary K-shot counterparts in all evaluated conditions. ProtoNet-1shot-full has the lowest mean error rates on MNIST. For Thai digits with 100 training images per class, ProtoNet-10shot-full reduces the mean error from 37.85% for the CNN to 35.01%, with a statistically significant difference in an exploratory paired t-test ($p = 0.016$). Improvements vary across classes, and confusions among Thai classes 4, 5, 8, and 9 remain. These results support evaluating episodic support size and final prototype construction separately when training data are limited.**

***Index Terms*—handwritten digit recognition, convolutional neural network, prototypical network, few-shot learning, metric learning, Thai digit recognition, small training samples**

I. Introduction

CNNs have achieved strong performance in handwritten digit recognition, with MNIST serving as a standard benchmark [1]. Their performance with limited labeled data remains important, particularly for non-Latin scripts with smaller public datasets. In statistical pattern recognition, estimating classifier parameters becomes difficult when the training sample size is small relative to the feature dimensionality [2], [3]. Let N be the number of labeled training samples per class and d the feature dimensionality. Jain *et al*. emphasize their ratio and report $N/d > 10$ as a practical design guideline [4], citing Jain and Chandrasekaran [2]. This guideline is not a universal sample-size requirement for deep networks.

In this study, a small training sample situation is defined by $N/d < 1$, with d referring to the input feature dimensionality. A class covariance estimated from N centered training vectors has rank at most $N-1$; thus, it is singular under this condition. A statistical discriminant that directly requires its ordinary inverse cannot be constructed without a modification such as regularization or a generalized inverse. Here, the 28×28 inputs have $d = 784$, and $N = 100$, 200, 300, or 400 gives ratios from 0.128 to 0.510. All four settings therefore meet the definition. This dimensionality refers to input pixels, not the learned 64-dimensional embedding.

Few-shot learning offers several approaches to classification with limited labels [5]–[7], as reviewed in [8], [9]. Prototypical Networks learn an embedding space and represent each class by the mean embedding of its support samples [10]. We compare a CNN with ordinary K-shot and full-prototype settings on MNIST and Thai handwritten digits. The full setting retains K-shot episodic training but constructs final prototypes from all training images.

The contribution is an empirical comparison that:

- evaluates seven classifier settings using common balanced splits on two digit datasets;
- distinguishes episodic support size from final prototype size;
- identifies class-dependent gains and persistent confusions that limit a single-prototype representation.

These findings inform classifier and prototype selection under a limited labeling budget.

II. Related Work

CNNs learn spatial and hierarchical image features and provide an established baseline for handwriting recognition [1]. Few-shot approaches include Matching Networks [5], MAML [6], Relation Networks [7], and Prototypical Networks [10]. Surveys identify embedding quality and class representation as central issues in metric-based classification [8], [9]. Meta-Baseline demonstrates the effectiveness of simple classifiers on learned embeddings [11], while Sahay and Coustaty apply an enhanced Prototypical Network to handwritten Urdu characters and numerals [12]. Our full-prototype setting similarly uses

TABLE I
DATASETS USED IN THIS STUDY.

Dataset	Classes	Image size	Role
MNIST	10	28×28	Standard benchmark
Thai digits	10	28×28	Non-Arabic digit dataset

nearest-class-mean classification in an episodically learned embedding space.

Thai handwriting studies report substantial variation in writing styles and fine structural differences involving loops, curls, and concavities [13]. Shape similarity and handwriting variation also contribute to residual errors in CNN-based Thai digit recognition [14]. Burapha-TH provides Thai character, digit, and syllable images while retaining scanning noise and acquisition artifacts [15]. These properties motivate examining class-wise behavior as well as overall error rates.

III. Datasets and Experimental Conditions

A. Datasets

MNIST provides a standard handwritten digit reference [1], and Burapha-TH provides a public Thai handwriting resource with a different visual structure and acquisition process [15]. Thai numerals represent decimal values 0–9 and remain part of the school curriculum [16]. Figure 1 shows examples; class labels refer to the numerical values of the Thai glyphs. Both datasets are treated as 10-class problems with 28×28 grayscale inputs (Table I).

The MNIST pool combines its original training and test partitions, totaling 70,000 images. The reported results therefore use new holdout splits rather than the official MNIST test partition. The Thai pool contains 10,673 digit images, with 972–1123 images per class, labeled from the source directory names. Thai images are converted to grayscale and resized directly using bilinear interpolation. Pixel values are divided by 255 for both datasets. The selected budget of up to 400 training and 400 test images per class is feasible for every Thai class.

B. Experimental Protocol

For each dataset, five random image-level holdout trials are used. In each class, 400 images form a training pool and another 400 form a non-overlapping test set. Nested training subsets contain 100, 200, 300, or 400 images per class. All seven classifiers share the same saved subsets and test set within a trial. Classes are balanced, and all ten labels occur in training and testing; this is a closed-set comparison rather than novel-class evaluation.

Test images are excluded from encoder training and prototype construction. The split unit is the source image index, without grouping by writer or image-content identity. Different trials may reuse images. Mean error rates and sample standard deviations are computed over the five trials. Table II and Fig. 2 summarize the protocol [17], [18].

TABLE II
EXPERIMENTAL CONDITIONS.

Item	Setting
Training samples/class	100, 200, 300, 400
Test samples/class	400
Number of classes	10
Class balance	Equal (empirical prior: 0.1/class)
Number of holdout trials	5 (common splits)
Image size	28×28 pixels
Raw input dimensionality	784
Data augmentation	Not used
Dropout	Not used

TABLE III
THE SEVEN CLASSIFIER SETTINGS. HERE N IS THE AVAILABLE TRAINING SIZE PER CLASS. THE CNN PREDICTS WITH A SOFTMAX OUTPUT AND HAS NO CLASS PROTOTYPES.

Classifier	Episodic support size K	Final prototype images/class
CNN	–	–
ProtoNet-1shot	1	1
ProtoNet-5shot	5	5
ProtoNet-10shot	10	10
ProtoNet-1shot-full	1	N
ProtoNet-5shot-full	5	N
ProtoNet-10shot-full	10	N

IV. Methods

Table III lists the seven settings: one CNN and ordinary/full Prototypical Networks for $K = 1$, 5, and 10. Ordinary and full variants differ in the images used to construct final prototypes.

A. CNN Classifier

The CNN is trained from scratch using the architecture in Table IV: three convolutional layers, two max-pooling layers, a 64-unit ReLU layer, and a 10-unit softmax output. Training uses Adam [19], 50 epochs, a learning rate of 10^{-3}, and a batch size of 32. Data augmentation and dropout are not used.

B. K-shot Prototypical Network

The encoder maps an image x to an embedding $f_\theta(x)$. Each training episode draws K support images and 10 query images

TABLE IV
CNN ARCHITECTURE.

Layer	Output shape	Setting
Input	$28 \times 28 \times 1$	Grayscale image
Conv2D	$26 \times 26 \times 32$	3×3, ReLU
MaxPooling	$13 \times 13 \times 32$	2×2
Conv2D	$11 \times 11 \times 64$	3×3, ReLU
MaxPooling	$5 \times 5 \times 64$	2×2
Conv2D	$3 \times 3 \times 64$	3×3, ReLU
Flatten	576	–
Dense	64	ReLU
Dense	10	Softmax

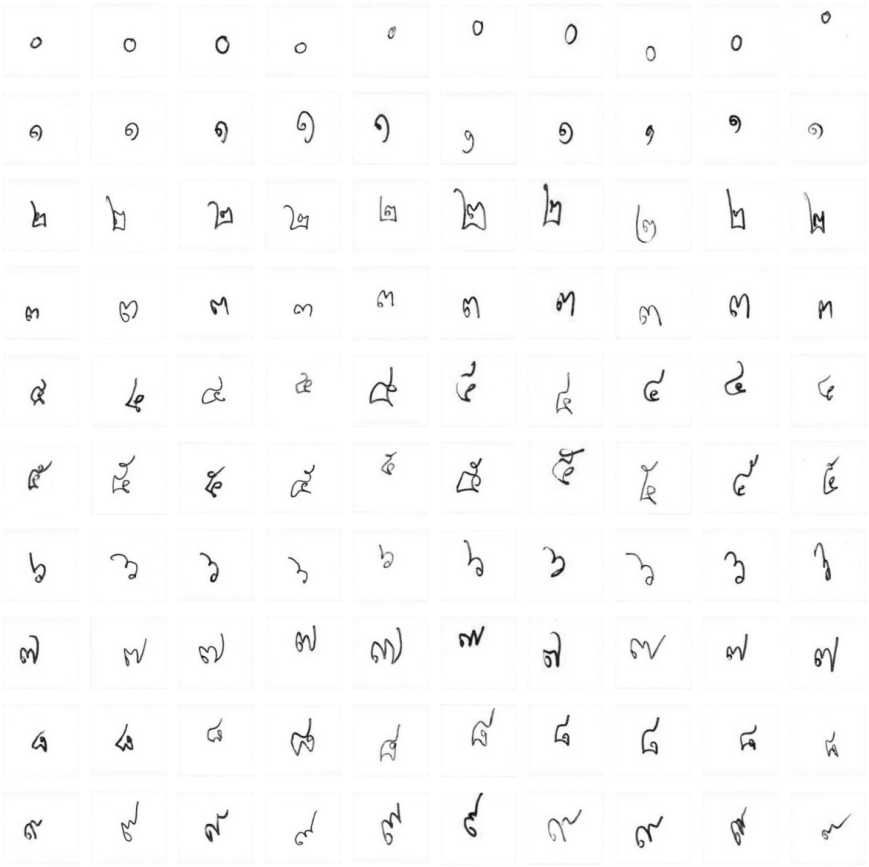


Fig. 1. Examples of Thai handwritten digit images. The rows from top to bottom correspond to the Thai digit classes representing the numerical values 0 through 9, respectively, and each row shows ten handwriting samples.

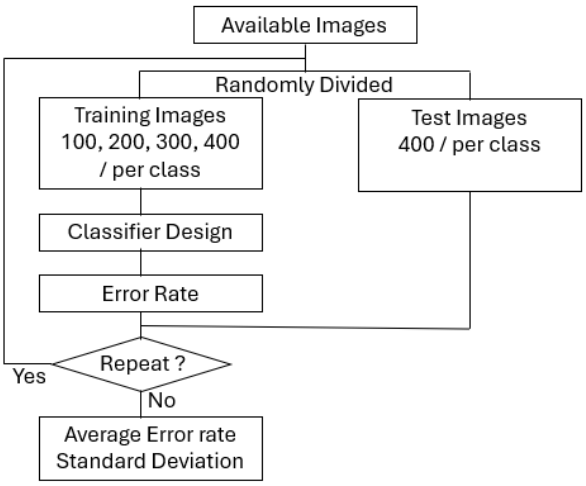


Fig. 2. Flow of error rate estimation using the holdout method.

per class from the training set. The prototype of class k is

$$c_k = \frac{1}{|S_k|} \sum_{(x_i, y_i) \in S_k} f_\theta(x_i), \tag{1}$$

where S_k is its support set. Classification uses squared Euclidean distance:

$$\hat{y} = \arg\min_k \|f_\theta(x) - c_k\|^2. \tag{2}$$

During training, negative distances serve as logits for the cross-entropy loss on query images. Support and query images are sampled without replacement in each episode; the held-out test set is used only for final evaluation.

The encoder uses the CNN's convolutional layers and flattening operation, followed by a 64-unit linear embedding layer. It is trained for 50 epochs of 50 episodes each (Table V). Thus, the encoder can draw on the whole training split even though each episode uses only K support images per class. The CNN and ProtoNet differ in both their training objective and output structure.

C. Full-Prototype Setting

After training, ordinary K-shot classification randomly selects K training images per class without replacement to construct final prototypes. Full-prototype classification instead uses all training images:

$$c_k^{\text{full}} = \frac{1}{|T_k|} \sum_{(x_i, y_i) \in T_k} f_\theta(x_i), \tag{3}$$

TABLE V
PROTOTYPICAL NETWORK SETTINGS.

Item	Setting
K-shot settings	1-shot, 5-shot, 10-shot
Number of classes/episode	10
Query samples/class/episode	10
Embedding dimension	64
Epochs	50
Episodes/epoch	50
Optimizer	Adam
Learning rate	10^{-3}
Distance	Squared Euclidean

TABLE VI
MEAN ERROR RATES AND STANDARD DEVIATIONS ON MNIST. UPPER ROWS: MEAN ERROR RATE (%); LOWER ROWS: STANDARD DEVIATION.

	Number of training samples per class			
Classifier	100/class	200/class	300/class	400/class
CNN	4.68	3.31	2.54	2.21
	0.38	0.29	0.06	0.13
ProtoNet-1shot	4.96	3.51	2.87	2.71
	0.61	0.58	0.60	0.30
ProtoNet-5shot	4.34	3.43	3.06	2.61
	0.42	0.32	0.50	0.29
ProtoNet-10shot	4.52	3.47	2.74	2.42
	0.62	0.57	0.18	0.31
ProtoNet-1shot-full	3.26	2.25	2.23	1.85
	0.33	0.15	0.36	0.29
ProtoNet-5shot-full	3.79	3.04	2.70	2.28
	0.28	0.40	0.38	0.22
ProtoNet-10shot-full	4.18	3.16	2.52	2.24
	0.71	0.46	0.26	0.41

TABLE VII
MEAN ERROR RATES AND STANDARD DEVIATIONS ON THAI HANDWRITTEN DIGITS. UPPER ROWS: MEAN ERROR RATE (%); LOWER ROWS: STANDARD DEVIATION.

	Number of training samples per class			
Classifier	100/class	200/class	300/class	400/class
CNN	37.85	25.21	20.42	17.42
	1.43	0.63	2.69	2.12
ProtoNet-1shot	45.72	37.19	35.05	36.38
	4.50	3.84	2.42	3.16
ProtoNet-5shot	37.18	28.31	23.25	21.22
	1.34	1.53	1.50	0.82
ProtoNet-10shot	35.84	26.14	21.44	19.16
	1.90	1.51	1.19	1.80
ProtoNet-1shot-full	41.27	28.64	24.62	24.60
	3.04	1.58	1.01	1.44
ProtoNet-5shot-full	36.04	26.67	20.04	17.32
	1.47	0.86	0.94	1.47
ProtoNet-10shot-full	35.01	24.88	19.48	17.61
	2.08	2.16	1.19	1.14

TABLE VIII
EXPLORATORY TWO-SIDED PAIRED T-TEST FOR THAI DIGITS WITH 100 TRAINING IMAGES PER CLASS.

Item	Value
CNN mean error rate	37.85%
ProtoNet-10shot-full mean error rate	35.01%
Mean difference	2.85 percentage points
SD of differences	1.59 percentage points
t statistic	4.01
Degrees of freedom	4
p value	0.016

where T_k contains the training samples of class k. Prediction uses the nearest prototype. For a given K, both variants use the same encoder architecture and episodic training procedure; only final prototype construction changes. All models use fixed training schedules without validation-based selection or early stopping.

V. EXPERIMENTAL RESULTS

A. MNIST Results

Table VI reports mean error rates and sample standard deviations in the upper and lower rows of each cell. Errors generally decrease with more training images. ProtoNet-1shot-full has the lowest mean error rates at all four training sizes: 3.26%, 2.25%, 2.23%, and 1.85%, compared with 4.68%, 3.31%, 2.54%, and 2.21% for the CNN. Each full-prototype setting has a lower mean than its ordinary K-shot counterpart.

B. Thai Handwritten Digit Results

Thai digit errors are higher than MNIST errors under the same protocol (Table VII). ProtoNet-10shot-full has the lowest mean error rates at 100–300 training images per class. At 100 images, its mean is 35.01%, compared with 37.85% for the CNN and 45.72% for ordinary ProtoNet-1shot. At 400 images, ProtoNet-5shot-full has the lowest mean (17.32%), close to the CNN (17.42%) and ProtoNet-10shot-full (17.61%). Full prototypes improve on their ordinary counterparts throughout, but their advantage over the CNN depends on training size.

C. Paired Comparison

For Thai digits with 100 training images per class, the CNN and ProtoNet-10shot-full errors are paired by their common trial splits. The mean reduction, computed before rounding, is 2.85 percentage points. A two-sided paired t-test gives $t(4) = 4.01$ and $p = 0.016$, indicating a significant difference at the 5% level for this comparison (Table VIII).

This comparison is exploratory, and no correction for multiple comparisons is applied. The result provides supporting evidence under the paired t-test assumptions; only five trials are available, and images may recur across trials.

D. Confusion Matrix Analysis

Tables IX and X show averaged normalized confusion matrices for Thai digits with 100 training images per class. Rows denote true classes and columns predicted classes.

For the CNN, classes 0 and 1 are frequently confused: the 0-to-1 and 1-to-0 misclassification rates are 0.21 and 0.17. Confusions also occur among classes 4, 5, 8, and 9. Compared with the CNN, ProtoNet-10shot-full increases the correct classification rates from 0.55 to 0.77 for class 0 and from 0.38 to 0.52 for class 4. However, the class-5-to-4 error rises from 0.16 to 0.29, while class-8-to-4 confusion remains about 0.16–0.17. For class 9, confusion with class 5 falls from

TABLE IX
AVERAGED NORMALIZED CONFUSION MATRIX OF CNN FOR THAI HANDWRITTEN DIGITS WITH 100 TRAINING SAMPLES PER CLASS.

True \ Pred.	0	1	2	3	4	5	6	7	8	9
0	0.55	0.21	0.04	0.07	0.02	0.01	0.02	0.01	0.06	0.00
1	0.17	0.63	0.06	0.05	0.01	0.02	0.05	0.01	0.01	0.00
2	0.01	0.03	0.80	0.03	0.00	0.01	0.10	0.00	0.01	0.00
3	0.08	0.05	0.04	0.73	0.01	0.00	0.01	0.04	0.02	0.01
4	0.02	0.02	0.02	0.01	0.38	0.16	0.01	0.05	0.20	0.14
5	0.01	0.01	0.02	0.00	0.16	0.60	0.02	0.01	0.05	0.11
6	0.01	0.05	0.11	0.01	0.00	0.02	0.79	0.00	0.00	0.00
7	0.02	0.00	0.01	0.06	0.06	0.03	0.00	0.69	0.08	0.04
8	0.07	0.03	0.01	0.03	0.16	0.04	0.00	0.08	0.51	0.07
9	0.01	0.00	0.00	0.01	0.13	0.15	0.01	0.08	0.07	0.53

TABLE X
AVERAGED NORMALIZED CONFUSION MATRIX OF PROTONET-10SHOT-FULL FOR THAI HANDWRITTEN DIGITS WITH 100 TRAINING SAMPLES PER CLASS.

True \ Pred.	0	1	2	3	4	5	6	7	8	9
0	0.77	0.11	0.00	0.03	0.01	0.00	0.01	0.00	0.06	0.01
1	0.17	0.69	0.01	0.05	0.03	0.00	0.02	0.00	0.03	0.00
2	0.01	0.05	0.72	0.05	0.02	0.01	0.09	0.04	0.02	0.00
3	0.05	0.04	0.03	0.76	0.00	0.00	0.00	0.05	0.06	0.00
4	0.02	0.03	0.01	0.00	0.52	0.16	0.00	0.02	0.16	0.09
5	0.00	0.01	0.01	0.00	0.29	0.59	0.00	0.01	0.04	0.03
6	0.04	0.10	0.08	0.02	0.02	0.00	0.71	0.00	0.02	0.00
7	0.01	0.01	0.06	0.10	0.02	0.01	0.00	0.64	0.11	0.03
8	0.06	0.03	0.01	0.02	0.17	0.03	0.00	0.05	0.56	0.07
9	0.02	0.01	0.00	0.01	0.16	0.04	0.00	0.06	0.14	0.55

0.15 to 0.04, but confusion with class 8 rises from 0.07 to 0.14. The lower overall error therefore reflects substantial gains in some classes rather than uniform improvement.

VI. DISCUSSION

A. Prototype Construction

Using all training embeddings can reduce dependence on the particular final K-shot support subset. This offers a plausible explanation for the lower errors of full-prototype variants, although prototype variance was not measured directly. No additional labels are required beyond those used for encoder training.

The preferred episodic K differs between datasets: 1-shot-full has the lowest mean errors on MNIST, while 10-shot-full has the lowest mean errors on Thai digits at 100–300 images per class. More images for final prototype construction therefore need not imply a larger K during training. Full prototypes are useful when all labeled training images are available for final classification; ordinary K-shot classification applies when that reference set is restricted to K images per class.

B. Class-Dependent Errors

The Thai confusion matrices show that a better overall result can coexist with persistent or increased errors between particular classes. Thai handwriting varies in loops, tails, and stroke shape, and similar structures may make classes difficult to separate [13], [14]. A single mean embedding may not represent several distinct writing styles within a class. This could explain why full prototypes improve classes 0 and 4 while some confusions among classes 4, 5, 8, and 9 remain. It motivates evaluating multiple prototypes per class; the present results do not establish an image-specific cause for each error.

C. Scope and Limitations

The findings support full-prototype classification as an option under limited labeling budgets, with benefits that depend on the dataset, training size, and class. Comparison with the CNN does not isolate prototype averaging: the learning objectives differ, and 50 CNN epochs and 2500 ProtoNet episodes are not matched computational budgets. Stronger CNNs with regularization, augmentation, or transfer learning remain to be evaluated.

The higher Thai error should not be attributed solely to the numeral system. The datasets also differ in acquisition, handwriting variability, image polarity, and preprocessing; Burapha-TH retains acquisition artifacts [15], and resizing may remove fine details. Finally, the evaluation covers two datasets, balanced classes, and image-level splits. Writer-disjoint evaluation, class imbalance, and additional datasets are needed to assess broader applicability.

VII. CONCLUSION

Full-prototype variants achieved lower mean errors than their ordinary K-shot counterparts under all evaluated conditions. ProtoNet-1shot-full had the lowest mean errors on MNIST, while ProtoNet-10shot-full had the lowest mean errors on Thai digits at 100–300 training images per class. With 100 training images per class on the Thai dataset, it reduced the CNN mean error from 37.85% to 35.01%, with statistical

support from an exploratory paired t-test ($p = 0.016$). The benefits varied across classes, with persistent confusions among classes 4, 5, 8, and 9. These findings favor evaluating episodic support size and final prototype size separately. Multiple prototypes and broader evaluation conditions are promising directions for further study.

REFERENCES

[1] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998, doi: 10.1109/5.726791.

[2] A. K. Jain and B. Chandrasekaran, "Dimensionality and sample size considerations in pattern recognition practice," in *Handbook of Statistics*, P. R. Krishnaiah and L. N. Kanal, Eds. Amsterdam, The Netherlands: North-Holland, 1982, vol. 2, pp. 835–855, doi: 10.1016/S0169-7161(82)02042-2.

[3] S. J. Raudys and A. K. Jain, "Small sample size effects in statistical pattern recognition: Recommendations for practitioners and open problems," in *Proc. 10th International Conference on Pattern Recognition*, 1990, vol. 1, pp. 417–423, doi: 10.1109/ICPR.1990.118138.

[4] A. K. Jain, R. P. W. Duin, and J. Mao, "Statistical pattern recognition: A review," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 1, pp. 4–37, 2000, doi: 10.1109/34.824819.

[5] O. Vinyals, C. Blundell, T. Lillicrap, K. Kavukcuoglu, and D. Wierstra, "Matching networks for one shot learning," in *Advances in Neural Information Processing Systems*, vol. 29, 2016, pp. 3630–3638.

[6] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *Proc. International Conference on Machine Learning*, PMLR, vol. 70, 2017, pp. 1126–1135.

[7] F. Sung, Y. Yang, L. Zhang, T. Xiang, P. H. S. Torr, and T. M. Hospedales, "Learning to compare: Relation network for few-shot learning," in *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 1199–1208.

[8] Y. Liu, H. Zhang, W. Zhang, G. Lu, Q. Tian, and N. Ling, "Few-shot image classification: Current status and research trends," *Electronics*, vol. 11, no. 11, article 1752, 2022, doi: 10.3390/electronics11111752.

[9] X. Li, X. Yang, Z. Ma, and J.-H. Xue, "Deep metric learning for few-shot image classification: A review of recent developments," *Pattern Recognition*, vol. 138, article 109381, 2023, doi: 10.1016/j.patcog.2023.109381.

[10] J. Snell, K. Swersky, and R. Zemel, "Prototypical networks for few-shot learning," in *Advances in Neural Information Processing Systems*, vol. 30, 2017, pp. 4077–4087.

[11] Y. Chen, Z. Liu, H. Xu, T. Darrell, and X. Wang, "Meta-Baseline: Exploring simple meta-learning for few-shot learning," in *Proc. IEEE/CVF International Conference on Computer Vision*, 2021, pp. 9062–9071.

[12] R. Sahay and M. Coustaty, "An enhanced prototypical network architecture for few-shot handwritten Urdu character recognition," *IEEE Access*, vol. 11, pp. 33682–33696, 2023, doi: 10.1109/ACCESS.2023.3263721.

[13] O. Surinta, M. F. Karaaba, L. R. B. Schomaker, and M. A. Wiering, "Recognition of handwritten characters using local gradient feature descriptors," *Engineering Applications of Artificial Intelligence*, vol. 45, pp. 405–414, 2015, doi: 10.1016/j.engappai.2015.07.017.

[14] K. Khunratchasana and T. Treenuntharath, "Thai digit handwriting image classification with convolutional neural networks," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 27, no. 1, pp. 110–117, 2022, doi: 10.11591/ijeecs.v27.i1.pp110-117.

[15] A. Onuean, U. Buatoom, T. Charoenporn, T. Kim, and H. Jung, "Burapha-TH: A multi-purpose character, digit, and syllable handwriting dataset," *Applied Sciences*, vol. 12, no. 8, article 4083, 2022, doi: 10.3390/app12084083.

[16] Ministry of Education, Thailand, *Basic Education Core Curriculum B.E. 2551 (A.D. 2008)*. Bangkok, Thailand: Office of the Basic Education Commission, 2008.

[17] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, 2nd ed. New York, NY, USA: Wiley-Interscience, 2001.

[18] K. Fukunaga, *Introduction to Statistical Pattern Recognition*, 2nd ed. San Diego, CA, USA: Academic Press, 1990.

[19] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. International Conference on Learning Representations*, 2015.