Article

Enhancing handwritten numeric string recognition through incremental support vector machines

Rani Kurnia Putri[1], Muhammad Athoillah[2,*]

[1] Department of Mathematics Education, Universitas PGRI Adi Buana Surabaya, Jawa Timur 60234, Indonesia
[2] Department of Statistics, Universitas PGRI Adi Buana Surabaya, Jawa Timur 60234, Indonesia
* Corresponding author: Muhammad Athoillah, athoillah@unipasby.ac.id

Abstract: Handwritten digit recognition systems are integral to diverse applications such as postal services, banking, and document processing in our digitally-driven society. This research addresses the challenges posed by evolving datasets and dynamic scenarios in handwritten digit recognition by proposing an approach based on incremental support vector machines (ISVM). ISVM is an extension of traditional support vector machines (SVM) designed to handle scenarios where new data points become available over time. The dataset includes handwritten images (numbers "0" to "6") and trials introducing new classes ("7", "8", and "9"). Evaluation utilizes k-fold cross-validation for robustness. Digital image processing involves converting images into numeric data using the histogram method. The result showed the positive outcomes of using ISVM in handwritten digit recognition, emphasizing its adaptability to incremental learning and its ability to maintain robust performance in the face of evolving datasets, which is crucial for real-world applications.

Keywords: digital image processing; incremental learning; pattern recognition; dynamic machine learning

1. Introduction

The significance of handwritten digit recognition systems cannot be overstated, as they play a pivotal role in a myriad of critical applications spanning postal services, banking, and document processing. In an era of digital transformation, where vast amounts of information are conveyed through handwritten numeric strings, the accuracy and efficiency of recognition systems become paramount [1–3].

Despite considerable advancements in pattern recognition, the intricate nature of handwritten digits poses ongoing challenges [4]. Variations in writing styles, distortions introduced during the writing process, and inherent noise within the data contribute to the complexity of the recognition task [5]. Therefore, the continual enhancement of handwritten digit recognition systems becomes imperative to address these challenges and ensure the seamless integration of handwritten data into various facets of our digitally-driven society [6,7]. As such, this research endeavors to contribute to the refinement of recognition systems, specifically through the exploration of Incremental Support Vector Machines, aiming to not only overcome existing limitations but also elevate the adaptability and accuracy of these systems in the face of evolving datasets and dynamic real-world scenarios [8].

Conventional methods for recognizing handwritten digits commonly employ machine learning algorithms, and support vector machines (SVM) have demonstrated effectiveness in this application. SVMs are robust classifiers recognized for their capacity to manage high-dimensional data and intricate non-linear relationships. Numerous prior investigations have attested to the proficiency of SVM in successfully addressing the challenges associated with handwritten digit recognition [9,10]. For example, Mahmoud Shams et al. [11], demonstrate the collaborative impact of combining convolutional neural networks (CNN) and support vector machines (SVM) in recognizing Arabic handwritten characters. This fusion leverages the feature extraction strengths inherent in CNN and the discriminative prowess of SVM, with the overarching goal of improving the precision and efficiency of the character recognition process. Additional examples are observable in the findings derived from research carried out by Rajnoha et al. [12] and Katiyar et al. [13]. Both studies proved that SVM can recognize handwriting with fairly high accuracy. However, the incorporation of a more dynamic learning model for SVM, allowing the addition of new data without necessitating the repetition of prior learning, remains a persistent challenge in the current landscape.

This paper focuses on addressing the limitations of conventional SVM in the context of handwritten numeric string recognition by proposing an approach based on incremental support vector machines (ISVM). Incremental learning is essential in real-world scenarios where the system needs to adapt to accommodate additional data without retraining the entire model [14,15]. By exploring the potential of ISVM, we aim to enhance the adaptability and performance of handwritten numeric string recognition systems. The objective of this research is to investigate the feasibility and effectiveness of employing ISVM for continuous learning in the context of handwritten digit recognition. We aim to contribute to the existing body of knowledge by providing insights into the advantages of incremental learning techniques, specifically ISVM, in handling evolving datasets and improving the accuracy of handwritten numeric string recognition systems.

2. Related work

2.1. Dataset

In the initial phase of this study, the dataset employed consisted of handwritten images depicting numbers ranging from "0" to "6". The dataset comprised a total of 7000 images, with each category encompassing 500 images. The data was partitioned into 90% for training purposes and 10% for testing purposes. Additionally, to expand the scope of classes, three trials were conducted. The first trial introduced a new class featuring handwritten images of the number "7", totaling 750 images. The second trial introduced a class showcasing handwritten images of the number "8", comprising 500 images. The third trial added a class depicting handwritten images of the number "9", amounting to 250 images. Similar to the initial dataset, each of these new classes adhered to a distribution of 90% for training and 10% for testing.

To assess the effectiveness of the developed method, the researchers employed the k-fold cross-validation technique, a widely-used resampling method in machine

learning and statistical modeling for evaluating model performance and generalization [16]. In this process, the dataset was partitioned into ten subsets, where nine subsets were allocated for training (denoted as x) and one subset for testing (denoted as y) in each iteration. This iterative procedure was repeated ten times, ensuring varied combinations of training and testing data with each repetition. By adopting this approach, the research aimed to enhance the robustness of the assessment during the pattern recognition test process [17,18].

2.2. Digital image processing

Recognizing handwriting poses a challenge within the realm of automatic pattern recognition, particularly in the context of digital images. Digital image recognition focuses on developing algorithms and systems that enable machines to interpret and understand visual information within digital images. Through the application of machine learning algorithms during a training phase with labeled examples, the system learns to associate specific features with particular objects or patterns. Once trained, the system can classify or identify objects in new images [19,20].

In the context of our research, the process of converting images into numeric data using the histogram method is fundamental. The process involves representing the image as a histogram and extracting relevant statistical features. Here's a simplified representation of this process: Let I be the grayscale image with pixel intensities ranging from 0 to 255. The histogram, denoted as H, is constructed to represent the frequency distribution of pixel intensities in I.

$$H(i) = \frac{\text{Frequency of intensity } i}{\text{Total number of pixels in } I} \tag{1}$$

The normalized histogram H is then used to calculate statistical features. Let μ, σ, *skew*, and *kurt* represent the mean, standard deviation, skewness, and kurtosis, respectively:

$$\mu = \sum_{i=0}^{255} i.H(i) \tag{2}$$

$$\sigma = \sqrt{\sum_{i=0}^{255} (\mu - i)^2.H(i)} \tag{3}$$

$$skew = \frac{\sum_{i=0}^{255}(i - \mu)^3.H(i)}{\sigma^3} \tag{4}$$

$$kurt = \frac{\sum_{i=0}^{255}(i - \mu)^4.H(i)}{\sigma^4} \tag{5}$$

These statistical features, along with other relevant characteristics, can be organized into a one-dimensional vector V, serving as the numeric representation of the image:

$$V = [\mu, \sigma, skew, kurt, \dots] \tag{6}$$

This vector V captures essential information about the image's pixel intensity distribution, providing a basis for further analysis, such as machine learning tasks or quantitative assessments in the context of our research. This approach proves

particularly relevant in situations where pixel intensity distribution holds significant implications for the features we aim to extract and analyze in our research [21].

2.3. Incremental support vector machine

Incremental support vector machines are an extension of traditional support vector machines designed to handle scenarios where new data points become available over time. SVMs are powerful machine-learning models used for classification and regression tasks, particularly in high-dimensional spaces [22]. However, they traditionally require the entire dataset to be available at once for training. Incremental SVM addresses this limitation by allowing the model to be updated incrementally as new data points arrive. This is particularly useful in dynamic or streaming environments where the dataset is not fixed, and new observations are continually added. The incremental learning process involves updating the existing SVM model with the new data, ensuring that the model adapts to changes or additions in the dataset [23].

In SVM, finding the best hyperplane involves determining the optimal decision boundary that maximally separates different classes in the feature space. The best hyperplane is the one that maximizes the margin between the classes [24]. In mathematical terms, the expression for determining this hyperplane is formulated by the following equation [25].

$$\max_{\alpha} L_D = -\frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j + \sum_{i=1}^{n} \alpha_i \tag{7}$$

under the condition of

$$\sum_{i=1}^{n} \alpha_i y_i = 0, 0 \leq \alpha_i \leq C, \forall i \tag{8}$$

According to Zhang et al. [26], after running the SVM training, several data points x_i are obtained with $a_i = 0$, indicating that these data points lie outside the margin. Conversely, there are data points x_i that satisfy $0 \leq a_i \leq C$, signifying that these points precisely lie on the separating plane $f(x) = \pm 1$. Additionally, data points with $C = \alpha$ are situated within the margin and can be expressed as follows:

$$\alpha_i = 0 \rightarrow |f(x_i)| \geq 1 \tag{9}$$
$$0 \leq \alpha_i \leq C \rightarrow |f(x_i)| = 1 \tag{10}$$
$$\alpha = C \rightarrow |f(x_i)| \leq 1 \tag{11}$$

It is crucial to note that each data point lying on the separating plane $f(x) = \pm 1$ is referred to as a support vector. Then by assuming

$$H_{i,j} = y_i y_j \tag{12}$$

then Equation (7) can be written as

$$\max_{\alpha} - L_D = \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i H_{ij} \alpha_j - \sum_{i=1}^{n} \alpha_i \tag{13}$$

If derived with respect to α, it is obtained

$$P_i = \frac{\partial L}{\partial \alpha_i} = \frac{1}{2} \sum_{i,j=1}^{n} H_{ij} \alpha_j - 1 \tag{14}$$

Zhang et al. [26] also described that if the addition of new data was s, then P_i changes as follows:

$$\Delta g_i = H_{i,s}\Delta \alpha_s + \sum_{i,j=1}^{n} H_{ij}\alpha_j \tag{15}$$

$$0 = y_s\Delta \alpha_s + \sum_{j \in 1}^{n} y_j\Delta \alpha_j \tag{16}$$

Therefore, if new data s is added to a system, where s is a support vector, the result will be:

$$P_s = H_{ss}\Delta \alpha_s + \sum_{s,j}^{n} H_{sj}(\alpha_j + \Delta \alpha_j) - 1 = 0 \tag{17}$$

so that the equation obtained is

$$\Delta \alpha_s = \frac{\sum_{j \in s}^{n} H_{sj}\alpha_j - 1}{H_{ss}} \tag{18}$$

$$\alpha = (\alpha + \Delta \alpha, \alpha_s)^T \tag{19}$$

So, if there is an addition of data where the data is a support vector, then α is updated according to Equations (18) and (19).

3. Result and discussion

The assessment of the application's performance involves the computation of key metrics, including Accuracy, Precision, and Recall. Accuracy gauges the proximity between the predicted values and the actual values. Precision quantifies the accuracy level in matching user-requested information with the system's responses. Meanwhile, Recall measures the system's effectiveness in rediscovering relevant information [27]. These metrics collectively provide valuable insights into how well the application is performing in terms of providing accurate and relevant results.

Based on the information provided earlier, this study initially examined handwriting recognition using data related to numbers zero to six, and here is the average result:

Table 1 shows that one striking observation is the exceptionally high precision of 99.45% for the digit one, indicating remarkable accuracy in correctly predicting instances of this specific digit. Conversely, the precision for the digit five is considerably lower at 46.04%, suggesting challenges in accurately identifying instances of this digit. The recall values add further nuance to the assessment, with the digit one demonstrating a high recall of 96.50%, indicating the model's effectiveness in capturing a significant proportion of actual instances of this digit. However, recall values for other digits vary, with higher rates for some (e.g., digit six with 84.30%) and lower rates for others (e.g., digit two with 46.20%). The overall accuracy across all digits remains consistent at 92.51%, highlighting the model's robustness in correctly identifying digits in general. The average precision of 82.17% and average recall of 73.79% provide a comprehensive overview of the model's overall effectiveness in this initial phase of the study, showcasing both strengths, such as the high precision for digit one, and areas for improvement, such as the lower precision for digit five.

Table 1. Performance of the initial stage (7 prior classes).

Class/number	Precision	Recall	Accuracy
Zero	94.99%	57.20%	93.41%
One	99.45%	96.50%	99.43%
Two	91.03%	46.20%	91.41%
Three	96.77%	73.80%	95.91%
Four	88.98%	78.70%	95.56%
Five	46.04%	79.80%	83.41%
Six	57.93%	84.30%	88.43%
Average	82.17%	73.79%	92.51%

The provided results pertain to a subsequent experiment where a new class, representing digit seven, was introduced into the SVM model initially developed using the proposed methodology. Here is the average result.

Table 2 shows the average precision, recall, and accuracy for each digit class. Notable changes are observed compared to the first table as follows:

- Performance on digit seven: The introduction of the new class is evident in the results for digit seven, showing a precision of 40.70%, a recall of 87.20%, and an accuracy of 85.42%. This indicates the model's ability to identify instances of the new class, but with lower precision compared to some existing classes.
- Changes in existing classes: Some existing classes, such as digit zero and digit five, show variations in precision, recall, and accuracy compared to the initial experiment. For instance, the precision for digit zero increased to 96.52%, while the precision for digit five decreased to 50.13%.
- Overall performance changes: The average precision, recall, and accuracy across all classes are 76.50%, 67.55%, and 91.73%, respectively. Comparing these average values to those in the first table, there is a noticeable shift in the model's overall performance due to the introduction of the new digit seven class.

Table 2. Performance of the 1st update class.

Class/number	Precision	Recall	Accuracy
Zero	96.52%	52.50%	93.60%
One	99.49%	95.70%	99.38%
Two	91.24%	44.40%	92.22%
Three	94.68%	73.80%	96.13%
Four	93.58%	69.30%	95.57%
Five	50.13%	28.50%	87.26%
Six	45.66%	89.00%	84.25%
Seven	40.70%	87.20%	85.42%
Average	76.50%	67.55%	91.73%

In brief, the inclusion of new classes has influenced the model's performance, affecting not only the accuracy of recognizing the new class but also introducing

changes to the performance metrics for existing classes. Nevertheless, this initial incremental learning experiment demonstrates that the established model continues to exhibit strong performance, comparable to the performance of the original main model. This underscores that, up to this point, the addition of the new class is yielding positive and satisfactory outcomes.

In the subsequent experiment, we reassess the model by adding another new class (digit eight). Overall, this addition of a new class (digit eight) has influenced the model's performance, resulting in alterations in metrics for both the newly added and existing classes, as we see in **Table 3**. The model remains flexible, showcasing its adaptability. However, the effects on existing classes and the overall average metrics underscore the dynamic nature of the model as it adjusts to the inclusion of extra classes, all the while maintaining satisfactory performance. Here's the detail of the result:

Table 3. Performance of the 2nd update class.

Class/number	Precision	Recall	Accuracy
Zero	93.48%	43.70%	92.90%
One	99.80%	97.00%	99.61%
Two	89.55%	45.00%	92.62%
Three	96.47%	73.10%	96.46%
Four	92.46%	71.00%	95.88%
Five	58.99%	26.90%	88.92%
Six	49.49%	84.80%	87.20%
Seven	31.04%	91.07%	80.23%
Eight	87.02%	61.20%	97.20%
Average	77.59%	65.97%	92.34%

To enhance our comprehensive understanding of the outcomes associated with incorporating this class into the utilized method, presented below are the results of the subsequent experiment, which involves adding another new class (digit 9) to the pre-established model.

As observed in **Table 4**, these findings indicate relatively consistent overall performance. The model created in this iteration maintains commendable results, boasting an average precision of 77.82%, recall of 64.58%, and accuracy of 92.76%. This underscores that, even with the introduction of a third new class, the employed method continues to exhibit strong performance, showing minimal deviation from the initial SVM model.

Table 4. Performance of the 3rd update class.

Class/number	Precision	Recall	Accuracy
Zero	94.15%	43.70%	93.13%
One	99.70%	95.80%	99.47%
Two	90.91%	44.70%	92.89%
Three	97.23%	73.10%	96.60%
Four	93.26%	69.40%	95.95%
Five	59.10%	25.40%	89.27%
Six	58.26%	69.80%	90.36%
Seven	25.90%	95.33%	75.05%
Eight	65.64%	67.40%	96.18%
Nine	94.10%	61.20%	98.73%
Average	77.82%	64.58%	92.76%

4. Conclusion

This research focused on enhancing handwritten digit recognition systems, acknowledging their critical role in various applications. The exploration of incremental support vector machines (ISVM) aimed to address the challenges posed by evolving datasets and dynamic real-world scenarios. This framework initially employed a dataset ranging from "0" to "6" and demonstrated the effectiveness of ISVM in adapting to incremental data. The integration of new classes, specifically digits "7", "8", and "9", showcased the adaptability of the ISVM model. Despite variations in precision, recall, and accuracy across different classes, the overall performance remained robust. The model consistently demonstrated its ability to accommodate additional classes, maintaining satisfactory outcomes.

The proposed ISVM approach proved to be a valuable tool in this context, providing insights into the advantages of incremental learning techniques for improving the accuracy of handwritten numeric string recognition systems. The results encourage further exploration and development of incremental learning approaches for pattern recognition tasks in the era of digital transformation.

Author contributions: Conceptualization, RKP and MA; methodology, MA; software, MA; analysis and validation, RKP; writing—original draft preparation, RKP and MA; writing—review and editing, MA. All authors have read and agreed to the published version of the manuscript.

Conflict of interest: The authors declare that they have no conflict of interest.

References

1. Athoillah M, Putri RK. Handwritten Arabic Numeral Character Recognition Using Multi Kernel Support Vector Machine. Kinetik: Game Technology, Information System, Computer Network, Computing, Electronics, and Control. 2019; 4(2): 99-106. doi: 10.22219/kinetik.v4i2.724

2. Memon J, Sami M, Khan RA, et al. Handwritten Optical Character Recognition (OCR): A Comprehensive Systematic Literature Review (SLR). IEEE Access. 2020, 8: 142642-142668. doi: 10.1109/access.2020.3012542

3. Diaz M, Ferrer MA, Impedovo D, et al. A Perspective Analysis of Handwritten Signature Technology. ACM Computing Surveys. 2019, 51(6): 1-39. doi: 10.1145/3274658

4. Athoillah M. K-Nearest Neighbor for Recognize Handwritten Arabic Character. Jurnal Matematika "MANTIK." 2019, 5(2): 83-89. doi: 10.15642/mantik.2019.5.2.83-89

5. Yogesh Y, Ghantasala GSP, Priya A. Artificial Intelligence Based Handwriting Digit Recognition (HDR) - A Technical Review. In: Proceedings of the 2023 International Conference on Device Intelligence, Computing and Communication Technologies, (DICCT); 17–18 March 2023; Dehradun, India. pp. 275–278. doi: 10.1109/dicct56244.2023.10110186

6. Khobragade RN, Koli NA, Lanjewar VT. Challenges in recognition of online and off-line compound handwritten characters: A review. In: Zhang YD, Mandal J, So-In C, Thakur N (editors). Smart Trends in Computing and Communications. Springer; 2020. Volume 165. pp. 375–383. doi: 10.1007/978-981-15-0077-0_38

7. Liu X, Lian Y. Handwriting identification: Challenges and solutions. Journal of Forensic Science and Medicine. 2018; 4(3): 167-173. doi: 10.4103/jfsm.jfsm_81_17

8. Yang Q, Gu Y, Wu D. Survey of incremental learning. In: Proceedings of the 2019 Chinese Control And Decision Conference (CCDC); 3–5 June 2019; Nanchang, China. pp. 399–404. doi: 10.1109/ccdc.2019.8832774

9. Nayak J, Naik B, Behera HS. A Comprehensive Survey on Support Vector Machine in Data Mining Tasks: Applications & Challenges. International Journal of Database Theory and Application. 2015, 8(1): 169-186. doi: 10.14257/ijdta.2015.8.1.18

10. Cervantes J, Garcia-Lamont F, Rodríguez-Mazahua L, et al. A comprehensive survey on support vector machine classification: Applications, challenges and trends. Neurocomputing. 2020, 408: 189-215. doi: 10.1016/j.neucom.2019.10.118

11. Shams M, Amira A, Wael Z. Arabic Handwritten Character Recognition based on Convolution Neural Networks and Support Vector Machine. International Journal of Advanced Computer Science and Applications. 2020, 11(8). doi: 10.14569/ijacsa.2020.0110819

12. Rajnoha M, Burget R, Dutta MK. Offline handwritten text recognition using support vector machines. In: Proceedings of the 2017 4th International Conference on Signal Processing and Integrated Networks (SPIN); 2–3 February 2017; Noida, India. pp. 132–136. doi: 10.1109/spin.2017.8049930

13. Katiyar G. Off-Line Handwritten Character Recognition System Using Support Vector Machine. American Journal of Neural Networks and Applications. 2017, 3(2): 22. doi: 10.11648/j.ajnna.20170302.12

14. Athoillah M, Imah EM, Irawan MI. Multi kernel-based classification method with incremental learning for image retrieval. In: Proceedings of the International Conference on Mathematics: Pure, Applied and Computation, ICoMPAC 2015; 3 December 2015; Surabaya, Indonesia.

15. Athoillah M. Face Recognition Using Multi Kernel SVM with Incremental Learning (Malaysian). Jurnal Online Informatika. 2018, 2(2): 84. doi: 10.15575/join.v2i2.109

16. Peryanto A, Yudhana A, Umar R. Image Classification Using Convolutional Neural Network and K Fold Cross Validation (Indonesian). Journal of Applied Informatics and Computing. 2020, 4(1): 45-51. doi: 10.30871/jaic.v4i1.2017

17. Xu Y, Goodacre R. On Splitting Training and Validation Set: A Comparative Study of Cross-Validation, Bootstrap and Systematic Sampling for Estimating the Generalization Performance of Supervised Learning. Journal of Analysis and Testing. 2018, 2(3): 249-262. doi: 10.1007/s41664-018-0068-2

18. Zhang Y, Yang Y. Cross-validation for selecting a model selection procedure. Journal of Econometrics. 2015, 187(1): 95-112. doi: 10.1016/j.jeconom.2015.02.006

19. Braga-Neto U. Fundamentals of Pattern Recognition and Machine Learning. Springer International Publishing, 2020. doi: 10.1007/978-3-030-27656-0

20. Ye Q, Doermann D. Text Detection and Recognition in Imagery: A Survey. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2015, 37(7): 1480-1500. doi: 10.1109/tpami.2014.2366765

21. Ajam A, Forghani M, AlyanNezhadi MM, et al. Content-based image retrieval using color difference histogram in image textures. In: Proceedings of the 2019 5th Iranian Conference on Signal Processing and Intelligent Systems (ICSPIS); 18–19 December 2019; Shahrood, Iran. pp. 1–6. doi: 10.1109/icspis48872.2019.9066062

22. Somvanshi M, Chavan P, Tambade S, et al. A review of machine learning techniques using decision tree and support vector machine. In: Proceedings of the 2016 International Conference on Computing Communication Control and automation (ICCUBEA); 12–13 August 2016; Pune, India. pp. 1–7. doi: 10.1109/iccubea.2016.7860040

23. Chefrour A. Incremental supervised learning: algorithms and applications in pattern recognition. Evolutionary Intelligence. 2019, 12(2): 97-112. doi: 10.1007/s12065-019-00203-y

24. Pisner DA, Schnyer DM. Support vector machine. Machine Learning: Methods and Applications to Brain Disorders. Academic Press; 2020. pp. 101–121. doi: 10.1016/b978-0-12-815739-8.00006-7

25. Campbell C, Ying Y. Learning with Support Vector Machines. Springer Nature; 2022.

26. Zhang YC, Hu GS, Zhu FF, et al. A new incremental learning support vector machine. In: Proceedings of the 2009 International Conference on Artificial Intelligence and Computational Intelligence; 7–8 November 2009; NW Washington, DC, United States. doi: 10.1109/aici.2009.342

27. Juba B, Le HS. Precision-Recall versus Accuracy and the Role of Large Data Sets. Proceedings of the AAAI Conference on Artificial Intelligence. 2019, 33(01): 4039-4048. doi: 10.1609/aaai.v33i01.33014039