Chinese Handwritten Digit Recognition Using Machine Learning

Abstract— **The recognition of handwritten digits is among the most popular applications in image classification. Chinese language is not just some simple alphabet symbols. It has unique set of characters called logograms and versatile handwriting strokes. Handwriting style of different individuals affects the structure and representation of the digits and makes the problem of digit recognition more difficult. Chinese Handwritten digits' recognition is a challenging task for machine learning due to its variety of patterns, size and strokes according to each individual. Although decent recognition accuracy has been reported by these classifiers such as such as Logistic Regression (LR), Linear Discriminant Analysis (LDA), K-nearest Neighbors (KNN), and Decision Tree (DT), handwriting digit recognition is still an open research problem to be explored. We have developed multiple machine learning models for Chinese handwritten digit recognition. We have analyzed previously conducted studies, and developed multiple machine learning models. Our developed models are multi-class classifiers, trained on a Chinese digit's image dataset. Then, we evaluated the performance of the models and compared the results. The proposed CNN model has achieved an accuracy of 98.93% for recognition of handwritten Chinese digit images.**

Keywords — Chinese Handwritten Digit, Machine Learning, Convolutional Neural Network

I. INTRODUCTION

Chinese handwritten digits recognition has its significance in the 21st century and it is served as a medium of communication and capturing the information among human and computers in our daily lifetime. The image classification tasks are required in routine work e.g., optical character recognition, automating data entry tasks and so on. Recognition of handwritten digits is a popular implementation of image classification. Such recognition task has been actively improving in demand of higher accuracy and reliability [1]. However, Chinese characters are not easy as English to be typed using keyboard due to its complex characteristics with four acoustic tones and a logographic writing system [2]. On top of that, little research has been conducted on recognizing Chinese handwritten digits using machine learning techniques recently.

In this research, we aim to address the challenge of recognizing Chinese handwritten digits using machine learning techniques. Chinese characters, known as logograms, are composed from a unique set of strokes and patterns, making the task of digit recognition more sophisticated compared to traditional alphabetic language. Furthermore, the handwriting style of different individuals can enormously affect the structure and representation of the digits, adding complexity to the problem.

Throughout this research, we have developed multiple machine learning models for Chinese handwritten digit recognition to accomplish our objectives listed as below:

- To analyse formerly conducted studies for Chinese handwritten digit recognition
- To develop multi-class classification models
- To evaluate the performance of the models

Thus, we have implemented multiple algorithms for recognition of Chinese handwritten digits by training the models with the Chinese digits image dataset. This research aims to contribute to the field of Chinese handwritten digit recognition by developing accurate machine learning models and evaluating their performance. By using CNN and other state-of-the-art techniques, we hope to achieve high accuracy in recognizing Chinese handwritten digits, making this technology more usable in real-world scenarios.

II. LITERATURE REVIEW

Table 1 summarizes several studies that have been conducted on the topic of recognizing handwritten characters using machine learning techniques. Each row in the table represents a different study, with information on the dataset used and method or algorithm applied.

TABLE 1: STUDY OF MACHINE LEARNING MODELS IN RECOGNIZING HANDWRITTEN CHARACTERS

Ref	Dataset	Method /Algorithm
[3]	Chinese Characters	• Spiking Neural Network (SNN)
[4]	Arabic Numeric Characters	• Random Forest (RF) • Convolutional Neural Network (CNN)
[5]	Hindi Characters	• Decision Tree (DT), • Random Forest (RF) • k-nearest Neighbors (kNN)
[6]	Chinese Characters	• Convolutional Neural Network (CNN)
[7]	Arabic Numeric Characters	• Decision Tree (DT) • Support Vector Machine (SVM) • Random Forest (RF) • Naïve Bayes (NB) • k-Nearest Neighbor (kNN)
[8]	Bangla Numeric Characters and Symbols	• Convolutional Neural Network (CNN)
[9]	Arabic Numeric Characters and English Alphabet	• Backpropagation Neural Network (BNN)
[10]	Chinese Numeric Characters	• Labeled Projective Dictionary Pair Learning (LpDPL) • Dictionary Pair Learning (DPL) • k-Nearest Neighbor (kNN)

[3] The author proposed a SNN model architecture and applied different hyperparameters configurations for classification model. The maximum accuracy achieved was approximately 93% on the testing dataset.

[4] The author compared model performance in terms of accuracy and efficiency of the RF and CNN model. Hybrid-RF achieved higher accuracy of 90.82%, CNN achieved 90.33%. Both models achieved almost similar result in accuracy measurement. However, in terms of efficiency, hybrid-RF performed better in consumed training time than

CNN. The proposed hybrid-RF model was faster and solved disadvantages of manual feature selection of RF.

[5] Boorojerdi and his team showed the comparison study on DT, RF and KNN algorithm implementations. The study has reported accuracy of 93.5% for RF, 98% for kNN algorithm and only 83.4% for DT algorithm for recognition Hindi handwritten characters.

[6] The study proposed a 6 layers CNN model for recognition of Chinese handwritten characters. The author achieved an accuracy of 90.91% on the testing dataset with 5000 epochs and spent an approximately of 3 hours for training the model. The author also demonstrated a trade-off between the computational time and accuracy.

[7] In this study, multiple machine learning models were trained and compared in terms of accuracy for the digit recognition. SVM outperformed among other models with an accuracy of 95.88%. The study was carried out as an initial effort for serving as reference for future work. Thus, no further data augmentation techniques and deep learning models were utilized.

[8] The author applied the MathNET model based on CNN to successfully recognize Bangla numerals and mathematical symbols simultaneously. RMSprop optimizer and data augmentation methods are introduced in the proposed model to reach a high accuracy of 96.01% (train) and 96.50% (validation).

[9] This study proposed a BNN model to recognize handwritten digits on a large dataset. The author compared the results under different setting of hidden nodes and concluded that the best number of hidden nodes is 65 which can obtain the best accuracy of 96.0%.

[10] This study demonstrated how dictionary learning for classification worked in recognizing Chinese handwritten digits. The best result of the models was 98.53% accuracy in average for LpDPL model. Moreover, the authors also mentioned that the proposed LpDPL model tends to be more robust in differentiating certain complex Chinese digits.

These studies demonstrated the effectiveness of various machine learning techniques, such as SNN, RF, CNN, DT, SVM, NB, kNN, and BNN, in recognizing handwritten characters. However, the choice of algorithm and the dataset trained can greatly affect the results and the trade-off between computational time and accuracy. This can be seen in the studies that used CNN and SVM, which achieved high accuracy with large datasets but with longer training time [1]. The current study analyses various algorithms using a single similar dataset in recognizing Chinese handwritten digits.

A. *Related Algorithms*

Logistic Regression (LR): LR is a supervised learning algorithm that uses an estimated probability of an event to classify data [11]. In our multiclass supervised problem, we used regularized LR with the "sag" solver, as it can handle multinomial loss in our dataset.

Linear Discriminant Analysis (LDA): LDA is an extension of Principal Discriminant Analysis (PDA) used to classify multiple classes, rather than just two [12]. LDA is implemented in our recognizing model as it is best suited for class separation and assumes all classes are linearly separable.

k-Nearest Neighbors (kNN): kNN is a non-parametric approach mainly for regression or classification tasks, known as a lazy learner strategy [13]. It had been applied in biomedical signal analysis by comparing test sets with similar training sets [14].

Decision Tree (DT): DT is one of the popular supervised learning algorithms that creates a training model to predict class or targeted values based on decision rules derived from the training dataset. In our multi-classification problem, CART algorithm is implemented as it utilizes binary trees for the feature and threshold that return the highest information gain at every single node.

Naïve Bayes (NB): NB is a probabilistic algorithm that uses Bayes theorem to classify dependent variables by calculating the probability of belonging to a certain class based on independent variables [15].

Support Vector Machine (SVM): SVM is a one of the supervised learning algorithms that capable to distinguish data points in high-dimensional space through optimizing the margin between target classes. It is had been implemented in brain disorders research due to its simplicity and low risk of overfitting [16].

AdaBoost Classifier (AB): AB is one of the well-known ensemble methods that generate a stronger classifier from a group of weak classifiers. It is often used in classification problems and can select the classifier with the least error, while avoiding overfitting issues.

Gradient Boosting (GB): GB is a type of machine learning algorithms that produce a strong classifier by combining the weak learners. It is widely used in applications such as anomaly detection, image classification and natural language processing.

Random Forest (RF): RF is one of the famous ensemble methods that generate a final prediction by combining multiple decision trees, with a focus on reducing overfitting issues. It is known for its ability to handle high dimensional data and reduce overfitting compared to a single decision tree. Successfully used in economic analysis [17] and autism spectrum disorder predictions [18].

Ensemble Tree (ET): ET is a classifier that built by using a meta estimator that can fit a few randomized decision trees on a various subset of the dataset. Demonstration of obtaining high accuracy had been obtained in wireless sensor network fault detection and diagnosis [19].

Convolutional Neural Network (CNN): CNN belongs to one of the deep learning algorithms that is well-known for its ability to perform well in image recognition tasks. A computationally efficient model with special convolution and pooling operations for image recognition or detection problems. Successfully trained in self-driving cars, Q value classification and biomedical engineering applications such as genomic sequencing, bio and medical images analysis, brain, body, and machine interface, and gene expression analysis, as well as public and medical health management systems [20] [21].

III. Research Methodology

Supervised Machine Learning techniques are used on the labelled dataset. From the journal papers studied, we acknowledged that there were several algorithms used to train and measure the accuracy performance in Chinese handwritten digit recognition.

A. Data Collection

To train the model mentioned above, we utilized a dataset of images (Chinese handwritten digits). This dataset is derived from the original dataset collected at Newcastle University in UK, which contains actual 15,000 images of handwritten Chinese digits with 15 classes of Chinese digit character [22]. One hundred Chinese individuals participated in this data collection activity and each participant wrote 15 Chinese digits on a paper. The activity was repeated 10 times for each digit per person, to collect multiple variations of handwritten digits to get better training and testing datasets for algorithms.

B. Data Pre-processing

It is a vital step to improve data quality for machine learning process, which involves data cleansing, data reduction, data transformation and data integration depending on the quality of raw data. Since some algorithms such as LR, KNN and DT do not recognize images as input, we then further transformed those 15,000 images into pixels in tabular format. The transformed data consisted of 4,098 columns, in which 64×64 pixel (equivalent to 4096 column), a Chinese character column and a label value column.

Thus, two datasets in different representative formats were used for further process. As a result of pre-examine on the dataset using Jupyter Notebook, there was no missing values or outliers in the dataset. A total number of 15 unique classes of dependent variable were recorded, specifically {0,1,2,3,4,5,6,7,8,9,10,100,1000,10000,100000000}. We had a balance count in each class for the dependent variable.

C. Environment Setup

The study was conducted on a Windows10 X64 operating system as main machine. CPU was Intel® Core™i9-9900K CPU at 3.60 GHz, with a 32-GB running memory. Python programming language utilized in Anaconda Navigator Integrated Development Environment (IDE) to conduct the project. The project developed in Jupyter Notebook from the IDE mentioned. scikit-learn and TensorFlow libraries used to implement the algorithms as it provides easy built-in functions for developing those algorithms.

D. Model Development

Proportion of 80-20 for training and testing was used to split the dataset. To ensure reproducibility, seed was set to obtain the similar result in randomization. 11 machine learning algorithms applied at first stage, such as CNN, SVM, ET, RF, GB, kNN, LR, LDA, DT, AB, and NB were trained, tested and the results were compared. CNN outperformed the rest. Thus, we decided to further enhance CNN architecture for obtaining better accuracy and reduce computational time.

IV. Result and Discussion

From the journal studies, basic algorithms and CNN model trained and tested out on the earlier stage. The performance of each model is examined based on evaluation metrics of accuracy and time.

The architecture of proposed CNN model built for recognizing the Chinese handwritten digit is referencing from previous journal papers conducted. Fig. 1 below showed the macro view of the proposed CNN architecture.

In general, CNN-based models consist of main building blocks such as convolutional and max pooling layers. For initial design, we chose a CNN model with an architecture of three convolutional and max pooling layers. Table 2 illustrated our proposed CNN architecture, which consists of three convolutional layers, three max pooling layers, and a fully connected layer. Each layer is discussed into details as follows:

- The 1st convolution layer: This layer was designed with a kernel of size 3x3. A total of 32 feature maps will be generated since this layer was set for 32 kernels.
- The 2nd convolution layer: This layer was designed with a same kernel as the previous 1st layer. Thus, it will generate 32 feature maps again as the 1st layer did.
- The 3rd convolution layer: This layer is designed with a same kernel as the 1st and 2nd layers. Thus, it will generate 32 feature maps again as the 1st and 2nd layers did.
- The pooling layer: A pooling layer is designed for each convolution layer, associated with a window size of 2x2.

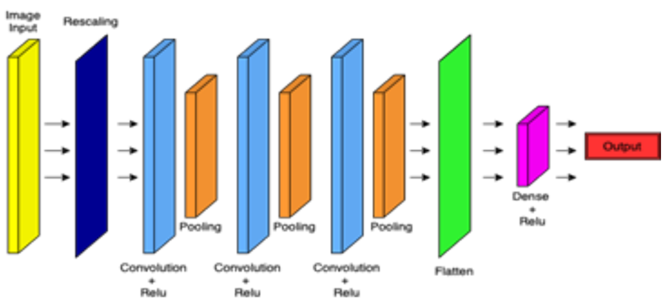

Fig. 1. Macro-view of Proposed CNN Architecture

TABLE 2. SUMMARY OF PROPOSED CNN ARCHITECTURE

Layer (type)	Output Shape	Param #
Rescaling	(64, 64, 1)	0
Convolution 1	(62, 62, 32)	320
Max Pooling 1	(31, 31, 32)	0
Convolution 2	(29, 29, 32)	9248
Max Pooling 2	(14, 14, 32)	160
Convolution 3	(12, 12, 32)	0
Max Pooling 3	(6, 6, 32)	0
Flatten	(1152)	0
Dense	(128)	147584
Output	(15)	1935

Since there is no apparent rule or guidance on how to select the best hyperparameters for CNN model, trial and error experimentation is performed to choose the most suitable settings. We could reference from previous research studies in deep learning and digit recognition for selecting hyperparameters setting. Thus, according to analysis of studies in Literature Review section, the hyperparameters settings are as follows:

- Non-linearity: Introducing the non-linearity function into CNN model is required as the relationship between input images and its classes is not linear. This issue can be eliminated by implementing a non-linear activation function into the CNN model. Rectified Linear Unit (ReLU) is typically adopted for image recognition problems. ReLU has the capability of allowing CNN in reducing computation time compared to other function [23].
- Rescaling layer: Known as a preprocessing layer, which mainly for rescaling the input values to a new range as output. In our implementation, it is set to rescale every input value (referring to image) by dividing with a scale factor and adding certain offset.

After the setting of the hyperparameters and layers of CNN, the model was compiled using Adam optimizer. Such optimizer allows CNN to set the weights and biases automatically and randomly. 10 epochs with a batch size of 32 is set for the CNN in training stage. Fig 2. plots the variations in training accuracy and loss and testing accuracy and loss with respect to the epochs.

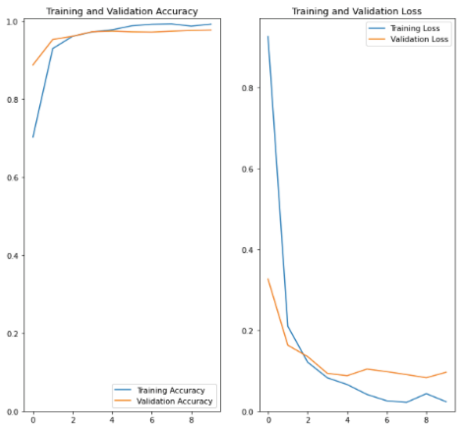

Fig. 2. Training Accuracy vs Validation Accuracy (Left), Training Loss vs Validation Loss (Right)

To achieve a higher accuracy for performance measure, data augmentation technique was applied to the initial architecture of CNN. After experimenting, our CNN model with data augmentation does not achieve a higher accuracy and lower loss than previously. Data augmentation may not always work in improving the accuracy of image recognition [16]. As shown in our study, it happened possibly due to several potential reasons, such as too much class for prediction, too large of input size etc. On top of that, we also experimented on different number of epochs to obtain the behavior on how it affects the accuracy. Table 3 below summarizes the accuracy on testing dataset and their loss according to epoch.

TABLE 3. SUMMARY OF ACCURACY AND LOSS FOR DIFFERENT EPOCH

Epoch	Testing Accuracy (%)	Testing Loss (%)
10	98.03	7.25
15	98.93	4.02
20	98.97	4.84

As shown table above, we can clearly observe that when the number of epoch increase, the accuracy on testing dataset increases simultaneously. However, when the epoch is increased from 15 to 20, the testing loss increases rapidly as well. In trade-off between accuracy and loss, it is an obvious inference that the best result is obtained from our proposed CNN model when epoch is 15. Because there is a high decrease of the testing loss when epoch increased from 10 to 15, and the accuracy increased by 0.9%, thus, our experiment did an early stopping at 20 epochs without further increasing the number of epochs.

After Multiple machine learning algorithms have been applied and the maximum accuracy achieved is 98.03% by CNN algorithm. The computational time for training the model is also recorded. Table 4. below shows the performance in terms of 5 measurements and time in seconds.

TABLE 4. OVERALL PERFORMANCE OF ALGORITHMS SORTED IN DESCENDING ORDER ACCORDING TO ACCURACY

Model	Accuracy (%)	Precision (%)	Recall (%)	AUC	F-Score	Time (sec)
CNN	98.93	100	32	63.17	86.65	44
SVM	71.80	72.54	71.80	95.82	71.56	2903
ET	69.53	70.89	70.13	95.44	69.74	19
RF	61.60	60.79	60.67	93.21	59.96	12
GB	59.03	58.24	58.13	91.98	57.66	3158
KNN	39.80	59.55	39.80	83.56	41.72	10
LR	39.13	38.66	39.03	80.89	38.67	513
LDA	37.73	38.37	37.73	75.88	37.14	88
DT	34.13	32.67	33.50	64.68	32.95	17
AB	24.67	23.64	24.67	71.55	23.26	52
NB	23.97	34.06	23.97	61.75	19.76	4

From table shown above, it is clearly demonstrated that CNN has outperformed among 11 algorithms with an accuracy of 98.93% achieved on testing dataset associated with an average of 44 seconds for computational time. Such result is expected due to the nature of our classification problem, which is image recognition. The proposed CNN model also demonstrated a trade-off between precision and recall, in which it obtained 100% in precision and 32% in recall. Precision can be interpreted as a measurement of quality while recall can be interpreted as a measurement of quantity. We achieved 100% in precision, in which the proposed CNN model returns more relevant results than the irrelevant ones. In the journal studies from previous research, CNN typically works well in image recognition problem. For example, a paper published by Saleh and his team on Arabic Sign Language Recognition through Deep Neural Network achieved a 99.57% accuracy with testing dataset. The paper implemented ResNet152 with 100 epochs for recognizing sign language on static image [24].

SVM is at 2nd ranking in terms of model accuracy, it obtained an accuracy of 71.80% on testing dataset for recognizing Chinese handwritten digit. The result is consistent with the journal studies in previous section in recognizing Chinese handwritten character. SVM outperformed other machine learning algorithms. However, due to the large number of input features, SVM took longer computational time for processing. Gholami and his team proclaimed that SVM is good approach in resolving the issues raised by having a limited amount of data for training, but not to a huge database [25]. It is consistent with the result obtained.

Ensemble Tree (ET) performed well and outperformed traditional models such as Decision Tree (DT) and Gradient Boosting (GB) in recognizing Chinese handwritten digits. This is likely due to the nature of the problem being image recognition and having many input features. Random Forest (RF) and Gradient Boosting (GB) performed relatively poorly in terms of accuracy, with 61.60% and 59.03% respectively. They also had longer computational times compared to ET. This can be attributed to the fact that these models are not well suited for image recognition problems. In terms of evaluation metrics, CNN also performed well in terms of precision, recall, AUC, and F-measure. However, it had a lower recall percentage of 32%. This can be attributed to the fact that the model returned more relevant results than irrelevant ones.

Overall, it is evidently illustrated that CNN performed the best among the algorithms trained in terms of accuracy and computational time, making it a suitable algorithm for recognizing Chinese handwritten digits. However, it is important to note that trade-offs between precision and recall exist, and further research can be conducted to improve these metrics.

V. Conclusion

To conclude the study, machine learning techniques are effective for Chinese handwritten digit recognition. To achieve the research objectives, we built 11 ML models and compared the models' performance based on evaluation metrics. The novelty of the study is that the proposed CNN model has achieved high accuracy of 98.93%. It has attained it without data augmentation regularization technique and no complex CNN higher layered architecture is implemented. Meanwhile, the training time is acceptable which is 44 seconds on average per epoch.

In short, research gap has been explored in recognizing Chinese handwritten digits with machine learning models. As compared to previous research paper in this field, we had proposed a simpler CNN model, with a limited number of layers in the architecture design. Simple structure of CNN model is crucial as not only reducing the complexity of the problem, but also computational power. Such proposed CNN model tends to reduce the computational time, but meanwhile increase the accuracy for recognition.

It is also worth mentioning that SVM outperformed other traditional ML algorithms in terms of accuracy (71.80%), but it takes longer time than others due to a huge number of input features. We can also conclude that KNN, LR, LDA, DT, AB, and NB are not suitable for such tasks due to very low accuracy. The findings of this study may pave a path towards other multi-class image recognition tasks, especially in handwritten character recognition.

The data pre-processing step was crucial in this study as it allowed for the image data to be transformed into a format that could be used by certain machine learning algorithms such as LR, KNN, and DT. This highlights the importance of properly preparing and cleaning the data before implementing any machine learning models. Furthermore, the use of an ensemble method like the AdaBoost classifier did not perform well in this study, which suggests that it may not be suitable for this specific task.

Evaluation metrics such as precision, recall, and AUC were utilized, in addition to accuracy, provides a more comprehensive understanding of the model's performance and can help identify any trade-offs between precision and recall. Additionally, the research methodology adopted in this study, which includes the use of multiple algorithms and the 80-20 split for training and testing, allows for a fair comparison of the models' performance, and increases the robustness of the findings.

This research was conducted on a specific dataset and the generalizability of the results is limited. Therefore, it would be beneficial to conduct further research on other datasets to confirm the robustness of the proposed model. Additionally, future research can also focus on investigating other techniques such as data augmentation and model optimization to improve the performance of the proposed model.

References

[1] Ahlawat, S., & Choudhary, A. (2020). Hybrid CNN-SVM classifier for handwritten digit recognition. *Procedia Computer Science*, *167*, 2554–2560. https://doi.org/10.1016/j.procs.2020.03.309

[2] Yang, G., Gao, X., & Lye, B. (2020). Teaching Chinese as a second or foreign language to non-Chinese learners in mainland China (2014–2018). *A Language In Focus*, *53*, 44-62. https://doi.org/10.1017/S0261444819000387

[3] Russo, N., Madsen, T., & Nikolic, K. (2022). Pattern recognition spiking neural network for Chinese characters classification. https://doi.org/10.36227/techrxiv.20702992.v1

[4] Xi, E. (2022). Image classification and recognition based on Deep Learning and Random Forest algorithm. *Wireless Communications and Mobile Computing*, *2022*, 1–9. https://doi.org/10.1155/2022/2013181

[5] Boroojerdi, S., & Rudolph, G. (2022). Handwritten multi-digit recognition with machine learning. *2022 Intermountain Engineering, Technology and Computing (IETC)*. https://doi.org/10.1109/ietc54973.2022.9796722

[6] Zhuang, Y., Liu, Q., Qiu, C., Wang, C., Ya, F., Sabbir, A., & Yan, J. (2021). A handwritten Chinese character recognition based on convolutional neural network and Median filtering. *Journal of Physics: Conference* Series, 1820(1), 012162. https://doi.org/10.1088/1742-6596/1820/1/012162

[7] Gope, B., Pande, S., Karale, N., Dharmale, S., & Umekar, P. (2021). Handwritten digits identification using MNIST database via machine learning models. *IOP Conference Series: Materials Science and Engineering*, *1022*(1), 012108. https://doi.org/10.1088/1757-899x/1022/1/012108

[8] Shuvo, S. N., Hasan, F., Ahmed, M. U., Hossain, S. A., & Abujar, S. (2020). Mathnet: Using CNN Bangla handwritten digit, mathematical symbols, and trigonometric function recognition. *Advances in Intelligent Systems and Computing*, 515–523. https://doi.org/10.1007/978-981-15-7394-1_47

[9] Yap, M. I., Moorthy, K., Daud, K. M., & Ernawan, F. (2021). Optical character recognition using backpropagation neural network for handwritten digit characters. *2021 International Conference on Software Engineering & Computer Systems and 4th International Conference on Computational Science and Information Management (ICSECS-ICOCSIM)*. https://doi.org/10.1109/icsecs52883.2021.00037

[10] Ameri, R., Alameer, A., Ferdowsi, S., Nazarpour, K., & Abolghasemi, V. (2022). Labeled projective dictionary pair learning: Application to handwritten numbers recognition. *Information Sciences*, *609*, 489–506. https://doi.org/10.1016/j.ins.2022.07.070

[11] Subasi, A. (2020). Practical machine learning for data analysis using Python. Elsevier Academic Press.

[12] Vaibhaw, Sarraf, J., & Pattnaik, P. K. (2020). Brain–computer interfaces and their applications. *An Industrial IoT Approach for Pharmaceutical Industry Growth*, 31–54. https://doi.org/10.1016/b978-0-12-821326-1.00002-4

[13] L. F. Cabeza (2022). Online Diagnosis of PEM Fuel Cell by Fuzzy C-Means Clustering. In *Encyclopedia of Energy Storage*. essay, Elsevier.

[14] Subasi, A., Khateeb, K., Brahimi, T., & Sarirete, A. (2020). Human activity recognition using machine learning methods in a smart healthcare environment. *Innovation in Health Informatics*, 123–144. https://doi.org/10.1016/b978-0-12-819043-2.00005-8

[15] Misra, S., & Li, H. (2020). Noninvasive fracture characterization based on the classification of Sonic Wave Travel Times. *Machine Learning*

for Subsurface Characterization, 243–287. https://doi.org/10.1016/b978-0-12-817736-5.00009-0

[16] Mechelli, A., & Vieira, S. (2020). *Machine learning: Methods and applications to brain disorders*. Elsevier Academic Press.

[17] Malik, A., Javeri, Y. T., Shah, M., & Mangrulkar, R. (2022). Impact analysis of COVID-19 news headlines on Global Economy. *Cyber-Physical Systems*, 189–206. https://doi.org/10.1016/b978-0-12-824557-6.00001-7

[18] Omar, K. S., Islam, M. N., & Khan, N. S. (2021). Exploring tree-based machine learning methods to predict autism spectrum disorder. *Neural Engineering Techniques for Autism Spectrum Disorder*, 165–183. https://doi.org/10.1016/b978-0-12-822822-7.00009-0

[19] Saeed, U., Jan, S. U., Lee, Y.-D., & Koo, I. (2021). Fault diagnosis based on extremely randomized trees in wireless sensor networks. *Reliability Engineering & System Safety*, *205*, 107284. https://doi.org/10.1016/j.ress.2020.107284

[20] Gonsalves, T., & Upadhyay, J. (2021). Integrated deep learning for self-driving robotic cars. *Artificial Intelligence for Future Generation Robotics*, 93–118. https://doi.org/10.1016/b978-0-323-85498-6.00010-1

[21] Satyanarayana Tallapragada, V. V. (2021). Deep learning and its applications in biomedical image processing. *Handbook of Deep Learning in Biomedical Engineering and Health Informatics*, 281–302. https://doi.org/10.1201/9781003144694-11

[22] Nazarpour, K., & Chen, M. (2019, December 21). *Handwritten Chinese numbers*. figshare. Retrieved January 25, 2023, from https://data.ncl.ac.uk/articles/dataset/Handwritten_Chinese_Numbers/10280831/1

[23] Liu, H. (2020). Rail Transit Channel Robot Systems. *Robot Systems for Rail Transit Applications*, 283–328. https://doi.org/10.1016/b978-0-12-822968-2.00007-3

[24] Saleh, Y., & Issa, G. F. (2020). Arabic sign language recognition through deep neural networks fine-tuning. *International Journal of Online and Biomedical Engineering (IJOE)*, *16*(05), 71. https://doi.org/10.3991/ijoe.v16i05.13087

[25] Gholami, R., & Fakhari, N. (2017). Support Vector Machine: Principles, parameters, and applications. *Handbook of Neural Computation*, 515–535. https://doi.org/10.1016/b978-0-12-811318-9.00027-2