

Classification of Standard FASHION MNIST Dataset Using KNN and DT

Abstract—The MNIST handwritten digit recognition dataset was originally used to evaluate common ML algorithms, however, its overall simplicity makes it a poor choice for evaluating modern supervised classification ML algorithms. More recently, this dataset has been replaced with the MNIST fashion dataset, which consists of a similar number of images, pixels, and classes. This article uses Python code to classify image instances taken from the dataset and implements two classifiers: a kNN classifier and a decision tree classifier. In this article, we try different parameters for each machine learning algorithm to get the best performance, and for each classifier, we compare and evaluate the results. Next, the performances of the best kNN classifier and the best decision tree classifier is compared. Different metrics are used to evaluate the performance of the training model.

Keywords—*MNIST fashion dataset; K-Nearest Neighbor(KNN); Decision Tree(DT); Image Classification*

I. INTRODUCTION

LeCun et al. first presented the MNIST dataset, which consists of 10-class handwritten digits, in 1998 [1]. The stellar emergence of deep learning techniques and their performance could not have been predicted at the time. Even though deep learning can now do a wide range of tasks, the straightforward MNIST dataset has surpassed CIFAR-10 [2] and ImageNet [3] in terms of popularity, according to Google Trends. Despite requests for it in the deep learning field, despite its simplicity, its usage does not appear to be declining.

Because of its vastness, MNIST has gained a lot of attention from deep learning researchers, who may use it to swiftly test and prototype their algorithms. All deep learning frameworks and machine learning libraries, such as Tensorflow and Scikit-Learn, offer helpful utility functions and practical examples that employ MNIST right out of the box.

The selection on Zalando's website is the foundation for Fashion-MNIST. Every fashion item on Zalando has a series of photos taken by professionals showing many facets of the item, such as front and back views, details, looks with a model, and looks in an outfit. The original image is a 762 x 1000 JPEG file with a light gray background (hexadecimal color: #fdfdfd). The original image is resampled with many resolutions, such as large, medium, small, thumbnail, and tiny, for effectively serving various frontend components.

Fashion-MNIST is a dataset of fashion product photos designed to replace MNIST and offer a more difficult option for testing machine learning algorithms. As seen in Fig. 1, Fashion-Photos MNIST is transformed into a format identical to that of the MNIST dataset, making it instantly compatible with any machine-learning tool capable of using the original MNIST dataset.

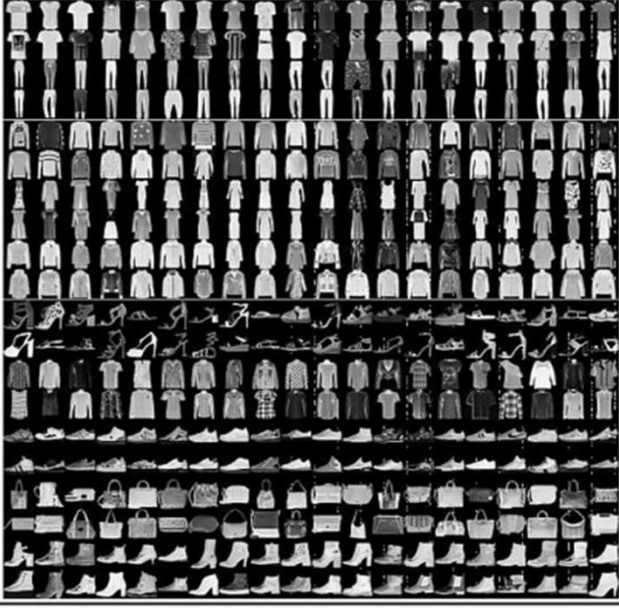
Label	Description	Examples
0	T-Shirt/Top	
1	Trouser	
2	Pullover	
3	Dress	
4	Coat	
5	Sandals	
6	Shirt	
7	Sneaker	
8	Bag	
9	Ankle boots	

Fig.1. Fashion-MNIST Dataset

II. FASHION-MNIST CLASSIFICATION BY KNN

KNN (K-Nearest Neighbor) is one of the simplest machine learning algorithms that can be used for classification and regression. It is a supervised learning algorithm. In Fig. 2, the idea is that if most of the K similar samples in the feature space (i.e., the closest samples in the feature space) belong to a certain category, the sample also belongs to this category. That is, the method determines the category [4] of the sample to be divided only according to the category of the nearest one or a few samples.

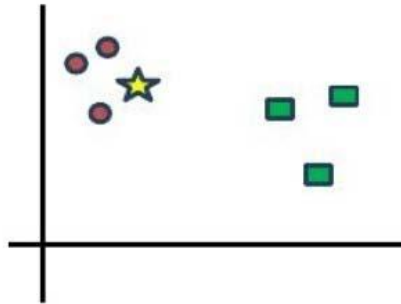


Fig.2. Working of KNN

A. The KNN is used to classify the following categories:

KNN is essentially based on a data-statistics method; many machine learning algorithms are also based on data-statistics methods. At the same time, KNN is instance-based learning, which belongs to lazy learning, that is, it does not have an obvious early training process, but when the program starts to run, the data set is loaded into memory, and it can directly start classification [5]. Among them, every time an unknown sample point is judged, you will vote for K the nearest point near the sample point. This is the meaning of K in KNN, and usually, K is an integer no greater than 20.

In general, the calculation process of the KNN classification algorithm is divided into five steps:

- 1) Calculate the distance between the point to be classified and the point of the known category.
- 2) Sort in order of increasing distance.
- 3) Select the K points with the smallest distance from the point to be classified.
- 4) Determine the number of occurrences of the category of the first K points.
- 5) The category with the highest occurrence of the first K points was returned as the predicted classification of the point to be classified.

B. The KNN is used to return to the following data:

The value of the point to be predicted is obtained by averaging the value of the nearest K point. The "nearest" can be the Euclidean distance or another distance. The specific effect depends on the data in the same way. As shown in Fig. 3, the x-axis is a feature, y is the value of the feature, and the red point is the known point. To predict the position of the first point, calculate the average of the nearest three red points (yellow box), and by analogy, get the green line.

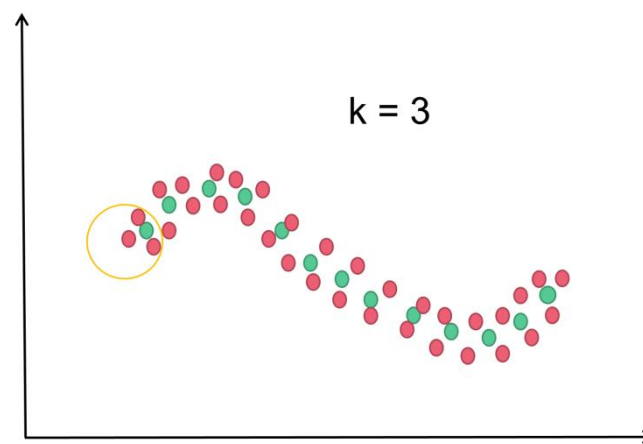


Fig.3. Selecting Nearest Data Points in KNN

Whether it is classification or regression, we can see several key points of the KNN algorithm:

- (1) The algorithm's hyperparameter K.
- (2) Distance measurement The distance between the sample points in the feature space is a reflection of the degree of similarity between the sample points.
- (3) Classified decision-making rules, and the minority shall obey the majority.

Among them, the choice of K value and the setting of distance measurement are several problems that should be paid attention to.

As the only bit hyperparameter in the KNN algorithm, the choice of K value will have an intuitively important impact on the prediction results of the final algorithm. If a smaller value of K is selected, it is equivalent to predicting with the training examples in a smaller neighborhood, and the approximation error of "learning" will decrease. Only the training examples with a closer input instance will affect the prediction results. However, the disadvantage is that the estimation error of "learning" will increase, and the prediction results will be very sensitive to the nearest neighbor instance points. If the adjacent instance point happens to be noise, the prediction goes wrong [6]. In other words, a decrease in the K value means that the overall model is very complex and prone to overfitting. If a large value of K

is chosen, it is equivalent to prediction using training examples in large neighborhoods. The advantage is to reduce the estimation error of learning, but the disadvantage is that the approximation error of learning will increase. At this time, the training example far from the input instance will also play a prediction role, so that the increase in the error K value of the prediction means that the overall model becomes simple and prone to underfitting. It can be assumed that the extreme condition $K = N$ means that regardless of the input instance, it is simply predicted to belong to the largest class of training instances. At this time, the model is too simple, completely ignoring a large amount of useful information in the training, and it is not desirable [7].

In applications, the cross-validation method is usually used to select the optimal K value. From the above analysis, the result, the general K value is relatively small. We will select the K value in a small range, and determine the one with the highest accuracy on the validation set as the final algorithm hyperparameter K .

The distance measure between two points in sample space represents the degree of similarity between the two points: a smaller distance indicates a higher similarity; instead, a greater distance means a lower similarity. In the KNN algorithm, there are three commonly used distances, namely, the Manhattan distance, the Euclidean distance, and the Minkowski distance.

The advantages and disadvantages of the KNN algorithm are as follows:
merits:

- 1) The algorithm is simple and the theory is mature, which can be used for both classification and regression.
- 2) Can be used for nonlinear classification.
- 3) There is no obvious training process, but when the program starts running, after the data set is loaded into memory, there is no need to train and directly predict, so the training time complexity is 0.
- 4) Since the KNN method mainly depends on the surrounding, limited adjacent samples, rather than the method to determine the category, it is more suitable than other methods for the overlapping sample sets of class domains.
- 5) This algorithm is suitable for the automatic classification of classes with large sample sizes, while those with small sample sizes are prone to misclassification using this algorithm.

shortcomings:

- 1) The distance between each test point and the training set should be calculated. When the training set is large, the calculation amount is quite large, and the time complexity is high, especially when the number of features is large.
- 2) Needing a lot of memory and having high spatial complexity.
- 3) There is a sample imbalance problem (some categories have a large number of samples, while other samples are very small), and the prediction accuracy of rare categories is low.
- 4) The lazy learning method, which does not learn, results in a slower prediction speed than algorithms such as logistic regression.

In this dataset, using KNN as a classifier where N takes different values, the accuracy is shown in Fig. 4. It can be found that the value of N is steadily increasing as the value of N gets larger and larger.

```

N= 3000 mean accuracy= 0.83355
N= 3500 mean accuracy= 0.8366285714285714
N= 4000 mean accuracy= 0.8391
N= 4500 mean accuracy= 0.84252
N= 5000 mean accuracy= 0.84405
N= 5500 mean accuracy= 0.8440666666666666
N= 6000 mean accuracy= 0.8473
Wall time: 3min 6s

```

Fig.4. Accuracy for Different N

The above results are shown as line charts, as shown in Fig. 5:

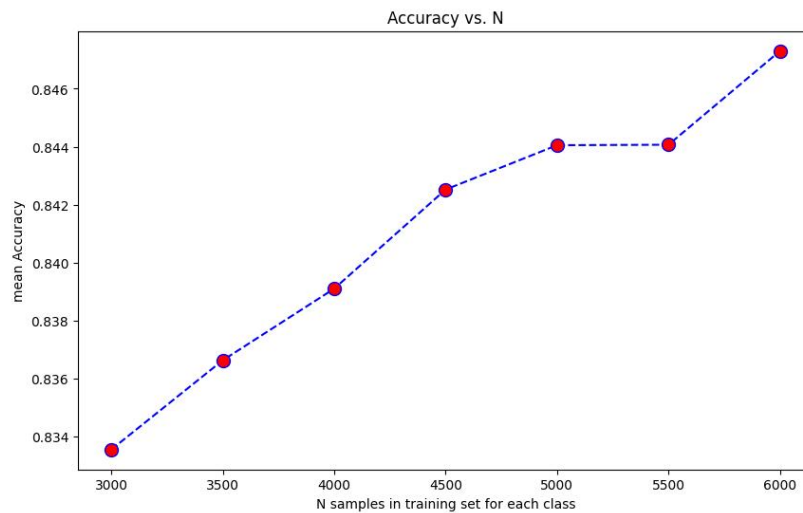


Fig.5. Accuracy vs.N

Next, look specifically at N = 6000, as shown in Fig. 6. An average accuracy of 0.8473 is obtained.

	precision	recall	f1-score	support
0	0.76	0.86	0.81	1000
1	0.99	0.95	0.97	1000
2	0.75	0.79	0.77	1000
3	0.91	0.88	0.89	1000
4	0.79	0.80	0.79	1000
5	0.99	0.78	0.87	1000
6	0.64	0.57	0.60	1000
7	0.85	0.94	0.89	1000
8	0.97	0.94	0.95	1000
9	0.87	0.96	0.91	1000
accuracy			0.85	10000
macro avg	0.85	0.85	0.85	10000
weighted avg	0.85	0.85	0.85	10000

N=6000, mean accuracy= 0.8473
Wall time: 29.1 s

Fig.6. N = 600

Use Kfold to select the training set, as shown in Fig. 7. You can see the first one with the highest average accuracy.

```

TRAIN: [10000 10001 10002 ... 69997 69998 69999] TEST: [ 0 1 2 ... 9997 9998 9999]
mean accuracy= 0.8513
TRAIN: [ 0 1 2 ... 69997 69998 69999] TEST: [10000 10001 10002 ... 19997 19998 19999]
mean accuracy= 0.8457
TRAIN: [ 0 1 2 ... 69997 69998 69999] TEST: [20000 20001 20002 ... 29997 29998 29999]
mean accuracy= 0.8473
TRAIN: [ 0 1 2 ... 69997 69998 69999] TEST: [30000 30001 30002 ... 39997 39998 39999]
mean accuracy= 0.8477
TRAIN: [ 0 1 2 ... 69997 69998 69999] TEST: [40000 40001 40002 ... 49997 49998 49999]
mean accuracy= 0.8429
TRAIN: [ 0 1 2 ... 69997 69998 69999] TEST: [50000 50001 50002 ... 59997 59998 59999]
mean accuracy= 0.8472
TRAIN: [ 0 1 2 ... 59997 59998 59999] TEST: [60000 60001 60002 ... 69997 69998 69999]
mean accuracy= 0.8474
Wall time: 1min 52s

```

Fig.7. Use Kfold to Select the Training Set

Next, select the index of the first fold as a training set, as shown in Fig. 8.

	precision	recall	f1-score	support
0	0.77	0.89	0.83	1009
1	0.99	0.96	0.97	972
2	0.73	0.78	0.76	951
3	0.91	0.86	0.88	1012
4	0.77	0.80	0.79	979
5	1.00	0.80	0.89	1008
6	0.65	0.56	0.60	1022
7	0.86	0.96	0.91	1037
8	0.97	0.94	0.96	974
9	0.89	0.96	0.93	1036
accuracy			0.85	10000
macro avg	0.85	0.85	0.85	10000
weighted avg	0.85	0.85	0.85	10000

Fig.8. Select the Index of the First Fold as Training Set

III. FASHION-MNIST CLASSIFICATION BY DECISION TREE

The decision tree algorithm is a method to approximate the values of the discrete functions. It is a typical classification method, first processing the data, using an induction algorithm to generate readable rules and decision trees, and then using the trees to analyze the new data. Essentially, a decision tree is a process of classifying data through a series of rules.

The decision tree method was first developed in the 1960s and used until the late 1970s. The ID3 algorithm proposed by J. Ross Quinlan aims to reduce the depth of the tree. But they ignored the study of leaf numbers. The C4.5 algorithm is improved based on the ID3 algorithm and greatly improves the default treatment of predictor variables, pruning technology, and derivative rules, which is suitable for both classification problems and regression problems [8].

CART and C4.5 support the treatment when the data features are continuously distributed, mainly by using binary segmentation to treat continuous variables, that is, finding a specific split value: the eigenvalue is greater than the split value, or the left subtree, or the right subtree. The principle of this split value is to reduce the "degree of confusion" in the divided subtree, and the C4.5 and CART algorithms have different definitions.

The ID3 algorithm was invented by Ross Quinlan and based on "Occam's razor": the smaller the decision tree is better than the larger decision tree (be simple theory) [9]. In the ID3 algorithm, according to the information gain of information theory, the feature with the largest information gain is selected each time. The ID3 algorithm can be used to divide nominal data sets without the pruning process. To remove the problem of excessive data matching, crop and merge adjacent leaf nodes that cannot produce large information gains (such as setting the information gain threshold). Using information gain is a disadvantage, that is, it tends to have many valuable properties; that is, in the training set, the more different values there are, the more likely it is to be taken as a split attribute, which is sometimes meaningless. Another ID3 can't handle the continuous distribution of data characteristics, so the C4.5 algorithm is used. The CART algorithm also supports a continuous distribution of the data features.

C4.5 is an improved algorithm of ID3 and inherits its advantages. C4.5 The algorithm uses the information gain rate to select attributes, which overcomes the disadvantage of selecting attributes with many values for pruning in the tree construction, can complete the discrete processing of continuous attributes and can process incomplete data. The classification rules produced by the C4.5 algorithm are easy to understand and have high accuracy but are weak in the process of tree construction. Also, because datasets must be scanned multiple times, C4.5 is only suitable for datasets that can reside in memory.

The CART algorithm uses the Gini index (choose the features with the smallest Gini index) as the splitting criterion, and it also includes the post-pruning operation. Although ID3 and C4.5 algorithms can mine as much information as possible in learning from the training sample set, their generated decision trees have many branches and a large scale. To simplify the size of the decision tree and improve the efficiency of generating decision trees, the decision tree algorithm CART appears, which selects the test properties according to the GINI coefficient.

The decision tree algorithm constructs a decision tree to discover the classification rules contained in the data. How to construct a decision tree with high precision on a small scale is the core content of the decision tree algorithm. The construction of a decision tree can be performed in two steps. The first step is decision tree generation, which is the process of generating decision trees from the training sample set. In general, the training sample data set is historical and integrated into data analysis and processing according to actual needs. The second step, the decision tree pruning: decision tree pruning is generated in the previous stage of the decision tree inspection, correction, and repair process, mainly with a new sample data set (called test data set) in the process of preliminary rules, will affect the accuracy of the balance branch out.

A. Decision tree learning

The goal is to build a decision tree model based on a given training dataset that enables it to correctly classify the instances. Decision tree learning is essentially the induction of a set of classification rules from the training dataset. A decision tree that can correctly classify the training data may have multiple nodes or may not. When choosing a decision tree, we should choose one with less contradiction to the training data and good generalization ability, and the conditional probability model should have a good fit not only with the training data but also with the unknown data. The loss function is usually a normalized maximum likelihood function. The core strategy is to minimize the loss function

as the objective function. Because it is an NP-complete problem to choose the optimal decision tree from all possible decision trees, decision tree learning usually adopts heuristics to solve this optimization problem, and the resulting decision tree is sub-optimal (sub-optimal). The algorithm of decision tree learning is usually a process of recursively selecting the optimal feature and splitting the training data according to this feature, making the best classification process for each subdataset. The pruning process includes feature selection, decision tree generation, and decision tree.

1) Pruning

The goal is to make the tree simpler and thus give it better generalization ability. The step is to remove two subdivided leaf nodes and bring them back to the parent or even higher nodes, and then change the parent or higher nodes to the new leaf node [10].

2) Generation of the decision tree

Corresponding to the local selection of the model, the pruning of the decision tree corresponds to the global selection of the model. Decision tree generation considers only local optima, and decision tree pruning considers global optima.

3) Feature selection

If the number of features is large, the features are selected at the beginning of the decision tree learning, leaving only the features that have sufficient classification power to treat the training data. (For example, by not choosing a name as a feature.)

B. Typical algorithm

Typical algorithms for the decision tree are ID3, C4.5, CART, etc. ICDM (the IEEE International Conference on Data Mining), the international authoritative academic organization and the international conference on data mining, selected the top ten classical algorithms in the field of data mining in December 2006, and the C4.5 algorithm ranked first. The C4.5 algorithm is a kind of classification decision tree algorithm in machine learning algorithms, and its core algorithm is the ID3 algorithm. The classification rules produced by the C4.5 algorithm are easy to understand and have high accuracy. However, in the process of tree construction, the data set needs to be successively scanned and sorted many times, which will lead to the inefficiency of the algorithm in practice.

DT is more commonly used in decision analysis, operations research, risk analysis, and data mining. DT is a popular tool in machine learning, too. DT appears as an upside-down tree-like structure. Each node decides on a particular attribute. Interested readers can see the details in [11].

After processing the MNIST dataset, the decision tree model was created with the `DecisionTreeClassifier` function of the `sklearn` library for training and prediction. There are many parameters in the `DecisionTreeClassifier` function. To get the optimal parameter setting, we need to set different decision tree parameters and find the best ones by training the resulting feedback after the prediction. After several experimental attempts, the best parameters of the decision tree were selected as follows: `splitter= "best"`, `min _samples_ split=2`, `min _samples _leaf=1`, and the rest of the parameters can default. The decision tree classifier is constructed using the best parameters and trained on the training set. After training the model, the model is used to predict the test set and compare it with the real label values. The prediction results of the decision tree classifier had an accuracy of 0.7732.

The decision tree applies to numerical and nominal types (discrete data, the results of variables are only valued in the finite target set), and can read the data set and extract the rules contained in some columns of data. There are many advantages to using decision tree models in classification problems. The decision tree calculation is not complicated, easy to use, and efficient. Decision trees can process data with unrelated characteristics and easily construct understandable rules, while rules are usually easy to interpret and understand. Decision tree models also have some disadvantages, such as difficulties in handling missing data, overfitting, and ignoring correlations between properties in the dataset.

IV. CONCLUSION

From the final results, the KNN classifier works better than the best decision tree classifier. However, their effect is not the best of all the algorithms, using the convolutional neural network Algorithms can easily improve the accuracy to more than 90% [12] [13].

REFERENCE

- [1] Y.LeCun, L.Bottou, Y.Bengio, and P.Haffner.Gradient-based learning applied to document recognition.Proceedings of the IEEE , 86(11):2278–2324, 1998.
- [2] A.Krizhevsky and G.Hinton.Learning multiple layers of features from tiny images.2009.
- [3] J.Deng, W.Dong, R.Socher, L.-J.Li, K.Li, and L.Fei-Fei.Imagenet: A large-scale hierarchical im age database.In Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on , pages 248–255. IEEE, 2009.
- [4] C.Sheppard, Tree-based Machine Learning Algorithms: Decision Trees, Random Forests, and Boosting.CreateSpace Independent Publishing Platform, 2017
- [5] A.D.Bhat, H.R.Acharya and S.H.R., "A Novel Solution to the Curse of Dimensionality in Using KNNs for Image Classification," 2019 2nd International Conference on Intelligent Autonomous Systems (ICoIAS), 2019, pp.32-36, doi: 10.1109/ICoIAS.2019.00012.
- [6] S.Cheamanunkul and Y.Freund, "Improved kNN Rule for Small Training Sets," 2014 13th International Conference on Machine Learning and Applications, 2014, pp.201-206, doi: 10.1109/ICMLA.2014.37.
- [7] T.Makkar, Y.Kumar, A.K.Dubey, Á.Rocha and A.Goyal, "Analogizing time complexity of KNN and CNN in recognizing handwritten digits," 2017 Fourth International Conference on Image Information Processing (ICIIP), 2017, pp.1-6, doi: 10.1109/ICIIP.2017.8313707.
- [8] A.Comeau and C.McDonald, "Analyzing Decision Trees to Understand MNIST Misclassification," 2019 IEEE MIT Undergraduate Research Technology Conference (URTC), 2019, pp.1-4, doi: 10.1109/URTC49097.2019.9660504.
- [9] K.Lavanya, S.Bajaj, P.Tank and S.Jain, "Handwritten digit recognition using hoeffding tree, decision tree and random forests — A comparative approach," 2017 International Conference on Computational Intelligence in Data Science (ICCIDS), 2017, pp.1-6, doi: 10.1109/ICCIDS.2017.8272641.
- [10] M.Bari, A.Ambaw and M.Doroslovacki, "Comparison of Machine Learning Algorithms for Raw Handwritten Digits Recognition," 2018 52nd Asilomar Conference on Signals, Systems, and Computers, 2018, pp.1512-1516, doi: 10.1109/ACSSC.2018.8645138.
- [11] Liu, Xu, Y., Duan, G., Qiu, C., & Tan, J.(2021).Accurate on-line support vector regression incorporated with compensated prior knowledge.Neural Computing & Applications, 33(15), 9005–9023. <https://doi.org/10.1007/s00521-020-05664-2>
- [12] M.Kayed, A.Anter and H.Mohamed, "Classification of Garments from Fashion MNIST Dataset Using CNN LeNet-5 Architecture," 2020 International Conference on Innovative Trends in Communication and Computer Engineering (ITCE), 2020, pp.238-243, doi: 10.1109/ITCE48509.2020.9047776.
- [13] E.Xhafera, E.Cina and L.Toti, "Classification of Standard FASHION MNIST Dataset Using Deep Learning Based CNN Algorithms," 2022 International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT), 2022, pp.494-498, doi: 10.1109/ISMSIT56059.2022.9932737.