Turkish Journal of Engineering

https://dergipark.org.tr/en/pub/tuje

e-ISSN 2587-1366

Text classification by machine learning algorithms using a new text feature extraction method based on image processing

Ahmet Çelik*[1], Deniz Kaptan[2]

[1]*Kütahya Dumlupınar University, Department of Computer Technology, Turkey, ahmet.celik@dpu.edu.tr*
[2]*Kütahya Dumlupınar University, Department of Electronic and Automation, Turkey, deniz.kaptan@dpu.edu.trl*

Cite this study: Çelik, A., & Kaptan, D. (2025). Text classification by machine learning algorithms using a new text feature extraction method based on image processing. Turkish Journal of Engineering, 9(4), 712-724.

https://doi.org/10.31127/tuje.1718023

Keywords
Artificial Intelligence
Machine learning
Image processing
Feature extraction
Character recognition
Digital data processing

Research/Review Article

Received:12.06.2025
Revised:12.09.2025
Accepted:17.09.2025
Published:30.10.2025

Abstract

Accurate text and character identification on documents using smart technologies is a very important method of obtaining data. The complex and irregular text and characters on the images, as well as the use of different writing styles, affect the text recognition success of both Artificial Intelligence (AI) and Machine Learning (ML) technologies. Manually transferring texts and characters from paper format documents to digital media creates a great waste of time and labor. In addition, when documents containing direct text are scanned and transferred in a computer environment, the texts cannot be edited. OCR (Optical Character Recognition) methods, which are proposed as a solution to this situation, are one of the Natural Language Processing (NLP) tasks. In particular, it has been observed that even in current artificial intelligence-based OCR software, the characters 0 and O are confused with each other. In this study, it is suggested that image pre-processing should be done on images containing characters in order to increase the success of character recognition. In the study, a new model was designed to increase the success of correctly recognizing 0 and O characters that are very similar to each other. In the study, image pre-processing was applied to the images of 408 characters. Classification successes were measured by using kNN, SVM and Logistic Regression algorithms on the data set. Additionally, the classification performance of 0 and O characters was measured on the artificial intelligence-based Google Documents tool. According to the results obtained, the success of recognizing 0 and O characters with the LR machine learning algorithm was realized at the rate of 1.00 according to the performance metrics.

1. Introduction

It is very difficult and time-consuming to digitize the contents of paper documents that contain a lot of text by hand. Another method is to transfer the information in paper documents to the computer system by scanning the documents and as images. However, when these data need to be re-examined, it is very difficult and time-consuming to read the contents of the document and examine them word by word. In addition, if the quality of the characters, font properties and images in the paper documents that are scanned and transferred to digital media is low. For this reason, Optical Character Recognition (OCR) software recognizes these texts as incorrect or incomplete.

The development of artificial intelligence (AI)-based applications that have the ability to successfully automatically read text in images is an active area of research [1]. Today, it is seen that difficulties are encountered in the recognition of irregular texts [2]. Complex backgrounds in text images, low resolution, insufficient or excessive brightness, and distortion in paper images containing text also reduce the success of existing character recognition methods [3]. The increase in technology has led to a rapid increase in the data to be processed. Classification and clustering algorithms are used to solve the problems caused by data dispersion and data multiplicity [4]. The large increase in text data has revealed the need for text classification [5]. The most effective method for automatically dividing a large number of text documents into one or more categories is to classify [6].

There are many difficulties when converting image format documents that contain text to a computer editable format. Documents in image format are divided into sections as lines, words and characters. One of the most common difficulties is the misclassification of similar characters due to the way which are written [7].

In addition, today, research is being carried out to recognize the characters in the air writing method. Air typing has been one of the key areas that people have turned to use in recent times, providing ease of communication between humans and machines, as well as the metaverse world. Advanced software based on machine learning has been used to accurately identify air writing characters [8].

Using machine learning and deep learning, the contents of documents can be classified on the basis of text classification. Data can be used to categorize into classes/groups/topics with the help of models trained using artificial intelligence methods. Today, text classification is widely used in web content, search engines, e-mail filtering, spam detection, intent detection, topic tagging, data classification, and sentiment analysis processing [9].

However, a successful text character recognition method should be used if the texts are on a printed document or image. Optical Character Recognition (OCR) methods are used to accurately detect the characters in printed documents or images and bring them into an editable text format. Image processing steps form the basis of character recognition methods [10].

Documents to be transferred to digital media may contain different writing styles, font sizes and font formats as well as handwritten texts and characters. It will be significantly beneficial to correctly define these texts by smart systems and convert them into editable text [11]. Today, the use of automated computer systems using machine learning algorithms has become mandatory. The Complex and big data sets must be analyzed, categorized, and stored with the help of prediction. This data analysis is very difficult for humans to do, questionable accuracy and takes lots of time [12]. Machine learning is a method of learning from historical data and making classification in the same way with the help of algorithms. Machine learning methods improve a measure of performance using existing data sets, causing an increase in computer usage. Machine learning methods are carried out with supervised, unsupervised, semi-supervised and reinforcement learning [13].

In supervised learning, there are predetermined classes, and one of these classes is predicted when the training examples are tested. Reinforcement learning, on the other hand, is the method used to get the most out of a particular reward. The popular way to increase reward in this learning algorithm is to choose actions that have been tried before and proven to be effective and efficient in generating the necessary reward. In unsupervised learning, within the data set, target learning is a learning method that does not have classes. Learning clusters are created by computer algorithms. In this method, the dataset shows how likely it is to detect a particular object in the future [13].

In a semi-supervised learning, certain sets or types are present in the input data. In this learning method, new data can be analyzed and new species that are not specified in existing labels can be identified [12].

Reinforcement learning (RL) is one of the most widely used machine learning techniques in decision prediction and classification tasks. In this method, the learning task can be performed even if there is very little information about the parameters. In the RL method, predefined rules are not used; instead, learning is achieved by using one's own experiences (by receiving rewards or punishments) [14]. There are plenty of studies where machine learning algorithms are used in different fields and success comparisons have been made with each other [15]. The relationship between machine learning algorithms and training data dimensions is also used from the UCI dataset platform [16].

Optical character recognition (OCR) operation is used to convert text images into editable text for digital documents based on computer vision methods [17]. Support Vector Machine (SVM), k Nearest Neighbor (kNN), Decision Tree, Neural Network and Naive Bayes machine learning algorithms are used in text, character and document recognition and classification [4]. In most of these studies, detailed pre-processing steps were not performed.

Jing et al. [6], used kNN to develop a density-based method by reducing the training data in order to increase the speed of the nearest neighbor algorithm in text classification processes. In this method, each sample data class is divided into several clusters and the sample noises are reduced. Highly similar documents are collected in the same class. According to the results obtained, it was shown that the method used could increase the computational speed of kNN text classification.

Lu et al. [18], used the kNN algorithm to classify text on the HADOOP platform. The kNN nearest neighbor algorithm is used in simple, effective, and linear classification operations. In the study, a pre-image processing method was not used and it was seen that the algorithm got fast results in the HADOOP environment and with the HADOOP simple programming models, fast processing of big data could be achieved.

Panhwar et al. [19] used an artificial neural network model and they applied text recognition on signage images in the city written in Urdu and English. Liu et al. [20] used a new model that can recognize Korean, Russian, Greek, and Chinese texts. In the study, text recognition achievements in different languages were compared. Shiferaw et al. [21] implemented a handwritten Amharic character recognition application. The study proposes a hybrid approach that combines convolutional neural networks (CNNs) with machine learning classifiers to improve recognition accuracy. In the study, the hybrid ResNet50 + SVM model achieved the highest accuracy. Im and Chan [22] implemented an application that can identify text from video images. The study used an automatic computer vision method based on an image processing method. In computer vision, the analysis of video data, especially inference by focusing on the information contained in these data has become an important research area.

Additionally, Natural Language Processing (NLP) is a field of research in which artificial intelligence systems are used to recognize data within a document. OCR is one of the most important tasks in the field of natural language processing. Character recognition is the foundation of text recognition [23]. When the recognition process is performed successfully, it will be useful to use Text Mining (TM) methods to obtain useful information

by extracting meaningful words from complex text sets containing large amounts of text information recorded in digital media [24]. In the literature, studies show that well-defined machine learning models outperform deep learning models [25].

In this study, attributes were extracted from the image data of 408 0 and O characters and a data set consisting of 4 attributes data of each character was created. Attribute data were obtained as a result of image preprocessing steps and recorded in the data set. Using this data set, the success of kNN, SVM and LR machine learning algorithms in correctly recognizing 0 and O characters was measured. In addition, feature analysis was performed using the box plot method in the study. In addition, the success of the Google Document tool, which uses advanced artificial intelligence technology, in classifying these characters was measured, and the success of distinguishing between 0 and O characters was compared with the developed model. In the study, a new method has been presented to find a solution to the problem of incorrect and incomplete reading of 0 and O characters on many OCR platforms today.

2. Materials and Methods

Machine learning algorithms (ML), which enable computers to learn new rules and gain experience from these rules, are defined as a subfield of artificial intelligence (AI)[19]. Machine learning algorithms are widely used to perform classification and prediction tasks in the fields of healthcare [26], Unmanned Aerial Vehicle (UAV) technology [27], industry [28], cyber security [29], art [30], energy [31], education [32] and social media [33] applications.

In this study, three machine learning algorithms were used to perform OCR recognition, which is one of the natural language processing areas. The flow chart of the designed model is shown in Figure 1. In the study, image processing steps were performed on images containing 0 and O characters in different fonts and formats, and a character identification model was designed with kNN, SVM and LR machine learning algorithms. In the first stage of the study, image processing steps were performed on images containing 0 and O character data and characters were detected. The attributes of the detected characters are obtained by using image processing methods. In the study, the data set consisting of 408 records was created.

In order to perform classification and prediction using machine learning methods, training and test data are needed. Training data is used to make the learning process take place, and test data is used to test learning achievement [34]. In the study, 50% (204) of the data were randomly selected from the data set and 50% (204) of the data were randomly selected as the test data. In the last stage kNN, SVM and LR algorithms were trained with the training data and 0 and O character classification was performed on the records selected randomly for testing.

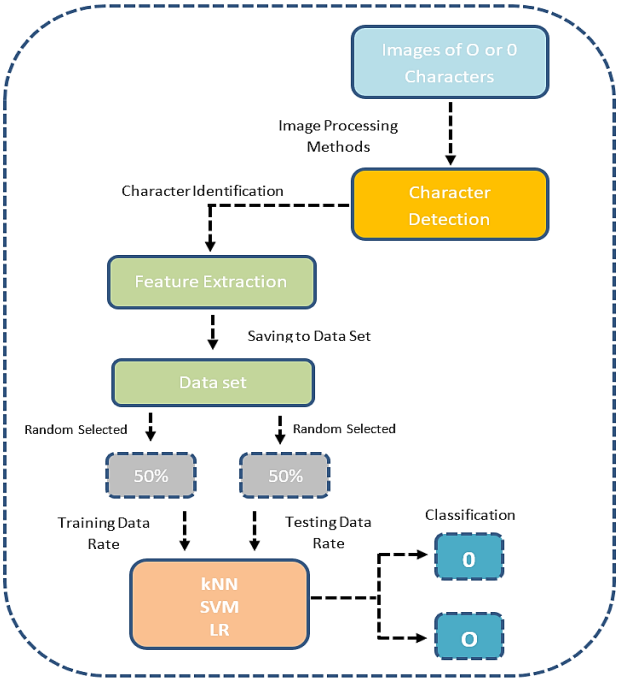
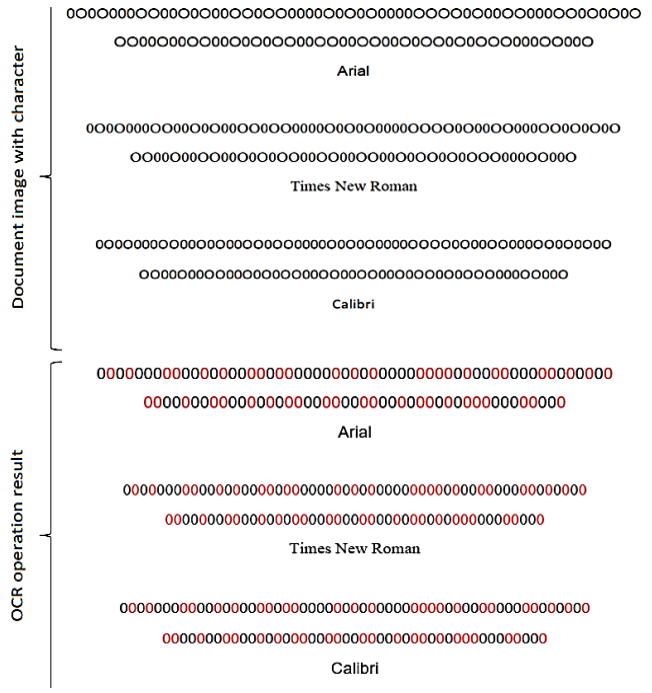


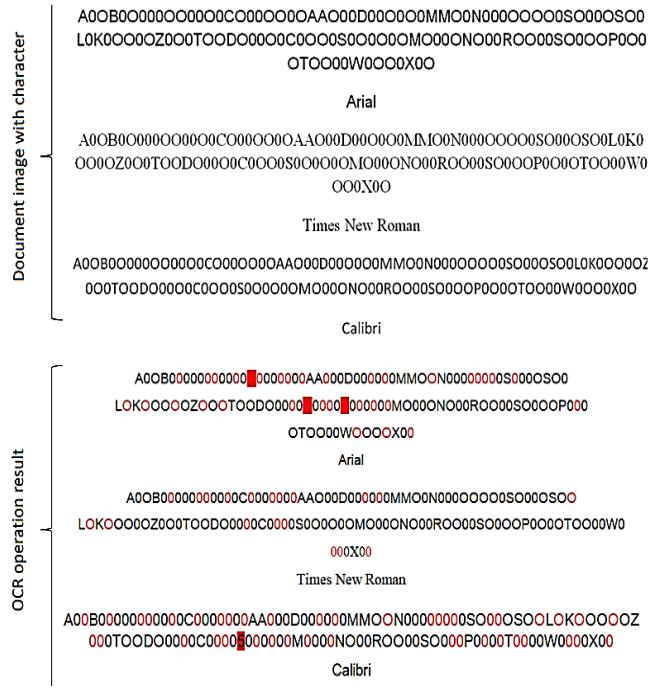
Figure 1. Flow chart of the system

Studies have shown that OCR software has low success in recognizing and classifying 0 and O characters. Since these characters are very similar to each other, it has been determined that their classification success is low.

In this study, the ability of Google Documents [35], which is one of the Google AI (Artificial Intelligence) tools, Google Bard [36] and Tesseract OCR [37], to read characters from images was tested. However, it has been observed that incomplete and erroneous OCR tasks still occur on these platforms. Using Google Documents (Google Drive interface), one of the Open-Source text-to-speech platforms, the optical character recognition test and results from a 100-character (0 and O) images are shown in Figure 2. The sample test image used in Figure 2.a has characters in Arial, Times New Roman, and Calibri typefaces. When we look at the success of the test data, it is seen that the letter O, which is 48 in each typeface, is defined as 0 (zero). This indicates that an incorrect OCR process has occurred. In Figure 2.b, the image with the 0 (zero) characters of the letter O and different characters over the alphabet were tested, but incorrect and incomplete determinations were encountered. In erroneous determinations, the O character is defined as 0. Erroneous detections are shown in red text and incomplete detections are shown in red background.



(a) Results of Google Documents OCR process of O and 0 characters



(b) Results of Google Documents OCR process of O, 0 and different alphabetic characters

Figure 2. Results of Google Documents OCR process

2.1. Image processing

In this study, firstly, the thresholding image processing method was applied to perform background extraction and the edge boundaries of the characters were made clear. The thresholding method is the basis of image processing methods. By using the threshold value determined by the thresholding method, it is possible to reveal objects from the image. The thresholding image process method is calculated using the following equation 1.

$$Tresholding = \begin{cases} Ip_{x,y} = 0, & Ip_{x,y(gray)} < Treshold_{Value} \\ Ip_{x,y} = 255, & Ip_{x,y(gray)} \geq Treshold_{Value} \end{cases} \quad (1)$$

$Ip_{x,y}$ is image pixel color value, $Ip_{x,y(gray)}$ is image pixel gray color value, $\text{Treshold}_{\text{Value}}$ is the value which is selected and usually using default 127. Figure 3 shows the interface where image processing steps are applied and attributes are extracted on the detected character images. Then the lower, upper, left and right boundaries of the characters were determined. After this step, the attribute data of 408 characters (204 0 and 204 O characters) obtained from this interface were recorded in the data set.

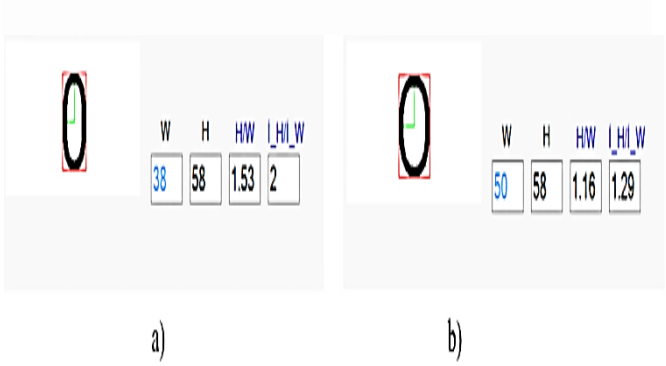


Figure 3. Interface for calculating character attributes: a) Attributes of character 0(Zero), b) Attributes of character O

The regions in which the character attributes are generated from the character it calculates are shown in Figure 4. In the figure, the *Width* (*W*) attribute, the width of the character, the *Height* (*H*), the height of the character, *Inside Width*(*I_W*) is the inner edge radius horizontally from the center point of the character, and *Inside Height* (*I_H*) is the vertical inner edge radius from the center point of the character.

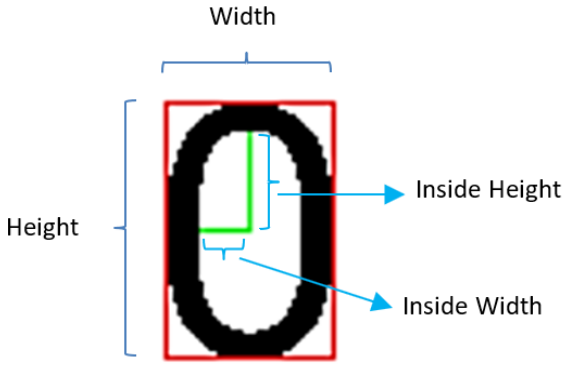


Figure 4. Display of attributes on the character image.

The dataset contains 204 0 characters and 204 O characters, including normal, italic and bold styles. Sample images in the dataset are shown in Figure 5. In the study, pictures with a resolution of 219x93 containing characters belonging to different fonts and formats were used.

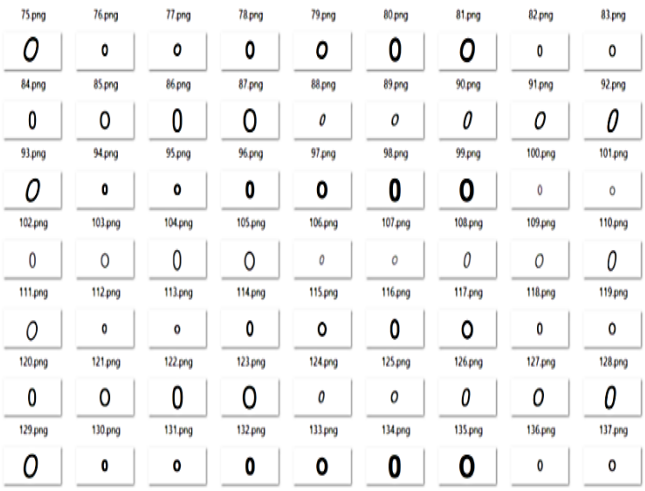


Figure 5. Sample character images in the dataset

The records in the dataset used in this study were created using 27 fonts. The image of each of the character registers is created one by one from a word operation program interface. The dataset contains a total of 408 characters of image recordings, all of which were used for training or testing. The character images manually generated by checking within the dataset. The images of the characters with a resolution of 219x93 were created in the dataset. In the study, the character sizes of each font were selected randomly by selecting 8-12-16. In addition, Normal (N) and Bold (B) fonts were applied to each sample character, and the Italic (I) font was used additionally on the character containing 9 fonts. Table 1 provides information about the fonts, fonts and font sizes of the characters in the dataset.

Table 1. Characters, fonts, typefaces and font sizes

Fonts	Typefaces	Font sizes
Calibri	N-I-B	8-12-16
Arial	N-I-B	8-12-16
Times New	N-I-B	8-12-16
Comic San	N-I-B	8-12-16
Tahoma	N-I-B	8-12-16
Courier New	N-I-B	8-12-16
Verdana	N-I-B	8-12-16
Lucida Sans	N-I-B	8-12-16
Book Antiqua	N-I-B	8-12-16
Bauhaus 93	N- -B	8-12-16
Cambria	N- -B	8-12-16
Californian FB	N- -B	8-12-16
Bodoni MT	N- -B	8-12-16
HoloLens MDL2 Assets	N- -B	8-12-16
Calisto MT	N- -B	8-12-16
Century	N- -B	8-12-16
Centaur	N- -B	8-12-16
Baskerville Old Face	N- -B	8-12-16
Imprint MT Shadow	N- -B	8-12-16
Ebrima	N- -B	8-12-16
Eras Demi ITC	N- -B	8-12-16
Gill Sans MT	N- -B	8-12-16
Footlight MT Light	N- -B	8-12-16
Gadugi	N- -B	8-12-16
Microsoft Himalaya	N- -B	8-12-16
Goudy Old Style	N- -B	8-12-16
Felix Titling	N- -B	8-12-16
Sembol	N- -B	8-12-16
Garamond	N- -B	8-12-16

2.2. Machine Learning Algorithms

In this study, kNN, SVM and LR machine learning algorithms were used in the classification process. These algorithms have also been used in classification and prediction processes in many studies. Machine learning algorithms use the experiences gained from the training data and the test data during prediction or classification operations on the dataset.

2.2.1. kNN (k Nearest Neighbor Algorithm)

kNN is a classification algorithm, proposed by Cover and Hart in 1967. In the kNN algorithm, data is divided into subgroups and new unclassified data are classified according to their proximity to previously classified records. This classification is carried out by looking at the voting of k nearest neighbor values. The proximity ratios of the data are calculated by using Euclid, Chebyshev, Manhattan, and Mahalanobis distance measurements in the kNN algorithm [34]. The kNN nearest neighbor algorithm is used in simple, efficient, and linear classification operations [18]. The kNN algorithm is a widely used algorithm in classification and clustering operations [5, 38].

2.2.2. SVM (Support Vector Machine)

SVM was developed in 1995 as a standard classification technique. The data set is divided into training and test sets while the classification processing. The training set consists of a target value and its associated attributes. The purpose of the SVM model is to deliver the test data (given training data) to the targeted data (predictions) using only the specified attributes [39].

SVM (Support Vector Machine) is one of the most widely used Supervised Learning algorithms for classification such as regression problems. In machine learning, it is used to solve classification problems. The SVM algorithm creates the best line or selection boundary that can classify N-dimensional space. This optimal selection boundary is known as the hyper-plane. With the SVM algorithm, support vectors that contribute to the creation of the hyper-plane are determined [40].

2.2.3. Logistic Regression

Logistic Regression is a logistic algorithm first discovered by David Cox in 1958. This algorithm is usually used for both classification and class probability estimation, depending on the logistic data distribution. In this algorithm, using the linear combination of the properties of the data, a nonlinear sigmoid function is applied to these properties. On the basis of Logistic Regression, the output variable is divided into two

classes, but some data sets need to be expanded to multiple classes. In such cases, it is referred to as multinomial Logistic Regression [41].

2.3. Evaluation Metrics

In the study, Accuracy, Recall, Precision and F score evaluation metrics were used. Evaluation metrics are used to determine or predict classes of test data using training data on a model.

Evaluation metrics are used to measure the performance of a trained classifier when tested on unseen data. Accuracy or error rate is one of the most commonly used metrics in practice and is used by many researchers to evaluate the generalizability of a classifier [42]. Complexity matrix parameters are taken into account in performance metrics (Figure 6).

Complexity on the matrix, True Positive (*TP*), True Negative (*TN*), False Positive (*FP*) and False Negative (*FN*). Data rates are used. In calculations, the best value ratio is 1.0 and the worst value is 0.0 [43, 44].

		Predicted	
		Positive (P)	Negative (N)
Actual	Positive (P)	TP	FN
	Negative (N)	FP	TN

Figure 6. General complexity matrix structure [43, 44]

The complexity matrix structure of the model designed in this study is shown in Figure 7. Positive results are represented by the character 0, and negative results are represented by the character O. In the model, the number that is actually 0 characters and is classified as 0 is defined by *TP* (True Positive), and the number that is actually 0 characters and is incorrectly classified as O characters is defined by *FN* (False Negative). In the model, the number that is actually the character O and is classified as O is defined by *TN* (True Negative), and the number that is actually the character O and is incorrectly classified as the character 0 is defined by *FN* (False Negative).

	Predicted 0	Predicted O
Actual 0	TP	FN
Actual O	FP	TN

Figure 7. Complexity matrix structure of the model

Accuracy (AUC) is obtained by dividing the sum of samples accurately classified by the classifier when tested with new data by the sum of all test data [45, 46].

$$Accuracy = \frac{TP + TN}{TP + TN + FN + FP} \quad (2)$$

Precision is calculated by dividing the number of data that is correctly classified (*TP*) by the total number of positive prediction data (*TP + FP*) [43-44].

$$Precision = \frac{TP}{TP+FP} \quad (3)$$

Recall is calculated by dividing the number of true positive classification (*TP*) data by the total number of positive data [43, 44].

$$Sensitivity(Recall) = \frac{TP}{TP+FN} \quad (4)$$

The F-Measure, or F-score, indicates the measure of accuracy of the test. Precision and sensitivity are calculated based on the results [43, 44].

$$F\ skor = 2 * \frac{Precision*Recall}{Precision+Recall} \quad (5)$$

2.1 Data Analysis Method

In this study, boxplot data analysis method was used to find the output of the samples [47]. Using the boxplot chart, different datasets are easily compared because the ranges and distribution of the data are shown [48]. In the boxplot data analysis method, the input (training) data set is divided into quarters. Figure 8 shows the structure of a Boxplot graph. The boxplot plot structure has a minimum value, lower quartile (LQ), median, upper quartile (UQ), and maximum value. Median and Mean values can also be compared in the boxplot graph [49].

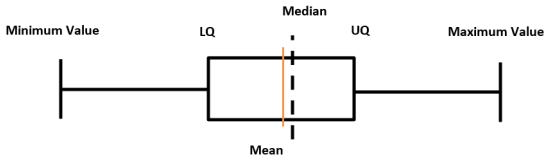


Figure 8. General Boxplot plot structure

Mean and Median values are used as statistics for describing the central trend of a dataset. The median value is used for all data values and is affected by extreme values that are much larger or smaller than the other. Additionally used only one or two of the middle values and is therefore not affected by extreme values [50]. When calculating the median first, the values of a dataset of size n are sorted from smallest to largest.

If n is odd, count value of data, the median is a value in position $(n + 1)/2$ and this is the median value function is calculated using the following equation 6.

$$Median = X_{(\frac{n+1}{2})} \quad (6)$$

if n is even, count the value of data, it is the average of the values in positions $n/2$ and $(n + 1)/2$. This is the median value function is calculated using the following equation 7.

$$Median = \frac{(X_{(\frac{n}{2})} + X_{(\frac{n+1}{2})}}{2} \quad (7)$$

The mean method is a fundamental concept, and the average is used as a basic technique/tool behind many of the measurements associated with the characteristics of the data [51]. It is also generally used to describe any characteristic of an individual/population/class processes. The Mean value is calculated by dividing the sum of all values in the sample plane by the number of

sample values [52]. The value of mean is calculated using by the following equation 8.

$$Mean = \frac{\sum_{i=1}^{n} X_i}{n} \quad (8)$$

3. Experimental results

The successful performances of the kNN, SVM and Logistic Regression algorithms used in the study are shown in Table 2. According to the results obtained, a success rate of 1.00 was obtained in all metrics in the Logistic Regression model.

Table 2. The performance rates of the kNN, SVM and Logistic Regression algorithms

Algorithms	AUC	CA	F1	Prec	Recall
kNN	0.994	0.985	0.985	0.985	0.985
SVM	1.00	0.995	0.995	0.995	0.995
Lojistic Regression	1.00	1.00	1.00	1.00	1.00

Complexity matrix values are used to calculate performance metrics. The complexity matrix of the classification prediction model obtained by the kNN learning algorithm used in the study is shown in Figure 9. In the figure, 105 of the 105 samples tested as the character 0 were correctly predicted as the character 0. However, since 2 of the O characters are estimated as 0 characters, it is seen that there are 106 of 0 characters in the matrix. Depending on this situation, out of the 99 samples of O characters tested, 97 were correctly predicted as O characters. In addition, since it is estimated that there is 1 of 0 characters as O character, it is seen that there are 98 of O characters in total.

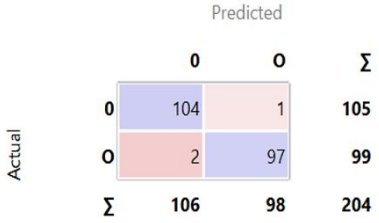


Figure 9. Complexity matrix obtained by kNN algorithm

The complexity matrix of the classification model used in the study, obtained by the SVM learning algorithm, is shown in Figure 10. In the figure, it is seen that all 105 of 0 characters samples tested were correctly classified as 0 characters. However, 1 of O character classified as 0 character, it is seen that there are 106, 0 characters in the matrix. Depending on this, it is seen that 98 of the 99, O character samples tested were correctly classified as O characters.

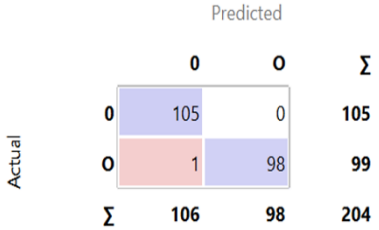


Figure 10. Complexity matrix obtained by SVM algorithm

The complexity matrix of the classification model obtained by the Logistic Regression learning algorithm used in the study is shown in Figure 11. In the figure, it is seen that all 105 of 0 character samples tested were correctly classified as 0 characters. In addition, it is seen that all of the 99 of O character samples tested were correctly classified as O characters.

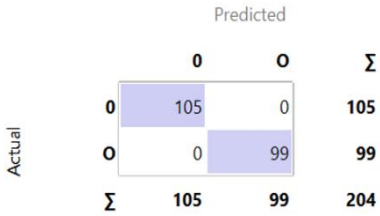


Figure 11. Complexity matrix obtained by Logistic Regression algorithm

By analyzing the ROC (Area Under the Curve) curve graph of a model, classification successes are calculated. According to the classification results of the models, if the area under the curve is large, it turns out that the classification success is also high. Accordingly, if the area under the curve is small, the forecast success turns out to be low.

Figure 12 shows the classification ROC curve graph of the character O obtained by kNN, SVM and Logistic Regression algorithms. In the figure, it is seen that the highest success is achieved with the Logistic Regression algorithm and the lowest success is achieved with the kNN algorithm.

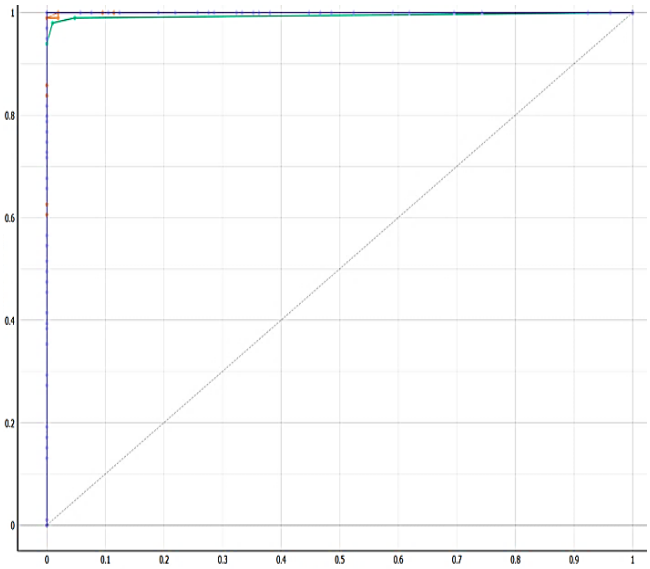


Figure 12.ROC curve of Success in Recognizing O Character of Algorithms

The ROC curve plot of the classification successes obtained by kNN, SVM and Logistic Regression algorithms of the character 0 is shown in Figure 13. In the figure, it is seen that the highest success is achieved with the Logistic Regression algorithm and the lowest success is obtained with the kNN algorithm.

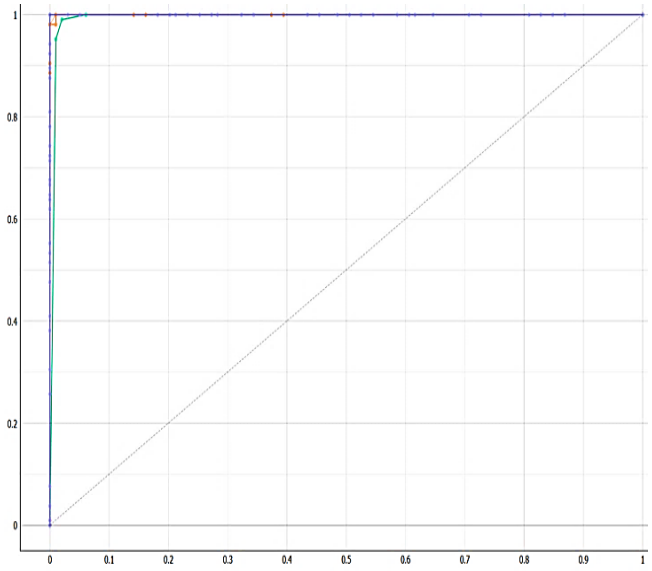


Figure 13.ROC curve of success in Recognizing 0 Character of Algorithms

In the ROC curve graphs, it was seen that the classification success of 0 and O characters was obtained with the highest Logistic Regression algorithm and the lowest with the kNN algorithm. In addition, with the Logistic Regression algorithm, it is seen that there is no misclassification and its success is 100% for both characters.

In this study, as a result of the calculation of the attributes, significant differences emerged for both characters. In particular, differences were observed in the Mean and Median values of the attributes. For the data analysis of the features used in this study, boxplot analysis method based on Mean and Median values was used.

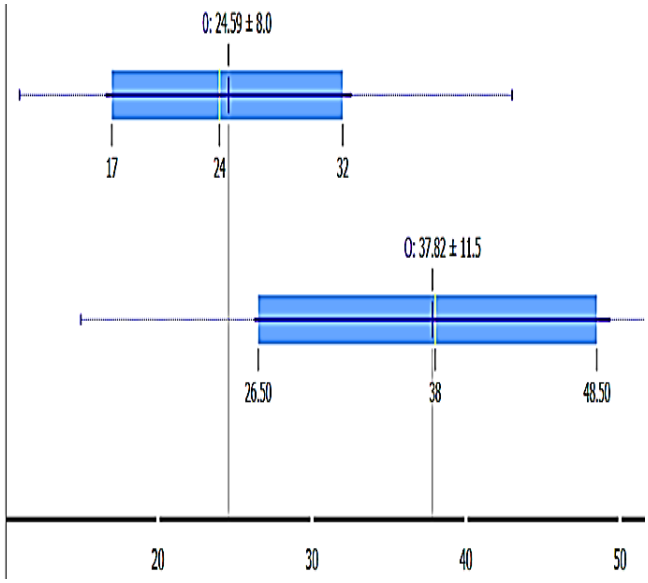
Figure 14 shows the boxplot plot based on the Mean and Median values of the attributes of the characters 0 and O.

In Figure 14.a, it is seen that the median value is 24 and the Mean value is 24.59 of the W attribute of the 0 character. In addition, it is seen that the Median value is 38 and the Mean value is 37.82 of the W attribute of the O character.

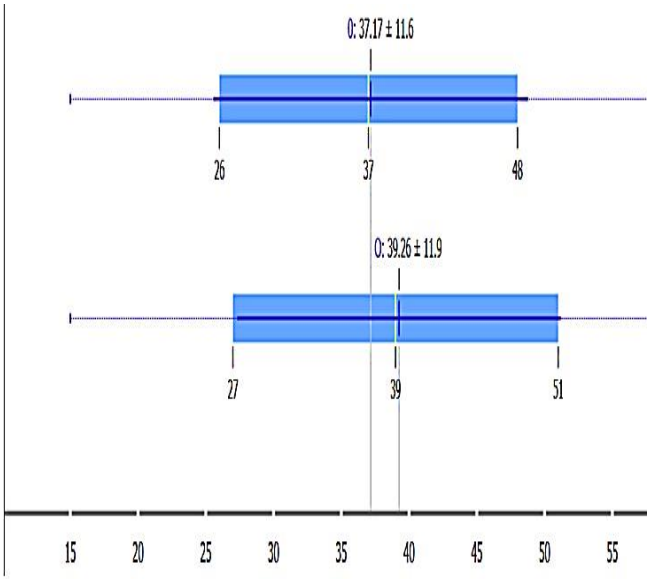
In Figure 14.b, it is seen that the Median value is 37 and the Mean value is 37.17 of the H attribute of the 0 character. In addition, it is seen that the Median value is 39 and the Mean value is 39.26 of the H attribute of the O character.

In Figure 14.c, it is seen that the Median value is 1.04 and the Mean value is 1.0419 of the *H*(*Height)/W(Width*) attribute of the O character. In addition, it is seen that the Median value is 1.53 and the Mean value is 1.5223 of the *H*(*Height)/W(Width*) attribute of the 0 character.

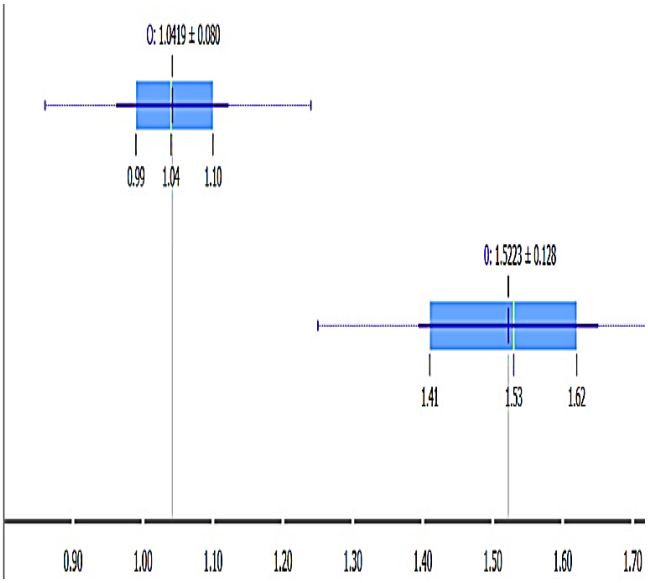
In Figure 14.d, it is seen that the Median value is 1.31 and the Mean value is 1.3631 of the *I_H*(Inside *Height)/I_W (Inside Width)* attribute of the O character. In addition, it is seen that the Median value is 2.43 and the Mean value is 2.8081 of the *I_H* (Inside Height)/I_W (Inside Width) attribute of the 0 character.



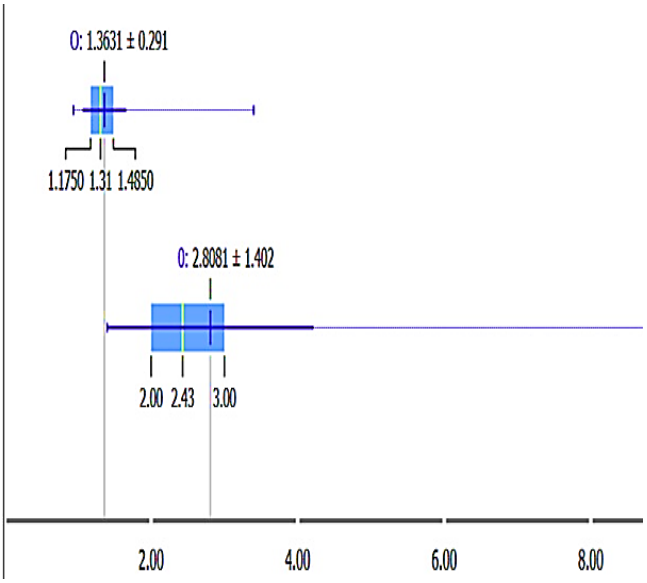
(a) *W* attributes of the characters 0 and O



(b)The *H* attributes of the characters 0 and O



(c)The *H (Height)/W (Width*) attribute of the characters 0 and O



(d)The *I_H* (*Inside Height)/I_W (Inside Width)* attribute of the characters 0 and O

Figure 14. Analyzing the attribute data in the boxplot plot.

4. Discussion

Text detection is one of the main problems of Natural Language Processing [53]. Once the texts are correctly identified, automatic text summarization tasks can also be performed [54]. Recently, many studies have been conducted on the performance of optical character recognition (OCR) operations using artificial intelligence and machine learning algorithms [55]. The comparison of the developed model and the studies in the literature is shown in Table 3.

Table 3. Comparison of the developed model with the researches in the literature

Researchers	Data	Methods	Name of success metric and method.
Rahman et al. [53]	Bangla text documents	Convolutional Neural Network (CNN) Long Short-Term Memory (LSTM)	Accuracy is 95.42% rate using Short-Term Memory.
Hadjadj and Sayoud [56]	Arabic text documents	SMO-SVM Bayes Net	Accuracy is 95% rate using SMO-SVM.
Alzoubi et. al. [5]	Turkish text classification	Support Vector Machine, Naïve Bayes, Long Term-Short Memory, Random Forest, Logistic Regression	Accuracy is 84% rate using Long Term-Short Memory methods.
Gowda and Kanchana [7]	Handwritten text written in the Canadian language	Random Forest SVM kNN	Accuracy is 95% rate using Random Forest.
Nahar et al. [8]	Classifying Arabic Letters	Neural Networks (NN) Random Forest	Accuracy is 88,8% rate using NN with VGG16
Panhwar et al. [19]	Written in Air Urdu and English of Singboad letters	kNN SVM Neural Networks (NN)	Accuracy is 85% rate.
Proposed Model	Specific Characters (0 and O)	Image preprocessing methods & kNN SVM Logistic Regression	A rate of 100% was achieved in all success metrics using Logistic Regression.

Rahman et al. [53] proposed a deep Learning-based study to classify Bangla text documents. Convolutional Neural Network (CNN) and Long Short-Term Memory (LSTM) algorithms were used for classification. In the study, the highest classification accuracy on the Prothom Alo dataset was obtained by using the Long Short-Term Memory (LSTM) algorithm of 95.42%.

Hadjadj and Sayoud [56] developed a model that detects texts in order to identify the citations of the authors in 7 books containing 2900 Arabic texts. In the first stage of the study, the texts in the book were divided into words. In the second stage, the words were parsed into characters and the first characters were detected and used as attribute data in the classifier algorithm. According to the results obtained, 100% successful detection was achieved according to the AUC metric using the SMO-SVM hybrid algorithm.

Alzoubi et al. [5] made a Turkish text classification model according to the inquiries of the customers of an institution. Support Vector Machine, Naïve Bayes, Long Term-Short Memory, Random Forest, and Logistic Regression algorithms were used in the study. By using the Long Term-Short Memory method, was achieved 84% accuracy rate and by using the Support Vector Machine algorithm was achieved a 78% accuracy rate.

Gowda and Kanchana [7] classified handwritten texts written in the Canadian language. They used the Convolutional Neural Networks method to extract the features of the texts. In the study, the characters that were segmented and whose attributes were generated were further classified using Support Vector Machines, K Nearest Neighbors, and Random Forest algorithms. According to the results obtained, 2000 handwritten documents were classified with an accuracy of 95% with the Random Forest, 96% with the Support Vector Machine, and 92% with the k Nearest Neighbor algorithm.

Nahar et al. [8] developed a model that combined a feature extraction process with deep learning methods. The machine learning (ML) and optical character recognition (OCR) methods were used to recognize Airborne Arabic Letters. In the study, the AHAWP dataset consisting of various writing styles and hand signal variations was used to train and evaluate the models. Deep convolutional neural networks (CNNs) models such as VGG16, VGG19, and SqueezeNet methods were used to extract features. Character recognition was performed with neural networks (NNs), random forest (RF), K-nearest neighbors (kNN) and support vector machine

(SVM) machine learning methods using the extracted attributes. According to the results obtained, it was shown that the proposed model achieved the highest accuracy of 88.8% using NN with VGG16.

Panhwar et al. [19] implemented text recognition on 500 signage images using a Neural Network model (NN). In the study, the images of the signs in the city were recorded by a mobile device and a data set was created. According to the results obtained, the classification of characters within the data set was carried out with a success rate of 85%.

In the literature, it is seen that studies have been carried out on the quality and management of dataset records, the size of training data, multi-language recognition and recognition of many fonts in many compilations and research using artificial intelligence and machine learning algorithms [57].

In this study, by using the Logistic Regression machine learning algorithm, the success rate of the text classification model was obtained as 1.00%. The algorithm of Logistic Regression is an algorithm that works on the basis of probability and has a high generalization performance. For this reason, it shows higher classification success compared to other machine learning algorithms. The success rate of this classification is higher than other studies in the literature. In addition, this study is proposed as a new and improved model combined with image processing methods. When other text classification studies in the literature are examined, it is seen that image pre-processing steps are not performed. In addition, it is seen that the new model developed in this study has a lower success rate than the classification success. In addition, it is seen that in other studies in the literature, similar text classifications are not made. There is no study in the literature in which different writing styles and writing styles are classified using artificial intelligence tools and machine learning algorithms. As a result, it shows that the model proposed in this study is a new model that is significantly different from the studies in the literature.

5. Conclusion

Intelligent technologies-based character recognition systems are used to character recognition tasks by reading and analyzing texts that converted to data into electronic format. In these systems, especially the success of recognizing and detecting similar characters is low. It is seen that these systems still need to be improved. In the character recognition tests, it was determined that the widely used Google Documents tool made errors in its task, especially in classifying the 0 and O characters, which are very similar to each other. For solution a new model based on image processing method was developed to increase the recognition success of existing OCR recognition systems for 0 and O characters in this study. Character classification success was tested by using kNN, SVM and Logistic Regression machine learning algorithms. According to the results obtained, the highest performance rates were achieved with the Logistic Regression algorithm, and the success of recognizing 0 and O characters was achieved. Based on the experience in this study, in the future studies the success of OCR recognition will be measure more samples of 0 and O characters. Additionally, improvements to the image pre-processing steps and by using more different character traits and optimization processes to the character classification success of different learning algorithms can be measured. In addition, research and applications will be carried out on other similar characters.

Conflicts of interest

The authors declare no conflicts of interest.

Author contributions

Ahmet Çelik: Methodology, Software, Data curation, Writing-Original, Software, Validation, Visualization, Investigation **Deniz Kaptan:** Draft preparation, Reviewing, Investigation and Editing.

References

1. Manwatkar, P.M. & Singh, K.R. (2015). A technical review on text recognition from images. *Proceedings of the IEEE 9th International Conference on Intelligent Systems and Control (ISCO), Coimbatore, India,* 1-5. https://doi.org/10.1109/ISCO.2015.7282362.
2. Prabu & Sundar, K.J.A. (2023) Enhanced attention-based encoder-decoder framework for text recognition. *Intelligent Automation & Soft Computing, 35(2),* 2071-2086.
3. Guan, T., Shen, W., Yang, X., Feng, Q., Jiang, Z., & Yang, X. (2023). Self-supervised character-to-character distillation for text recognition. *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Paris, France,* 19473-19484. https://doi.org/10.48550/arXiv.2211.00288.
4. Zhou, L., Wang, L., Ge X., Shi, Q. (2010). A clustering-Based KNN improved algorithm CLKNN for text classification. *Proceedings of the 2nd International Asia Conference on Informatics in Control, Automation and Robotics (CAR 2010), Wuhan, China,* 212-215. https://doi.org/10.1109/CAR.2010.5456668.
5. Alzoubi, Y., Topcu, A., & Erkaya, A. (2023). Machine learning-based text classification comparison: *Turkish language context. Applied Sciences, 13*(16), *9428.* https://doi.org/10.3390/app13169428.
6. Jing, Y., Gou, H., & Zhu, Y. (2013). An improved density-based method for reducing training data in KNN. *Proceedings of the International Conference on Computational and Information Sciences, Shiyang, China,* 972-975. https://doi.org/10.1109/ICCIS.2013.261.
7. Gowda, D.K., & Kanchana, V. (2022). Kannada handwritten character recognition and classification through OCR using hybrid machine learning techniques. *Proceedings of the IEEE International Conference on Data Science and Information System (ICDSIS), Hassan, India,* 1-6.

https://doi.org/10.1109/ICDSIS55133.2022.9915906.

8. Nahar, K.M.O., Alsmadi, I., Al Mamlook, R.E., Nasayreh, A., Gharaibeh, H., Almuflih A.S., & Alasim, F. (2023). Recognition of Arabic air-written letters: machine learning, Convolutional Neural Networks, and Optical Character Recognition (OCR) Techniques, Sensors, 23(23), 9475. https://doi.org/10.3390/s23239475.
9. Gaikwad, R., Mulchandani M., & Thakur R. (2024). Review On Text Classification Using Improved Deep Learning Models. *Proceedings of the 2nd International Conference on Computer, Communication and Control (IC4), Indore, India,* 1-5. https://doi.org/10.1109/IC457434.2024.10486233.
10. Çelik, A. (2021). Eğik karakter tanıma başarısını arttırmak için yeni bir yöntemin kullanılması. *Harran Üniversitesi Mühendislik Dergisi, 6*(1), 1-11. https://doi.org/10.46578/humder.720001.
11. Çelik, A. (2020). Optik karakter tanımada hata yayılım algoritmalarının performans kıyaslaması. *Journal of the Institute of Science and Technology, 10(4),* 2328-2340. https://doi.org/10.21597/jist.714810.
12. Lahmer, H., Oueslati, A.E., & Lachiri, Z. (2019). DNA Microarray analysis using machine learning to recognize cell cycle regulated genes. Proceedings of the 2019 *International Conference on Control, Automation and Diagnosis (ICCAD), Grenoble, France,* 1-5. https://doi.org/10.1109/ICCAD46983.2019.9037868.
13. Sasikala, B.S., Biju, V.G., & Prashanth, C.M. (2017). Kappa and accuracy evaluations of machine learning classifiers. *Proceedings of the 2nd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT), Bangalore, India,* 20-23. https://doi.org/10.1109/RTEICT.2017.8256551.
14. Ghafari, R., & Mansouri, N. (2025). Reinforcement learning-based solution for resource management in fog computing: A comprehensive survey. *Expert Systems with Applications, 276,* 127214. https://doi.org/10.1016/j.eswa.2025.127214
15. Çelik, A., & Demirel, S. (2023). Enhanced pneumonia diagnosis using chest X-Ray image features and Multilayer Perceptron and k-NN machine learning algorithms. *Traitement du Signal, 40*(3), 1015-1023. https://doi.org/10.18280/ts.400317.
16. Khan, M.M.R., Arif, R.B., Siddique, M.A.B. & Oishe, M.R. (2018). Study and Observation of the variation of accuracies of KNN, SVM, LMNN, ENN algorithms on eleven different datasets from UCI Machine Learning Repository. *Proceedings of the 4th International Conference on Electrical Engineering and Information & Communication Technology (iCEEiCT), Dhaka, Bangladesh, 124-129.* https://doi.org/10.1109/CEEICT.2018.8628041.
17. Zhang, Q., Liu, F., & Song, W. (2025). IMTLM-Net: improvedmulti-task transformer based on localization mechanism network for handwritten English text recognition. *Complex &Intelligent Systems, 11,* 125-143. https://doi.org/10.1007/s40747-024-01713-8
18. Lu, S., Tong, W., & Chen, Z. (2015). Implementation of the KNN algorithm based on Hadoop. *Proceedings of the International Conference on Smart and Sustainable City and Big Data (ICSSC), Shanghai, China,* 123-126. https://doi.org/10.1049/cp.2015.0265.
19. Panhwar, M.A., Memon, K.A., Abro, A. Zhongliang, D., Khuhro, S.A., & Memon, S. (2019). Signboard detection and text recognition using Artificial Neural Networks. *Proceedings of the IEEE 9th International Conference on Electronics Information and Emergency Communication (ICEIEC), Beijing, China,* 16-19. https://doi.org/10.1109/ICEIEC.2019.8784625
20. Liu, C., Yang, Qin, H.B., Zhu, X., Liu, C.L., & Yin X.C. (2022). Towards open-set text recognition via label-to-prototype learning. *Pattern Recognition, 134,* 109109. https://doi.org/0.1016/j.patcog.2022.109109.
21. Shiferaw, N.A., Mayaluri, Z.L., Sahoo, P.K., Panda, G. Jain, P., Rath, A., Islam, S., & Islam, M.T. (2025). Handwritten Amharic Character Recognition Through Transfer Learning: Integrating CNN Models and Machine Learning Classifiers. IEEE Access, 13, 52134-52148. https://doi.org/10.1109/ACCESS.2025.3553199
22. Im, S.-K. & Chan, K.-H. (2023). Study of small corpus-based NMT for image-based text recognition. *Proceedings of the 9th International Conference on Advanced Computing and Communication Systems (ICACCS), Coimbatore, India,* 1497-1501. https://doi.org/10.1109/ICACCS57279.2023.10112894.
23. Kang, Y. Cai, Z., Tan, C-W., Huang, Q., & Liu, H. (2020). Natural language processing (NLP) in management research: A literature review. *Journal of Management Analytics. 7*(2), 1-34. https://doi.org/10.1080/23270012.2020.1756939
24. Gülbandılar, E., Kızıltepe, S., & Yaylak, F. (2023). Pubmed platformunda cerrahi alanında yayınlanmış makalelerin metin madenciliği teknikleri ile incelenmesi. *Journal of ESTUDAM Information 4*(1), 24-28. https://doi.org/10.53608/estudambilisim.1224150.
25. Bailly, A., Blanc, C., Francis, E. Guillotin, T., Jamal, F., Wakim, B., & Roy, P. (2022). Effects of dataset size and interactions on the prediction performance of logistic regression and deep learning models, Computer Methods and

Programs in Biomedicine, 213, 0169-2607, https://doi.org/10.1016/j.cmpb.2021.106504.
26. Maza, D., Ojo, J. O., & Akinlade, G. O. (2024). A predictive machine learning framework for diabetes. *Turkish Journal of Engineering, 8*(3), 583-592. https://doi.org/10.31127/tuje.1434305.
27. Zeybek, M. (2021). Classification of UAV point clouds by random forest machine learning algorithm. *Turkish Journal of Engineering, 5*(2), 48-57. https://doi.org/10.31127/tuje.669566
28. Alcantara Suarez, E.J. & Monzon Baeza, V. (2023). Evaluating the role of machine learning in defense applications and industry machine learning and knowledge extraction 5(4), 1557-1569, https://doi.org/10.3390/make5040078.
29. Bhadauria, A.P.S., Singh M., Kumar, R., & Kumar, A., (2025). Real Time Intrusion Detection In Edge Computing Using Machine Learning Techniques. *Turkish Journal of Engineering, 9* (2), 385-393.
30. Börekci, A., & Sevli, O. (2023). A classification study for Turkish folk music makam recognition using machine learning with data augmentation techniques. *Neural Computing and Applications,* 1-19, https://doi.org/10.1007/s00521-023-09177-6.
31. Demirtop, A., & Sevli, O. (2024). Wind speed prediction using LSTM and ARIMA time series analysis models: A case study of Gelibolu. *Turkish Journal of Engineering, 8*(3), 524-536. https://doi.org/10.31127/tuje.1431629.
32. Pallathadka, H., Wenda, A., Ramirez-Asis, E. Asís-López, M., Flores-Albornoz, J., & Phasinam, K. (2023). Classification and prediction of student performance data using various machine learning algorithms. *Materials Today: Proceedings, 80*, 3782-3785, https://doi.org/10.1016/j.matpr.2021.07.382.
33. Sıngh, S., Kumar, K., & Kumar, B. (2024). Analysis of feature extraction techniques for sentiment analysis of tweets. *Turkish Journal of Engineering, 8*(4), 741-753. https://doi.org/10.31127/tuje.1477502.
34. Çelik, A. (2023). Determination of the Classification Success of KNN Algorithm Distance Metric Methods on Wheat Seeds Dataset. *Afyon Kocatepe Üniversitesi Fen Ve Mühendislik Bilimleri Dergisi, 23*(5),1142-1149. https://doi.org/10.35414/akufemubid.1263900.
35. Google Drive. Google Documents. https://www.google.com/intl/tr_tr/drive/
36. Google Bard. Bard Google's conversational AI tool-Google Bard. https://bard.google.com
37. Lertsawatwicha, P., Phathong, P., Tantasanee, N., Sarawutthinun, K., & Siriborvornratanakul, T. (2023). A novel stock counting system for detecting lot numbers using Tesseract OCR. *Int. j. inf. Tecnol, 15*, 393-398, https://doi.org/10.1007/s41870-022-01107-4.
38. Liu, L., Zhao, G., & Liang, W. (2023). Slope Stability Prediction Using k-NN-based optimum-path forest approach. *Mathematics, 11*(14), 3071. https://doi.org/10.3390/math11143071.
39. Sasikala, B.S., Biju, V.G., & Prashanth, C.M. (2017). Kappa and accuracy evaluations of machine learning classifiers. *Proceedings of the 2nd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT), Bangalore, India,* 20-23. https://doi.org/10.1109/RTEICT.2017.8256551.
40. Kurani, A., Doshi, P., Vakharia, A., & Shah, M. (2023). A comprehensive comparative study of Artificial Neural Network (ANN) and Support Vector Machines (SVM) on stock forecasting. *Annals of Data Science,* 10,183-208, https://doi.org/10.1007/s40745-021-00344-x.
41. Bartosik A. & Whittingham, H. (2021). The era of artificial intelligence. *Machine Learning and Data Science in the Pharmaceutical Industry. Academic Press.* https://doi.org/10.1016/B978-0-12-820045-2.00008-8.
42. Hossin, M. & Sulaiman, M.N. (2015). A review on evaluation metrics for Data Classification Evaluations. *International Journal of Data Mining & Knowledge Management Process, 5*(2), 01-11. https://doi.org/10.5121/ijdkp.2015.5201.
43. Vujovic, Z. (2021). Classification model evaluation metrics. *International Journal of Advanced Computer Science and Applications, 12*(6), 599-606. https://doi.org/10.14569/IJACSA.2021.0120670.
44. Tharwat, A. (2021). Classification assessment methods. *Applied Computing and Informatics, 17*(1), 168-192. https://doi.org/10.1016/j.aci.2018.08.003.
45. Foody, G.M. (2023). Challenges in the real world use of classification accuracy metrics: From recall and precision to the Matthews correlation coefficient. PLOS ONE, 18(10), 1-27. https://doi.org/10.1371/journal.pone.0291908.
46. Çelik, A. (2023). Buğday tohumu sınıflandırmasının karar ağacı algoritmasıyla gerçekleştirilmesi ve değişken eğitim verisine göre başarı kıyaslaması. *International Journal of Advanced Natural Sciences and Engineering Researches, 7*(11), 44-48.
47. Kampstra, P. (2008). Beanplot: A boxplot alternative for visual comparisonof distributions. *Journal of statistical software 28*(1), 1-9.
48. Abt, M., Loibl, K., Leuders, T., Dooren, W.V., & Reinhold, F. Understanding student errors in comparing data sets with boxplots. Educ Stud Math (2025). https://doi.org/10.1007/s10649-025-10387-z
49. Majaw, N., & Ahmed, S. S. (2023). Exploring data distributions using box and whisker plot analysis. *Proceedings of the 14th International Conference on Computing Communication and Networking Technologies (ICCCNT), Delhi, India,*

1–8. https://doi.org/10.1109/ICCCNT56998.2023.10308191.
50. Ross, S. M. (2014). *Introduction to probability and statistics for engineers and scientists.* Academic Press.
51. Chakrabarty, D. (2018). Generalized fG - mean: derivation of various formulations of average. *American Journal of Computation, Communication and Control, 5*(3),101-108.
52. Chakrabarty, D. (2021). Four formulations of average derived from pythagorean means. *International Journal of Mathematics Trends and Technology 67*, 97-118. https://doi.org/10.14445/22315373/IJMTT-V67I6P512.
53. Rahman, M. M., Aktaruzzaman Pramanik, M., Sadik, R., Roy, M., & Chakraborty, P. (2020). Bangla documents classification using transformer based deep learning models. *Proceedings of the 2nd International Conference on Sustainable Technologies for Industry 4.0 (STI), Dhaka, Bangladesh,* 1–5. https://doi.org/10.1109/STI50764.2020.9350394.
54. Alomari, A., Idris, N., Qalid, A., & Alsmadi, I. (2023). Improving Coverage and Novelty of Abstractive Text Summarization Using Transfer Learning and Divide and Conquer Approaches. *Malaysian Journal of Computer Science, 36*(3), 271–288. https://doi.org/10.22452/mjcs.vol36no3.4.
55. Meng, F., & Ghena, B. (2023). Research on text recognition methods based on artificial intelligence and machine learning. *Advances in Computer and Communication, 4*(5), 340-344, http://dx.doi.org/10.26855/acc.2023.10.014.
56. Hadjadj, H., & Sayoud, H. (2021). Arabic authorship attribution using Synthetic Minority Over-Sampling Technique and principal components analysis for imbalanced documents. *International Journal of Cognitive informatics and natural intelligence (IJCINI), 15*(4), 1-17. http://doi.org/10.4018/IJCINI.20211001.oa33
57. Meng, F., & Wang, C. A. (2024). Artificial Intelligence and Machine Learning Approaches to Text Recognition: A Research Overview. *Journal of Mathematical Techniques and Computational Mathematics, 3*(3), 01-05.