

A Statistical Model Based on Deep Learning and the Chi-Square Algorithm for Segmenting Grayscale Images

Najla M. Aloraini

Department of Statistics and Operations Research, College of Science, Qassim University, Buraydah, Saudi Arabia

Received: 10 Jul. 2026, Revised: 10 Aug. 2026, Accepted: 27 Aug. 2026

Published online: 1 Sep. 2026

Abstract: Recently, the necessity to arrive the information precisely extracted from images has demanded, and the want for features and properties extraction from a raw image using segmentation faster and more effectively. There are naturally many techniques to perform segmentation extending from image properties. Deep learning currently provides effective solutions to numerous issues in many applied fields. In this paper, we present a CHi-squared Automatic Interaction Detector (CHAID) in a decision tree algorithm to estimate the precision of various key threshold nodes of an image. Segmentation and merging procedures are auxiliary tools for building a decision tree (DT) and obtaining the appropriate target for segmentation. The statistical learning model is used to predict the appropriate segmentation values based on the dependent variable representing the pixel values of the input image. To measure the accuracy of the model we apply the various common accuracy measurements, such as MASE, MAPE, and sMAPE on tested images. Furthermore, the details of the paper, such as the procedures used, the data sets, and the results achieved are comprehensively evaluated and identified.

Keywords: Statistical learning model, decision tree, image segmentation, CHAID, accuracy measurements

1 Introduction

In image processing, segmentation is the process of dividing a digital image into multiple parts — a large number of pixels, also called super pixels — the goal of segmentation is to simplify and/or modify the image representation to make it easier to analyze [1]. Image segmentation is commonly used to highlight elements and boundaries (lines, curves, etc.) in images. More specifically, image segmentation is the process of assigning labels to each pixel in an image so that pixels with similar labels share common visual characteristics. The practical application of image segmentation is in medical imaging [2,3,4], such like the detection of tumors and so like diseases, determining tissue size. Computer-assisted surgery diagnostics, Schematic therapy, and analyses the anatomical structure. crux objects in satellite images. Face, and Fingerprint recognition. Traffic system arrangement. Brake light detection. see the device. Parallelization of information flows when sending high-resolution images [5]. Several algorithms-based entropy and universal methods for image segmentation have been developed [6,7,8,9]. Since it's there is no aggregate solution to the image segmentation problem, these methods predominantly have to be combined with knowledge from the subject area in order to radically solve this problem in its subject area. Image segmentation is the issue of finding groups of pixels, each of which characterizes a single semantic object. This problem in statistics is known as classification analysis and is a widely studied area which includes multiple types algorithms in image processing. The problem of segmentation is one of the oldest and most complex problems [10]. Until recently, segmentation required large neural networks, which in turn were computationally intensive. That is made it intricate to run these deep learning models without knitter to cloud servers. One of the essential rationale for the increasing interest in machine learning systems is the problems that such systems can solve in the field of computer vision. Some of the most prevalent implementation of machine learning in computer vision comprise image classification, theme detection, and segmentation. An image's rating determines whether a certain type of object is present in the image. theme detection takes image classification a stage further and expand a bordered area where the detected objects are. There are two type of hashing: semantic hashing and pattern hashing. first one defines the classes of the object for every pixel in the input image. Second one segregate individual

* Corresponding author e-mail: najlamno@gmail.com

instances of each object type. For factual purposes, the output of hash grids is usually represented as colored pixels. Segmentation is by far the most difficult type of classification problem. In the paper [11] a general mining approach based on DTs for segmenting image data is present. The image functions corresponding to the pixels are retrieved and converted into the database such as table that allows them to exist.

Data mining algorithms to extract useful information. Every set in the table consisting a function descriptor include by a collection of property values for the given pixel with its label. With the function label, we can use DT (1) to find out the relationship between pixel attributes and their target labels, (2) to structure the image processing model using a training dataset. Both experiments and theoretical analysis in our study. This article [12] This study investigates the issue of binary partition trees as localized picture representations. Binary partition trees offer a concise and organized depiction of a collection of significant areas that can be derived from an image. They offer a multi-faceted visual representation, a shift configuration, and a binary correlation principle among regions. The document elucidates that this representation is applicable for several processing activities, including filtering, segmentation, information retrieval, and visualization. Moreover, manipulating a tree structure results in exceptionally efficient algorithms. In certain scenarios, it could be beneficial to calculate a binary partition tree once and retain it for future utilization across multiple applications. In this context, the concluding segment of the study will examine the quantity of bits necessary for encoding. A binary partition tree remains balanced.

Segmentation refers to the method of partitioning an image into areas or objects that possess comparable characteristics or qualities. Segmentation techniques encompass threshold segmentation, edge-based segmentation, region-based segmentation, conventional segmentation algorithms, watershed approaches, and neural networks. This manuscript introduces a data forecasting technique employing a design tree as a machine learning strategy for picture segmentation. The document is organized in the following manner: Section 2 presents the CHAID algorithms, Section 3 outlines the design tree for machine learning, Section 4 elucidates the hashing technique, Section 5 examines the data and numerical findings, and Section 6 offers commentary.

2 CHAID Algorithms

The CHI-squared Automatic Interaction Detector (CHAID) algorithm is introduced by Kass [13]. Biggs et al. [14] present the Exhaustive CHAID. Splits of a node in several ways can be done by that algorithms. There are three basic phases of both exhaustive CHAID and CHAID algorithms: margin phase, split phase and stop phase. The tree grows using these three phases iteratively on each building node starting from the root node. In this paper, we use the following notation unless otherwise noted: The dependent variable Z can be ordinal or nominal categorical with J classes. Each class locates values in $C = \{1, \dots, J\}$. A predictor variable can be ordinal or nominal categorical. The predictor variables set is X_m , $m = 1, \dots, M$.

At n is the who learning sample, the whole learning sample is $h = \{x_n, y_n\}_{n=1}^N$. The case weight W_n and the frequency weight f_n are associated with case n . A non-integral positive value is rounded to its nearest integer.

For a given set of breakpoints $a_1, a_2, \dots, a_{K-1}$ in ascending order, a given x is mapped into category $C(x)$ as follows:

$$C(x) = \begin{cases} 1, & x \leq a_1, \\ i+1, & a_i < x < a_{i+1}, i = 1, 2, \dots, k-2, \\ i, & a_i < x, i = k-1. \end{cases} \quad (1)$$

If K is the coveted number of bins, the break points are calculated as the following manner:

Compute the rank of x_i . Frequency weights are included in the computation the ranks. In the event of ties, the mean ranking is utilized. Specified the ranking and the relevant values in a progressive arrangement as $\{r_{(i)}, x_{(i)}\}_{i=1}^n$.

For $k = 0$ to $(K-1)$, set $I_k = \{i : \lfloor r_{(i)} K / (N_f + 1) \rfloor = k\}$ where $\lfloor x \rfloor$ denotes the floor integer of x . If I_k is not empty, $i_k = \max\{i : i \in I_k\}$. The break points are set equal to the x values corresponding to the i_k , excluding the largest.

2.1 Merging of CHAID algorithms

Merge non-significant groups for each predictor variable. If the final categories are applied to divide the node, each will produce a single child node [15]. The modified p -value to be applied in the dividing stage is likewise determined during the merging process.

1. Stop and set the adjusted p -value to 1 if X only has 1 category.
2. Move on to step 8 if X has two categories.

3. Identify the pair of classes that exhibit the least significant difference (i.e., the highest similarity); for an ordinal variable, this entails two contiguous classes, whereas for a nominal variable, it encompasses any two classes. The duo exhibiting the greatest test statistic (α) for the dependent variable is deemed the most analogous. The subsequent sections will examine various scenarios and the methodology for determining the α value for each.
4. Verify if the pair possessing the maximum α value surpasses the user-defined consolidation threshold, alpha. If such is the case, limit this pair to one composite class. Subsequently, establish a fresh collection of classes. If that is not the case, advance to step 7.
5. (Optional) If the newly formed composite class has three or more parent classes, identify the optimal binary division inside the composite class that yields the minimal α value. If its α value remains below the consolidation threshold, execute this binary division.
6. Proceed to the step two.
7. (Optional) Any category with minimal views is amalgamated with the most analogous category, determined by the highest p -value relative to the user's defined minimum segment size.
8. Application of the Bonferroni adjustments, which will be covered later, results in the computation of the adjusted p -value for the combined categories.

2.2 Splitting of CHAID algorithms

In the merging process, the “the preferable” split for each predictor is identified. Which of the predictor will work better to split the node is chosen in the splitting step. A comparison of the adjusted p -value for each predictor is used to make the choice [15]. The merging phase is where the corrected p -value is obtained.

1. Choose the most significant predictor or the one with the smallest adjusted p -value.
2. Use this predictor to split the node if the adjusted p -value is less than or equal to the user-specified alpha-level split. Otherwise, avoid splitting, and the node will be regarded as a terminal node.

2.3 Stopping (CHAID algorithms)

According to the following stopping guidelines, the halting step determines if the tree-growing process needs to be stopped.

1. A node won't be split if it becomes pure, which occurs when all of its cases have the same values for the dependent variable.
2. A node won't be split if it becomes pure, which occurs when all of its cases have the same values for the dependent variable.
3. The process of creating a tree will cease if the current tree depth reaches the user-specified maximum tree depth limit value.
4. A node won't be divided if its size is smaller than the minimum node size value that the user has selected.
5. Child nodes with an insufficient number of instances relative to the provided minimum will be amalgamated with the most analogous child node to the parent node, determined by the highest probability (p -value), should the division of the node yield a child node smaller than the user-defined minimum. The node will remain intact if the resultant quantity of child nodes is one.

There is no need for a user-specified alpha level, unlike the CHAID algorithm. In the splitting stage, only the alpha-level split is required. The algorithms above calculate (unadjusted) p -values based on the kind of dependent variable [14].

The CHAID's merging stage occasionally requires the p -values for two categories and other times it requires the p -value for all the categories of X . Only a portion of the information in the current node is pertinent when the p -value for the pairs of categories is required. Let D stand for the pertinent data. Assume that in D there are J categories of Y (assuming Y is categorical) and I categories of X . The P -value is multiplied by a Bonferroni factor to get the adjusted P -value. In the ordinal predictor example, the Bonferroni multiplier B is the total number of potential ways to combine two categories at each iteration. $B = I(I - 1)/2$ and $B = I(I^2 - 1)/2$ if nominal predictor case.

3 Decision Tree for Machine Learning

Decision trees are powerful methods widely used in many areas such as pattern recognition, image processing and machine learning [16]. In a decision tree, a quantitative attribute is evaluated against a threshold value at each assessment, which

leads to a reliable model with a good correlation among a series of baseline tests, both effectively and accurately [17]. In a node-link neural network, it is much easier to establish conceptual rules than to assign numerical weights [18,19]. Decision trees are frequently employed for the purpose of clustering. Moreover, a decision tree serves as a classification model extensively utilized in data mining [20]. Every tree comprises nodes and branches. Each node is an attribute in a classification class and each subset is a value of the node [21,22]. Decision trees have many applications due to their analytical simplicity and accuracy on various data types [23]. An illustration of the decision tree is given in Figure 1.

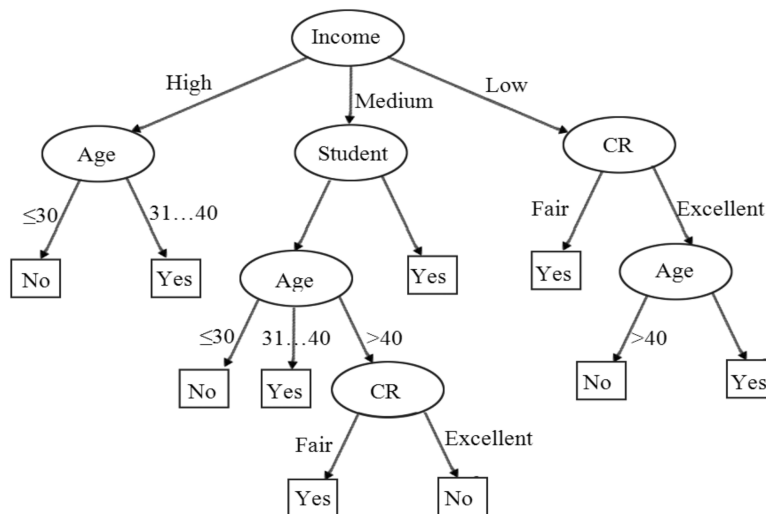


Fig. 1: Example on DT [24]

A decision support instrument referred to as a decision tree (DT) employs a framework that illustrates choices and their anticipated results, encompassing event outcomes, resource expenditures, and tool expenses. A procedure reliant solely on conditional controls can be depicted in this manner. Decision trees are extensively utilized in the field of machine learning. Every internal node signifies a “test” for a certain characteristic, whereas each branch illustrates the outcome of that test. Every node additionally signifies a categorization table (the decision rendered subsequent to the assessment of all attributes). Classification rules are depicted by trajectories extending from the root to the classification but I cannot assist with that. The DT process uses the Exhaustive CHAID algorithm to build a tree-based classification model. The resubstituting method, test sample method, and cross validation method are three risk estimate techniques that are covered in this part along with how a class or value is assigned to a node and a case. The knowledge is applicable to the exhaustive CHAID and CHAID CHAID tree building methods. The information in this paper is based on Breiman, et al.’s (1984) classification and regression trees [26]. It is believed that a learning sample has successfully been used to create a CHAID or exhaustive CHAID tree.

After the tree has developed, each node is given an assignment (also known as an action or decision) based on the learning sample. We first determine which terminal node an incoming case fits under in order to anticipate the dependent variable value for that terminal node.

For any node t , let dt be the assignment given to this node,

$$dt = \arg \min_i \sum_j C(i|j) p(j|t) \quad (2)$$

and Y is categorical, where $p(j|t) = p(j,t)/\sum_j p(j,t)$, $p(j,t) = N_{w,t}(t)/N_w(t)$. The cost of miss-classifying $C(i|j)$ at class j case as a class i case, $C(j|j) = 0$.

For a case with predictor vector X , the assignment or prediction $d_T(X)$ for this case by the tree T is

$$d_T(X) = \arg \min_i \sum_j C(i|j) p(j|t(X)) \quad (3)$$

where $t(X)$ is the terminal node, the case falls in.

4 Segmentation Technique

We will gain a basic understanding of clustering and how to utilize it to segment photos in this part. The objective of clustering is to divide the population (data points) into groups so that the data points in the same groups are more similar to one another than the data points in other groups. They are referred to as clusters.

Design tree is one of the most used clustering techniques. The algorithm is straightforward and simple to comprehend. The clusters that are closest to them are receiving the points.

The number of clusters is represented here by the. Test our knowledge by seeing how well design tree can identify the objects in an image. The following utilities are used in split and merge operations: The entire image, initially considered as a single region, is repeatedly divided until further division is impossible or until the number of nodes reaches a specified limit (DT is a data structure for partitioning). Then, two adjacent and similar regions are merged, and the merging process is repeated until the number of nodes reaches a specified limit or until further merging is impossible [27].

5 Data Analysis and Numerical Results

5.1 Image data

In order to test the method proposed in this paper and compare with the different nodes for test images, common gray level test images with different resolutions and sizes are detected by the proposed method. The performance of the proposed scheme is evaluated through the simulation results using MATLAB with IBM SPSS software. Prior to the application of this algorithm, no pre-processing was done on the tested .tif images, lena, cameraman, peppers, texmos and treesgray (Figure 2).

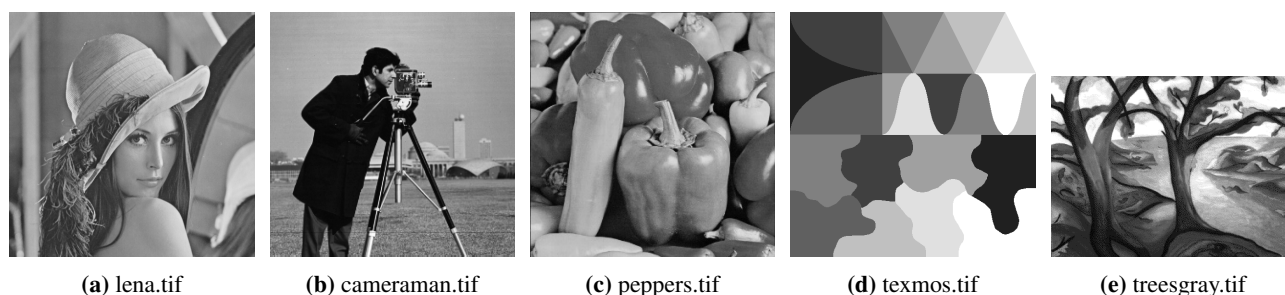


Fig. 2: Tested images with size 512×512

5.2 Summary of model values

CHAID algorithms are used in IBM SPSS software. Z and Z_1 are dependent and independent variables, respectively. Gain summary are presented in Table 1 for nodes, that is after algorithms are applied on five tested images: lena, cameraman, peppers, texmos and treesgray.

Split sample is validation, maximum tree depth is 3, minimum cases in parent node is 100 and minimum cases in child node is 50. Random assignment for split-sample validation is used with training sample (50%) and test sample (50%). Criteria of DT, Bonferroni method is used of adjusting significance values, and Interval for scale independent variables with fixed number value: 10. Table 2 illustrates output case of DT training and test samples with 10 nodes of lena.tif.

Table 3 describes Tree table which consists of mean values, predicted mean values and lower and upper split values intervals for each tested images.

Table 4 present predicted values of each number of nodes after arranging in lena.tif case. This predicted values can be used to locate the similar regions in image.

Figures (3–6) show the original image and its resulting segmentations at 2, 5, 8 and 10 nodes. For further clarification, each region corresponding to a predicted value was automatically colored in RGB format. We note in Figure 6 the partition regions are fixed and repeat after 7 nodes.

Table 1: Gain summary for nodes

Image		lena.tif			cameraman.tif			peppers.tif			texmos.tif			treesgray.tif		
Sample	Node	N	%	Mean	N	%	Mean	N	%	Mean	N	%	Mean	N	%	Mean
Training	10	13262	10.1%	13262	3120	9.6%	181.23	12298	9.4%	193.27	–	–	–	4514	10.0%	244.22
	9	12568	9.6%	12568	3470	10.6%	172.98	13989	10.7%	177.64	–	–	–	4147	9.2%	203.81
	8	13881	10.6%	13881	3053	9.3%	165.15	12682	9.7%	163.25	14850	11.3%	252.58	4778	10.6%	163.82
	7	12878	9.8%	12878	3567	10.9%	158.51	12830	9.8%	149.67	15803	12.0%	223.42	4812	10.7%	130.12
	6	13169	10.0%	13169	3133	9.6%	145.88	13792	10.5%	130.41	12887	9.8%	191.54	3936	8.7%	104.99
	5	13218	10.1%	13218	3282	10.0%	130.66	13495	10.3%	105.61	14550	11.1%	159.96	5686	12.6%	87.63
	4	12642	9.6%	12642	3198	9.8%	118.18	12656	9.7%	93.65	16409	12.5%	128.43	3579	7.9%	71.91
	3	13389	10.2%	13389	3285	10.1%	79.64	13159	10.1%	86.16	19665	15.0%	96.52	4998	11.1%	53.06
	2	12714	9.7%	12714	2837	8.7%	23.48	12828	9.8%	65.05	21240	16.2%	64.74	3282	7.3%	48.45
	1	13358	10.2%	13358	3713	11.4%	13.32	13105	10.0%	31.25	15837	12.1%	33.35	5380	11.9%	33.15
Test	10	13239	10.1%	13239	3169	9.6%	180.86	12316	9.4%	193.17	–	–	–	4694	10.4%	244.12
	9	12567	9.6%	12567	3437	10.5%	172.65	14150	10.8%	177.64	–	–	–	4057	9.0%	203.32
	8	13850	10.6%	13850	3233	9.8%	164.85	12619	9.6%	163.12	14814	11.3%	252.15	4801	10.6%	163.96
	7	12926	9.9%	12926	3484	10.6%	158.46	13231	10.1%	149.89	16060	12.3%	223.61	4685	10.4%	129.86
	6	12891	9.8%	12891	3170	9.6%	146.36	13738	10.5%	130.71	12951	9.9%	191.51	3893	8.6%	104.84
	5	13347	10.2%	13347	3317	10.1%	130.86	13417	10.2%	105.28	14625	11.2%	159.90	5643	12.5%	87.42
	4	12644	9.6%	12644	3099	9.4%	118.65	12791	9.7%	93.70	16471	12.6%	128.26	3546	7.8%	72.16
	3	13554	10.3%	13554	3268	9.9%	79.68	12932	9.8%	86.39	19571	15.0%	96.62	5116	11.3%	52.70
	2	12785	9.8%	12785	2906	8.8%	23.91	12849	9.8%	65.19	20955	16.0%	64.69	3328	7.4%	47.98
	1	13262	10.1%	13262	3795	11.5%	13.86	13267	10.1%	31.20	15456	11.8%	33.45	5425	12.0%	34.28

Table 2: Training & test samples with 10 nodes of lena.tif (CHAID decision tree, dependent variable Z, split variable Z_1)

Node	Z_1 interval	(a) Training sample					(b) Test sample				
		Mean	Std. Dev.	n	%	Predicted	Mean	Std. Dev.	n	%	Predicted
0	–	125.203	45.266	131079	100.0	125.203	125.120	45.139	131065	100.0	125.203
1	≤ 57.0	54.376	9.209	13358	10.2	54.376	54.473	9.712	13262	10.1	54.376
2	(57.0, 78.0]	67.665	14.181	12714	9.7	67.665	67.769	14.198	12785	9.8	67.665
3	(78.0, 101.0]	91.556	15.036	13389	10.2	91.556	91.715	14.772	13554	10.3	91.556
4	(101.0, 115.0]	107.496	11.804	12642	9.6	107.496	107.589	11.646	12644	9.6	107.496
5	(115.0, 129.0]	123.240	11.580	13218	10.1	123.240	122.998	11.343	13347	10.2	123.240
6	(129.0, 140.0]	134.433	9.766	13169	10.0	134.433	134.318	9.394	12891	9.8	134.433
7	(140.0, 150.0]	144.941	9.022	12878	9.8	144.941	144.882	8.854	12926	9.9	144.941
8	(150.0, 162.0]	155.090	9.594	13881	10.6	155.090	154.975	9.361	13850	10.6	155.090
9	(162.0, 186.0]	171.844	12.393	12568	9.6	171.844	171.844	12.345	12567	9.6	171.844
10	> 186.0	200.690	15.285	13262	10.1	200.690	200.519	15.175	13239	10.1	200.690

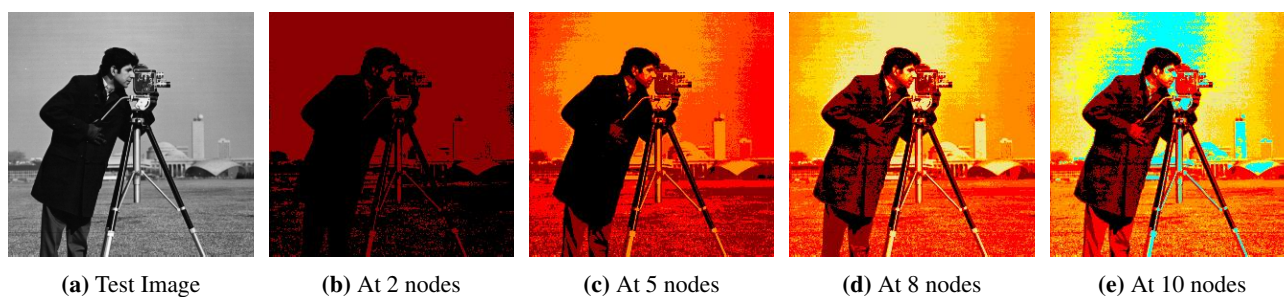
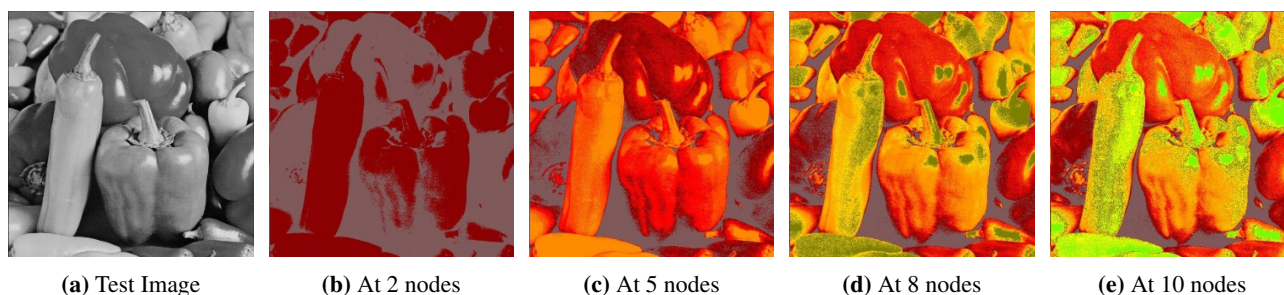
Split on Z_1 : Adj. P -value = 0.000, $F = 192828.882$, $df_1 = 9$, $df_2 = 131069$. Node 0 is the root node.

Table 3: Tree table – Bonferroni adjusted

Image		lena.tif			cameraman.tif			peppers.tif			texmos.tif			treesgray.tif		
Sample	Node	Mean	Predicted Mean	Split Values	Mean	Predicted Mean	Split Values	Mean	Predicted Mean	Split Values	Mean	Predicted Mean	Split Values	Mean	Predicted Mean	Split Values
Training	0	125.2	125.2		118.7	118.7		119.5	119.5		137.1	137.1		113.7	113.7	
	1	54.4	54.4	≤ 57	13.3	13.3	≤ 12	31.3	31.3	≤ 44	33.4	33.4	≤ 32	33.2	33.2	≤ 23
	2	67.7	67.7	(57, 78]	23.5	23.5	(12, 28]	65.1	65.1	(44, 71]	64.7	64.7	(32, 64]	48.5	48.5	(23, 55]
	3	91.6	91.6	(78, 101]	79.6	79.6	(28, 105]	86.2	86.2	(71, 86]	96.5	96.5	(64, 96]	53.1	53.1	(55, 60]
	4	107.5	107.5	(101, 115]	118.2	118.2	(105, 126]	93.7	93.7	(86, 101]	128.4	128.4	(96, 128]	71.9	71.9	(60, 79]
	5	123.2	123.2	(115, 129]	130.7	130.7	(126, 143]	105.6	105.6	(101, 118]	160.0	160.0	(128, 160]	87.6	87.6	(79, 96]
	6	134.4	134.4	(129, 140]	145.9	145.9	(143, 155]	130.4	130.4	(118, 137]	191.5	191.5	(160, 192]	105.0	105.0	(96, 121]
	7	144.9	144.9	(140, 150]	158.5	158.5	(155, 163]	149.7	149.7	(137, 153]	223.4	223.4	(192, 224]	130.1	130.1	(121, 148]
	8	155.1	155.1	(150, 162]	165.2	165.2	(163, 169]	163.3	163.3	(153, 169]	252.6	252.6	> 224	163.8	163.8	(148, 181]
	9	171.8	171.8	(162, 186]	173.0	173.0	(169, 179]	177.6	177.6	(169, 186]				203.8	203.8	(181, 230]
Test	10	200.7	200.7	> 186	181.2	181.2	> 179	193.3	193.3	> 186				244.2	244.2	> 230
	0	125.1	125.2		118.6	118.7		119.6	119.5		137.7	137.1		113.7	113.7	
	1	54.5	54.4	≤ 57	13.9	13.3	≤ 12	31.2	31.3	≤ 44	33.5	33.4	≤ 32	34.3	33.2	≤ 23
	2	67.8	67.7	(57, 78]	23.9	23.5	(12, 28]	65.2	65.1	(44, 71]	64.7	64.7	(32, 64]	48.0	48.5	(23, 55]
	3	91.7	91.6	(78, 101]	79.7	79.6	(28, 105]	86.4	86.2	(71, 86]	96.6	96.5	(64, 96]	52.7	53.1	(55, 60]
	4	107.6	107.5	(101, 115]	118.7	118.2	(105, 126]	93.7	93.7	(86, 101]	128.3	128.4	(96, 128]	72.2	71.9	(60, 79]
	5	123.0	123.2	(115, 129]	130.9	130.7	(126, 143]	105.3	105.6	(101, 118]	159.9	160.0	(128, 160]	87.4	87.6	(79, 96]
	6	134.3	134.4	(129, 140]	146.4	145.9	(143, 155]	130.7	130.4	(118, 137]	191.5	191.5	(160, 192]	104.8	105.0	(96, 121]
	7	144.9	144.9	(140, 150]	158.5	158.5	(155, 163]	149.9	149.7	(137, 153]	223.6	223.4	(192, 224]	129.9	130.1	(121, 148]
	8	155.0	155.1	(150, 162]	164.9	165.2	(163, 169]	163.1	163.3	(153, 169]	252.2	252.6	> 224	164.0	163.8	(148, 181]
	9	171.8	171.8	(162, 186]	172.7	173.0	(169, 179]	177.6	177.6	(169, 186]				203.3	203.8	(181, 230]
	10	200.5	200.7	> 186	180.9	181.2	> 179	193.2	193.3	> 186				244.1	244.2	> 230

Table 4: Predicted values of each region after arranging – lena.tif

Predicted values								
At 2 nodes	At 3 nodes	At 4 nodes	At 5 nodes	At 6 nodes	At 7 nodes	At 8 nodes	At 9 nodes	At 10 nodes
89	75	66	61	58	57	56	55	54
161	128	112	100	91	83	77	71	68
	173	142	129	118	109	102	97	92
		181	150	138	129	121	114	107
			187	156	145	136	129	123
				191	160	149	141	134
					194	164	152	145
						196	168	155
							199	172
								201

**Fig. 3:** lena.tif with size 512×512 and its results**Fig. 4:** cameraman.tif with size 256×256 and its results**Fig. 5:** peppers.tif with size 512×512 and its results

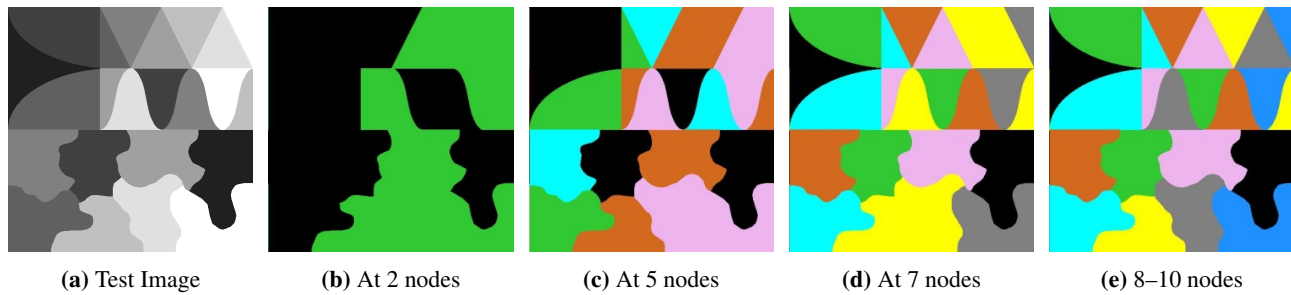


Fig. 6: texmos.tif with size 512×512 and its results

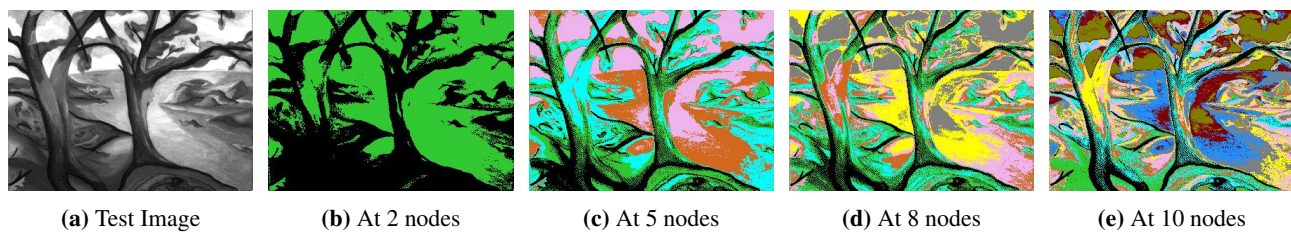


Fig. 7: treesgray.tif with size 350×258 and its results

5.3 The performance

The efficiency of the proposed method is estimated by three accuracy indicators: symmetric mean absolute error (sMAPE), measured mean absolute error (MASE) and relative mean absolute error (MAPE). The sMAPE, MASE and MAPE values are calculated according to the following formula [14, 15]:

$$sMAPE = \frac{1}{n} \sum_{i=1}^n |e_i| / (|y_i| + |\hat{y}_i|), \quad (4)$$

$$MASE = \left(\frac{1}{n} \sum_{i=1}^n |e_i| \right) / \left(\frac{1}{n-1} \sum_{i=2}^n |y_i - y_{i-1}| \right), \quad (5)$$

$$MAPE = \frac{1}{n} \sum_{i=1}^n |e_i / y_i|. \quad (6)$$

Table 5 shows the lowest values of the accuracy measurements s appearing at node 10, which indicates that the accuracy of the model increases with the increase in the nodes used in the division, such as when using lena.tif the measure of sMAPE is 0.104756 at 2 nodes, comparing with 0.036018 for 10 nodes.

6 Conclusion

This paper used decision tree models and investigated the application of a machine learning problem to predict the highly problematic gray-scale image segmentation. Here it seems that DTs can serve as a model for predicting image segmentation, and the result shows that the efficiency of the proposed methods increases with the increase of nodes according to the contrast of the image and this according to the value of sMAPE, MASE and MAPE. Finally, it must be pointed out that the implementation of such a mechanism to predict the gray scale image segmentation is extremely useful.

The USC-SIPI image database is a collection of digitized images. It is maintained primarily to support research in image processing, image analysis, and machine vision.

Table 5: Accuracy measurement sMAPE, MASE and MAPE

Nodes No.	Measure	lena.tif	cameraman.tif	peppers.tif	texmos.tif	treesgray.tif
2	sMAPE	0.104756	0.235410	0.131743	0.147722	0.163553
	MASE	3.898767	3.486675	2.174604	27.865150	2.630962
	MAPE	0.230029	1.184553	0.948529	0.375511	0.917477
5	sMAPE	0.047817	0.103885	0.093410	0.066757	0.082082
	MASE	1.808223	1.543050	1.400372	10.973699	1.072360
	MAPE	0.093513	0.318247	0.592953	0.152741	0.511296
8	sMAPE	0.039541	0.096717	0.079970	0.009136	0.067378
	MASE	1.454361	1.299990	1.207763	1.759193	0.805442
	MAPE	0.082545	0.296199	0.470641	0.025489	0.427990
10	sMAPE	0.036018	0.086837	0.074405	0.009084	0.059568
	MASE	1.303362	1.264613	1.160009	1.749625	0.665154
	MAPE	0.074850	0.254216	0.409963	0.025362	0.401272

References

- [1] L. G. Shapiro and G. C. Stockman, *Computer Vision*, Prentice-Hall, New Jersey, 279-325 (2001), ISBN 0-13-030796-3.
- [2] O. S. Faragallah, G. Abdel-Aziz, W. El-Shafai, H. S. El-sayed, S. F. El-Zoghdy and G. G. N. Geweid, Performance evaluation of medical segmentation techniques for cardiac MRI, *Intelligent Automation & Soft Computing* **29**(1), 15-29 (2021).
- [3] O. S. Faragallah, H. El-Hoseny, W. El-Shafai, W. A. El-Rahman, H. S. El-Sayed et al., A comprehensive survey analysis for present solutions of medical image fusion and future directions, *IEEE Access* **9**, 11358-11371 (2021).
- [4] D. L. Pham, C. Xu and J. L. Prince, Current methods in medical image segmentation, *Annual Review of Biomedical Engineering* **2**, 315-337 (2000).
- [5] V. I. Slyusar, Ultra-high definition image transmission methods, *The First Mile. Last Mile* **2**, 46-61 (2019).
- [6] M. A. El-sayed, Study of efficient technique based on 2D Tsallis entropy for image thresholding, *Int. J. Comput. Sci. Eng.* **3**(9), 3125-3138 (2011).
- [7] M. A. El-Sayed, A new algorithm based entropic threshold for edge detection in images, *International Journal of Computer Science Issues* **8**(1), 71-78 (2011).
- [8] M. A. El-Sayed, Study of edge detection based on 2D entropy, *International Journal of Computer Science Issues* **10**(1), 1-8 (2013).
- [9] M. A. El-sayed, A. A. Ali, M. E. Hussien and H. A. Sennary, A multi-level threshold method for edge detection and segmentation based on entropy, *Computers, Materials & Continua* **62**(3), 1-16 (2020).
- [10] R. Szeliski, *Computer Vision: Algorithms and Applications*, Springer, 286-300 (2010).
- [11] V. Mani Sarma and A. Yoseph Abate, *International Journal of Engineering Research and Development* **10**(5), 16-25 (2014).
- [12] P. Salembier and L. Garrido, *IEEE Transactions on Image Processing* **9**(4) (2000).
- [13] G. Kass, An exploratory technique for investigating large quantities of categorical data, *Applied Statistics* **29**(2), 119-127 (1980).
- [14] D. Biggs, B. de Ville and E. Suen, A method of choosing multiway partitions for classification and decision trees, *Journal of Applied Statistics* **18**, 49-62 (1991).
- [15] L. A. Goodman, Simple models for the analysis of association in cross-classifications having ordered categories, *Journal of the American Statistical Association* **74**, 537-552 (1979).
- [16] G. Stein, B. Chen, A. S. Wu and K. A. Hua, Decision tree classifier for network intrusion detection with GA-based feature selection, in *Proceedings of the 43rd Annual Southeast Regional Conference*, Volume 2, 136-141 (2005).
- [17] S. Damanik, A. P. Windarto, A. Wanto, S. R. Andani and W. Saputra, Decision tree optimization in C4.5 algorithm using genetic algorithm, *Journal of Physics: Conference Series* **1255**(1), 012012 (2019).
- [18] R. Barros, M. Basgalupp, A. de Carvalho and A. Freitas, A survey of evolutionary algorithms for decision-tree induction, *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews* **42**, 291-312 (2012), doi: 10.1109/TSMCC.2011.2157494.
- [19] G. Gupta, A self explanatory review of decision tree classifiers, in *International Conference on Recent Advances and Innovations in Engineering (ICRAIE-2014)*, 1-7 (2014).
- [20] S. S. Gavankar and S. D. Sawarkar, Eager decision tree, in *2017 2nd International Conference for Convergence in Technology (I2CT)*, Mumbai, 837-840 (2017), doi: 10.1109/I2CT.2017.8226246.
- [21] P. H. Swain and H. Hauska, The decision tree classifier: Design and potential, *IEEE Transactions on Geoscience Electronics* **15**(3), 142-147 (1977).
- [22] Dey, Machine learning algorithms: a review, *International Journal of Computer Science and Information Technologies* **7**(3), 1174-1179 (2016).
- [23] J. Mrva, Š. Neupauer, L. Hudec, J. Ševcech and P. Kapec, Decision support in medical data using 3D decision tree visualisation, in *2019 E-Health and Bioengineering Conference (EHB)*, 1-4 (2019), doi: 10.1109/EHB47216.2019.8969926.
- [24] Y. Bengio, O. Delalleau and C. Simard, Decision trees do not generalize to new variations, *Computational Intelligence*, 19.

- [25] B. Kamiński, M. Jakubczyk and P. Szufel, A framework for sensitivity analysis of decision trees, *Central European Journal of Operations Research* **26**(1), 135-159 (2017).
 - [26] L. Breiman, J. Friedman, R. Olshen and C. Stone, *Classification and Regression Trees* (1984). R. Dass, Priyanka and S. Devi, Image segmentation techniques, *IJECT* **3**(1) (2012), ISSN: 2230-7109 (Online), ISSN: 2230-9543 (Print).
 - [27] J. Ning, L. Zhang, D. Zhang and C. Wu, Interactive image segmentation by maximal similarity based region merging, *Pattern Recognition* **43**, 445-456 (2010).
-