I.J. Information Technology and Computer Science, **2022, 5, 26-43**
Published Online on October 8, 2022 by MECS Press (http://www.mecs-press.org/)
DOI: 10.5815/ijitcs.2022.05.03

Design and Implementation of an Improved Obstacle Avoidance Model for Land Mower

Samuel M. Alade
Nnamdi Azikiwe University/ Department of Computer Science, Awka, Nigeria
E-mail: sm.alade@unizik.edu.ng

Adebayo S. Afonrinwo
Lagos State Ministry of Education, Lagos State, Nigeria
E-mail: bayosyntax@yahoo.com

Received: 13 February 2022; Revised: 25 May 2022; Accepted: 19 July 2022; Published: 8 October 2022

Abstract: The paper presents the design, simulation and evaluation of an improved obstacle avoidance model for the lawnmower. Studies has shown that there has been few or no work done has on the total minimization of computational time in obstacle avoidances of land mower. Sample image data were collected through a digital camera of high resolution. The obstacle avoidance model was designed using the Unified Modelling Language tools to model the autonomous system from the higher-level perspective of the structural composition of the system. Automata theory was used to model two major components of the system, which are the conversion of the colour image to binary and the obstacle recognizer components by neural network. The model was simulated using the and evaluated using the false acceptance rate and false rejection rate as performance metrics. Results showed that the model obtained False Acceptance Rate and False Rejection Rate values of 0.075 and 0.05 respectively. In addition, the efficiency of the proposed algorithm used in the present work shows that the time taken to avoid obstacles was faster when compared with another existing model.

Index Terms: Obstacle Avoidance, Autonomous System, Image Recognition, Simulation, Land mower, Finite Automata, Obstacle Detection.

1. Introduction

The alluring nature of man’s ecosystem is as significant and substance to the development and formation of a total man [1]. The ecosystem consists of living (plant and animal) and non-living (earth, grass, soil, atmosphere) organisms, where grasses are described as thin and translucent and often glow in the sunlight and are used to cover the ground in a lawn and other places in our environment. Therefore, the need to prune the tall grasses is necessary to improve the radiance of our surrounding ecosystem. Nevertheless, as man advances intellectually and mentally, the cutting of the tall alluring green plants as a matter, of course, becomes necessary to undertake. The advancement of technology has made the development of handmade tools into a motorized system. The set of interrelated components or parts or a set of principles and procedures can change the nature of the machine in reaction to completely unexpected events during work. Such interrelated parts, otherwise called systems are called autonomous systems [2].

According to [2] the word "autonomous" refers to the machine or the system that can think and take decisions based on the information received from the camera sensor. Autonomous (independent) systems technology is genuinely an alteration and changeover with possible advantages in both expenses incurred and the hazards that come with it. Examples of autonomous systems are a robot, mobile software, vacuum cleaner and lawn mower. Autonomous systems are used in environments that are difficult for humans to enter, such as deep water, space, uncharted territory, and radioactive zones and so on. A greater need exists for autonomous items that can function without being remotely controlled by a person. In order to do this, it is important for autonomous operations to be able to avoid unexpected impediments in three dimensions (3D) or two-plane space (2D). Hence, autonomous systems depend on knowledgeable but reasonable inference abilities to take appropriate actions even in unanticipated events [3].

Recently, due to the advancement and improvement in the performance of computers and sensors, creation of autonomous systems has been made possible. This is because it enables the expansion of the operational range of autonomous robots and the decrease of human tiredness brought on by long periods of control operation, technology for the autonomous operation of robots is also in great demand as a customer need [4]. Moreso, autonomous systems are gradually indispensable tools being used in our daily lives. According to [5], obstacle detection and avoidance have been a fundamental challenge in designing and specifying the requirement for any operation in a real environment that is, cutting or mowing a field of grass, where stones, as well as pebbles, are present. Obstacle avoidance attempts to detect objects that the autonomous system may encounter while traversing its path. Obstacles to be avoided may include trees, rocks, debris, stones, and so on. Maintaining (grass mowing) parks, sporting fields, golf courses and driving ranges is labour intensive [6].

A lawn mower is a machine designed to cut grass and the garden. Lawn mowers are categorized into four major types: manual lawn mower, motorized lawn mower, hover lawn, electric lawn mower and other types of lawn mowers [7]. Cutting grass is time and energy-consuming, therefore there is an interest in strategies to dispense with such assignments. The standard approach for managing a yard cutter is to pay another person to perform it. There is another choice, be that as it may, and that alternative is mechanical autonomy (robotics), which had been in existence since the 1950s deeply rooted as an immutable used in the transformation of human activities and operations.

Nowadays, the robotics industry has reacted to this foreseen interest by making a straightforward automated lawnmower for cutting grass and navigating around an obstacle. The grass cutter does not cut in a specific pattern but it uses copper wire laid around its perimeter to allow its operations. The mower moves in a linear direction until it perceives the copper wire or runs into the obstacle. However, it is unable to perform well in fields that have complex shapes and sizes.

Studies has shown that many autonomous robot systems have been developed for land mower. However, finite state automata-based method has not been applied. This is because the existing models developed for land mower robots are computationally demanding and not adequate to reduce computation time expended in locating obstacles on the lawnmower path in two-dimensional arrays. More so, from the literature review conducted, it can be deduced that no published work has ever been reported on the total minimization of computational time in obstacle avoidances due to the fact that their processes are too cumbersome and most of the obstacle avoidance models for lawn mowers randomly cut lawn which is probabilistic. The need to apply a simplified model-driven technique that is linear, known as FSM that combines image processing techniques with neural network to process digital colour lawn images is considered in this study

Over the years, there have been several efforts exerted by researchers and scholars in designing an obstacle avoidance model. Studies on obstacle detection and avoidance have shown and been reported in literature, particularly in the form of a farm tractors, vehicles and motorized robots [1,8] that existing obstacle avoidance models for autonomous systems are complex [9]. However, with the rigorous and tasking nature of mowing, the specification required for an extensive processing and computational time in searching the data structure for the presence or absence of obstacles along the path of the autonomous system is yet to be determined. Therefore, the need for a computational model to simplify this process is desired. The idea of an autonomous system (lawn mower), the advancement in Global Positioning System (GPS) technology and the computerization of automated mower has enhanced the cutting of lawns without human intervention, the capability to perceive the environment is an important aspect for an automated mower. Any mobile robot that must reliably operate in a dynamic environment must be able to detect obstacles and avoid them. Several sensors are capable of detecting obstacles such as the sonar system which has been used for several years for obstacle detection for mobile robots [10].

Previously, random path planning is used by existing autonomous lawn mowers, and it is accomplished through the use of a land map, a possibility graph, and a search. The location of roads, fields, obstacles, and other topographic details of the area where the vehicle will be driven are all stored on the terrain map. There are nodes and edges in the possibility graph, which depicts potential routes through the map. In contrast, a *" search algorithm, as described in [2], employs edge costs to identify the best routes through the graph. However, this approach is ineffective and a huge waste of time and energy. The machine might not be set up with the correct tyre adjustments and power distribution to its two engines to move the lawnmower in a straight line. As a result, the lawnmower would steer itself toward the same track using this technique on the entire grassy area on the field requiring extensive mowing, which required a lot of time, money, and energy. Similarly, edges that traveled through a recognized obstacle were rejected in order to avoid it. This straightforward method guarantees that barriers will be escaped. However, it reduces the graph's connectivity and might without cause block access to certain areas of the map. The tractor's path planning makes use of edge-building strategies that account for the shape of obstacles and create edges that go around them.

Moreso, existing autonomous lawnmowers randomly cut patterns and back away from encountered obstacles [11]. As a result, a method to determine avoidance paths based on the geometry of barriers is required to increase the effectiveness of obstacle avoidance operations. Some of the methods to find avoidance paths are force fields such as the vector field histogram (VHF), potential field (PF), texture analysis, histogramic in-motion mapping (HIMM), color/hue, sensors, bubble band, fixed sonar, optical flow, and edge detection are some of the techniques used to determine avoidance courses [12-17]. The potential field puts its theory into practice by turning the data gathered by its sensor into actions. The vector field histogram (VHF), on the other hand, enables the detection of unidentified or unknown objects and prevents collisions while simultaneously guiding the mobile robot toward the destination. This approach defines potential functions on a destination location and obstacle position and determines the movement direction based on the gradients of the functions in order to find a path traveling to the destination position while avoiding obstacles [18]. In contrast, real-time map construction is possible using histogramic in-motion mapping. The Cartesian histogram and rectangular grid used by the VHF and APF methods to represent data in two-dimensional array are updated through quick in-motion sampling of the onboard range sensors.

[19] proposed a probabilistic roadmap for obstacle avoidance for a holonomic robot, which can be customized to run more efficiently and work on various problems. Several model approaches [20-22] are available; however, their operations

are computationally intensive. Self-determining and independent lawnmowers have as of late gained popularity. However, cannot cut parallel lines or other progressively muddled and complicated terrains. Despite this challenge, it still cannot detect yet, other more complicated patterns and cannot sense obstacles from a distance, hence the study.

Nonetheless, many of these previous works discussed here are computationally demanding and are not adequate to reduce computation time expended in locating obstacles on the lawnmower path in two-dimensional arrays. More so, from the literature review conducted, it can be deduced that no published work has ever been reported on the total minimization of computational time in obstacle avoidances. This is because their processes are too cumbersome and most of the obstacle avoidance models for lawn mowers randomly cut lawn which is probabilistic. Nonetheless, a robust but simplified finite state automata model that is linear, which combines image processing techniques with neural networks to process digital colour lawn images is considered in this study to get over the constraints and difficulties of such data-driven processes of existing approaches. As it is seen, the major benefits of employing FSA are its universality (application is not restricted to Real-time Operating System) and effectiveness in system prototyping.

This study is aimed at developing an improved obstacle avoidance model using formal language and automata-based technique. Hence, the objectives of the research are to examine the existing obstacle avoidance technique employed in designing land mower, model and simulate the process behaviour using Unified modelling language and the combination of deterministic finite automata model (DFA) and image processing techniques with Neural network to evaluate the behavioural properties of the model and process digital colour lawn images. Therefore, the paper presents a simulation of an obstacle avoidance model of land mower behaviour. This paper is organized as follows: Section 2 discusses the overview of autonomous land mower in relation to obstacle avoidance, the theory employed in the development of improved obstacle avoidance as well as the related work. Section 3 presents Unified Modelling Language model and formal language and automata theory employed for the modelling of land mower component behaviour with JFLAP tool while Section 4 presents the methodology. The simulation and the experimental set up are discussed in section 5, result and discussion in section 6 and the final section gives a concluding summary of the paper.

2. Literature Review

In this section, some of the concepts which are required for better understanding of the paper are briefly described. In this present work, formal language and automata theory is used to describe language and grammar as computational objects used in the design of machines.

An autonomous system or agent is an intelligent self-independent agent that can work on behalf of the system without any interference from that ownership [23]. They are software entities that, utilizing some knowledge or a representation of the user's objectives, perform a certain set of activities on behalf of a user or another software program or agent with a certain level of independence or autonomy, using some knowledge or representation of the user's goals or desires. Examples include autonomous robots (mower, farm tractor, vehicle), various software agents, and many computer viruses. Consequently, the capacities expected of an independent robot include picking up data about nature, working continuously without human intervention, moving all or part of itself through its operational conditions without assistance, and avoiding situations that could be harmful to people, property, or itself, unless those conditions are specified in its design.

Lawn mowing is an important part of the process necessary to keep terrain with grass well-manicured and tidy. Places such as soccer fields, gardens, golf courses and many other places require constant lawn mowing and grass cutting to maintain areas. [24] described mowing as one of the complicated tasks with a large amount of time and effort needed to complete it. In the same context, a lawn mower is a machine containing one or more blades rotating, which is used to cut grass to an even height. The blades may be powered either by hand, an electric motor or an internal combustion engine to spin their rotating blades. An autonomous lawn mower is not a novel concept. Now, the straightforward but tedious work of harvesting and plowing a field may be delegated to an autonomous tractor that never has to rest and does the tasks it is given: first, how the mower perceives and reacts to its surroundings.

Formal language and Automata theory (FLT) is a system of ideas intended to describe language and grammar as computational objects. In other words, the theory is a method used for specifying and defining many diverse activities represented in an infinite set according to a specific set of rules [25]. The concept of formal language and automata theory is founded on set theory and its mathematical properties, where it provides methods for identifying relations among factor, particularly on the presumption that every object of the universe of discourse is closely associated with information [14]. Furthermore, several complex problems like navigation behaviour of Ants, dining philosophy, biomolecular and DNA computing issues, programming language structure and compiler design have been solved using formal language and automata theory [26,27]. In this study, Finite State Automata (FSA) was employed to abstract the model to its simplest form for ease of understanding. Formal definition of a deterministic finite state automation discussed in (1).

Finite Automata: finite automation is formally defined as a quintuple

$$A = (Q, \Sigma, \partial, qo, F) \tag{1}$$

where

Q= {q_1, q_2, q_3, …, q_n}is a finite set of states,
∑= {white (W), red (R), brown (S), black (BK), green (G) and blue(B)} is a finite set called input alphabets, which represents the colour of interest of the input alphabet,
∂ is the mapping function,
q0 is the initial state, and
F ꜫQ which is the set of the final state.

2.1. Related Work

[28] developed a behaviour-based lawn mower robot controller using a sonar camera with Global Positioning System to detect and avoid obstacles. In an attempt, the author proposed ultrasonic sensors for obstacle avoidance and humidity sensor for checking the moisture level on the lawn. The author used the Artificial potential method in converting information collected from sensors into behavioural responses to design and implement a lawn mower robot controller using Finite State Automation. Consequently, the simulation was carried out by making use of a player mobile robot simulator. [16] presented obstacle avoidance for vegetation control using autonomous mowing. In a bid to ensure the efficiency of the mower, a novel autonomous mower enhanced with laser was developed to detect and avoid an obstacle on the testbed. Also, the algorithm designed was evaluated based on quantitative performance measures. The work shows relevance in that it provides an algorithm used in obstacle avoidance. However, the autonomous mower can be used in an outdoor environment but with relatively low speed in processing the time in detecting the obstacles and their avoidance.

[29] investigated the challenge of moving obstacles in the process of obstacle avoidance during navigation. The work also presented the design and validation of a reactive method which considers obstacle velocities to achieve safe visual navigation outdoor. [4] investigated the difficulty encountered by humans while operating in harsh environment. The author developed a robot by using a potential method to establish control logic and verify it through simulation to find an obstacle avoidance path. [5] explores the relationships or connections between the environmental conditions/ factors and edge detector performance in order to exploit edge detector performance as a crucial component of detection and avoidance (DAA) operations in the unmanned aerial systems. The author employed an in-between method, using both visual and quantitative method, where a set of data images containing six different photos was collected from objects an unmanned aerial vehicle through its camera was validated using the edge detection algorithm and implemented using MATLAB. [21] presented a procedure that takes advantage of the efficient path planning capabilities by proposing modified Rapidly Exploring Random Tree that used hypersphere (balls) for planning, otherwise called Ball tree. The model was able to make effective use of computational time to enhance the solution toward a best and low-cost feasible/shortest path to the goal in configuration space. However, the processing step can take minutes and at the computation expense of rejection sampling.

Also, [23] developed an autonomous land mower to remove the challenge of mowing the lawn. The author employed the encoded path planning algorithm by using sensors and computer vision to outline the mowing boundaries and place the lawnmower position during use to avoid obstacle. The system was designed with several components like the microcontroller, motors, camera and others to save and manipulates the positional data to determine the path of the mower and implemented using Simple CV tool. [30] addressed the challenge of ensuring accuracy for a robot. To achieve the aim, the authors designed an algorithm for obstacle avoidance to detect and avoid obstacles. More so, the authors experimented by using the Arduino microcontroller hardware component on the Arduino software IDE environment and implemented using C-programming language. In the study, the authors designed an algorithm and flowchart to detect the presence of any object within a specified distance. The obstacle detection module was constructed using three (3) ultrasonic or sonar sensors to determine the distance required and specified as well as compare the distance between the objects identified and detected. However, the result showed that the combination of the three sensors provides high accuracy in identifying surrounding obstacles. This paper provides basic information in developing a model for avoiding obstacles and a base for further improvement to increase the accuracy and adaptability of obstacle detection in several environments

[31] presented embedded robust visual obstacle avoidance on an autonomous lawnmower. To avoid an obstacle, a prototype autonomous lawnmower was developed having a camera (752*480), which is non-contact obstacle avoidance. The prototype was developed using a low-cost colour camera and on an ARM-based processing board. The result showed that the proposed lawnmower can work without human intervention in all seasons. The obstacle avoidance performance was evaluated for efficiency. The result showed that 88% accuracy was achieved and an 11.9 false Positive rate as compared with the hand mower which gave an accuracy of 86%. This work provides a high obstacle avoidance accuracy of objects, particularly with the earlier visual detection of people, animals, toys and other objects. However, it has lower reliability in detecting other objects such as plants and trees with colours like green and brown. [11] presents a mobile robot using histogramic in-motion mapping method for real-time map building in motion to avoid obstacle. The author made use of sensors that is modeled in a two-dimensional histogram grid to produce maps and instantaneous information for us by the avoidance algorithm employed. Similarly, [13] presents a real-time mobile robot using force field method, particularly the Vector field force to avoid obstacle while simultaneously steering toward the target. However, [11,12] in separate work stated that the methods employed in avoiding obstacle are computational intense but the less accurate when robot is static.

The annual Robotic Lawn Mower Competition, held by the Institute of Navigation starting 2004 to 2012, was a significant event in the past. Although several of the robots performed well in terms of mowing coverage and obstacle avoidance, the prototype machines did not take cost, weather proofness, or manufacturing constraints into account. Particularly, the majority of participants employ pricey sensors like D-GPS [32], which employs a path planning algorithm and is implemented in Java. [9,33] used image data obtained from camera to segment obstacles and grass. The authors used image data to create a straightforward yet effective grass segmentation algorithm based on colour and texture parameters. However, because the test set was so minute, robustness and generality were in doubt. [34] employed field programmable gate array to design and implement an autonomous land mower with the aim of doing the task of human cutting the grass using sensors.

[35] proposes a 3-Dimensinal reactive obstacle avoidance method using audio and laser-based sensing methods to locates/detects the obstacle obstruction in a UAV's path, thus causing the craft to linger there/ on while it determines the optimum escape path, and then gives the UAV the appropriate instructions to follow. The author of [36] suggested a solution where the pathways are based on various line colors. A robot that can distinguish between different colors can choose the best line on its own to get to the target. Both changing situations and poor lighting make this approach ineffective. Additionally, the robot absolutely lacks dynamic skills and is entirely dependent on the sight of the lines, not accounting for obstacles that may be present on the lines themselves.[37] proposed a Potential Field method for determining the shortest route from the starting to the destination points in avoiding obstacle in a robot. for mobile robot with obstacle avoidance. [14] discussed three main approach that can be employed in obstacle avoidance and path planning. The author concluded that these approaches based on their advantages and disadvantages can be synthesized to achieve an autonomous navigation of robot in any environment be it static or dynamic, indoor or outdoor. [38] presented the developed an improved obstacle detection system employing an Arduino Mega 2560 embedded system development board and tools using an Arduino board and sensors. [39] developed a self-independent effective collision avoidance algorithm to detect and avoid obstacle in a vicinity using an ultrasonic sensor and controlling the movement of the vehicle with the view of reducing the original path. Saraswathi and colleagues [40] created an obstacle-detecting robot in 2015 using IR sensor, wheel encoder and controlled by an Arduino microcontroller.

[41] investigated the difficulties encountered in the implementing a mobile multi-robotic system with respect to collision avoidance among agents and other obstacles in a disorderly environment as well as the convergence of each robot to its desired destination. [42] applied finite automata model to design the decision-making system of a mobile robots. In the first place, the robot and environmental information all around are all abstracted into some special symbols and states used in the design of the system similarly, [43] designed, simulated and implemented a robot controller using the logic-labelled finite-state automata using Virtual Robot Experimentation Platform and MiEditLLFSM modelling tool.

3. Materials and Methods

3.1. Unified Modelling language Tool

An object-oriented design and model-based approach formulated using Unified Modelling Language (UML diagrams) and finite automation tool (JFLAP) were employed in this study. The UML design for the obstacle avoidance model offers a standard way of showing the system's architectural blueprints as well as the structure and the logical flow of process or activities involved in the obstacle avoidance of the land mower.

3.2. Java Formal Language and Automata Package (JFLAP)

This is an interactive visualization software tool available for the purpose of teaching mathematical structure, and certain axiomatic rules (formal language), automata theory and provides methods for identifying relations among factor, particularly on the presumption that every object of the universe of discourse is closely associated with information [14]. The JFLAP tool will allow us to construct and work with automata, grammars, L-systems, and regular expressions. In addition, the JFLAP tool can help simulate input strings on deterministic and non-deterministic automata, create parse trees and tables for grammars, and display sequential L-system expansions. Besides, it offers interactive visualizations for building a land mower finite automaton. Again, complicated instances can be readily solved using step-by-step simulation run with JFLAP with less time by giving feedback on cases that have been solved. Furthermore, accuracy of the automata system/model built can be verified.

3.3. Model Design

Model design enables faster development of dynamic systems, and it dictates the standards, specifications and strategies to be followed throughout the building and evaluation of a model. Though the object-oriented design approach (formulated using Unified Modelling Language (UML) diagrams) was employed in this study, the formal language technique was also used to model lower-level descriptions of the two major components of the system. The UML design for the obstacles avoidance model for lawnmowers provides a formal way of showing the system's architectural blueprints. As illustrated in Fig. 1, the UML class diagram shows the static structural view of the system using objects, attributes, operations and relationships among objects. The classes are represented with boxes in three compartments; the top-most part contains the name of the class; the middle section contains the attribute of the class while the third compartment represents methods or object's behaviour. The classes are AutonomousSystem, NeuralNetwork and ImageProcessor. NeuralNetwork class is, however, a component of the *AutonomousSystem* class. The ImageProcessor class performs functions such as preprocessing the image, extracting colour components of an image, computing lawn area, and so on.

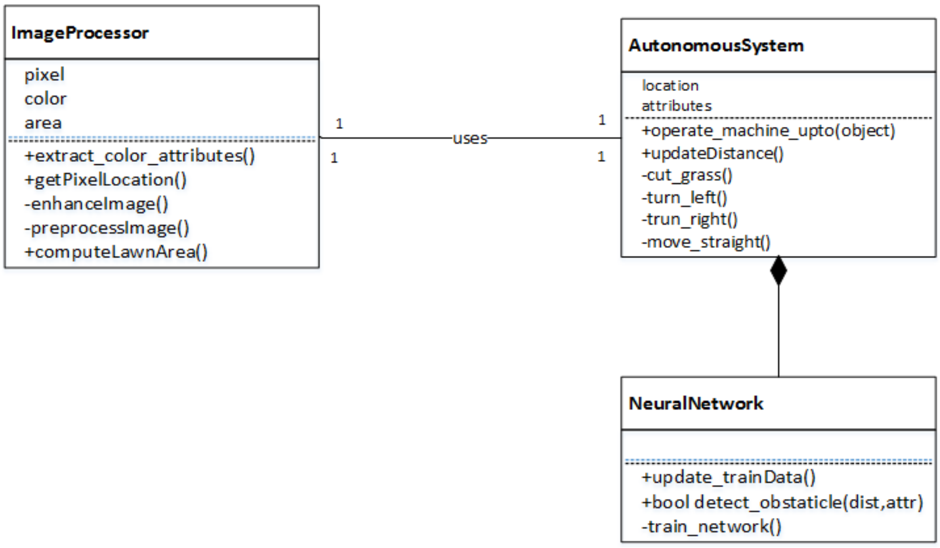


Fig.1. Class diagram of the model

The UML sequence diagram is an interactive diagram depicting how processes operate with one another. It depicts the flow of messages, sequence of events, and actions between objects, and shows time successions that are not effectively delineated in other diagrams. It analyses object interactions which are arranged in time sequence and how they exchange information from one stage to another. The sequence diagram for the obstacle avoidance model is shown in Fig 2.

The model has three (3) participating objects: AutonomousSystem, Imageprocessor and the Neural networks. The *ImageProcessor* object carries out preprocessing operations on the image data acquired using the digital camera. On the activation of the object's lifeline, the *AutonomousSystem* object initiates calls to the ImageProcessor object to return the area of lawn to cut as well as the area's colour attributes. The *NeuralNetwork* object is used to determine the presence or otherwise of obstacles. The obstacles avoidance model (*AutonomousSystem)* checks for the lawn area to cut. If the area currently covered is less than the total lawn area, the model passes the current block of pixels information to the NeuralNetwork object to determine whether or not that portion has an obstacle. If there is an obstacle, the model initiates the appropriate method (turn_left(), turn_right()) to avoid that obstacle, otherwise, the model moves in a straight path until either another obstacle is detected or the entire lawn area has been covered.

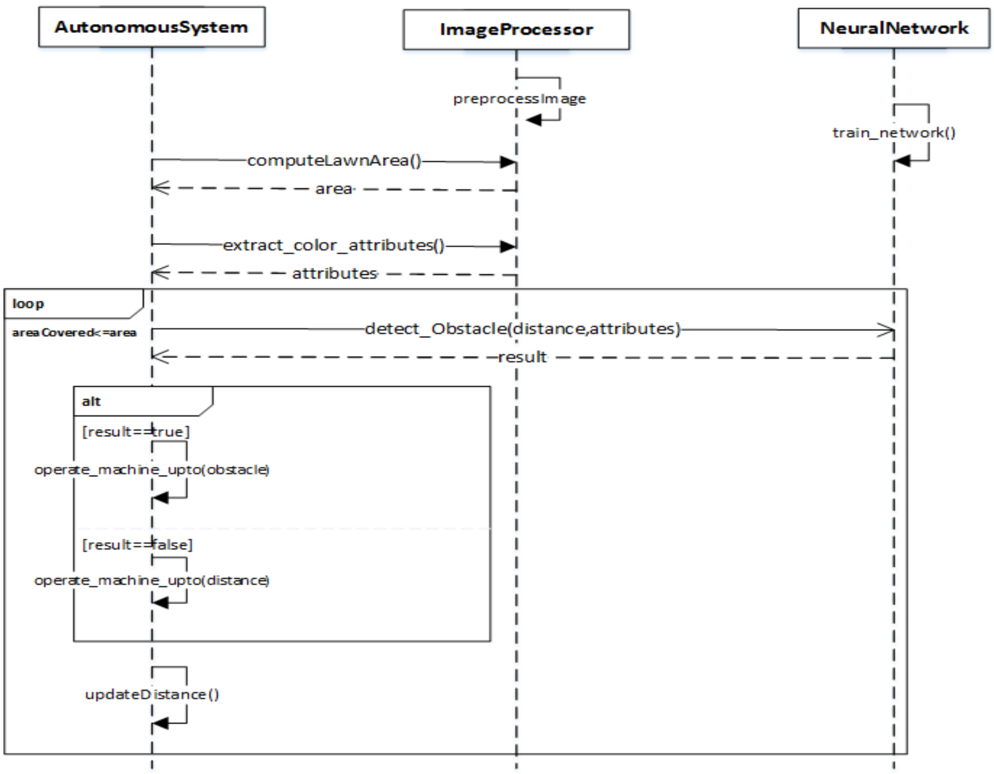


Fig.2. Sequence Diagram of the model

3.4. Obstacle Avoidance Algorithm

Obstacle avoidance in a static or real-time environment can be extremely challenging, hence it is important that there is a delay between receiving sensor data and acting on it. Obstacle avoidance algorithm for the present work is achieved by using a collision-free navigation algorithm (a combined global and local avoidance) that manages both recognized and unrecognized obstacles in the lawn's coverage area. To ensure that all obstacles may be safely avoided and other block cut, the avoidance algorithm steer the autonomous system around the obstruction so that it may swiftly retrace its steps back to the predetermined course in finite time under the following assumptions that:

1) The workspace is bounded, curved, and partitioned into equal sized blocks (x, y, t). The Cartesian coordinates x and y are linked to each pixel block in the form of f(x, y, t).
2) At any given time, $t\varepsilon N_0$, where t stands for a time unit and N_0 is the set of natural numbers, a block of pixels can exist in one of two states. The block can either be empty (f(x, y, t) = 0) or occupied by an obstacle (f(x, y, t) = 1).
3) In addition, an obstacle can only occupy a block once; otherwise, a collision would result. In automata, these rules are known as transition rules and, together with the dimensions of the workspace, the states of the cells, and the neighbourhood of each block, they define a finite state automaton.

The user interface automaton connects with the global navigation automaton, the abstraction with the highest level of abstraction, which receives target waypoints and constructs routes or asking for additional guidance when a path is locked. furthermore, the global navigation system keeps track of the mission while computing direction and position errors. The status repository now includes this data, which the local navigation then uses. The local navigation subsystem then chooses the appropriate step and navigates the path accurately. It satisfies the global navigation subsystem's request for a route by creating a path for the robot that avoids collisions using data from the proximity camera sensors. The importance of the efficiency with response to time is underlined in this study. In addition, as a computer device has a finite resource, it is important to take this into account when evaluating how efficient is the algorithm. The efficiency of the algorithm employed in the present work estimates the quantity of computational resources by taking into account the time and memory requirements for each task.

Time complexity is a term used to describe an algorithm's effectiveness. It establishes a relationship between the number of steps and the input length (or problem size). The memory needs of an algorithm are known as its space complexity. Theoretical estimations of the number of algorithmic resources needed to solve a particular computer problem are provided by algorithm analysis, a key component of the larger computational complexity theory. These estimates serve as the standards for evaluating the algorithms' effectiveness [15]. Asymptotic analysis is used to measure complexity in theoretical algorithm analysis. The big-oh notations (O) show the complexity of the algorithm. Calculating the efficiency of an algorithm can be extremely challenging. This is due to the fact that how an algorithm is implemented in actual computer codes has a significant impact on how complex the algorithm is. Also, the scenarios to which the method is used frequently affect the overall number of computations. Assumptions about how an algorithm is implemented, such as the operations employed and their computing cost to the algorithm, are typically necessary for this computation [16]. while keeping a safe distance from the goal position/obstacle. The big-oh notations define the execution time required to perform the obstacle avoidance on the set of data in a linear form. This implies that faster updates to the solution are produced by algorithms with lower computing complexity.

4. Methodology

Toward accomplishing the objectives of the study on the development of an improved obstacle avoidance model for a lawnmower, digital image sample data about obstacles was collected from an outdoor environment using a digital camera (resolution of 640*840) to reduce the noise contained in the image data set. The collected sample image data was a 2D image function f (x, y), where x and y are spatial coordinates and f is the amplitude of any pair of coordinates (x, y). The obstacle in the sample image was reduced to one-third (1/3) of the original size to avoid unnecessary delay in processing time. Similarly, the image data collected and reduced was pre-processed to enhance the quality of the image to make some features easier to see by adjusting the colours. In the study, the intensity or colour adjustment was carried out.

During the experiment, FSA was used to model the two important components in the model. First, the colour image to binary was carried out using Deterministic Finite Automata (DFA) with the following assumptions that the last pixel of the image has the blue colour and the object is stationary. To make effective use of the theory of formal language and automata theory for specification, the DFA model for the low-level description of the two major components of the system was formulated using FSA. Here, FSA is defined as the collection of a finite number of states having one input state, probably an intermediate state and some final output state from which input strings are formed and a finite set of transitions as expressed in (1).

In this study, the colour of interest representing input alphabet are white (W), red (R), brown (S), black (BK), green (G) and blue (B). qo represents the initial/start state; $F \in$ Q is the set of accept states; and ∂ is a mapping function ∂:

$$Q * \Sigma \rightarrow Q$$

$$Q = \{q0, q1, q2, q3\}$$

$$\Sigma = \{R, W, S, BL, BK, G\}$$

$$qo = \{q0\}$$

$$F = \{q3\}$$

$$\partial : Q * \Sigma \rightarrow Q$$

Then, the instantaneous mapping is shown as follows and the table depicted.

$$\partial \text{ (q0, R/W/S/BK)} \rightarrow \text{q1;}$$

$$\partial \text{ (q1, R/W/S/BK)} \rightarrow \text{q1;}$$

$$\partial \text{ (q1, G)} \rightarrow \text{q2}$$

$$\partial \text{ (q1, R/W/S/BK)} \rightarrow \text{q2;}$$

$$\partial \text{ (q2, G)} \rightarrow \text{q2;}$$

$$\partial \text{ (q2, BL)} \rightarrow \text{q3}$$

$$\partial \text{ (q1, BL)} \rightarrow \text{q3}$$

Table 1. State Transition Table for Converting Colour Image to Binary

States	R/W/S/BK	G	B L	States
q1	q1	—	—	q0
q1	q1	q2	q3	q1
q2	q1	q2	q3	q2

Table 1 shows the transition table for the instantaneous mapping of the states. Fig. 3 shows the transition diagram for converting a colour image to binary Each labelled circle represents a state. This machine has four (4) states. The arrow indicates that the machine starts in state qo and transits to an intermediate state before it gets to its final state q3. The purpose of converting a colour image to binary is to simplify the image processing operations.

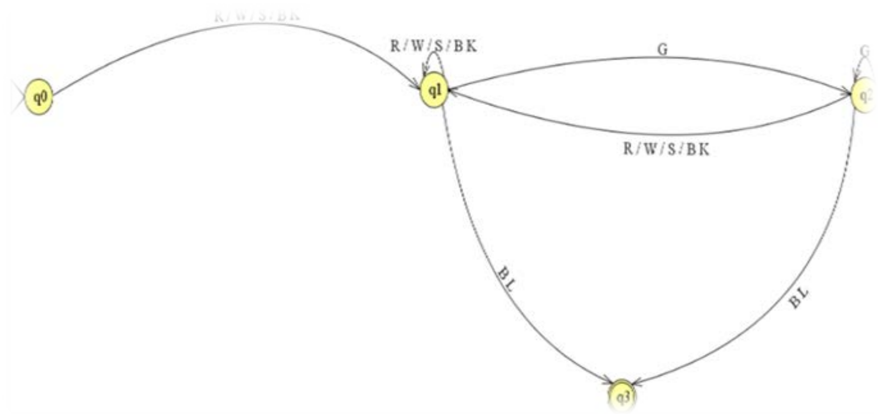


Fig.3. State diagram for colour image conversion to binary

Second, the authors determined the block of pixels to be safe or cut. In the process of determining the block of pixels to save (those containing obstacles) or cut (those without obstacles), a non-deterministic finite automaton (NFA) was used. In contrast to a Deterministic Finite Automata (DFA), where there is a unique next state for a transition from a state on an input symbol, here a finite automaton with nondeterministic transitions is considered in (2). A transition is non-deterministic if there are several (possibly zero) next states from a state on an input symbol or without any input.

Formally, a non-deterministic finite automaton (NFA) is a quintuple defined in (2) as

$$A = < Q, \Sigma, qo, F, \partial > \quad (2)$$

where;
Qis a finite set of states
Σ is a finite set of the input alphabet
q0 is called the start state
F ε Q is the set of accepted state
∂is a mapping function ∂: $Q * \Sigma \rightarrow 2^{Q}$

With assumptions that the machine scans eight (8) pixels (representing a block) at a time, the pixel contains a 0 or1 and the object is stationary, we have the followings:

$$Q = \{q0, q1, q2, q3, q4, q5, q6, q7\}$$

$$\Sigma = \{R, W, S, BK, G\}$$

where; R(red) / W(white) / S(brown) / BK (black) =1
and G =0

$$qo = \{ qo \}$$

$$F = \{q3, q4, q5, q7\}$$

$$\partial: Q * \sum \rightarrow 2^{Q}$$

The instantaneous mapping is shown

$$\partial\ (qo, 10101011) \rightarrow q3$$

$$\partial\ (qo, 10001000) \rightarrow q$$

$$\partial\ (qo, 00010000) \rightarrow q7$$

$$\partial\ (qo, 00000000) \rightarrow q5$$

$$\partial\ (qo, 11111111) \rightarrow q3$$

q3 and q4 = save; q5 and q7=cut

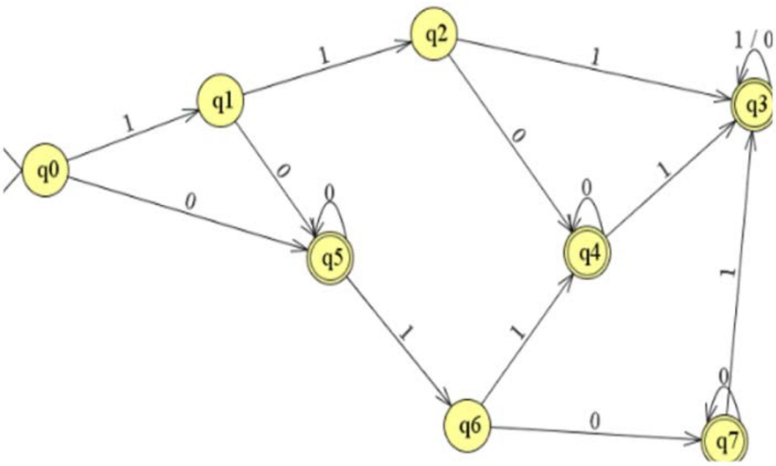


Fig.4. Screenshot Showing the Transition of States using JFLAP Simulation Tool

Fig.4 shows the transition diagram for determining the block of pixels to cut or save using finite automata theory. Each labelled circle represents a state {q0, q1, q2, q3, q4, q4, q5, q6, q7}. This machine has eight (8) states. The arrow indicates that the machine starts in state q0 reaching its final states (q3, q4, q5, and q7) where a decision is made by the model either to cut or save the block of pixels. In other words, when the machine is in states q3 and q4, the model saves the block of pixels but the moment the state transits to either state q5 or state q7, the block of pixels will be cut. States with double circles are the accepting states. If the number of ones in the input block of pixels is greater than two, the machine goes to the final or accept states q3 or q4 (block is saved), while in other accepting states (q5 or q7) where the number of ones in a block is less than three, the block is cut. State q0 is the initial state at which the system is expecting a block of pixels as input. Each of the arrows from states q3, q4, q5 and q7 loops right back into their respective states, meaning that once the machine is in this state (for instance state q3), it stays there provided it gets 1/0 as input. Table 2 show the mapping of input to the appropriate accept state.

Table 2. Examples of Input States mapped to Accept States

Initial state (q0)	Input	Final state (F)
q0	10101011	q3
q0	10001000	q7
q0	00000000	q5
q0	00010000	q7
q0	11111111	q3

The neural network (NN) was used in recognizing obstacles in images. NN consists of a set of interconnected input neurons and output units, where each input neuron has a weight (w) associated with it. A part of input data is fed concurrently into a layer of processing units making up the input layer. The weighted outputs of the processing units are, in succession given to the hidden layer. The resultant weighted outputs of the last hidden layer serve as input to units making up the output layer which issues the prediction of the network for a given set of samples. During the learning stage, the network created learns by adjusting the weights to have the option to accurately say what will happen to the output target of a given set of input based on present indications. Feed-forward perceptron being employed as a widely used type of neural network is chosen to be actualized in this experiment to deal with the outcomes obtained from the automata-based hypothesis. In the same context, the feed-forward network allows connection to traverse in a unidirectional way throughout the neural network module. The neural net uses a single layer log-sigmoid feedforward which comes after by an output layer of linear neuron (‘purelin') to identify and output value corresponding to the target (-1 to cut and 1 to safe). The designed network is presented with a set of input vectors producing an output of similar or equivalent target vectors.

Colour image data that was collected were first converted to binary forms, while the resulting binary images were then processed in blocks (8 pixels per block) using the procedure discussed in Section 3. The sum or number of ones that can appear in a block of eight pixels are 0, 1, 2, 3, 4, 5, 6, 7, 8. Zero (0) means all the pixels are turned off, that is, no one 1is occurring and eight (8) means all the pixels are on (1 in all pixels). After a close evaluation of roughly fifty per cent of the processed images used, it was observed that for an obstacle to be present in a block, not less than three pixels of the block must have a 1. If the number of 1 (ones) within a block is less than 3, that block is cut (this is represented as -1), otherwise (number of 1 greater or equal to 3), the block is saved (this is represented as 1). Hence, the two states; safe and cut are represented in this study as 1 and -1, respectively, as shown in Table 4 To define a fitting problem for the proposed model to avoid obstacles in MATLAB, a set of input data is arranged as a vector and another set of target data is arranged into a second vector. For example, the fitting problem for the proposed model with a dimensional element input vector and a dimensional element target is as follows:

```
input = [0 1 2 3 4 5 6 7 8]; # input vector, number of ones in a block of pixels
target = [-1-1-1 1 1 1 1 1 1]; # target vector, -1 (cut), 1(safe)
```

Table 3. Network Inputs and Outputs

Input	Neural network output	Meaning
0	-1	Cut block
1	-1	Cut block
2	-1	Cut block
3	1	Safe block
4	1	Safe block
5	1	Safe block

These vectors are duplicated arbitrarily to give more sample data to enhance network performance as shown in Table 3.

```
d1= [ 0 0   1   1   2   2   3   3   3   4   4   4   5   5   5   5]
                    6    6   6   7   7   8   8 8]
```

d2= [-1 -1 -1 -1 -1 -1 -1 -1 -1 1 1 1 1 1 1]
1 1 1 1 1 1 1 1 1]

5. Implementation

In this study, a Rapid application development (RAD) approach was utilized to deliver the improved model of a land mower (required outcomes). This is because the approach affords the model design to be modularized, developed with higher quality within a short time frame aligned with quick feedback from users and user's requirement. Moreover, MATLAB gives a great RAD condition that helps numerical computation, simulation, evaluation and visualization in less time**.** The Matrix Lab tool was used for image processing, creating intelligent systems, data analysis, developing algorithms. In addition, the tool provides development environment that aids the design of complex systems.

5.1. Determining Obstacles Colours

The colour range (minimum to maximum) of obstacles was determined using the Image Toolbox. The *imtool* function in MATLAB was used to display the true colour of the image. The Image Tool indicates data in connection with the area and estimation of respective elements in a picture in the bottom left-hand side corner of the toolbox. The pixel worth and location information represent the pixel under the current location of the pointer.

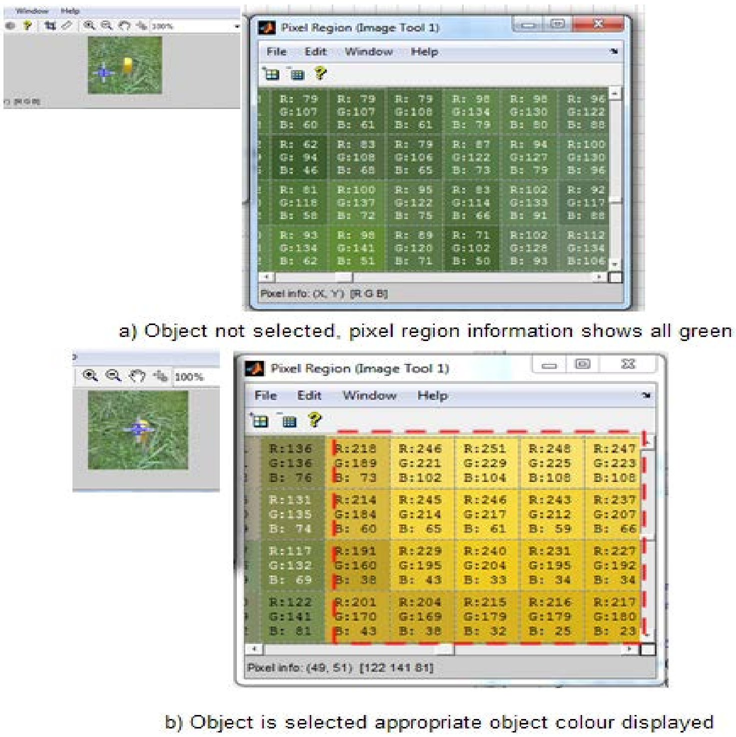


Fig.5. Image Tool Window Displaying Obstacles' Colour Values

Table 4. Colour range used to define Obstacles in Sample Data

Lawn Images	Colour range defining obstacles from minimum to maximum		
	Red	Green	Blue
1	183-255	161-255	136 -255
2	71-91	83-106	71- 96
3	126-158	112-171	134 - 143
4	182-205	163-182	114-167
5	167-231	130-188	133-187
6	201-255	159-241	152-255
7	163-243	163-226	176-243
8	136-144	101-141	100-154
9	187-255	161-255	165-252
10	179-255	149-245	13-146

The Image Tool revises this information as one hover the pointer over the image. Fig.5 illustrates how the information is displayed. Fig. 5(a) shows values (red, green and blue segments of colour) of pixels. The green constituents dominate the other two components (red and blue), given the reason why the colour at the given locations was green. This portion of the image represents part of the lawn that will be cut by the autonomous robot (proposed model) during the mowing operation. Fig. 5(b) also shows the values (red, green and blue components) of the pixels when the pointer is partially on an obstacle (brownish colour). This portion contains the obstacle which should be avoided during mowing. Table 4 depicts the colour range (minimum to maximum for colour components representing obstacles) of ten (twenty percent) lawn images randomly selected from the sample image data. Only this fraction of data) was used to capture information of all the sample images used in this study (see Table 4); thus, used in the proposed model to represent pixels information defining obstacles of interest.

Images were first read into the MATLAB environment through the use of imread function. Then a scale transformation (*imresize* function) was applied to the original image to reduce its size. Intensity adjustment operation (contrast adjustment) was applied using *imadjust* function on images to improve the results of image analysis techniques. Fig. 6 depicts the block diagram of the proposed model. The following sub-sections describe the two important modules of the model.

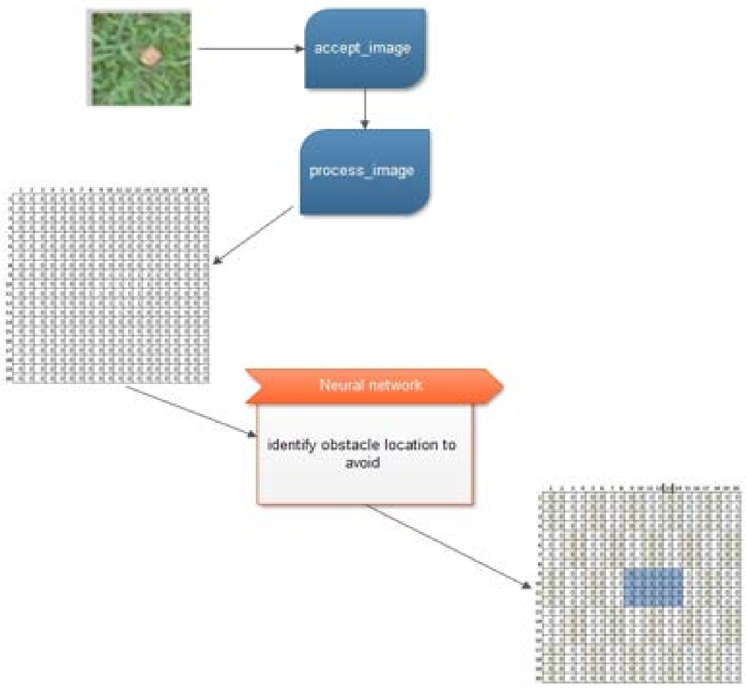


Fig.6. Schematic Diagram of the Proposed Model

5.2. Conversion of Colour Image to Binary

The procedure for converting a colour image to binary is a key component of the proposed model as shown in Algorithm 1. A detailed description was given in this sub-section. The procedure accepts a 3D colour image and returns a 2D binary equivalent (*temp*) of the image. The colour image is converted to binary using information depicted in Table 4. The pixels of the image are scanned from the beginning to the ending. If a given pixel's colour (red, green and blue) of an image match those in Table 4 and glow, the corresponding pixel location in the target image (t*emp*) is set to one (1) otherwise it is set to zero (0). The algorithm *convert2binary* summarises this operation. Fig. 7 depicts an example of a colour image (Fig. 6 in binary, Fig. 7 with the matrix representation and Fig. 8 of the binary form.

(a) Colour image (b) Binary image (c) Matrix form of binary image

Fig.7. Colour Image to Binary

Algorithm 1: Convert2binary (int, image)

```
szi=size (image)
        for m=1 to szi(1) // rows
        for n=1 to szi(2) // columns
        if (image(m,n,1)>182  & image (m,n,1)<256) & (image(m,n,2)>160 &image (m,n,2)<255) &(image(m,n,3)>
135 & image (m,n,3)< 256)
        temp (m,n)=1;
else if (image(m,n,170 & image (m,n,1)<92) & (image(m,n,2)>82 & image (m,n,2)<107) &(image(m,n,3)>70 & image
(m,n,3)<97)
        temp(m, n)=1;
: // other alternatives
:
else
        temp( m, n)=0;
end
end
end
return temp
```

5.3. Identifying Obstacles in Images

This procedure for identifying obstacles in images accepts as input the binary form (zeros and ones) of a colour image produced using the algorithm described in Algorithm 2. It processes the binary image in blocks (usually 8 pixels per block as default) by counting the number of ones in a block and passing the result to the neural network module to determine whether or not there is an obstacle in the block. The neural network returns 1 if the number of ones is greater than two (2) and -1 if the number of ones is less than three. The coordinates of those blocks in which the neural network returns a 1(one) are then saved while other blocks are marked for cutting by the autonomous agent. This procedure is described in Algorithm 2 *identify_obstacle.* Fig. 8 depicts this process with a 20 by 20 dimension of a binary form of the colour image, where blocks of pixels in colours white and light grey (to simplify visualization) represent the area to be cut while blocks in blue represent the location of the obstacles which must be avoided by the proposed model.

Algorithm 2: Identify_obstacle (int, temp)

```
p_i=0
for m=1 to row step 2
for n=1 to column step 4 // process 8 pixels as default
        ap=j
        s=network(image(ap,m)+image(ap+1,m)+image(ap+2,m)+image(ap+3,m)+image (ap,m+1)+...
        image (ap+1,m+1)+image (ap+2,m+1)+image (ap+3,m+1))
                if (s>2)
    // possible location of obstacle, save coordinates of block
    //(ap,m), (ap+1,m), (ap+2,m), (ap+3,m),  (ap,,+1),(ap+1,m+1),(ap+2,m+1),(ap+3,m+1)
         p=p_m+1; //increment block of pixels to be saved
else // s== -1, number of 1s less than three
         // block of pixels to cut
         q=q+1;     // increment counter of blocks to be cut
                end
        end
end
```

5.4. Implementation Details

The proposed model was successfully implemented using MATLAB. The program was run on a machine with the following configurations: 6GB RAM, Intel Core (TM) i5-2450M at clock speed 2.5GHz, 64-bit Operating system and 100GB Hard disk. The model was configured to operate in three modes (slow, normal and fast) depending on prior knowledge about the environment it operated on. Normal is the default mode where eight pixels per block are processed at a time. Slow mode (4 pixels per block) is used for terrain with many obstacles. Fast mode (16 pixels per block) is suitable for an environment with very small or no obstacles. In contrast to existing models that randomly search individual pixels for obstacles, the proposed model moves on a straight path and operates on a block of pixels. Sample image data collected through a digital camera were first preprocessed (image enhancement) to enhance results. The original image data could be scaled to several dimensions- 80*80, 100*100, or 120*120 to simulate the cutting of different sizes of lawns. The image was converted to a binary form which was then fed to the neural network module. The neural system utilized the number of 1s in a square of pixels to determine the presence of a snag in a square block. The portion of the image having at least two 1s per block is identified and avoided during mowing operation.

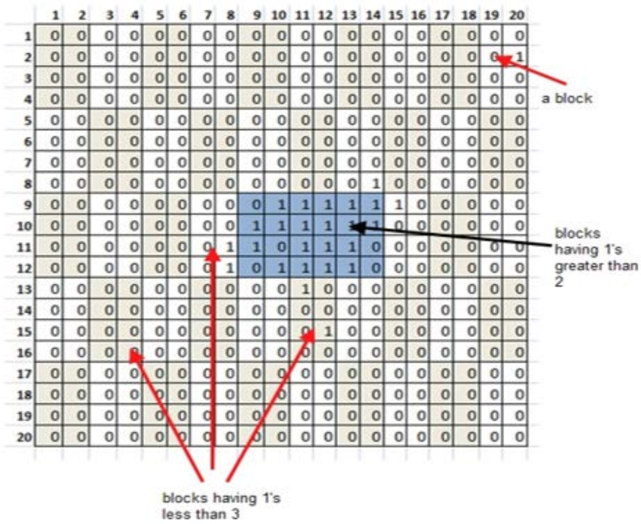


Fig.8. Binary form of a colour image

5.5. Model Evaluation

After modeling and analysis of the lawn mower operations, the simulation model and its performance are assessed to confirm the accuracy, efficiency and validity of the proposed modelled system. The measure of performance of the model is necessary in the estimation procedure as well as in testing models. To achieve this, two evaluation metrics were used. The first is the False Acceptance Rate (FAR) and the second is the False Rejection Rate (FRR). The FAR gives the rate at which the system recognizes obstacles that are not present. This is described mathematically using the expression given in (3). The False Rejection Rate (FRR) measures the rate at which the system fails to recognize obstacles that are present. The mathematical notation expressing this is given in (4).

$$FAR = \frac{t-p}{p} \tag{3}$$

where;
t = obstacles not present but recognized as present
p = total number of obstacles

$$FRR = \frac{p-h}{p} \tag{4}$$

where;
p = total number of obstacles
h = valid number of obstacles recognized

A total of twenty images (containing obstacles ranging from 1 to 10) from sample data were randomly selected. The total number of obstacles in the randomly selected images was manually counted and recorded as p=40. Each of the randomly selected images was then given as input to the model to process. The purpose of this is to determine how efficient is the model in correctly recognizing obstacles in those images. The number of obstacles (both present and not present) recognized by the model was noted and recorded as t = 43. The number of valid obstacles presents and correctly recognized by the model was also noted and recorded as h=38. The values for p, t and h were then substituted in (3) and (4) respectively to give a measure of the model's accuracy.

Given the total number of obstacles in the randomly selected sample as forty (40), then

$$FAR = \frac{t-p}{p} = \frac{43-40}{40} = 0.075$$

Similarly,

$$FRR = \frac{p-h}{p} = \frac{40-38}{40} = 0.05$$

6. Results and Discussion

Fig. 9 demonstrates two instances of the proposed step-by-step procedure for discovering obstacles that impede smooth movement. The images on the left-hand side represent the completely new colour input images, the dominant images which are equ-distant from other images are the binary equivalents of pictures on the left, while images on the right-hand side illustrate the corresponding model's output images of the original images indicating obstacles together with their locations.

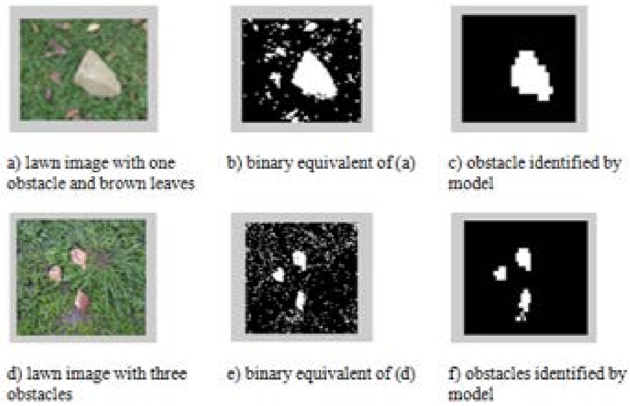
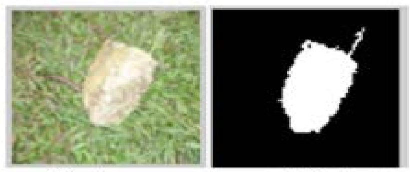


Fig.9. Proposed model's output

Apart from showing the obstacle's location, other information produced by the model is depicted in Fig. 9. Additional information on images such as name, image dimension, number of blocks to cut in the picture, number of blocks to safe in the image, and the estimated time (in seconds) to process the lawn images are also available. Fig. 10. illustrates four other examples with original colour input images on the left, the location of obstacles in the middle and the relevant information on images on the right. For the first three images in Fig. 10, the model was able to accurately detect the number of obstacles present in each but failed to detect all the obstacles in the last image. This may be attributed to few data (20% of the sample) used to represent the colour range defining obstacles in this study.



(a) Image (b) Output

```
Identifying obstacle(s) in ramdomly picked lawn image 'dat3.jpg'
'dat3.jpg' dimension is 80 by 80
Blocks to be saved 174
Blocks to cut 626
Total blocks 800
Total elapsed time 3.440734e+00
```

(c) Additional information

Fig.10. Proposed model's output with additional information

The FAR and FRR values (0.075 and 0.05 respectively) obtained for the proposed model gave a high measure of the reliability of the model in detecting an obstacle. The result presented in the present study is in consonance with the findings of [33,31] who in separate studies reported that the false positives rate (FAR) was not sufficiently large 0.46 and 0.57 respectively bringing about a failure of the land mower robot to fail in identifying obstacles. Similarly, the model was also benchmarked (for processing time) with an existing model [4], which randomly checks obstacles along its path while in operation. Different lawn image dimensions were given to both present work and Yashiro’s work and the time required to process these images was recorded. The existing model on the average case will visit half of the total pixels that make up the image while the proposed model operates on a block of pixels at a time. Fig. 11. depicts processing times of the two models at different image dimensions (sizes). The time taken to avoid obstacles in the proposed model (i.e., about 0.34s) was faster when compared with the findings of [9, 31] which yielded 10s and 0.97s Also, the time taken to avoid obstacles in the proposed model was faster than compared with an existing model. The implication of the study is to develop a model which will reduce the computation time expended in locating obstacles in lawns. Based on the accuracy and the reliability obtained from the model developed, it can seamlessly be incorporated into lawn mowers for detecting obstacles within its coverage area

The findings revealed by this present work is in contrast with the findings reported in the works of [5] who implemented the edge detection robot using MATLAB tool as employed in the present study. However, the computational time involved in processing the obstacle detection in the study increases to several minutes which is not useful in real time application. The result obtained from the present study is in consonance with the findings of [28] who also designed and implemented an autonomous lawn mower controller using FSA-based technique. However, the author employed other methods of potential and global positioning system. In the same context, [29] also based his work on one of the two assumptions given. Nevertheless, [4] implemented the edge detection robot using MATLAB tool as employed in the present study. However, the computational time involved in processing the obstacle detection. Moreover, the present study employed the digital image sample data about obstacles collected from an outdoor environment using a digital camera (resolution of 640*840) as was used in previous study of [33].

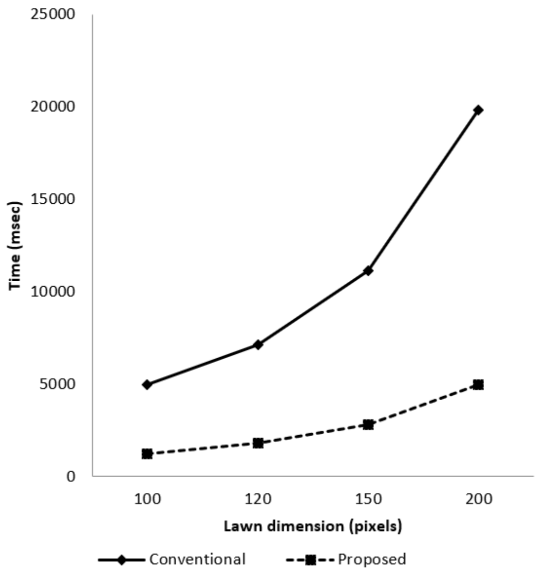


Fig.11. Time taken to avoid obstacle

7. Conclusions

This present work modelled and simulated a lawn mower system using finite state automata theory to understand the behavioural characteristics of the autonomous system. The model was designed using the unified modelling language (UML) and formal method of finite state automation. The model's intelligence was created using image processing techniques and neural networks. Using MATLAB tool based on the Algorithm created to convert colour images to binary equivalent. The model processes the binary image in blocks (usually 8 pixels per block as default) by counting the number of ones in a block and passing the result to the neural network module to determine whether or not there is an obstacle in the block. The model was implemented and simulated in a MATLAB environment. The model was also evaluated in terms of performance time (FAR and FRR). Improving the time taken in the detection that causes a collision and, in turn, impedance to motion has been a major priority in automation and robotics that seeks to employ technology in detecting and identifying obstacles during the process of moving. Hence an improved model that aims at ensuring a timely and smooth cutting of grass has been introduced. Based on the accuracy and the reliability obtained from the model developed, it can seamlessly be incorporated into lawn mowers for detecting obstacles within its coverage area.

The result of the performance evaluation of the proposed model has shown that the model will run conveniently with small computation time. This study will interest the research community and should be recommended for all users who are conscious of keeping their environment beautiful. To make the *obstacle avoidance model* accessible to people, full implementation of this model for the autonomous system should be considered. It is also recommended that the program code developed in this study should be made available to students. More so, it can be used as a teaching aid for students that are interested in computer vision and image processing. Future work for the obstacle avoidance model would be aimed at taking an in-depth study of the various algorithms and methods which have been used for avoiding obstacles, the issues limiting their performance and optimization would also be considered. The present step-by-step procedures could be improved from numerous points of view. It will be good to examine whether the colour schemes apart from RGB perform for this application. Samples of good encouraging colour spaces are luma component (Y′) and two chrominance (UV), standardized RGB, and opposing colours.

References

[1] B. Okafor, "Simple Design of self-powered Lawn Mower". *International Journal of Engineering and Technology*, vol. *3*, no.10, pp. 933-938, 2013.
[2] D. P. Watson, and D.H. Scheidt, "Autonomous Systems" *Johns Hopkins APL Technical Digest*, vol.26, no.4, pp.368-376, 2005.
[3] R. Simmons, J. L. Fernandez, R. Goodwin, S. Koenig and J. O'sullivan, "Lessons learned from Xavier", *IEEE Robotics & Automation Magazine*, vol. *7, no.* 2, pp.33-39, 2017.

[4] Y. Yashiro, K. Eguchi, S. Iwasaki, Y. Yamauchi, and M. Nakata, "Development of Obstacle Avoidance Control for Robotic Products Using Potential Method", *Mitsubishi Heavy Industries Technical Review*, vol. *51, no.*1, pp. 34-39, 2014.

[5] N. Allen, A. Tabassum, W. Semke, W and J. Neubert,"Repeatability of edge detectors in various environmental conditions" In *2017 International Conference on Unmanned Aircraft Systems (ICUAS),* IEEE, pp. 912-919, 2017..

[6] N. Nourani-Vatani, M. Bosse, J. Roberts and M. Dunbabin, "Practical Path Planning and Obstacle Avoidance for Autonomous Mowing" In *Proceedings of the Australasian Conference on Robotics and Automation* pp. 1-9, 2006.

[7] A. Dipin, and T. K. Chandrashekhar, "Solar Powered Vision Based Robotic Lawn Mower", *International Journal of Engineering Research and Reviews*, vol.*2, no.*2, pp.53-56., 2014.

[8] K. W. Gray and K. Baker, "*Obstacle Detection and Avoidance for an Autonomous Farm Tractor*", (Unplished Master's Thesis). Utah State University, Department of Electrical and Computer Engineering, USA. 2000.

[9] K. A. Daltorio, A. D. Rolin., J. A. Beno, B. E. Hughes, A. Schepelmann, M. S. Branicky, R. D. Quinn and J. M. Green, "An obstacle-edging reflex for an autonomous lawnmower", In *IEEE/ION Position, Location and Navigation Symposium, IEEE,* pp. 1079-1092, 2010.

[10] P. E. Hart, N. J. Nilsson, and B. Raphael, " A Formal basis for the heuristic determination of minimum cost path", IEEE Trans. System. Sci. Cybern, Vol. 4, no. 2, pp. 100-110, 1968.

[11] J. Borenstein and Y. Koren, (1989). Real-time obstacle avoidance for fast mobile robots. *IEEE Transactions on systems, Man, and Cybernetics*, vol. *19, no.*5, pp.1179-1187, 1989.

[12] J. Borenstein and Y. Koren, Y, "Histogramic in-motion Mapping for Mobile Robot Obstacle Avoidance". *IEEE Transactions on Robotics and Automation*, vol. 7, no. 4, pp. 535-539, 1991.

[13] J. L. Crowley, "World Modeling and Position Estimation for a Mobile Robot using Ultrasonic Ranging", In *Proceedings, 1989 International Conference on Robotics and Automation* pp. 674-680, 1989. IEEE.

[14] R. Kuc and B. Barshan, "Navigating vehicles through an unstructured environment with sonar" In *Proceedings, 1989 International Conference on Robotics and Automation,* IEEE, pp. 1422-1426, 1989 R.

[15] Oroko J, Nyakoe G (2014) Obstacle avoidance and path planning schemes for autonomous navigation of a mobile robot: a review. In: Proceedings of Sustainable Research and Innovation Conference, vol 4, pp. 314–318

[16] B. C. Creed, A. Arsenault, A., S. A. Velinsky and T. A. Lasky, "Obstacle Avoidance for Autonomous Mowing", Mechanics *Based Design of Structures and Machines*, vol. *40, no.* 3, pp. 334-348, 2012.

[17] Bhavesh, V. A. (2015). Comparison of various obstacle avoidance algorithms. *Int. J. Eng. Res. Technol*, *4*, 629-632.

[18] K. W. Gray and K. Baker, "*Obstacle Detection and Avoidance for an Autonomous Farm Tractor*", (Unplished Master's Thesis). Utah State University, Department of Electrical and Computer Engineering, USA. 2000.

[19] L. E. Kavraki, M. N. Kolountzakis, and J. C. Latombe, "Analysis of Probabilistic Roadmaps for Path Planning", In *Proceedings of IEEE International Conference on Robotics and Automation*, Vol. 4, pp. 3020-3025, 1996.

[20] S. M. LaValle, and J. J. Kuffner, "Randomized Kinodynamic Planning" *The International Journal of Robotics Research*, vol. *20, no.*5, pp.378-400, 2001.

[21] A. Shkolnik, M. Levashov, I. R. Manchester, I. R., and R. Tedrake, (2011), "Bounding on Rough Terrain with the LittleDog robot" *The International Journal of Robotics Research*, vol. *30, no.*2, pp.192-215, 2011.

[22] V. Milanés, D. F. Llorca, J. Villagrá, J. C. Pérez, I. Fernández, C. Parra, Gonzalaz and M. A. Sotelo, M. A, "Intelligent Automatic Overtaking System using Vision for Vehicle Detection", *Expert Systems with Applications*, vol.*39, no.*3, pp.3362-3373, 2012.

[23] A. Cochrum, J. Corteo, J. Oppeland, and M. Seth, "An Autonomous Lawnmower", The ManScaper, 1-8, 2013.

[24] T. J. S. Sivarao, H. Anand, H. and F. Minhat, "Review of Automated Machines towards Devising A New Approach in Developing Semi-Automated Grass Cutter" *International Journal of Mechanical and Mechatronics Engineering IJMMEIJENS 2010.*

[25] W. T. Fitch and A. D. Friederici,, "Artificial Grammar Learning meets Formal Language Theory": An Overview. *Philosophical Transactions of the Royal Society B: Biological Sciences*, vol. 367, no. 1598, pp. 1933-1955, 2012.

[26] I. M. Chiswell, "A course in formal languages, automata and groups". Springer Science & Business Media, 2008.

[27] J. E. Hopcroft, "Introduction to Automata Theory*", Languages, and Computation,* India: Pearson Education. 1-537, 2008.

[28] M. Wasif, "Design and implementation of autonomous Lawn-Mower Robot Controller", In *2011 7th International Conference on Emerging Technologies*, IEEE, pp. 1-5, 2011.

[29] Cherubini, F. Spindler and F. Chaumette, "Autonomous Visual Navigation and Laser-Based moving Obstacle Avoidance", *IEEE Transactions on Intelligent Transportation Systems*, vol.*15, no.* 5, pp.2101-2110, 2014.

[30] F. Tobassum, S. Lopa, M. M. Terek, and B. J. Ferdosi, " Obstacle avoiding robot", Global Journal of Research in Engineering, vol. 17, no. 1, 2017.

[31] M. Franzius, M. Dunn, M. Einecke and M. Dirnberger "Embedded Robust Visual Obstacle Detection on Autonomous Lawn Mowers", In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops,* pp.44-52, 2017.

[32] Newstadt, G., Green, K., Anderson, D., Lang, M., Morton, Y., & McCollum, J. (2010). Miami RedBlade III: A GPS-aided autonomous lawnmower. *Positioning*, *8*(1), 25-34.

[33] Schepelmann, A., Snow, H. H., Hughes, B. E., Merat, F. L., Quinn, R. D., & Green, J. M. (2009, November). Vision-based obstacle detection and avoidance for the cwru cutter autonomous lawnmower. In *2009 IEEE International Conference on Technologies for Practical Robot Applications* (pp. 218-223). IEEE.

[34] N. Ahmad, N. bin Lokman and M. H. Abd Wahab, (2016, November). Autonomous Lawnmower using FPGA implementation. In *IOP Conference Series: Materials Science and Engineering,* Vol. 160, No. 1, p. 012112, IOP Publishing.

[35] S. Hrabar, (2011) "Reactive obstacle avoidance for rotorcraft uavs". In: 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 4967–4974, 2011.

[36] K. M Hasan, A. Al-Nahid, K. J. Reza, S. Khatun, M.R. Basar, "Sensor based autonomous color line follower robot with obstacle avoidance". In: 2013 IEEE Business Engineering and Industrial Applications Colloquium (BEIAC), pp. 598–603, 2013.

[37] M. Zohaib, M. Pasha, R. A. Riaz, N. Javaid, M. Ilahi, and R. D. Khan, "Control strategies for mobile robot with obstacle avoidance", J Basic Appl Sci Res vol. 3, no. 4, pp. 1027–1036, 2013.

[38] V. A. Akpan, and A. S. Eyefia, (2021). Autonomous Vehicle with Machine Vision and Integrated Sensor Suite Based on Internet-of-Things Technologies. *sensors*, *15*, 19.

[39] Yasin, J. N., Mohamed, S. A., Haghbayan, M. H., Heikkonen, J., Tenhunen, H., & Plosila, J. (2021). Low-cost ultrasonic based object detection and collision avoidance method for autonomous robots. *International Journal of Information Technology*, *13*(1), 97-107.
[40] A. Saraswathi, A. A. Abass, S. I. Mathi and A. Eswaran, “Obstacle detecting robot using arduino and sensor”, S., vol. 3, 2015.
[41] E. J. Rodríguez-Seda. and C. K Rico, “Cellular automata based decentralized cooperative collision avoidance control for multiple mobile robots”. In *2019 IEEE International Symposium on Measurement and Control in Robotics (ISMCR)*, pp. A3-3, 2019, IEEE.
[42] J. Yang, J. Yang, and W. Wang, W. (2011). “An Efficient Path Planning Method Based on State Automata Model for Mobile Robots”, In *Advances in Automation and Robotics, Vol. 2* pp. 181-188, 2011, Springer, Berlin, Heidelberg.
[43] R. Hussain, T. Zielinska, and R. Hexel, “Finite state automaton-based control system for walking machines”, International *Journal of Advanced Robotic Systems*, vol. *16, no.* 3, 2019, 1729881419853182.

Authors’ Profiles

Samuel Mayowa, ALADE has received his Ph. D. degree in Intelligent Systems Engineering and Master of Computer Science and Technology from the Department of Computer Science and Engineering at Obafemi Awolowo University (OAU), Ile-Ife, Nigeria. He received his B.Tech. Computer Science from Engineering Faculty at Ladoke Akintola University of Technology, Ogbomoso, Nigeria. He is currently a lecturer in the Department of Computer Science at Nnamdi Azikiwe University, Awka, Nigeria. His research interest includes: Intelligent Systems, Pattern Recognition and Artificial Intelligence, Speech and Language processing with focus on Language teaching and Learning system, Object Detection, Object Classification, Machine Learning, Deep Learning, AI, Image Processing and Analysis, Information Systems Digital Multimedia, computational modelling and Narratives. He is a member of Nigerian Computer Society (MNCS), International Association of Engineers (IAENG) and Association of Computing machinery.

Adebayo Segun, **AFONRINWO** has received the Master degree Master of Computer Science and Technology from the Department of Computer Science and Engineering at Obafemi Awolowo University (OAU), Ile-Ife, Nigeria. He also received his B.Tech. Computer Science from Engineering Faculty at Ladoke Akintola University of Technology, Ogbomoso. His current works as a Senior Education Officer at Lagos State Ministry of Education.

How to cite this paper: Samuel M. Alade, Adebayo S. Afonrinwo, "Design and Implementation of an Improved Obstacle Avoidance Model for Land Mower", International Journal of Information Technology and Computer Science(IJITCS), Vol.14, No.5, pp.26-43, 2022. DOI:10.5815/ijitcs.2022.05.03