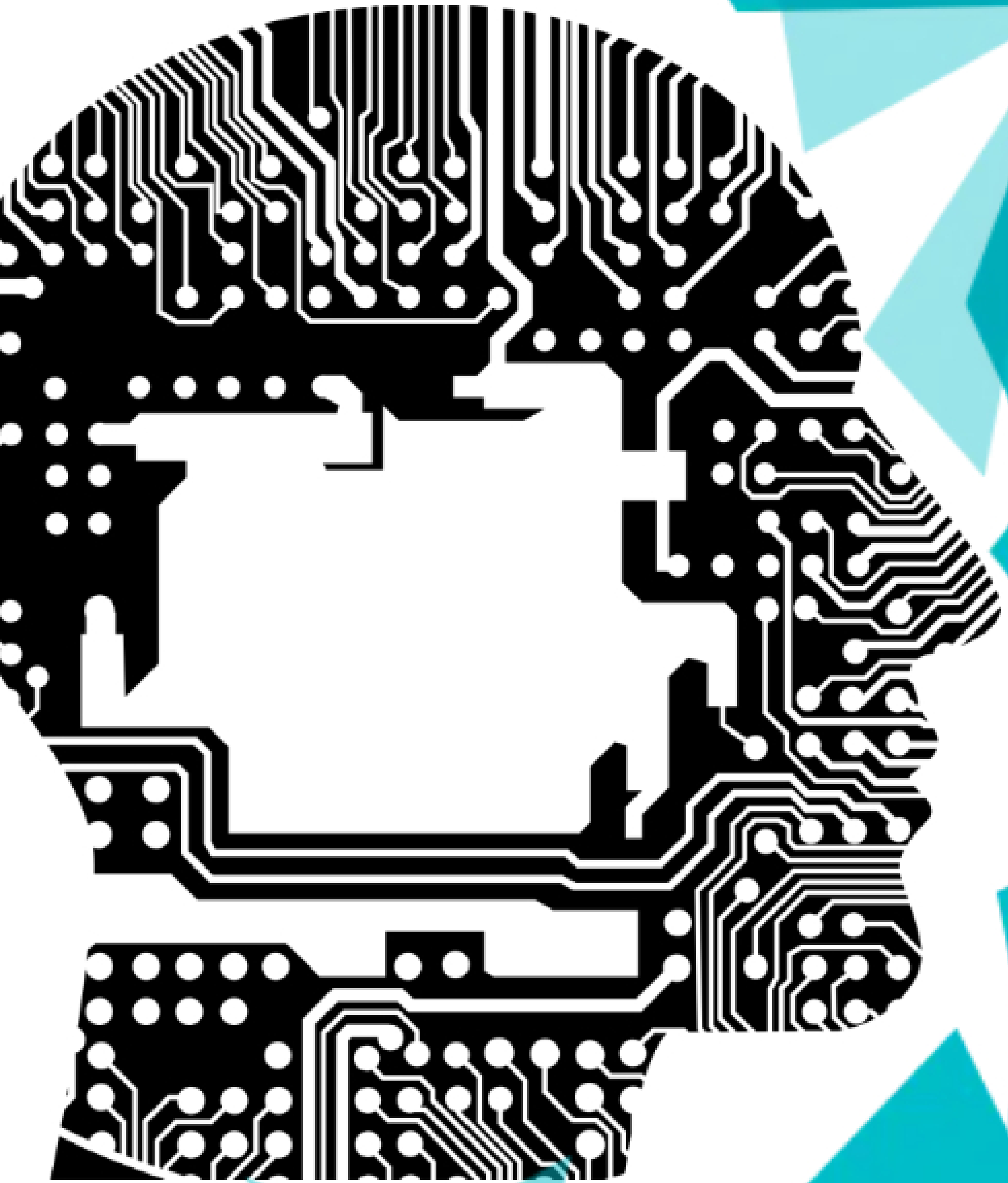
IMIENS



Intelligent Methods In Engineering Sciences

Vol 1 (1)

2022

International Journal
Intelligent Methods In Engineering Sciences
IMIENS
OPEN ACCESS
ISSN : 2979-9236

Vol. 1 No. 1 (2022)

Intelligent Methods in Engineering Sciences 1(1): 001-007, 2022

IMIENS

INTELLIGENT METHODS
IN ENGINEERING SCIENCES

https://www.imiens.org

International Open Access

December, 2022

e-ISSN 2979-9236

Research Article

https://doi.org/10.58190/imiens.2022.1

Complex Support System for Visually Impaired Individuals

Yavuz Selim TASPINAR [a,*], ***Murat SELEK*** [b]

[a] *Selcuk University, Doganhisar Vocational School, Konya, Turkiye*
[b] *Konya Technical University, Vocational School of Technical Sciences, Konya, Turkiye*

ARTICLE INFO

Article history:
Received 22 August 2022
Accepted 15 September 2022

Keywords:
deep learning
object detection
support system
visually impaired

ABSTRACT

It is very difficult for visually impaired individuals to avoid obstacles, to notice or recognize obstacles in distance, to notice and follow the special paths made for them. They continue their lives by touching these situations or finding solutions with the help of a walking stick in their hands. Due to these safety problems, it is difficult for visually impaired individuals to move freely and these situations affect individuals negatively in terms of social and health. In order to find solutions to these problems, a support system has been proposed for visually impaired individuals. The vision support system includes an embedded system with a camera with an audio warning system so that the visually impaired individual can identify the objects in front of him, and a circuit with an ultrasonic sensor so that he can detect the obstacles in front of him early and take precautions. The object recognition system is realized with convolutional neural networks. The Faster R-CNN model was used and in addition to this, a model that we created, which can recognize 25 kinds of products, was used. With the help of the dataset we created and the network trained with this dataset, the visually impaired individual will be able to identify some market products. In addition to these, auxiliary elements were added to the walking sticks they used. This system consists of a camera system that enables the visually impaired individual to notice the lines made for the visually impaired in the environment, and a tracking circuit placed at the tip of the cane so that they can easily follow these lines and move more easily. Each system has been designed separately so that the warnings can be delivered to the visually impaired person quickly without delay. In this way, the error rate caused by the processing load has been tried to be reduced. The system we have created is designed to be wearable, easy to use and low-cost to be accessible to everyone.

1. Introduction

With the increase in the number of people in the world, the number of people per square meter in public living areas, especially in cities, is also increasing. The world population is approximately 7.3 billion and the number of visually impaired individuals in this population is 253 million in total. About 36 million of this number are completely blind [1]. In Turkey, this number is around 220 million. It is very difficult for these individuals to perceive obstacles in their living spaces, to recognize the objects in front of them, and to take precautions by noticing the auxiliary tools made for them. Visually impaired individuals use a cane to detect obstacles and recognize their surroundings. They may need other people to learn new environments they are not familiar with. In order to get rid of these problems, the use of sensors has become widespread. However, having information about the type of disability will enable the visually impaired individual to progress more safely. With the development of deep learning, progress has been made in the field of computer vision and studies to facilitate the lives of visually impaired individuals have accelerated [2]. It has been developed in systems with sensors capable of real-time object recognition and audible warning [3]. Applications that can run on low-cost smartphones have been developed so that they can find their way indoors and outdoors [4]. Some studies have been carried out so that they can move freely and recognize and identify the obstacles in front of them. Systems capable of GPS, obstacle detection and object detection have been designed. In these studies, ultrasonic sensors were generally used for obstacle detection [5]. In another study, a GPS assisted system was designed to enable visually impaired individuals to move by using ultrasonic sensors and vibration motors. Distance sensors and liquid sensors are used to make the walking sticks they use more functional. When these systems come into contact with liquid, they warn the user with a buzzer [6]. There are studies in the literature that have multiple

* **Corresponding Author:** ytaspinar@selcuk.edu.tr

wearable obstacle sensors and can warn the user in obstacle detection and liquid detection. There are studies that can detect the localization of objects with sensors that can detect depth using a Kinect or another RGB-D camera [7]. Computer vision is used in object detection and identification. However, the fact that these mobile systems run object recognition algorithms also brings some problems. Due to their processor and memory capacities, they cannot show an advanced level of success. It is foreseen that these problems can be overcome in time. In order for visually impaired people to recognize objects, systems that segment the objects in the scene by using depth sensor cameras and the depth of the images and transmit this to the user have been developed [8]. Although real-time image and video processing can be done by computers, in embedded systems these processes are often not possible or are performed slowly. Therefore, more powerful embedded systems have emerged. Embedded systems with GPU processors can identify objects faster than other embedded systems by performing real-time image and video processing. Existing deep learning models are tested on embedded systems and the fastest way to detect objects is carried out [9]. In these developments, it makes a great contribution to the work done for visually impaired people. In order to solve the problems by examining the studies in the literature, a deep learning-based camera vision support system, an obstacle detector with an ultrasonic sensor, a camera system that detects the roads made for the visually impaired and a sensor system that allows these roads to be followed easily have been proposed. The system includes some differences and improvements in addition to the studies in the literature. The embedded system used in the vision support system was created with Nvidia Jetson TX2, which can easily run deep learning algorithms. The circuit at the tip of the cane is a color sensor circuit that allows to follow the roads made for the visually impaired. There is a camera system in the middle of the cane and this system is a system that allows the user to notice the lines on the road. Multiple ultrasonic sensors are also designed to ensure that the user does not crash into surrounding obstacles. All these systems are designed to be operated easily by the user and are designed independently of each other in order to minimize the error rate. The materials and methods used in the second part of our study, the experimental results in the third part, and the results in the fourth part are given.

2. Material and Methods

In this section, the modules that make up the system are given in order. Firstly, the object identification system (ODS) sub-module, which enables the user to identify and vocalize objects, secondly, the obstacle detection sub-module with multi-sensors that allows the user to recognize the obstacles in front of them, and thirdly, the yellow line detection sub-module on the cane, which allows to detect the lines made for the visually impaired on the road. module, and finally, the yellow line tracking module at the tip of the cane, which enables the user to follow the yellow line. The modules that make up the system are shown in Figure 1. The use of the system created is shown in Figure 2.

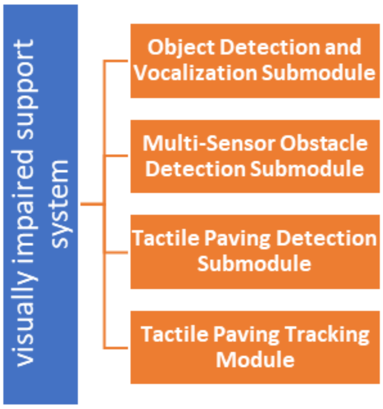

visually impaired support system
Object Detection and Vocalization Submodule
Multi-Sensor Obstacle Detection Submodule
Tactile Paving Detection Submodule
Tactile Paving Tracking Module

Figure 1. Visually Impaired Support System Modules

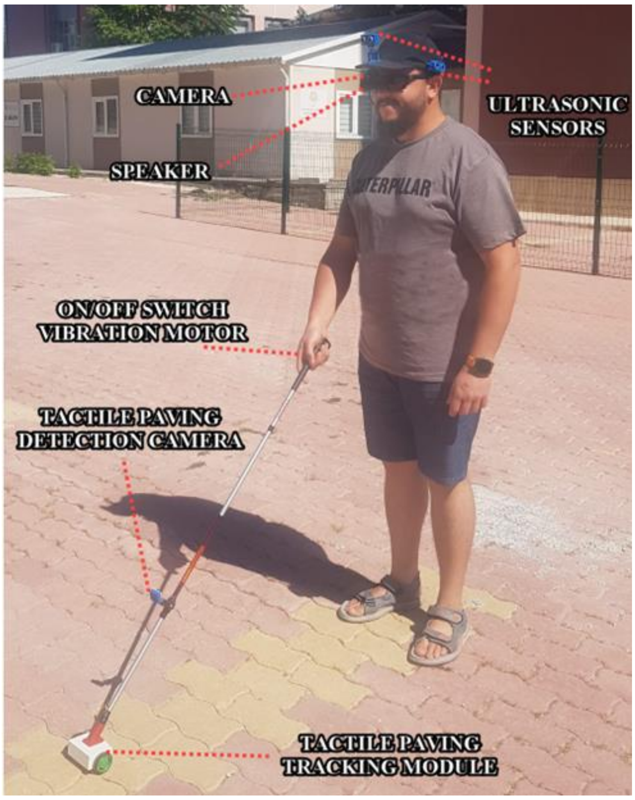

CAMERA
ULTRASONIC SENSORS
SPEAKER
ON/OFF SWITCH VIBRATION MOTOR
TACTILE PAVING DETECTION CAMERA
TACTILE PAVING TRACKING MODULE

Figure 2. Visually Impaired Support System

2.1. Object Detection and Vocalization Submodule

This module consists of a hat, a camera placed on this hat, and the Nvidia Jetson TX2 embedded system to which the camera is connected. Real-time object identification is made with the images coming from the camera. In order to make this detection, the SSD MobilNet V1 [11] model, which was trained with the COCO dataset [10], was used. There are several reasons why we use this model. The first of these is that this model has a shorter image resolution time than other trained models. On the other hand, classification success is lower than other models. However, since our object recognition study will be performed on an embedded system, this model has been used to avoid delays in object recognition and vocalization. In addition to this model, our model that we trained with our own dataset was used. In our dataset, there are 9000 images in 25 categories containing market products in Turkey. With this dataset, it is aimed that visually impaired people can recognize the products in the markets. Python programming language and Tensorflow Object Detection API [12] are used to run these models. This tool is used more frequently in object detection operations to be made with bounding box than other applications. The models we use in our study perform the object recognition process by working one after the other. Figure 3 shows the flow chart of object recognition and vocalization processes.

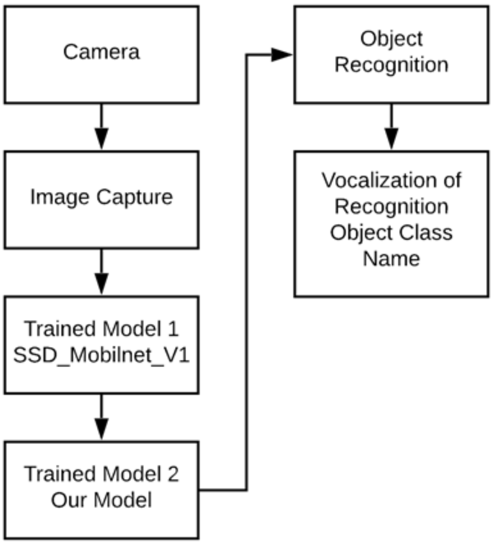


Figure 3. Object Recognition and Voiceover Process Flow Chart

MobilNet model structure was used in the training of our own model. The reason for using both models in the system can be shown as being able to recognize more objects, the ready model can be easily replaced with models with higher classification success when desired, and the model we have created can be continuously improved. Object detection is made by processing the frames coming to the SSD MobilNet V1 model. There may be objects that cannot be defined in this model. The same frame that entered the first model is processed by the second model, that is, by our model, and the objects are defined. Object class names defined in both models are voiced. Vocalization of class names was made with the Python TTS library, and because the names of the classes were in English, they were translated into Turkish and voiced. By selecting the desired language, voice over in that language can be done easily. A high resolution camera was chosen so that the images from the camera could be clear, but the high resolution brought with it processing load, causing the system to run slowly in some cases. The module contains Logitech C930e camera, Nvidia Jetson TX2 and headphones. The Logitech C930e camera is capable of capturing images in full HD quality and has a 90-degree viewing angle. Nvidia Jetson TX2 has a 4-core ARM57 CPU, 256 Nvidia Cuda-core GPU, 8GB of 128-bit LPDDR4 memory. Thanks to the GPU, image processing can be done quickly. By using these tools, object recognition is performed and voiced by performing operations on the frames coming from the camera in real time [13]. By setting the number of frames per second to 1, the user is prevented from being constantly disturbed by noise. If desired, the number of frames per second can be reduced. However, this can create a security vulnerability for the user.

2.2. Multi-Sensor Obstacle Detection Submodule

Ultrasonic sensors are sensors consisting of two modules that can detect an object, wall or other obstacle in front of it with sound waves. They can calculate distances using high-frequency sound waves that the human ear cannot hear. By vibrating one of the modules with an electrical signal, the module emits a sound wave. The second module creates an electrical signal at its output by vibrating with the sound waves reflected from the obstacle. The distance of the obstacle is calculated by calculating the time it takes for the sound wave to hit an obstacle and return. Because ultrasonic sensors can measure distance, they are also used in different areas such as measuring liquid levels in the tank. Ultrasonic sensors can measure distances between 2 cm and 400 cm, but the sound waves produced must hit an obstacle and bounce back. The sound waves produced may not return for some reason. The softness of the surface on which the sound wave will reflect can make it difficult for ultrasonic sensors to detect the obstacle. The speed of the sound wave emitted by the sensor is 343 m/s. Accordingly, the time required to measure a distance of 1 meter is 6 milliseconds. That's quite enough time for our work. The angle at which the ultrasonic sensor used in our study measures is 30 degrees. For this reason, many sensors are used in our obstacle detection module. It has been tried to minimize the possibility of the user hitting an obstacle without creating any blind spots. The sensor placed in front of the flat is placed on a servo motor, and the servo motor is rotated 30° at 2 second intervals, allowing the sensor to scan a wider area. Figure 4 shows the placement and detection areas of the ultrasonic sensors used in the obstacle detection module.

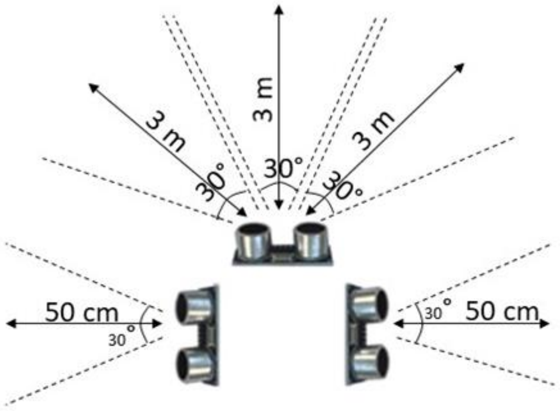


Figure 4. Placement and Detection Areas of Ultrasonic Sensors

There are tree sensors placed on the hat. The front sensor is designed to prevent the visually impaired person from hitting the obstacle in front of him. The object recognition module warns the user by recognizing 80 objects, but this warning system is needed for all obstacles other than these 80 objects. As seen in Figure 2, the viewing angle of each sensor is limited to 30°. For this reason, ultrasonic sensors have been placed on the right and left of the hat against the dangers from the right and left. The sensing distance of the previous ultrasonic sensor is 3 meters, and the sensing distance of the adjacent sensors is 1 meter. These distances can be easily changed if desired. Ultrasonic sensors on the sides are placed in order to prevent the user from hitting his head left and right, and to enable him to notice the objects passing by. The user is warned by 3 vibration motors, right, left and front. Right vibration motor for warnings coming from the right, left vibration motor for warnings coming from the left, front vibration motor for warnings coming from the front works according to the distance of the obstacle. As the obstacle approaches, the vibration motors vibrate more frequently, informing the user that the obstacle is approaching. It is designed in such a way that the warnings coming from the ultrasonic sensor in the front can be changed with the help of a button when desired, with the voice warning system. Right, left and front sensors can be disabled separately.

2.3. Tactile Paving Detection Submodule

Tactile paving are placed on the sidewalks, shopping malls and walking areas of various places so that visually impaired individuals can walk more quickly and safely. Visually impaired people can easily move forward by touching the reliefs on these paving with their walking sticks. However, in order to follow these paving, they must first be aware of their existence. Various studies have been carried out in the relevant literature on the determination of these paving. These paving are generally produced in yellow and yellow tones. Colors and designs may vary from country to country. However, it is used as yellow in Turkey. The reason why it is made in these colors is to attract the attention of people who are not visually impaired so that they do not put different objects on these roads. In this study, Raspberry Pi 3 B+ embedded system, Pi Camera compatible with this system, power supply, headphones and buzzer were used for the detection of paving. The images coming from the camera are processed by the Python OpenCV image processing library and the presence of paving is detected. As a method, background subtraction technique and color extraction were used. The background subtraction technique is generally used to detect or track objects by removing them from the background. It is a faster method for object detection than other methods. Along with this method, the color sorting method was also used. With the color extraction method, only objects in the specified color range are displayed by thresholding on the image. When paving in yellow and yellow tones are detected, an audible warning is given to the user. There are two options for the audible warning. The first option is buzzer and buzzer sounds when paving is detected. The second option is an audible warning with a headset, and it warns the user in the presence of paving by playing the desired mp3 sound. A mini speaker can be used instead of a headphone. The user can start and stop this system at any time with the help of a key. It is inevitable that the paving will get dirty over time and the color scale has been kept wide in order to reduce the detection error rate. The color tones used are shown in Figure 5.

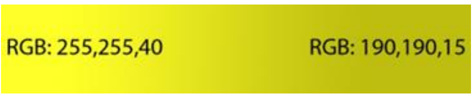


Figure 5. Color Scale Used in Tactile Paving Detection

The camera is placed in the middle of the cane and is positioned so that it can easily see the paving on the floor. The area that the camera scans continuously is shown in Figure 6.

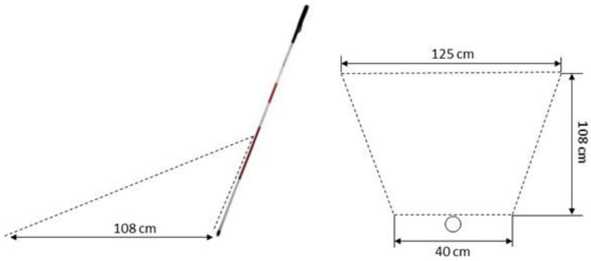


Figure 6. Camera Scan Area

When the tactile paving are detected, the user can follow the paving by running the tactile paving tracking module.

2.4. Tactile Paving Tracking Module

The yellow tactile paving on the pavements are made so that visually impaired citizens can find their way by feeling the small notches on the paving that they touch with their walking sticks on the pavement. In this way, they can continue on the road without falling into the pits, without getting on the road, without hitting something on the curves and they can be protected from other harmful effects. Tactile paving applications are made in order to facilitate the life of visually impaired individuals and to enable them to travel on the roads without being harmed. They try to walk by moving their canes on the paving and feeling the notches on it. However, due to this walking cane process, their progress on these paving is slower. In addition, in the parts of the pavement where there are no tactile paving, the floor may not be smooth and the visually impaired person may mistake them for tactile paving and go in different directions. Studies have been carried out to follow these paving with image processing techniques. Tracking systems have also been developed in order to provide easy access to frequently used places such as toilets and sinks with RFID. In this study, a tactile paving tracking module with a color sensor is recommended so that visually impaired individuals can progress on tactile paving much faster. While designing this module, cost and accessibility came to the fore as in other modules. Arduino Pro Mini, color sensor, power supply and vibration motor were used in the design of the circuit. The module is designed to be easily disassembled and attached to the cane tip. There are strong neodymium magnets on the module and the tip of the cane. The user module is enough to attach it to the tip of the cane. The module is stored in a protective case made in a 3D printer so that it is not affected by external factors. Two wheels are placed on this case. By means of these wheels, the module can move quickly on tactile paving. When the module goes out of the tactile paving, the vibration motor placed on the handle of the cane works and warns the user. In the parts where there is no tactile flooring or when the user does not want to use this module, he can disable the module with the button on the handle of the cane. Since the sensor in the module does not need any external light source, it can also be used in unlit environments. The circuit diagram of the module is shown in Figure 7. The module attached to the cane tip is shown in Figure 8.

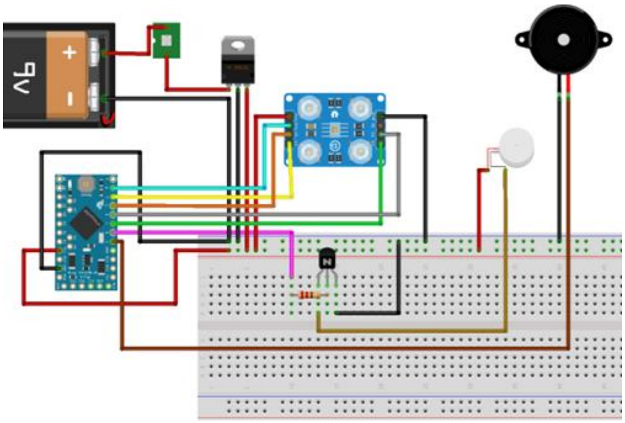


Figure 7. Tactile Paving Tracking Module Circuit Diagram

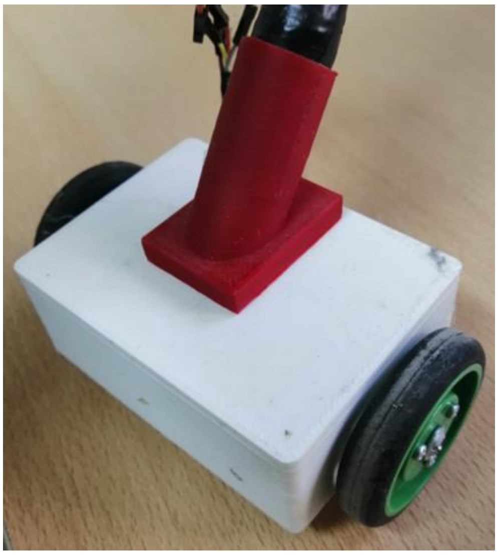


Figure 8. Tactile Paving Tracking Module

In case the color sensor used in this module cannot detect color due to external factors, the part of the module that touches the ground is covered with a soft material from the sides to touch the ground.

3. Experimental Results

All modules of our support system, which consists of many modules, are designed to not slow down each other's speeds and to give the user the fastest notification. It has been designed in such a way that visually impaired people can meet their daily needs by using equipment that will require minimum power, and they can easily change their power units when necessary. A support system was created with the aim of providing a support system that all visually impaired people can easily purchase, taking into account the minimum cost principle. Each module has been tested within itself. Experiment results are given in this section.

3.1. Object Detection and Vocalization Submodule Experimental Results

This module is the one that will help the user in the support system the most and is open to development. Thanks to the Nvidia Jetson TX2 embedded system GPU processor in this module, it provides a fast operation on the frames taken from the videos. However, it does not comply with the minimum cost policy considered in the design of the support module. However, the Nvidia Jetson Nano system, which is cheaper than this device, can be used instead of this embedded system. This embedded system is used in our system so that the tests can be done easily and quickly. Two models were used in our visual system and these models were tested for success. Our first model is the SSD Mobilnet V1 model, which includes weights previously trained with the COCO dataset. With this model, 80 objects can be defined. The object recognition success of the SSD Mobilnet V1 model is 72.4%. This model has been used because it can run faster on the embedded system. It is important for the detection of objects as well as the classification rate. The object identification rate was obtained by taking the arithmetic average of the identification rates of the objects. In the same way, tests were carried out with our model created with our own dataset containing 24 classes. The results of the tests performed with SSD Mobilnet V1 and our own model are shown in Table 1. There is no object that both models can define in common. 85% of the generated dataset was used to train the model and 15% for testing. When using models, the process of placing them in the processor, that is, the allocate process, is performed. Since the two models work one after the other, there is a time to allocate the models on the processor. Each model is allocated once on the first run of the program on the processor. These times are shown in Table 1.

Table 1. Success Rates and Allocate Times of Models

Models	Accuracy (%)	Allocation Time (second)
SSD MobilNet V1	72.4	3.19
Our Model	90.2	3.48

A large number of objects can be defined by taking advantage of the pre-trained model, and more objects can be defined with the model trained using the dataset we have created. The product can be defined and voiced from 25 different product packages.

3.2. Multi-Sensor Obstacle Detection Submodule Experimental Results

The sensor placed on the front of the hat in the obstacle detection module rotates at an angle of 30 degrees every 2 seconds with the help of a servo motor. Ultrasonic sensors placed on the sides of the hat scan a scanning area of 30 degrees. In this way, it is ensured that the user can take precautions by noticing all the obstacles in front of him, to his right and to his left. Since the movable sensor placed at the front rotates every 2 seconds, it has been tested against the possibility of missing the obstacles. In addition, the error rate was determined by measuring the differences between the distance measured by the ultrasonic sensor and the actual distance. The distance measured by the ultrasonic sensor and the actual distance are shown in Table 2.

Table 2. Distance Measured by Ultrasonic Sensor and Actual Distance

Distance (cm)	200	150	100	75	50	25	10
Distance Measured by Ultrasonic Sensor (cm)	196	146	99	74	49	24.5	9.75
Error rate (%)	2	2.66	1	1.33	2	2	2.5

When the data obtained in Table 2 are examined, it is seen that the error rate is the least in the measurements made at 75 and 100 cm. The distances measured by the ultrasonic sensor may vary depending on weather conditions, electronic noise in the environment where the sensor is located, and the type of obstacle. The sensors placed on the right and left sides of the hat, which enable the user to notice the objects on the right and left, are adjusted to measure 50 cm. According to the data in Table 2, it can warn the user by measuring with an error of 1 cm at a distance of 50 cm. This value is acceptable for our system. The error rate for the ultrasonic sensor placed on the front of the hat and checking the distance of 200 cm every 2 seconds is 2%. All microcontrollers, sensors and servo motor on the flat are fed from the same power source in order not to weigh the user down. The power consumed when the servo motor is connected and the ultrasonic sensor is connected without any auxiliary elements on the control circuit are shown in Table 3.

By measuring the power consumed by the Arduino and its components, it was observed how much power the obstacle detection system would consume. In this way, how long the user will need the battery can be calculated. As a result of the tests, one battery can operate the obstacle detection system for 8 hours.

3.3. Tactile Paving Detection Submodule Experimental Results

Successful results have been achieved with minimum cost and minimum hardware understanding in this module, which we designed in order to recognize the tactile paving made for the visually impaired. Raspberry Pi 3 B+, the cheapest embedded system, easily ran the OpenCV image processing library. With the camera connected to the Raspberry Pi 3 B+, there was no need for extra cooling in our system, which constantly takes images from the ground. This is an advantage in terms of power

Table 3. Power Consumption of Arduino and Components

	Volt		Amper		Watt	
	Min.	**Max.**	**Min.**	**Max.**	**Min.**	**Max.**
Arduino without component	2.800	2.860	0.011	0.014	0.040	0.069
Arduino with Servo Motor	2.800	2.860	0.020	0.065	0.054	0.199
Arduino with Ultrasonic Sensor	2.800	2.860	0.014	0.017	0.069	0.084

consumption.

Raspberry Pi 3 B+ can run smoothly in the module where we use a rechargeable power supply. When operating under maximum power, the power supply module can operate for 9 hours. Real field tests were carried out in order to test the reliability of the module's process of detecting tactile paving and informing the user. Tactile paving are made in shades of yellow in Turkey, and their colors can change over time by being affected by natural conditions. Our preference for a wide color scale that can detect paving with changing colors reduces the error rate of the system.

3.4. Tactile Paving Tracking Submodule Experimental Results

This module has been proposed in order to follow the tactile paving in the fastest way. There are image processing and tracking systems in the literature, but we have designed a simpler, very low cost and accessible module for visually impaired individuals to move on tactile flooring. This module is designed to fit between bubbles on tactile paving. In this way, the user will be able to feel these bubbles and proceed safely. In order for the module to detect tactile paving, it must be at a certain distance from the paving. As a result of the measurements and tests, the distance of the sensor on the module should be a maximum of 25 mm from the paving. Considering this value, the position of the sensor on the module has been adjusted. Although the sensor located at the bottom of the module has its own illumination, it is affected by the external light and gives false warning. In order to prevent this, the part where the sensor is located is closed so that it will not receive light from the outside, as seen in Figure 7. In field tests, tactile paving could be followed successfully with the help of the module at the tip of the cane.

4. Conclusions

In order for visually impaired individuals to live their social lives without any problems, environmental barriers should be minimized. With the support system we have created based on this problem, the work of visually impaired individuals will be easier. In this way, they will be able to devote more time to social life. With the camera placed on the cane, the tactile paving on the road made for the visually impaired can be detected. The presence of tactile paving is reported audibly to the visually impaired individual. With the tactile paving tracking module placed on the tip of the cane, it can easily follow the yellow lines on the road without moving the cane. When it goes out of the yellow line, the cane vibrates and gives a warning. This module can be disabled with the button on the walking stick. With the help of a hat equipped with ultrasonic sensors and a camera, the visually impaired individual can move forward without hitting an obstacle. Whichever direction the obstacle is approaching, the part of the hat where the obstacle is located vibrates and warns the visually impaired individual. With the help of the camera on the front of the hat, 80 different objects are detected and voiced. In addition, a dataset consisting of 9000 images containing market products in Turkey was created. With the model created using this dataset, 25 different market products can be identified and voiced.

The proposed system has been tested in real life and its operability has been tested. Object recognition models are flexible. For this reason, their training can be carried out so that they can recognize more objects. Embedded systems used in the created system can be easily changed by users in case of failure due to time. The proposed system can be transformed into a product so that all visually impaired people can access it.

References

[1] Islam, M.M., M.S. Sadi, K.Z. Zamli, and M.M. Ahmed, *Developing walking assistants for visually impaired people: A review.* IEEE Sensors Journal, 2019. **19**(8): p. 2814-2828.

[2] Kuriakose, B., R. Shrestha, and F.E. Sandnes, *Tools and technologies for blind and visually impaired navigation support: a review.* IETE Technical Review, 2022. **39**(1): p. 3-18.

[3] Simões, W.C., G.S. Machado, A. Sales, M.M. de Lucena, N. Jazdi, and V.F. de Lucena, *A review of technologies and techniques for indoor navigation systems for the visually impaired.* Sensors, 2020. **20**(14): p. 3935.

[4] Choi, J., S. Jung, D.G. Park, J. Choo, and N. Elmqvist. Visualizing for the non-visual: Enabling the visually impaired to use visualization. in Computer Graphics Forum. 2019. Wiley Online Library.

[5] Real, S. and A. Araujo, Navigation systems for the blind and visually impaired: Past work, challenges, and open problems. Sensors, 2019. **19**(15): p. 3404.

[6] Zhang, J., K. Yang, A. Constantinescu, K. Peng, K. Müller, and R. Stiefelhagen. Trans4Trans: Efficient transformer for transparent object segmentation to help visually impaired people navigate in the real world. in Proceedings of the IEEE/CVF International Conference on Computer Vision. 2021.

[7] Tapu, R., B. Mocanu, and T. Zaharia, *Wearable assistive devices for visually impaired: A state of the art survey.* Pattern Recognition Letters, 2020. **137**: p. 37-52.

[8] Manjari, K., M. Verma, and G. Singal, *A survey on assistive technology for visually impaired.* Internet of Things, 2020. **11**: p. 100188.

[9] Aruna, M.A., M.B. Mol, M. Delcy, and P.D.M. ME, *Rduino Powered Obstacles Avoidance For Visually Impaired Person.* International Journal of Engineering and Information Systems (IJEAIS), 2018. **3**(2).

[10] Lin, T.-Y., M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C.L. Zitnick. *Microsoft coco: Common objects in context*. in *European conference on computer vision*. 2014. Springer.

[11] Li, Y., H. Huang, Q. Xie, L. Yao, and Q. Chen, *Research on a surface defect detection algorithm based on MobileNet-SSD.* Applied Sciences, 2018. **8**(9): p. 1678.

[12] Sai, B.K. and T. Sasikala. Object detection and count of objects in image using tensor flow object detection API. in 2019 International Conference on Smart Systems and Inventive Technology (ICSSIT). 2019. IEEE.

[13] Taspinar, Y.S. and M. Selek, *Object recognition with hybrid deep learning methods and testing on embedded systems.* International Journal of Intelligent Systems and Applications in Engineering, 2020. **8**(2): p. 71-77.

Intelligent Methods in Engineering Sciences 1(1): 008-012, 2022

IMIENS

INTELLIGENT METHODS
IN ENGINEERING SCIENCES

https://www.imiens.org

International Open Access

December, 2022

e-ISSN 2979-9236

Research Article

https://doi.org/10.58190/imiens.2022.2

Motor Imagery BCI Classification with Frequency and Time-Frequency Features by Using Different Dimensions of the Feature Space Using Autoencoders

Esra KAYA [a,*], ***Ismail SARITAS*** [a]

[a] *Selcuk University, Faculty of Technology, Electrical and Electronics Engineering, Konya, Turkiye*

ARTICLE INFO

Article history:
Received 27 September 2022
Accepted 10 October 2022

Keywords:
ANN
Autoencoder
BCI
EEG
EMD
Hilbert-Huang
Wavelet

ABSTRACT

Brain-Computer Interfaces (BCIs) enable the users to directly communicate with machines based on various desired purposes through brain signals without moving any body parts. Thus, they have become very useful for prostheses, electric wheelchairs, virtual keyboards, and other studies like survey applications and emotion classifications. In this study, EEG signal processing was performed on the BCI Competition III-3a dataset, which contains motor imagery (MI) signals with four classes. Features of the non-stationary EEG signals belonging to three subjects were extracted using Power Spectral Density (PSD) with welch method, Wavelet Decomposition (WD), Empirical Mode Decomposition (EMD) and Hilbert-Huang Transform (HHT). From extracted 900 features, feature space dimension reduction was realized using Autoencoder, an unsupervised learning algorithm. The average accuracy obtained with Artificial Neural Network (ANN) is 74.5% for all binary classifications, which is generally a good result because of the non-stationary nature of EEG signals. 801 features yielded the best classification performance, obtained using an autoencoder with 400 hidden layer neurons.

1. Introduction

The BCI is a direct communication path with or without a cable, which allows bi-directional information flow between the brain and an external device, where the brain signals are information carriers. BCIs are often used to research, map, support, increase or repair human cognitive or sensory-motor functions. BCIs are used in many applications, such as prostheses and electric wheelchairs, virtual keyboards, survey applications, and emotion classifications.

Electroencephalography (EEG) is the most used for BCI applications and physiological signals, which are electrical potential representations of brain signals. EEG signal is a harmless signal to human health and is more advantageous and easier to use than other techniques that help us obtain and analyze brain signals [1-4]. Motor Imagery (MI) signals are the EEG signals that occur when the user of the BCI system imagines the necessary movement of a body part to use a specific purpose machine [5].

In literature, Royer et al. conducted a study on the control of a virtual helicopter in a three-dimensional space using MI signals with four classes: right-hand movement to go right, left-hand movement to go left, both hands up to rise, and both hands down to descend or rest. %67 of flight time was closer to the intended path while using BCI in this study [6]. In another study, Bhattacharyya et al. designed an Interval Type-2 Fuzzy classifier to classify EEG signals obtained by imagining a total of five wrist and finger movements presented with audio and visual stimulation. They extracted Extreme Energy Ratio (EER) features and obtained 86.45% and 78.44% accuracies for offline and online classifications, respectively [7]. Ang and Guan identified a strategy for detecting MI signals for control and rehabilitation purposes. 29 of 34 chronic stroke patients were suitable for BCI use. Within the calibration sessions used for training and subsequent test sessions, subjects were assigned two or more MI tasks, such as left- and right-hand movements. The accuracy rates obtained from the signal analyzed by the common spatial pattern filter bank were 79.8% for offline and 69.5% for online classifications [8]. Mahajan and Bansal developed a neuro-rehabilitation control application that processes EEG signals through Arduino. EEG signals were obtained over a total of 5 sessions which consisted of 20s periods, from each of the three male and seven female subjects. Peak amplitude values were used as features in the signals, classified as comfortable condition and blink. If the

*** Corresponding Author:** esrakaya@selcuk.edu.tr

maximum peak amplitude value exceeds the threshold value, the led lamp is turned on with Arduino Uno, corresponding to the blink [9]. Mistry et al. conducted a study to realize wheelchair control using visual triggering for individuals with motor cortex disabilities. The signal intensity was determined by calculating the target frequencies, mean values, and SNR values with Fourier transform of EEG signals received from the parietal lobe and occipital lobes of 4 individuals with visual triggers consisting of 4 different vibration speeds (7 Hz, 9 Hz, 11 Hz, and 13 Hz). From 7 Hz LED for left, 9 Hz LED for forward, 11 Hz LED for right, and 13 Hz LED for backward, only two LEDs were active each time, first right and left decision, then forward and backward decision. The average accuracy was 79.4%, and a simple path follow-up using all the classes took 5 minutes and 9 seconds [10]. Athif et al. introduced a new method called WaveCSP, which combines wavelet decomposition and Common Spatial Pattern (CSP) concepts to extract features for more robust classification of EEG signals. The classification of the right-hand and left-hand MI signals has given 63.5% average accuracy with the k-Nearest Neighbor (kNN) classifier [11]. A new network structure called QNet was proposed by Fan et al., which learns the attention weight of EEG channels, time points, and feature maps. The method provided an 82.9% accuracy rate for the classification of right-hand and left-hand MI signals [12]. Xiao et al. proposed a channel selection algorithm based on coefficient-of-variation for right-hand and left-hand MI EEG signal classifications by dividing channels into different categories according to their contributions to the feature extraction process. They have achieved an average accuracy of 74.30% [13]. As seen from the literature, many studies use different features and methods for various BCI control applications. However, the accuracy results are not at the desired levels yet. Thus, BCI technology needs more effective feature extraction, feature selection, and classification algorithms for more accurate daily use.

This study used the IIIa dataset consisting of four class MI EEG signals created for the BCI Competition in the BCI laboratory of Graz University of Technology, Austria. We have extracted features using Power Spectral Density (PSD) with the welch method, Wavelet Decomposition, Empirical Mode Decomposition (EMD), and Hilbert-Huang Transform (HHT). The dimensions of the feature space increase if there are more electrode channels in an EEG cap and if more feature extraction methods are used to represent a signal. Thus, in this study, the feature space was reduced using Autoencoder neural network, an unsupervised learning algorithm [14]. The classification was realized with 5-Fold Cross-Validated ANN. The autoencoder hidden layer sizes were changed to see the effects of different size feature spaces on the binary classification of EEG signals. It was seen that the Autoencoder could reduce the size of the existing feature space and represent the feature space more effectively.

2. Material and Methods

2.1. BCI Competition III-3a Dataset

BCI Competition III-3a dataset was obtained in the BCI laboratory of Graz University of Technology, Austria. The dataset contains MI signals with four classes: right-hand movement to go right, left-hand to go left, tongue movement to go up, and feet movement to go down [15, 16]. The sequence of the events in the dataset was shown to 3 subjects according to the paradigm shown in Figure 1.

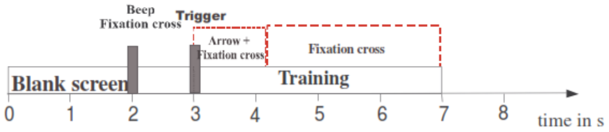


Figure 1. The paradigm of BCI Competition IIIa Dataset [15, 16]

The EEG signals of the three subjects based on the designated paradigm were obtained from a 60-electrode EEG Cap [15, 16]. The electrode positions corresponding to various regions of the brain are shown in Figure 2. In this study, EEG signals that belong to 9 electrode channels were used, which were 6, 8, 20, 22, 28, 31, 34, 48, and 50 electrode placements. These electrodes correspond to the brain's frontal lobe, motor cortex, and parietal lobe, where activities are related to control applications.

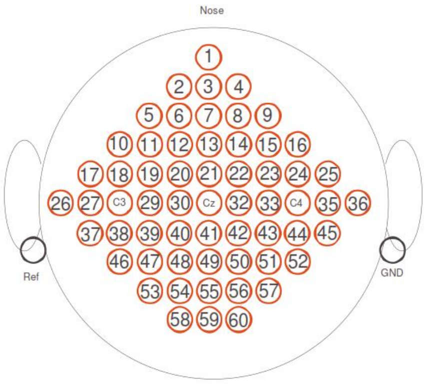


Figure 2. Electrode Positions of a 60 Electrode Cap [15, 16].

The EEG segments for four classes belonging to the first, second, and third subjects were 360, 240, and 240, respectively. Half of these segments are labeled, and the other half is unlabeled [15, 16]. In this study, the labeled segments consisting of 2560 samples were used. For segments with fewer than 2560 samples, the averages of the samples were taken to complete the missing data. A total of 409 labeled segments that were used in this study were 175, 118, and 116 for the first, second, and third subjects, respectively.

2.2. EEG Signal Processing and Feature Extraction

Before we used the signals from the stated nine electrodes, we applied the Common Average Reference (CAR) method, subtracting the average value of all electrodes from the signals of all electrodes. In addition, baseline correction is applied for each electrode separately by subtracting the average of the signal from the signal itself belonging to one electrode.

Then, in order to protect the signals' significant parts and reveal their outline, the 9th-degree db4 wavelet decomposition was applied, and the detail coefficients of the four lowest levels were subtracted from the main signal; thus, filtering was performed. After filtering the signals, we separated the signals into their bands (delta, theta, alpha, beta, gamma) using elliptic filters to extract features defining them. An example of EEG bands belonging to the first class (right hand) from the first electrode of the first subject is shown in Figure 3.

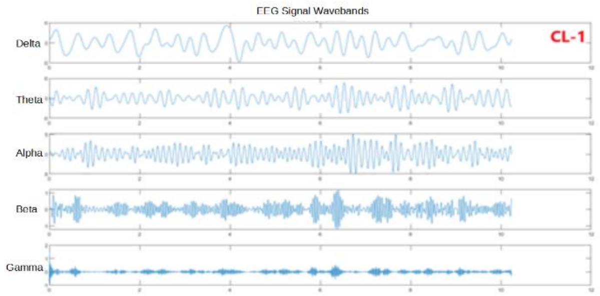


Figure 3. EEG Bands belonging to Class-1of the first electrode of the first subject

Welch method, which is the averaged and modified version of the periodograms [17], was applied to the signals of each electrode channel, and Power Spectral Density (PSD) values were found. These values were calculated for each EEG band separately, and the mean, standard deviation, skewness, kurtosis, and logarithmic energy entropy values of PSDs were used as frequency domain features.

Then, the 9th-degree Daubechies-db4 wavelet coefficients were obtained because wavelet transform can give information about the changes in a signal or an image and the time location of the occurring changes, unlike Fourier transform, which loses the time information [18]. Mean, standard deviation, skewness, kurtosis, and logarithmic energy entropy values of wavelet coefficients were calculated separately for each EEG band and used as time-frequency features.

Empirical Mode Decomposition (EMD) method considers the oscillations in signals in a very local manner [19]. The decomposition is based on the understanding that the signal can contain many simple oscillations at significantly different frequencies superimposed on each other [20]. The results of the EMD method are components defined as Intrinsic Mode Functions (IMF), which define the oscillations in a signal. These IMFs must satisfy the following conditions: The number of extremes and zero-crossings must either be equal or differ at most by 1, and the mean value of the enveloping signals defined as local maxima and local minima must be zero at any given data point [20]. After applying EMD with piecewise cubic Hermite interpolating polynomial method for envelope construction to all EEG bands, the mean, standard deviation, skewness, kurtosis, and Shannon entropy values were calculated for the resulting maximum 5 IMFs and used as time-frequency features.

Hilbert-Huang Transform (HHT) reveals the Hilbert spectrum of a signal sampled at a specific frequency and specified by IMFs resulting from EMD. HHT is useful for analyzing signals that contain a mixture of signals whose spectral components changes in time. The time-frequency representation of a signal with HHT does not contain spurious oscillations. Thus, the signal is in its more natural and physically meaningful form [20]. After applying HHT to all EEG bands, the mean, standard deviation, skewness, kurtosis, and Shannon entropy values were calculated and used as time-frequency features.

As a result of the feature extraction process, a total of 900 features were obtained from 5 EEG bands of 9 electrode channels belonging to 3 subjects. Five statistical calculations from 5 EEG bands of 9 channels for one feature extraction method result in 5x5x9 = 225 features, resulting in 225x4 feature extraction methods total of 900 features.

2.3. Feature Space Dimension Reduction and Classification

After obtaining the total amount of 900 features, we have decided to reduce the feature space dimension because 900 features need lots of computation time. The process should be fast and effective. So, we have used Autoencoder neural networks for feature space dimension reduction. Autoencoders are unsupervised learning algorithms that automatically learn features from unlabeled data by applying backpropagation and setting the number of target values equal to the number of inputs. Autoencoders are also useful for discovering interesting structures in the data by placing constraints on the network, such as limiting the number of hidden units [14]. Suppose the data is not reconstructed at the end of the procedure. In that case, the encoder and decoder weights and biases based on hidden layer size can be used as a reduced version of the feature space dimension.

We have selected 50, 100, 150, 200, 250, 300, 350, 400, and 450 neurons for a hidden layer of Autoencoder. The autoencoders used the Scaled Conjugate Gradient function for backpropagation and Mean Squared Error with L2 and Sparsity Regularizers for performance function. The encoder weights, decoder weights, and decoder biases were used as feature space which consists of 101, 201, 301, 401, 501, 601, 701, 801, and 901 features for all hidden layers, respectively.

After reducing the feature space dimension, we have classified the EEG samples belonging to 3 subjects divided into four categories: right-hand, left-hand, tongue, and feet. The classifications were realized as binary. Of the 409 labeled samples in total, there are 101 samples from the first category (right-hand), 100 samples from the second category (left-hand, 104 samples from the third category (tongue), and, finally, 104 samples from the fourth category (feet). The classifier is a pattern recognition ANN with 200 hidden neurons trained with a resilient backpropagation algorithm. The ANN was applied with 5-fold cross-validation. The 409 samples were divided randomly as 70% for training, 15% for testing, and 15% for validation. The transfer functions of the network were chosen as a logarithmic sigmoid function for the input layer and a tangent sigmoid function for the output layer. The performance function was Mean Squared Error function. The flow chart of the study is shown in Figure 4.

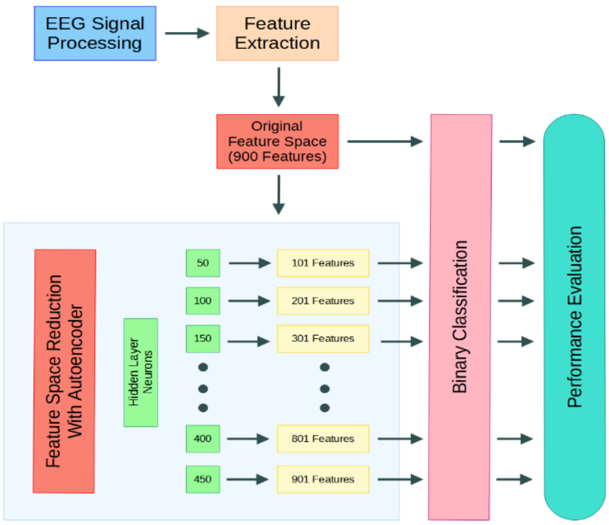


Figure 4. The flow chart of the study.

3. Results and Discussion

After all the binary classifications were realized, the testing accuracy rates based on different feature spaces obtained using autoencoders are given in Table I. The results are shown as a different representation in Figure 5, where the changes in results can be seen based on the change in feature space size.

According to the results shown in Table I, the original feature space and feature space with 901 elements obtained with autoencoders have similar accuracy rates. The maximum accuracy rate of 74.5% was obtained with 801 features. However, features of 401 and 501 obtained with autoencoders with 200 and 250 neurons are also close to the maximum accuracy rate of 71.5% and 70.8%, respectively.

On the other hand, right-hand and left-hand classification is the most classified classification pair in literature, and the maximum accuracy rate obtained for this pair is 86.7% with 401 features. The maximum accuracy of 80.6% and 90.3% for right-hand and tongue, and right-hand and feet pairs, respectively, were obtained with 801 features. For the left-hand and tongue pair, the maximum accuracy of 83.9% was obtained with 701 features from the Autoencoder's 350 neurons. Finally, 80.6% maximum accuracies were obtained with 501 features of the Autoencoder's 250 neurons for the left-hand and feet pair and tongue and feet pair.

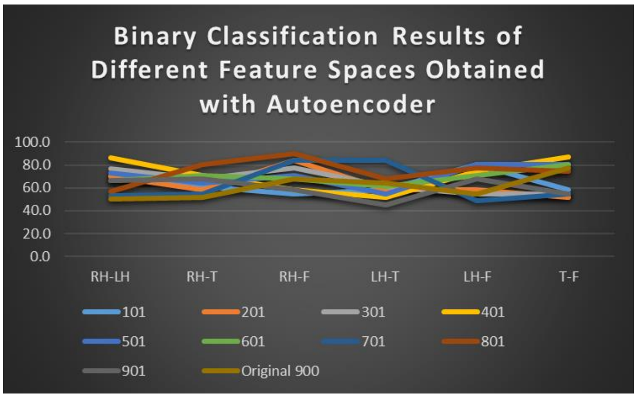


Figure 5. Binary classification results of different feature spaces were obtained using an Autoencoder with a different number of neurons in the hidden layer.

The lowest standard deviation value of binary classifications was 5.9, obtained for 601 features, which shows that the change in accuracies for all binary

Table 1. Testing accuracy rates (%) of binary classifications based on feature spaces obtained using autoencoders with different hidden layer sizes. (RH: Right-Hand, LH: Left-Hand, T: Tongue, F: Feet)

Classification Categories	Feature Space Size Created with Autoencoder									Original Feature Space (900)
	101	201	301	401	501	601	701	801	901	
RH-LH	70.0	70.0	76.7	86.7	73.3	66.7	53.3	56.7	66.7	50.0
RH-T	61.3	58.1	67.7	71.0	64.5	71.0	54.8	80.6	67.7	51.6
RH-F	54.8	83.9	77.4	58.1	71.0	67.7	83.9	90.3	58.1	67.7
LH-T	58.1	58.1	61.3	51.6	54.8	61.3	83.9	67.7	45.2	64.5
LH-F	80.6	58.1	54.8	74.2	80.6	71.0	48.4	77.4	67.7	54.8
T-F	58.1	51.6	54.8	87.1	80.6	80.6	54.8	74.2	54.8	77.4
Average Acc. (%)	63.8	63.3	65.5	71.5	70.8	69.7	63.2	74.5	60.0	61.0
Std. Dev.	8.9	10.7	9.3	13.3	9.1	5.9	14.8	10.5	8.3	9.8

classifications is less than for other feature space sizes. The average accuracy for 601 features was 69.7%, obtained using an autoencoder with 300 hidden layer neurons. The results show that the Autoencoder does not only change the size of feature space but changes the weights of the features because the average accuracy results change randomly.

4. Conclusion

The maximum average accuracy for all binary classifications was 74.5%, with 801 features obtained using an autoencoder with 400 hidden layer neurons. This result is acceptable because the nature of the signal is non-stationary, so it is hard to characterize EEG signals. However, the original feature space has 900 features, so there is not much reduction of the feature space using an autoencoder. On the other hand, the feature space with 401 features is the closest one, with 71.5% accuracy. Also, the literature's most used classification pair of right-hand and left-hand has an 86.7% accuracy rate with 401 features.

This study shows that the average accuracy rates of binary classifications for features obtained using autoencoders with different size hidden layers change randomly. Thus, it can be said that an autoencoder does not change only the feature space size but the weights of features while representing the original feature space in another way.

In another study, the most useful features can be found in the original feature space and use an autoencoder to represent them for a more compact and effective feature space.

Author's Note

Part of this work was presented at the 9th International Conference on Advanced Technologies ICAT'2020 Istanbul, Turkiye.

References

[1] M. Hämäläinen, R. Hari, R. J. Ilmoniemi, J. Knuutila, and O. V. Lounasmaa, "Magnetoencephalography—theory, instrumentation, and applications to noninvasive studies of the working human brain," Reviews of modern Physics, vol. 65, no. 2, p. 413, 1993.

[2] R. Srinivasan, "Methods to improve the spatial resolution of EEG," International journal of bioelectromagnetism, vol. 1, no. 1, pp. 102-111, 1999.

[3] P. M. Vespa, V. Nenov, and M. R. Nuwer, "Continuous EEG monitoring in the intensive care unit: early findings and clinical efficacy," Journal of Clinical Neurophysiology, vol. 16, no. 1, pp. 1-13, 1999.

[4] F. Yasuno et al., "The PET radioligand [11 C] MePPEP binds reversibly and with high specific signal to cannabinoid CB 1 receptors in nonhuman primate brain," Neuropsychopharmacology, vol. 33, no. 2, pp. 259-269, 2008.

[5] J. Decety and D. H. Ingvar, "Brain structures participating in mental simulation of motor behavior: A neuropsychological interpretation," Acta psychologica, vol. 73, no. 1, pp. 13-34, 1990.

[6] A. S. Royer, A. J. Doud, M. L. Rose, and B. He, "EEG control of a virtual helicopter in 3-dimensional space using intelligent control strategies," IEEE Transactions on neural systems and rehabilitation engineering, vol. 18, no. 6, pp. 581-589, 2010.

[7] S. Bhattacharyya, M. Pal, A. Konar, and D. Tibarewala, "An interval type-2 fuzzy approach for real-time EEG-based control of wrist and finger movement," Biomedical Signal Processing and Control, vol. 21, pp. 90-98, 2015.

[8] K. K. Ang and C. Guan, "EEG-based strategies to detect motor imagery for control and rehabilitation," IEEE Transactions on Neural Systems and Rehabilitation Engineering, vol. 25, no. 4, pp. 392-401, 2016.

[9] R. Mahajan and D. Bansal, "Real time EEG based cognitive brain computer interface for control applications via Arduino interfacing," Procedia computer science, vol. 115, pp. 812-820, 2017.

[10] K. S. Mistry, P. Pelayo, D. G. Anil, and K. George, "An SSVEP based brain computer interface system to control electric wheelchairs," in 2018 IEEE International Instrumentation and Measurement Technology Conference (I2MTC), 2018: IEEE, pp. 1-6.

[11] M. Athif and H. Ren, "WaveCSP: a robust motor imagery classifier for consumer EEG devices," Australasian physical & engineering sciences in medicine, vol. 42, no. 1, pp. 159-168, 2019, doi: https://doi.org/10.1007/s13246-019-00721-0.

[12] C.-C. Fan, H. Yang, Z.-G. Hou, Z.-L. Ni, S. Chen, and Z. Fang, "Bilinear neural network with 3-D attention for brain decoding of motor imagery movements from the human EEG," Cognitive Neurodynamics, vol. 15, no. 1, pp. 181-189, 2021, doi: https://doi.org/10.1007/s11571-020-09649-8.

[13] R. Xiao, Y. Huang, R. Xu, B. Wang, X. Wang, and J. Jin, "Coefficient-of-variation-based channel selection with a new testing framework for MI-based BCI," Cognitive Neurodynamics, vol. 16, no. 4, pp. 791-803, 2022, doi: https://doi.org/10.1007/s11571-021-09752-4.

[14] A. Ng, "Sparse autoencoder," CS294A Lecture notes, vol. 72, no. 2011, pp. 1-19, 2011.

[15] A. Schlögl, G. Müller, R. Scherer, and G. Pfurtscheller, "BIOSIG-an Open Source Software Package for biomedical Signal Processing," in 2nd OpenECG Workshop, 2004: . pp. 77-78.

[16] B. Blankertz et al., "Bci competition iii," Fraunhofer FIRST. IDA, http://ida. first. fraunhofer. de/projects/bci/competition_iii, 2005.

[17] S. Villwock and M. Pacas, "Application of the Welch-method for the identification of two-and three-mass-systems," IEEE Transactions on Industrial Electronics, vol. 55, no. 1, pp. 457-466, 2008.

[18] R. Choudhary, S. Mahesh, J. Paliwal, and D. Jayas, "Identification of wheat classes using wavelet features from near infrared hyperspectral images of bulk samples," Biosystems Engineering, vol. 102, no. 2, pp. 115-127, 2009.

[19] G. Rilling, P. Flandrin, and P. Goncalves, "On empirical mode decomposition and its algorithms," in IEEE-EURASIP workshop on nonlinear signal and image processing, 2003, vol. 3, no. 3: Citeseer, pp. 8-11.

[20] N. E. Huang and Z. Wu, "A review on Hilbert-Huang transform: Method and its applications to geophysical studies," Reviews of geophysics, vol. 46, no. 2, 2008.

Intelligent Methods in Engineering Sciences 1(1): 013-017, 2022

IMIENS

INTELLIGENT METHODS
IN ENGINEERING SCIENCES

https://www.imiens.org

International
Open Access

December, 2022

e-ISSN 2979-9236

Research Article

https://doi.org/10.58190/imiens.2022.3

Autonomous Car for Indian Terrain

S. T. Patil [a], ***Aryan Aher*** [a,*], ***Aarushi Bhate*** [a], ***Adnan Shaikh*** [a],

Sandhya Vinukonda [a]

[a] *Computer Engineering Dept., Vishwakarma Institute of Technology, Pune, India*

ARTICLE INFO

Article history:
Received 11 October 2022
Accepted 27 October 2022

Keywords:
Autonomous Vehicle (AV)
Convolutional neural network
Deep learning
Neural Network

ABSTRACT

In recent years, autonomous vehicle (AV) technology has improved dramatically. Self-driving cars have the potential to transform urban mobility in India by offering sustainable, convenient, and congestion-free transportation. However, India confronts challenges such as potholes and the need for enhanced lane detection to make autonomous vehicles a reality. In countries like India, lanes are incomplete, causing potential confusion with broken lanes and the need for advanced object detection. Many traffic rules are not followed, leading to potential fatalities if the objects are not correctly detected. The project's central goal is to create a Convolution Neural Network (CNN) model that can scan and identify its surroundings and move. To achieve this, we have experimented with various CNN layers to achieve maximum accuracy and implemented real-time footage-to-image conversion by processing and standardizing the dataset. This paper proposes a project accomplished by training CNN with a dataset of images and videos to perform advanced lane identification, pothole recognition, and sophisticated object detection.

1. Introduction

Self-driving cars are technological developments in the automotive field. Self-driving cars are the future of humanity, but they are the most expensive cars. Here, we focused on two applications of Automated Vehicle and designed their prototype vehicles. The only major problem is that when there is heavy traffic, the driver must always apply the brakes, accelerators, and clutches to get to his destination slowly. We have proposed a solution to relax the driver in this situation. This makes the vehicle intelligent, keeps a certain distance from surrounding vehicles and obstacles, makes decisions automatically, and moves.

In recent years, autonomous vehicles have become a reality and are practiced in many cosmopolitan cities. But this does leave room for the question of the efficiency or the reliability of these cars in countries with rugged terrain. In this research paper, we aim to tackle possible problems faced by autonomous vehicles in rough terrain and adjust existing capabilities to the various possible situations that could be met. In this research paper, we have primarily focused on keeping India as a terrain location. As you see, it isn't easy to tact with things permitted while driving autonomous vehicles. The items are as follows: -

i. The high number of potholes in India poses a problem of cars slipping or losing balance, leading to accidents.

ii. With the high prevalence of wildlife in India comes the problem of animals coming onto the road, leading to increased accident rates.

iii. In Indian traffic, there are many issues, like detecting the traffic signs on the board due to obscurity. In such cases, the recognition would fail and lead to fatalities.

2. Literature Review

According to the publication [1], the research aims to develop a self-driving automobile that uses a CNN model to make decisions based on picture input from the camera. The amount of training data and the quality of the object detection model are directly linked to the accuracy and efficiency of a self-driving automobile. In the realm of transportation, this concept asks for a more modern and secure future for all residents.

In paper [2], one approach for self-governing driving is presented under the stimulated condition: the methodologies use deep learning strategies and end-to-end figuring out how to imitate automobiles. The main frame of the driver cloning algorithm is the Nvidia neural network. This image comprises five convolution layers, one levelingS layer, and four fully linked layers. The

*** Corresponding Author:** aryan.aher19@vit.edu

steering angle is the outcome we get. The usage of autonomous mode results in successful autonomous driving over a preset stimulation path, with the model trained using fewer data sets.

The paper [3] presented a deep imitative reinforcement learning (DIRL) framework to train end-to-end driving strategies to accomplish vision-based autonomous automobile racing. They combined IL and RL, using IL to initialize the policy and model-based RL to enhance it further by interacting with an uncertain-aware world model.

The application of the CNN deep learning algorithm for recognizing the surrounding environment in producing the automatic navigation required for autonomous cars is discussed in the paper [4]. In an environmental simulation using the self-driving car simulator, the suggested approach of autonomous cars employing CNN deep learning can operate smoothly without error and is highly stable without oscillation.

[5] in the paper, the project's goal is to contribute to this study by developing a driving simulator for a device that can recognize speed limit signs and make decisions that make driving more comfortable and safer. This research proposes a Yolo-based approach to traffic sign identification in the Clara Stimulator. This project required using a real-time CNN to detect and recognize CARLA speed signals. The car was connected to an RGB camera sensor every five frames, which collected environmental data. Animal detection systems aim to avoid accidents caused by animal-vehicle collisions. Humans are killed, injured, and their property is damaged. Animal Detection Using Template Matching Algorithm

In this work [6], several object detection techniques were reviewed. Regarding efficiency, the suggested system has a low false positive and false negative rate. Matching Templates Template matching is a technique for recognizing tiny picture areas that should correspond to the template image. To achieve template matching, normalized cross-correlation is implemented. Cross-correlation in signal processing is a measure of similarity between two waveforms as a time-slack component applied to one waveform. This is also known as the sliding point product or/and sliding inner product. Template matching is typically used to search a long-duration signal for an identifiable characteristic. The template may change owing to lighting and exposure conditions for applications utilizing image processing techniques to determine a picture's brightness. Hence the images must first be normalized. This is typically accomplished at each stage by removing the mean and dividing by the standard deviation. We addressed the feature-based template matching approach utilizing NCC in this study.

The proposed system in the paper uses rotation, scale, translation, and illumination invariant properties to strengthen the system. In this study, the traffic sign is identified using SURF features-based recognition. To match the extracted features of the Indian Traffic Sign Data (ITSD) base with the extracted features from the annotated region of an acquired image, surf features are extracted. Due to its speed and reliability, the SURF algorithm is used.

Koch and Brilakis [7] proposed a method that uses a histogram shape-based threshold to separate defect and non-defect regions in an image. Based on a perspective view, the authors estimate that a pothole's shape is approximately elliptical. The authors stress the importance of using machine learning in future research.

In 2017, a study in Taoyuan, Taiwan, used a data analytic approach that included correlation and regression analysis; [8] the results showed that regions with a high frequency of road potholes had a higher rate of traffic accidents. Potholes caused irrevocable damage to pizzas during delivery. Therefore, one of the top pizza businesses in the United States gave a special grant to correct them at a few sites in 2018 [9].

3. Methods

3.1. CNN

[10] In computer vision, convolutional neural networks (CNN) are now widely used. CNN is well-liked because of its consistent, practical outcomes in object identification and recognition. A group of separate filters makes up the convolution layer. The filter covers the entire image, and the dot product is calculated between the filter and various portions of the input image. Feature maps are produced after each filter has been separately convolved with the picture. Deriving a feature map has many applications, one of which is shrinking the size of the image while maintaining its semantic content.

3.2. Neural Network

[12] The photographs are given to the model as input. The model is fed photos of various sizes (note dimension) using OpenCV. The model's first layer is a convolution layer with 32 filters of the same size and extent (3x3). This layer is followed by a 20 percent dropout layer, which prevents the model from becoming overfitting. Next, a 64-filter convolution layer with a 3x3 dimension is added. In this research, we demonstrate a self-driving vehicle that uses monocular vision and a CNN model to make decisions based on picture input from the camera. Accuracy and efficiency for a driverless automobile are directly correlated with the volume of training data and the caliber of the object detection model.

3.3. Deep Reinforcement Learning

[12] Deep reinforcement learning combines artificial neural networks with reinforcement learning architectures

to enable software-defined agents to learn the best possible actions in a virtual environment and achieve their own goals. Deep Reinforcement Learning (DRL) is used to solve a variety of challenges, including Example: Recently complex board games and computer games. However, using DRL to solve actual robotics tasks is more complicated. The preferred approach is to train the agent in the simulator and transfer it to the real world. However, simulator-trained models tend to perform poorly in real-world environments due to the differences.

4. Problem Statement

In countries across the world, autonomous vehicles have become commonplace but are yet to reach most of the world and still have reliability problems on a large scale. There are still accidents caused by autonomous vehicles that constantly question such vehicles' safety, with terrain that is not always good and road and lane detection which may be more complicated than in most cosmopolitan cities.

Concerning India in mind, there are a lot of possible issues that could be faced by autonomous cars, such as the sign boards covered by dirt and incomplete lane lines along with twisted roads which can lead to confusion in the decision making which could lead to an error of considering the unfinished lane as a broken lane. There is also a need for improved object detection methodology to detect even motorcycles in high quantities in India. Autonomous vehicles should also be able to do pothole recognition which could also cause imbalance leading to accidents.

5. Objectives

Our project aims to provide a solution to possible problems that could be faced by autonomous vehicles in difficult situations and terrains, as well as enhance the existing features. The project focuses on using computer vision to implement various object detection algorithms to detect traffic signs, advanced lane detection, and obstacles along the road. Also, to manage the car’s speed when certain things come across the vehicle, it should reduce its speed and make decisions accordingly while driving.

5.1. Lane Detection

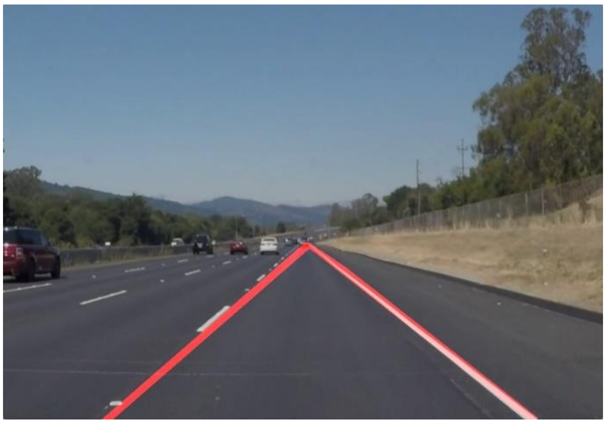


Figure 1. Lane detection

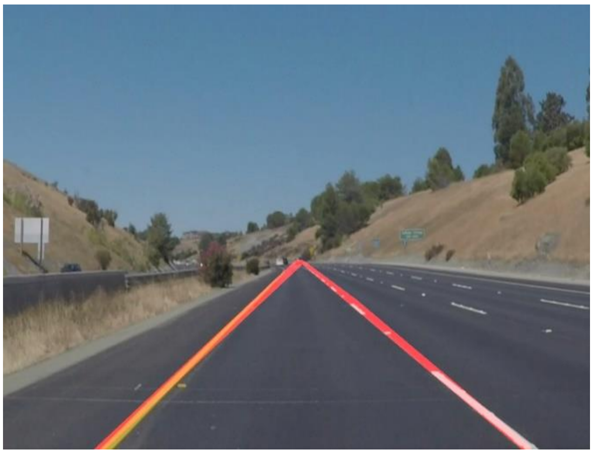


Figure 2. Lane detection

Because most vehicle road accidents occur due to the driver missing the vehicle path, safety is the primary goal of all road lane detecting systems. As a result, various vision-based road identification algorithms have been created to avoid vehicle collisions. A horizontal straight line is drawn to detect a lane that crosses the extended section at red locations, represented by red circles. The points are close and clustered in a group based on their distances.

5.2. Object Detection

They detect objects on the road. The three critical sensors utilized by self-driving cars work in tandem, much like the eyes and brain of a human. These sensors include cameras, radar, and lidar. When utilized simultaneously, they give the automobile a good picture of its surroundings. They help the vehicle determine the position, speed, and 3D forms of objects in its surroundings.

Object detection is a computer vision job that is utilized in a variety of consumer applications, such as surveillance and security systems, mobile text recognition, and illness diagnosis with MRI/CT scans. Object detection is a critical component of autonomous driving. Autonomous vehicles rely on the perception of their environment to enable safe and reliable driving. Object detection algorithms are used by this perceptual system to precisely determine things in the vehicle's vicinity, such as pedestrians, autos, traffic signs, and obstacles. Deep learning-based object detectors are crucial for detecting and localizing these things in real time. This article discusses cutting-edge object detectors and open issues for their integration into self-driving automobiles.

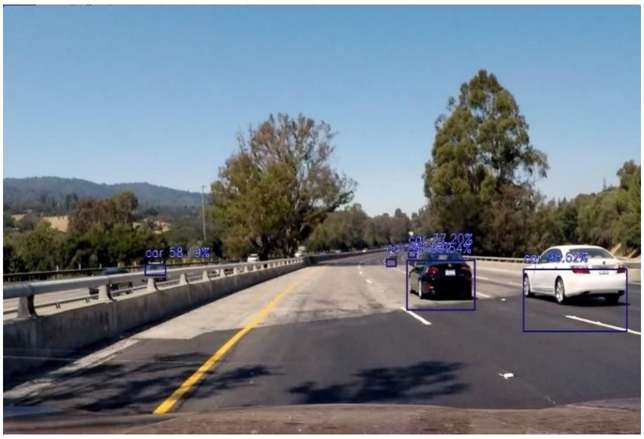


Figure 3. Object detection

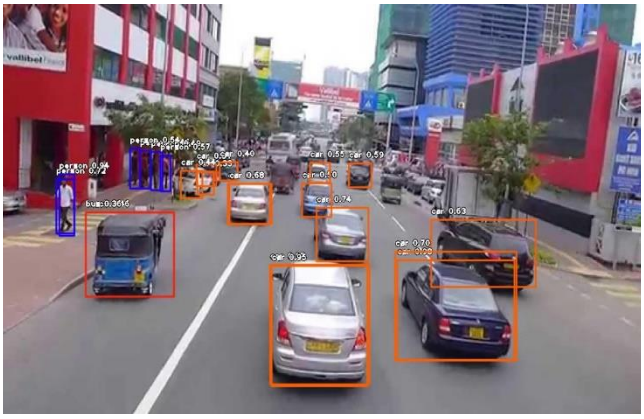


Figure 4. Object detection

5.3. Pothole Detection

Figure 4. Pothole on a road

The goal is to predict if there will be potholes in a certain number of frames. Detects road depressions using a live video feed processed by the CNN model. The video will be converted to a specific number of frames. After that, all images are preprocessed. Image pre-processing involves converting all images from color to grayscale (to reduce processing power) and resizing all images to the same size. H. 300 x 300 pixels, produce the output value corresponding to each image from the dataset used for training.

All the images from the dataset are processed and divided into training and testing datasets.

The processed image is passed to a CNN model for pothole detection. The CNN model is a sequential model having two convolutional layers with Relu as an activation function followed by an average pooling layer.

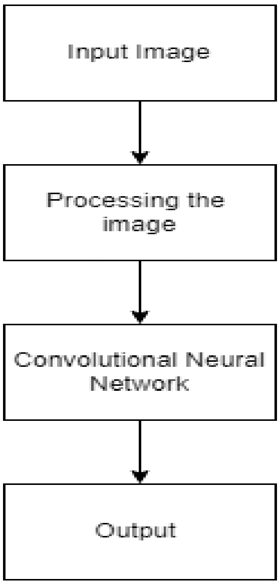


Figure 5. Flow for pothole detection

5.4. Signboard Detection

To create a system that can detect and recognize text and symbols on traffic panels based on street-level pictures. On the roadside, there are numerous text-based traffic signboards. It takes much work to capture all the signboards manually. Most of the current Automatic Signboard Recognition Systems (ASRs) are based on symbols. The need is to conduct additional research on text-based ASR.

6. Results and conclusion

The project shows an approach to developing a system for autonomous vehicles to operate on Indian roads and terrain. The fundamental idea behind the project is to create an autonomous car that can sense its environment and move without human input. This paper proposes Car automation, which is accomplished by recognizing the road, signals, obstacles, and stop signs, responding and making decisions such as changing the course of a vehicle, stopping at red signals, and moving on green calls using machine learning techniques.

6.1. Future Scope

Based on the planning of our project, there can be some recommendations to improve the features of the system to make it more users friendly and effective:

i. It has a possibility of further enhancement by using hardware to implement basic models.

ii. It can be further enhanced by increasing the model's accuracy.

iii. Decision-making can be increased and managed according to the new information and object detection.

Acknowledgment

We want to thank and respect the supporters of this project. We acknowledge the efforts of those who have contributed significantly to our project. We express our gratitude and appreciation to Professor ST Patil for his valuable time and guidance during this project. We express our sincere and deep gratitude to the Vishwakarma Institute of Technology, Pune, for allowing us to work on such a fascinating project.

References

[1] N. Sanil, P. A. N. venkat, V. Rakesh, R. Mallapur and M. R. Ahmed, "Deep Learning Techniques for Obstacle Detection and Avoidance in Driverless Cars," 2020 International Conference on Artificial Intelligence and Signal Processing (AISP), 2020, pp. 1-4, doi: 10.1109/AISP48273.2020.9073155.

[2] Chirag Sharma , S. Bharathiraja , G. Anusooya, "Self Driving Car using Deep Learning Technique", International Journal of Engineering Research & Technology (IJERT) Volume 09, Issue 06 (June 2020).

[3] P. Cai, H. Wang, H. Huang, Y. Liu and M. Liu, "Vision-Based Autonomous Car Racing Using Deep Imitative Reinforcement Learning," in *IEEE Robotics and Automation Letters*, vol. 6, no. 4, pp. 7262-7269, Oct. 2021, doi: 10.1109/LRA.2021.3097345.

[4] I. Sonata, Y. Heryadi, L. Lukas, και A. Wibowo, 'Autonomous car using CNN deep learning algorithm', *Journal of Physics: Conference Series*, τ. 1869, τχ. 1, σ. 012071, Απριλίου 2021.

[5] Y. Valeja, S. Pathare, D. Patel and M. Pawar, "Traffic Sign Detection using Clara and Yolo in Python," *2021 7th International Conference on Advanced Computing and Communication Systems (ICACCS)*, 2021, pp. 367-371, doi: 10.1109/ICACCS51430.2021.9442065.

[6] N. Banupriya, S. Saranya, R. Swaminathan, S. Harikumar, και S. Palanisamy, 'Animal detection using deep learning algorithm', J. Crit. Rev, τ. 7, τχ. 1, σσ. 434–439, 2020.

[7] C. Koch and I. Brilakis, "Pothole detection in asphalt pavement images," *Advanced Engineering Informatics*, 01-Feb-2011. [Online].

[8] B. -H. Lin and S. -F. Tseng, "A predictive analysis of citizen hotlines 1999 and traffic accidents: A case study of Taoyuan city," 2017 IEEE International Conference on Big Data and Smart Computing (BigComp), 2017, pp. 374-376, doi: 10.1109/BIGCOMP.2017.7881696.

[9] D. O'Carroll, "For the love of pizza, Domino's is now fixing potholes in roads," *Stuff*, 12-Jun-2018. [Online]. Available: https://www.stuff.co.nz/motoring/104643123/for-the-love-of-pizza-dominos-is-now-fixing-potholes-in-roads. [Accessed: 30-Oct-2022].

[10] S. Uchida, S. Ide, B. K. Iwana and A. Zhu, "A Further Step to Perfect Accuracy by Training CNN with Larger Data," 2016 15th International Conference on Frontiers in Handwriting Recognition (ICFHR), 2016, pp. 405-410, doi: 10.1109/ICFHR.2016.0082.

[11] M. Egmont-Petersen, D. de Ridder, and H. Handels, "Image processing with neural networks-A Review," *Pattern Recognition*, 19-Jun-2002. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0031320301001789. [Accessed: 30-Oct-2022].

[12] Y. Li, "Reinforcement learning in practice: Opportunities and challenges," *arXiv.org*, 22-Apr-2022. [Online]. Available: https://arxiv.org/abs/2202.11296v2. [Accessed: 30-Oct-2022].

Intelligent Methods in Engineering Sciences 1(1): 018-022, 2022

IMIENS

INTELLIGENT METHODS
IN ENGINEERING SCIENCES

https://www.imiens.org

International Open Access

December, 2022

e-ISSN 2979-9236

Research Article

https://doi.org/10.58190/imiens.2022.4

Transfer Learning-Based Benchmarking Study for Diagnosis of COVID-19 from Lung CT Scans

Mücahit Akar [a], *Kadir Sabanci* [b], *Muhammet Fatih Aslan* [b,*]

[a] *MVD Machinery, Konya, Turkiye*
[b] *Electrical and Electronics Engineering, Karamanoglu Mehmetbey University, Karaman, Turkiye*

ARTICLE INFO

Article history:
Received 11 November 2022
Accepted 27 November 2022

Keywords:
COVID-19
convolutional neural networks
CT scan
deep learning
fine-tuning
transfer learning

ABSTRACT

The virus known as Severe Acute Respiratory Syndrome Coronavirus 2 (SARS-CoV-2) or Coronavirus Disease 2019 (COVID-19), which emerged from the city of Wuhan in the People's Republic of China, has affected the whole world. This disease, which is categorized as an epidemic disease, continues to increase despite the various measures taken. It is aimed to reduce death and infected people rates with vaccination studies, inspection and early diagnosis. On the other hand, new types of coronavirus cases are emerging and people are kept under surveillance to prevent the spread of the virus. By keeping the infected people under quarantine, the transmission of the epidemic to more people is prevented. For this reason, early diagnosis kits and tests are vital. Today, various abnormalities are detected by specialists thanks to medical imaging tools. On the other hand, this process is performed on medical images using image processing techniques. Thanks to methods such as image classification, image segmentation, image quantification and various operations such as object detection, localization and quantitative analysis on the object are performed. In this study, it is aimed to detect COVID-19 on lung CT scan images with deep learning methods. CNN-based state-of-art deep learning models, which were pre-trained with millions of images and applied transfer learning method for a similar problem, were used in this study. This process was performed by choosing VGG19, ResNet152 and MobileNetV2 models and the results were compared. According to the performance criteria, validation accuracy of 93.53%, 95% and 87.28% was obtained from VGG19, ResNet152 and MobileNetV2 models, respectively. These results show that these models give good results for the detection of COVID-19 from lung CT scan images.

1. Introduction

On December, 31, 2019, a new viral pneumonia outbreak has emerged in the city of Wuhan, the People's Republic of China. After the report published on the subject, World Health Organization (WHO) learned about the new coronavirus case for the first time. The disease caused by the new coronavirus known as SARS-CoV-2/COVID-19 [1]. According to the WHO report, this disease can be transmitted in many different ways, such as liquid particles coming out of their mouths and noses when talking, sneezing, coughing, yawning among people [2]. Thus, the virus spread rapidly and still does. According to the WHO report dated September 1, 2021, a total of 4.517.240 deaths and 217.558.771 confirmed cases were reported [3]. Therefore, early detection of the corona virus and taking measures such as quarantine against it and thus keeping the virus under control before it spreads to more people play a critical role. In this context, many diagnostic and detection methods are used. Molecular tests such as Polymerase Chain Reaction (PCR), isothermal nucleic acid amplification, antigen tests, antibody tests are just a few of them [4]. PCR is one of the frequently used molecular test methods to diagnose virus since the beginning of COVID-19. It is still used but has some limitations. Some these make this method expensive because the test method is complicated, delays in test reports, specialist laboratory personnel and special equipment are required [5]. This situation has led researchers to diagnostic kits with high detection success and less expensive, and in addition to all these, medical imaging methods have begun to come to the fore. T. Ai, Z. Yang and et al. [6] investigated the correlations between Chest CT and Reverse-Transcription Polymerase Chain Reaction (RT-PCR) tests in their study and reported that Chest CTs have high sensitivity for COVID-19 diagnosis and can be considered as the primary tool for current COVID-19 diagnosis.

* **Corresponding Author:** mfatihaslan@kmu.edu.tr

Medical imaging provides many advantages for the diagnosis of COVID-19. The data collected thanks to the medical images make facilitated the diagnosis with image processing techniques, computer vision, deep learning and various artificial intelligence applications such as CT image analysis [7], automatic coronavirus disease detection using X-Ray images [8], lung infection quantification of COVID-19 [9] and multi-class segmentation of COVID-19 chest CT images [10].

Convolutional neural networks (CNNs) are very successful networks in making meaning from image. Computer vision problems with CNN models used in various fields give very accurate results. There are various successful state-of-art CNN models such as ResNet [11], Xception [12], AlexNet [13], VGG [11] and many more [14].

In this study, research was carried out on the detection of COVID-19 disease on dataset which consists of lung CT scan images with the transfer learning method by using some state-of-art models.

2. Materials and Methods

Dataset, pre-processing, data augmentation, transfer learning and fine-tuning phases will be explained in this section.

2.1. CT Scan Dataset

In this study, the SARS-CoV-2 CT scan dataset [15] was used. The dataset consists of 2482 CT scan images collected from 1252 positive cases (COVID-19) and 1230 negative cases (non-COVID-19). This dataset consists of data from real patients collected from hospitals in Sao Paulo, Brazil. Some of the samples from dataset are shown in Fig. 1.

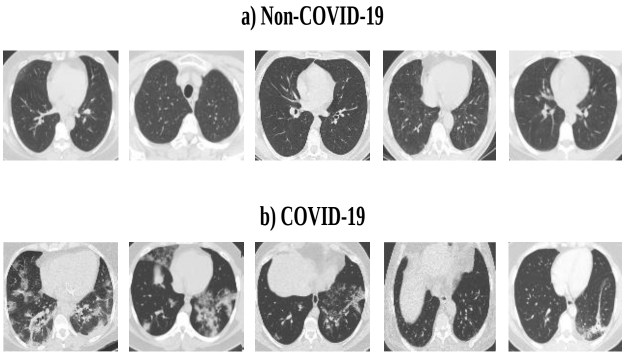


Figure 1. Samples from dataset [15].

2.2. Preprocessing

At this stage, the images in the CT Scan dataset serve to clean up the parts of the model that are not needed for better results. Here, thresholding has been applied on the images using Otsu's method [16] and both unnecessary values have been eliminated and background and foreground have been separated from each other.

In addition, all the images in the dataset were converted to a single format (.png) and the input size determined during the model training phase was adjusted to a fixed size (e.g., 224 x 224 x 3) (see Figure 2)

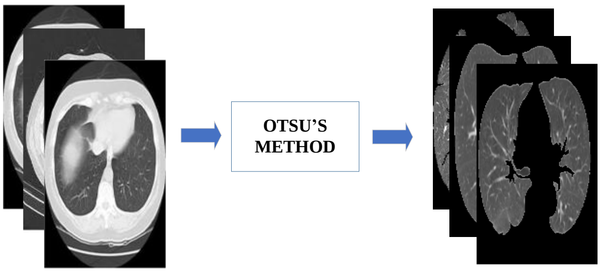


Figure 2. Otsu's method (thresholding)

2.3. Data Augmentation

Data augmentation is a method of increasing the amount of data by applying various changes (such as rotation, flipping, zooming, etc.) to the data in the existing dataset [17]. In this way, it helps to reduce the overfitting encountered during model training and the number of data has been increased. In this study, flipping and rotation processes were applied on images, both horizontally and vertically, as shown in Figure 3.

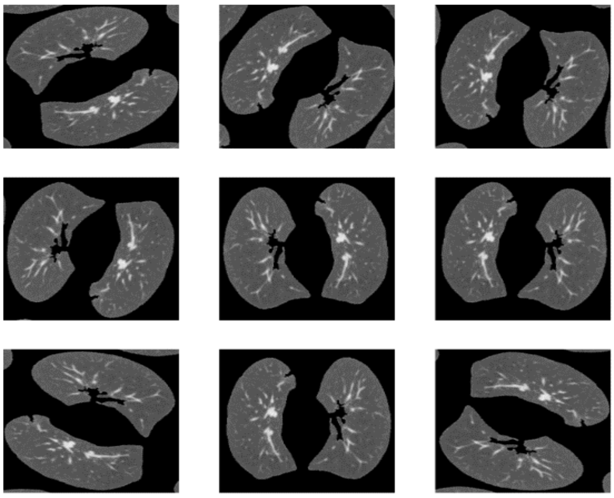


Figure 3. Data augmentation method shown on an image in the dataset (both rotated and flipped).

2.4. Transfer Learning

Transfer learning is a research problem in which the knowledge gained for one problem is used for another related problem. This allows us to achieve higher success rates with fewer datasets and faster learning for the model.

In this study, transfer learning was applied on the dataset we have in order to achieve faster learning and better results. Pre-trained CNN models were used to perform this operation. Some of these models used in this study are as follows; ResNet152 [18], VGG19 [11] and MobileNetV2 [19]. With these models, the binary class problem (COVID-19 and non-COVID-19) has been solved on the dataset consisting of lung CT scan images. The hyperparameter values selected for the created models are shown in Table 1.

Table 1. Some of Hyperparameters of Resnet152, Vgg19 and Mobilenetv2 Models

Hyperparameters	Values
Learning Rate	0.0001
Optimizer	Adam
Batch-Size	16
Epochs	30

During the model training, the binary cross entropy value for the loss function and the 0.3 dropout value for the regularization were selected. In addition, the model is set to interrupt the epoch automatically when the validation loss value falls below a certain value.

After the model training are completed, the graphs showing the accuracy and loss values of the CNN models are shown in Fig. 4, Fig. 5 and Fig. 6.

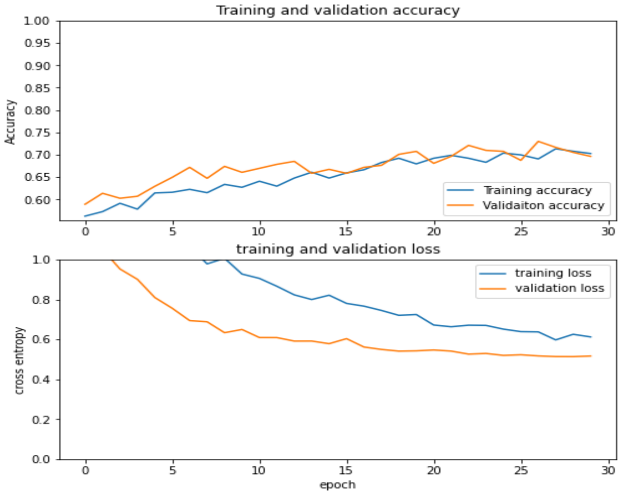


Figure 4. Accuracy and loss graph of VGG19 model for training and validation.

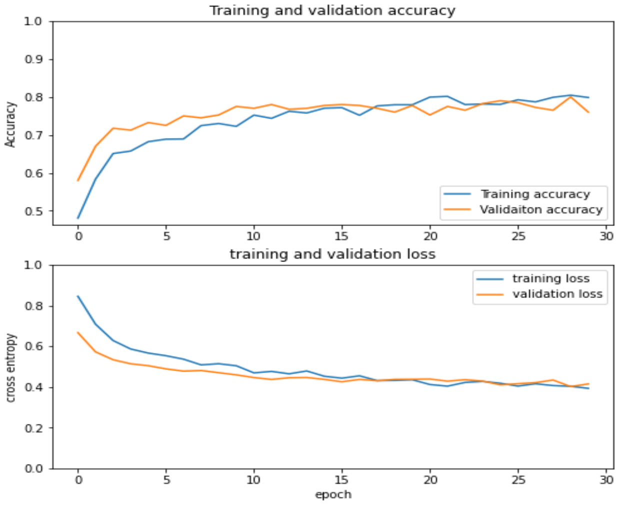


Figure 5. Accuracy and loss graph of ResNet152 model for training and validation.

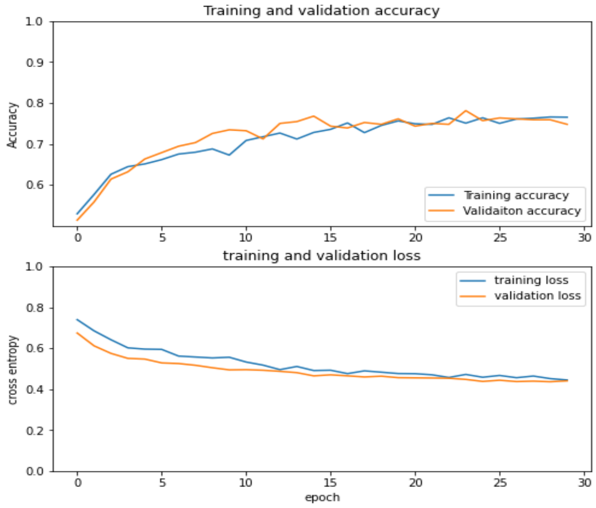


Figure 6. Accuracy and loss graph of MobileNetV2 model for training and validation

2.5. Fine Tuning

Fine-tuning is a process that takes a pre-trained model for a particular task and then adjusts or modifies the model to perform a similar task. In this way, the performance values of the model can be increased even more. Computer vision, face recognition, object detection, action and activity detection are being developed thanks to CNNs [20]. Therefore, CNN models are frequently used in transfer learning. Simply a CNN model consists of two main parts. These consist of the convolutional base part that is responsible for extracting features and the classifier part that provides the classification [21, 22]. It is possible to train models in three different ways, as seen in Fig. 7, in the convolutional base and classifier sections. The first is to train the entire model. The second is to freeze some layers in the model and train the remaining layers. The third is to freeze only the convolutional base part of the model and train the classification part.

3. Results

In this study, only the classifier part is trained during the model training before fine-tuning phase. Afterwards, some layers in the convolutional base part of the model were unfreeze and the model were retrained to perform fine-tuning.

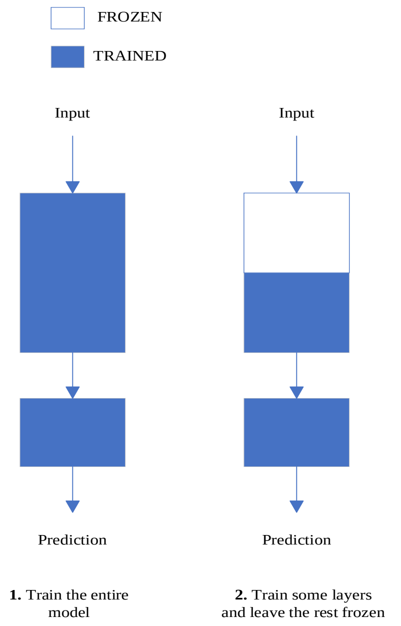

FROZEN
TRAINED
Input
Input
Input
Prediction
Prediction
Prediction
1. Train the entire model
2. Train some layers and leave the rest frozen
3. Train the only classifier

Figure 7. Fine-tuning strategies. 1. Train the entire model. 2. Train some layers and leave the rest frozen. 3. Train the only classifier.

In the fine-tuning phase, the epoch value is selected as 30 for retraining the networks, but for early stop operation, the training will automatically interrupt the iteration when the validation loss (val_loss) value falls below a certain value. In addition, at this stage, the learning rate (learning_rate) value will automatically and gradually decrease itself to obtain a more robust validation accuracy (val_accuracy). The frozen layers were selected as the first 8 layers for VGG19 model, the first 250 layers for ResNet152 model, and finally the first 90 layers for MobileNetV2 model.

After fine-tuning, the model results are shown respectively in the graph in Fig. 8, Fig. 9, Fig. 10 depending on the accuracy and loss values and compared with the values before fine-tuning phase.

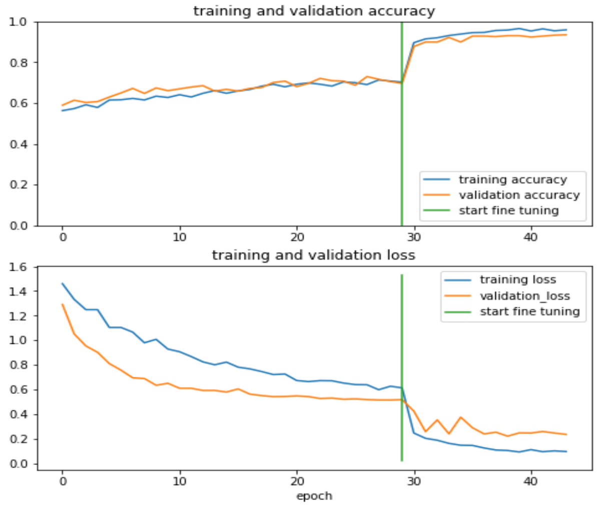


Figure 1. Accuracy and loss graphs of VGG19 model for training and validation after fine-tuning.

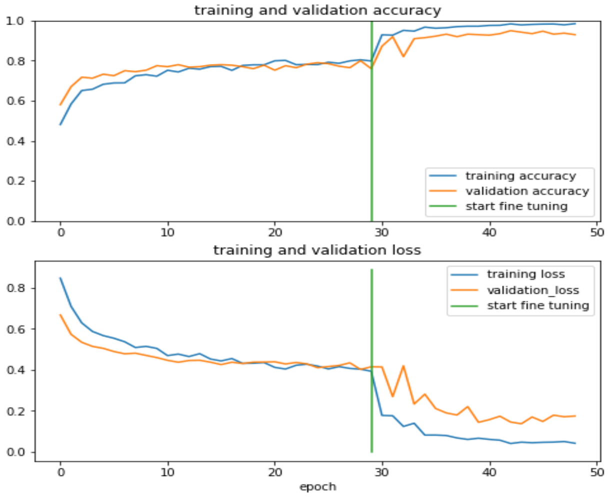


Figure 2. Accuracy and loss graphs of ResNet152 model for training and validation after fine-tuning.

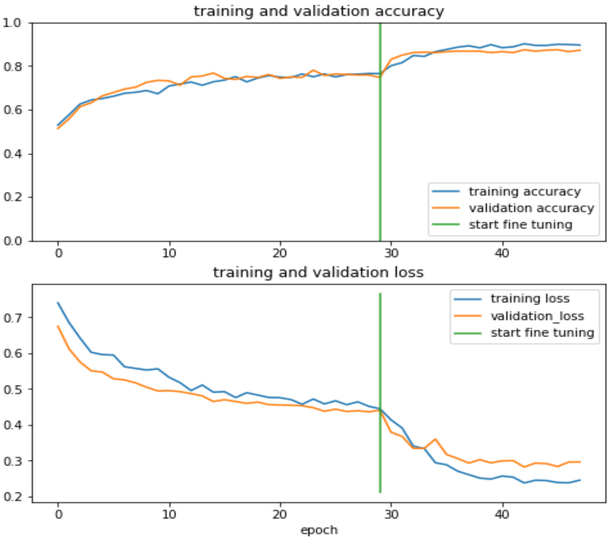


Figure 3. Accuracy and loss graphs of MobileNetV2 model for training and validation after fine-tuning.

Pre-trained CNN-based neural network models such as VGG19, ResNet152 and MobileNetV2, which were tried in this study, and then fine-tuning stage successfully classified the binary class problem (COVID-19 and non-COVID-19 classes). Obtained performance metrics from the CNN models are shown in Table 2. According to the results, ResNet152 obtained the best validation accuracy. Thus, the images reserved for the validation in the dataset were detected with 95% validation accuracy.

Table 2 Performance Metrics of CNN Models

Models	Accuracy	Loss	Validation Accuracy	Validation Loss
VGG19	0.9792	0.1815	0.9353	0.2343
ResNet152	0.9583	0.0397	0.95	0.1753
MobileNetV2	0.8838	0.2960	0.8728	0.2967

4. Discussions

The results obtained in the study show that COVID-19 can be detected quickly and without overfitting. CNN models in different architectures can provide different accuracy in COVID-19 detection. Different pre-processing methods can be tried to improve the dataset to be trained for the model. In this way, model success can be increased.

5. Conclusion

The results obtained from the trained VGG19, ResNet152 and MobileNetV2 models are 93.53%, 95% and 87.28%, respectively. Among these models, the ResNet152 model achieved higher validation accuracy than the other two models. In order to obtain higher results, different CNN models can be used, more data can be collected and the model can achieve higher results, and different pre-processing, data augmentation and hyperparameter selection can be made. Thus, the performance criteria obtained from the models can be compared with other models.

References

[1] "Coronavirus disease (COVID-19)." https://www.who.int/emergencies/diseases/novel-coronavirus-2019/question-and-answers-hub/q-a-detail/coronavirus-disease-covid-19 (accessed Sep. 02, 2021).

[2] "Coronavirus disease (COVID-19): How is it transmitted?" https://www.who.int/news-room/q-a-detail/coronavirus-disease-covid-19-how-is-it-transmitted (accessed Sep. 02, 2021).

[3] "WHO Coronavirus (COVID-19) Dashboard | WHO Coronavirus (COVID-19) Dashboard With Vaccination Data." https://covid19.who.int/ (accessed Sep. 02, 2021).

[4] "COVID-19 testing - Wikipedia." https://en.wikipedia.org/wiki/COVID-19_testing (accessed Sep. 03, 2021).

[5] "COVID-19 diagnostic testing: advantages and disadvantages." https://www.myamericannurse.com/covid-19-diagnostic-testing/ (accessed Sep. 03, 2021).

[6] T. Ai et al., "Correlation of Chest CT and RT-PCR Testing for Coronavirus Disease 2019 (COVID-19) in China: A Report of 1014 Cases," Radiology, vol. 296, no. 2, pp. E32–E40, 2020, doi: 10.1148/radiol.2020200642.

[7] O. Gozes, M. Frid, H. Greenspan, and D. Patrick, "Rapid AI Development Cycle for the Coronavirus (COVID-19) Pandemic : Initial Results for Automated Detection & Patient Monitoring using Deep Learning CT Image Analysis Article Type : Authors : Summary Statement : Key Results : List of abbreviati," arXiv:2003.05037, 2020, [Online]. Available: https://arxiv.org/ftp/arxiv/papers/2003/2003.05037.pdf.

[8] A. Narin, C. Kaya, and Z. Pamuk, "Automatic detection of coronavirus disease (COVID-19) using X-ray images and deep convolutional neural networks," Pattern Anal. Appl., vol. 24, no. 3, pp. 1207–1220, 2021, doi: 10.1007/s10044-021-00984-y.

[9] F. Shan et al., "Lung infection quantification of COVID-19 in CT images with deep learning," arxiv.org, Accessed: Sep. 03, 2021. [Online]. Available: https://arxiv.org/abs/2003.04655.

[10] X. Chen, L. Yao, and Y. Zhang, "Residual Attention U-Net for Automated Multi-Class Segmentation of COVID-19 Chest CT Images," vol. 14, no. 8, pp. 1–7, 2020, [Online]. Available: http://arxiv.org/abs/2004.05645.

[11] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 3rd Int. Conf. Learn. Represent. ICLR 2015 - Conf. Track Proc., pp. 1–14, 2015.

[12] K. R. Avery et al., "Fatigue Behavior of Stainless Steel Sheet Specimens at Extremely High Temperatures," SAE Int. J. Mater. Manuf., vol. 7, no. 3, pp. 1251–1258, 2014, doi: 10.4271/2014-01-0975.

[13] T. F. Gonzalez, "Handbook of approximation algorithms and metaheuristics," Handb. Approx. Algorithms Metaheuristics, pp. 1–1432, 2007, doi: 10.1201/9781420010749.

[14] A. Khan, A. Sohail, U. Zahoora, and A. S. Qureshi, "A survey of the recent architectures of deep convolutional neural networks," Artif. Intell. Rev., vol. 53, no. 8, pp. 5455–5516, 2020, doi: 10.1007/s10462-020-09825-6.

[15] "SARS-COV-2 Ct-Scan Dataset | Kaggle." https://www.kaggle.com/plameneduardo/sarscov2-ctscan-dataset (accessed Sep. 10, 2021).

[16] "Otsu's method - Wikipedia." https://en.wikipedia.org/wiki/Otsu%27s_method (accessed Sep. 04, 2021).

[17] "Data augmentation - Wikipedia." https://en.wikipedia.org/wiki/Data_augmentation (accessed Sep. 04, 2021).

[18] V. Sangeetha and K. J. R. Prasad, "Syntheses of novel derivatives of 2-acetylfuro[2,3-a]carbazoles, benzo[1,2-b]-1,4-thiazepino[2,3-a]carbazoles and 1-acetyloxycarbazole-2-carbaldehydes," Indian J. Chem. - Sect. B Org. Med. Chem., vol. 45, no. 8, pp. 1951–1954, 2006, doi: 10.1002/chin.200650130.

[19] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L. C. Chen, "MobileNetV2: Inverted Residuals and Linear Bottlenecks," Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., pp. 4510–4520, 2018, doi: 10.1109/CVPR.2018.00474.

[20] A. Voulodimos, N. Doulamis, A. Doulamis, and E. Protopapadakis, "Deep Learning for Computer Vision : A Brief Review," vol. 2018, 2018.

[21] K. Sabanci., M.F. Aslan., E. Ropelewska., M.F. Unlersen, A. Durdu. "A Novel Convolutional-Recurrent Hybrid Network for Sunn Pest–Damaged Wheat Grain Detection," Food Anal. Methods pp. 1748–1760, 2022. https://doi.org/10.1007/s12161-022-02251-0

[22] M. Koklu, M. F. Unlersen, I. A. Ozkan, M. F. Aslan, and K. Sabanci, "A CNN-SVM study based on selected deep features for grapevine leaves classification," Measurement, vol. 188, p. 110425, 2022/01/01/ 2022,

Intelligent Methods in Engineering Sciences 1(1): 023-026, 2022

IMIENS

INTELLIGENT METHODS
IN ENGINEERING SCIENCES

https://www.imiens.org

International
Open Access

December, 2022

e-ISSN 2979-9236

Research Article

https://doi.org/10.58190/imiens.2022.5

A Voice Recognition Based Game Design for More Accurate Pronunciation of English

Emre Avuçlu [a,*], ***Murat Köklü*** [b]

[a] *Department of Software Engineering, Aksaray University, Aksaray, Turkiye*
[b] *Department of Computer Engineering, Technology Faculty, Selçuk University, Turkiye*

ARTICLE INFO

Article history:
Received 16 November 2022
Accepted 29 November 2022

Keywords:
English pronunciation
Game programming
Technology
Voice recognition

ABSTRACT

The development of technology has made it considerably easier for people to meet a number of needs. Without technology, it is no longer possible to run certain applications. Nowadays, in many countries, the process of speaking English poses some problems in terms of different situations, such as allocating time for people. In this study, a game-based application was developed to help a person anywhere in the world pronounce English better. The application was implemented in C# programming language. The Speech.dll library was used to introduce voice commands to the system and to perform other necessary operations. Voice commands can be sent by the user via the wireless headset from anywhere in the shooting area. There is no need to wait at the computer while using the application because the developed application gives voice feedback to the user that it is right or wrong after the voice recognition process. In this application letter, word or sentence exercises can be done. The program aims to improve the level of pronunciation of people who want to improve their English-speaking skills.

1. Introduction

The use of technology has become an indispensable part of today. People have constantly developed technology to benefit some applications for their own benefit. Controlling any application with a computer is very easy thanks to the software. Today, some arrangements are made with technology so that people can live more comfortably. It is possible to see examples of this in every field. From medicine to the automotive industry, computerized control, diagnostics, etc., in almost every field, are done by computerized control software. Thus, it is easier to get fast, reliable, quality results. Studies on voice, speech, speaker recognition are as follows:

Today, technology can be defined as an effective part of directing discoveries by using data sharing in the most effective way [1]. Different voice recognition algorithms have been used on MATLAB, they have used "Open", "Close", "Start" and "Stop" instruction sets [2-3]. Using PIC 16F876, attempts were made to recognize voice recognition commands under 4 different conditions [4]. It has been tried by setting up different algorithms on a phone simulation. It has been observed that the results obtained vary according to the way the sound is pronounced [5]. A voice recognition-based security system has been developed [6]. A structure named EllaVoice has been developed by using improved dynamic time warping algorithms [7]. A program has been developed for NASA by using Mel frequency Kepstrum coefficients algorithm and dynamic time warping and hidden markov model algorithms separately [8]. A remote-controlled robot design has been made using the RS 232 connection with voice command [9]. More than 80% success has been achieved in voice recognition on the letters "a", "e" and "i" [10]. Separate tests were conducted on male and female users [11] and a voice recognition system was used [12].

A speech recognition system independent of text and speaker has been developed on the Turkish language using Artificial Intelligence techniques [13]. Turkish word recognition system has been developed [14]. By using Artificial Neural Networks and Dynamic Time Warping algorithms separately, home applications working with voice command were made [15]. A successful recognition rate was found for 10 people in the simulation environment on Matlab [16]. Mobile vehicle design was made using different voice recognition algorithms [17]. They applied music and speech recognition [18]. Class Non-Principal Component Analysis was compared with Vector Quantization algorithm [19]. They used different voice recognition algorithms [20]. Successful results were obtained in the study that performed 40 commands [21]. The numbers uttered between 0 and 9 were first perceived

* **Corresponding Author:** emreavuclu@aksaray.edu.tr

by the sound detection system independently of time, and then sound processing techniques were used [22]. It was controlled by voice commands of a remote-control car [23]. It has been tried to determine the English pronunciation of the numbers 0-9 [24-25]. An attempt was made to control a submarine model moving underwater with voice commands [26]. In the study, based on the calls made to the call center, 5 different emotion detections were tried to be tested by 30 people and 70% success was achieved [27]. The success rate of the system, which can make simultaneous comparisons without registration, was 67.5% [28]. A voice-controlled robot was designed using Artificial Neural Network algorithms [29]. In the literature, different studies have been carried out on the development of voice recognition and English pronunciation [30-31].

In this study, a voice recognition-based application was developed to improve English pronunciation. No matter where the purpose of this application is in the world, a program has been written that can be used by people of all ages to improve their English speaking.

2. Materials and Methods

The design of the application made in this study consists of a series of stages. A number of processes are carried out in the process from the recognition of voice commands to the execution of the necessary actions. The flow chart of the developed application is as shown in Figure 1.

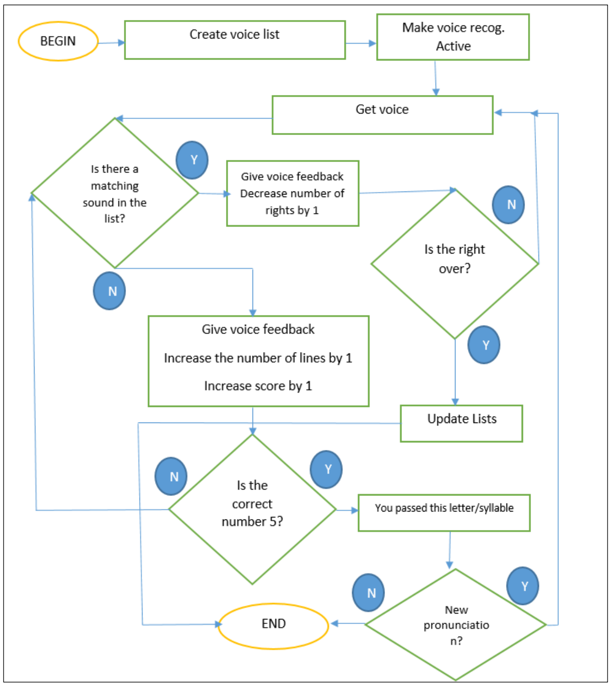


Figure 1. Flow chart of the application

The application made in this study is programmed with the C# programming language. Regarding voice recognition, the following libraries must be added to the system first. After installing the required SDK version on the computer, the definitions required for the voice recognition related engine to work are defined as shown in the code block below.

```
    SpeechSynthesizer SpeechSynth = new
SpeechSynthesizer();
    PromptBuilder Builder = new PromptBuilder();
    SpeechRecognitionEngine myRecognize = new
SpeechRecognitionEngine();
```

The process of defining which sounds the recognition engine will be sensitive to during the voice recognition process is defined as in the code block below. New definitions can be made to the list as letters or syllables.

```
    Choices SpeechList = new Choices();
    SpeechList.Culture = new
System.Globalization.CultureInfo("en-US");
    SpeechList.Add(new string[] { "one", "2", "3", "a", "b", "c",
"d", "e", "f", "g", "h", "i", "j", "k", "l", "m", "n", "o", "p", "q", "r",
"s", "t", "u", "v", "w", "x", "y", "z" });
     Grammar gr = new Grammar(new
GrammarBuilder(SpeechList));
```

After the necessary definitions are made, the sound begins to be heard from the outside. If the received sound is in our list, it returns the necessary answer to us. Otherwise, the catch block is executed and the system is waiting for the sound again.

```
        try
         {
          myRecognize.RequestRecognizerUpdate();
          myRecognize.LoadGrammar(gr);
          myRecognize.SpeechRecognized +=
sRecognize_SpeechRecognized;
          myRecognize.SetInputToDefaultAudioDevice();

myRecognize.RecognizeAsync(RecognizeMode.Multiple);

          }
         catch
          {
             pictureBox2.Visible = true;
             MessageBox.Show("error", e.ToString());
               return;
           }
```

In the developed application, first the letter or syllable to be practiced is selected. This process is illustrated in Figure 2.

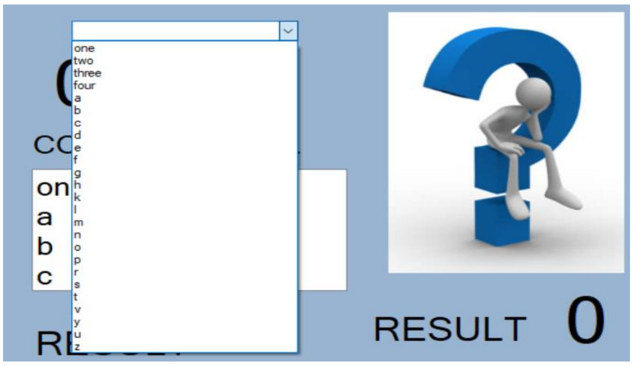


Figure 2. Word selection process in application

The selected letter or syllable is automatically set in the "Input Text" as shown in Figure 3.

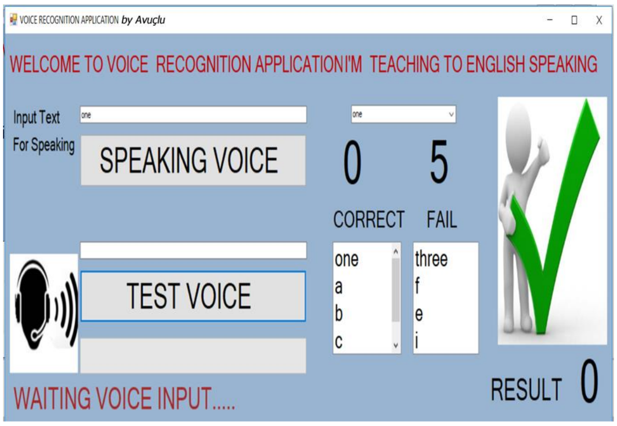

VOICE RECOGNITION APPLICATION by Avuçlu
WELCOME TO VOICE RECOGNITION APPLICATIONI'M TEACHING TO ENGLISH SPEAKING
Input Text
For Speaking
one
one
SPEAKING VOICE
0
5
CORRECT
FAIL
TEST VOICE
one
a
b
c
three
f
e
i
WAITING VOICE INPUT.....
RESULT
0

Figure 3. Pronunciation listening process

Before starting to pronounce this letter or syllable, the user listens to how this sound is produced by pressing the "Speaking voice" button. The code block that performs this operation is as follows.

```
    private void button1_Click(object sender, System.EventArgs e)
        {
            Builder.ClearContent();
            Builder.AppendText(textBox1.Text);
            SpeechSynth.Speak(Builder);
        }
```

After the listening process is finished, the “Test Voice” button is pressed. The word "One" begins to pronounce. If he pronounces the word "one" correctly 5 times in a row without making any mistakes, the word "one" is added to the "correct" list. If it is not pronounced correctly 5 times in a row, this time the word "one" is added to the "fail" list. The user earns points for each word included in the "correct" list.

A small high-pitched sound is given as a feedback to inform the user after the correct pronunciation. After mispronunciation, a different treble sound is given to the user as a feedback. So the user knows how to behave. After adding the necessary libraries for the audio feedback process, the following code block is run.

```
    axWindowsMediaPlayer1.URL = "C:\\ProgramFiles\\Sound\\correct.mp3";
    axWindowsMediaPlayer1.URL = "C:\\ProgramFiles\\Sound\\fail.mp3";
```

If the voice recognition engine cannot recognize the voice command given from the outside, it shows the voice it defines as close in the text box. Thus, the user is given both audio and visual feedback.

Correct and incorrectly pronounced words obtained during the trial process are shown in Table 1.

Table 1. Expressions tested in application

Trials	Correct	Fail
one	one	
a	a	
b	b	
c	c	
Three		Three
f		f
e		e
i		i

When we look at Table 1, it is seen that the words that are difficult to express are more in the list of incorrect expressions.

3. Conclusions

In this study, an application was developed for people anywhere in the world to pronounce English words and letters correctly. The Speech.dll library was used to perform the voice recognition process in the application developed in the C# programming language. In the feedback process, the AxInterop.WMPLib.dl library was used to run the necessary codes for voice commands. With the use of the developed application, children will reach a more permanent and impressive pronunciation level at an early age. Developed in a game style, the application will both interest children and help them improve their English pronunciation. The application has been tested on different children and adults and it has been observed that the pronunciation levels have improved. As a result, the English speaking pronunciation of each person who will use this application will increase.

References

[1] İnternet: Teknoloji, https://tr.wikipedia.org/wiki/Portal: Teknoloji, 2016.

[2] Karakaş, M., Computer Based Control Using Voice Input, Yüksek Lisans Tezi, Dokuz Eylül Üniversitesi, 2010.

[3] Muda, L., Begam, M., Elamvazuthi, I., "Voice Recognition Using Mel Frequency Cepstral Coefficient and Dynamic Time WarpingTechniques", Journal of Computing, Cilt 2, 2010.

[4] Fezari, M., Salah, M.B., "A Voice Command System for Autonomous Robots Guidance", IEEE AMC, 2006.

[5] Baygün, M.K., Yaldır, A.K., “Linear Predictive Coding ve Dynamic Time Warping Teknikleri Kullanılarak Ses Tanıma Sistemi Geliştirilmesi”, Pamukkale Üniversitesi, 2009.

[6] Bala, A., Kumar, A., Birla, N., "Voice Command Recognition System Based on MFCC and DTW", International Journal Of Engineering Science and Technology, Cilt 2, 7335-7342, 2010.

[7] Hrncar, M., "Voice Command Control for Mobile Robots", Department of Control and Information Systems Faculty of Electrical Engineering Univercity of Zilina, 2000.

[8] Price, J., Eydgahi, A., "Design of Matlab Based Automatic Speaker Recognition Systems", 9th International Conference on Engineering Education T4J-1, 2006.

[9] Zhizeng, L., Jinghing, Z., "Speech Recognition and Its Application in Voice-based Robot Control System", Proceedings Of International Conference On Intelligent Mechatronics, 2004.

[10] Öztürk, B., Çakar, T., Gerçek Zamanlı Ses Tanıma, Bitirme Projesi, İstanbul Üniversitesi Mühendislik Fakültesi Elektrik/Elektronik Mühendisliği Bölümü, 2007.

[11] Demirci, M. D., Bilgisayar Destekli Ses Tanıma Sistemi Tasarımı, Yüksek Lisans Tezi, İstanbul Üniversitesi Fen Bilimleri Enstitüsü, 2005.

[12] Phoophuangpairoj, R., "Using Multiple HMM Recognizers and the Maximum Accuracy Method to Improve Voice Controlled Robots", International Symposium on Intelligent Signal Processing and Communication Systems (ISPACS), December 2011.

[13] Kirişçioğlu, F., Karabacak, E., Çetintürk, Ç., "Bilgisayar Destekli Bir Dil Programı", Turkish Speech Recognition Platform TREN.

[14] Aşlıyan, R., Günel, K., Yakhno, T., "Dinamik Zaman Bükmesi Yöntemiyle Hece Tabanlı Konuşma Tanıma Sistemi", Çanakkale Onsekiz Mart Üniversitesi, Akademik Bilişim, 2008.

[15] Bayğın, M., Karaköse, M., "Gerçek Zamanlı Ses Tanıma Tabanlı Akıllı Ev Uygulaması" IEEE 978-4673-0056, 2012.

[16] Dede, G., Sazlı,M.H., "Biyometrik Sistemlerin Örüntü Tanıma Perspektifinden İncelenmesi ve Ses Tanıma Modülü Simülasyonu", Savunma Bilimleri Enstitüsü.

[17] Edizkan, R., Tiryaki, B., Büyükcan, T., Uzun, İ., "Ses Komut Tanıma ile Gezgin Araç Kontrolü", Dumlupınar Üniversitesi, Akademik Bilişim, 2007.

[18] Bolat, B., Küçük, Ü., Yıldırım, T., “Aktif Öğrenen PNN ile Konuşma/Müzik Sınıflandırma”, Akıllı Sistemlerde Yenilikler ve Uygulamaları Sempozyumu, 2004.

[19] Ertaş, F., Hanilçi, C., Konuşmacı Tanıma Sistemi İçin Yeni Bir Sınıflandırıcı, Uludağ Üniversitesi Elektronik Mühendisliği.

[20] Meral, O., Doğrusal Öngörülü Kodlama ve Adaptif Algoritma Tabanlı Konuşmacı Tanıma, Yüksek Lisans Tezi, İstanbul Üniversitesi Fen Bilimleri Enstitüsü, 2008.

[21] Asyalı, M.H., Yılmaz, M., Tokmakçı, M., Sedef, K., Aksebzeci, B.H., Mittal, R. ,“Design and Implementation of a Voice Controlled Prosthetic Hand”, Turk J. Elec. Eng. and Comp., Vol.19, No.1, 2011.

[22] Babui, G., Kumar, H., Vanathi, P.T., “Performance Analysis of Hybrid Robust Automatic Speech Recognition System”, IEEE 978-1-46731318-6, 2012.

[23] Leechor, P., Pornpanomchai, C., Sukklay, P., “Operation of a Radio Controlled Car by Voice Commands”, 2010 2nd International Conference on Mechanica and Electronics Engineering (ICMEE 2010), 2010.

[24] Abushariah, A.A.M., Gunawan, T.S., Khalifa, O.O., “English Digits Speech Based on Hidden Markov Models”, International Conference on Computer and Communication Engineering (ICCCE 2010), 2010.

[25] Jiang, Z., Huangi, H., Yang, S., Lu, S., Hao, Z., “Acoustic Feature Comparsion of MFCC and CZT based Cepstrum for Speech Recognition”, 2009 Fifth International Conference on Natural Computation, 2009.

[26] Ferrando, F., Nouveau, G., Philip, B., Pradeilles, P., Soulenq, V., Stean, G.V. P., Courmontagne, “A Voice Recognition System for a Submarine Pilotting”, IEEE 1-4244-2523-5, 2009.

[27] Petrushin, V.A., "Emotion in Speech Recognition and Application to Call Centers", Andersen Consulting, 2000.

[28] Öztürk, N., Ünözkan, U., "Microprocessor Based Voice Recognition System Realization", IEEE 978-1-4244-6904-8/10, 2010.

[29] Özdemircan, M.Z., Robot Control With Voice Command, Bitirme Projesi, Yıldız Teknik Üniversitesi Bilgisayar Mühendisliği Bölümü, 2008.

[30] Avuçlu, E., Taşdemir, Ş., An Application For Web Browser Control With Voice Commands, (26.08.2019 -30.08.2019), Yayın Yeri:8th International Conference on Advanced Technologies (ICAT’19) , 2019.

[31] Taspinar, Y. S., Koklu, M., & Altin, M. (2020). Identification of the English Accent Spoken in Different Countries by the k-Nearest Neighbor Method. International Journal of Intelligent Systems and Applications in Engineering, 8(4), 191-194.