EAI/Springer Innovations in Communication and Computing

Luis Carlos Méndez-González
Luis Alberto Rodríguez-Picón
Iván Juan Carlos Pérez Olguín *Editors*

Innovation and Competitiveness in Industry 4.0 Based on Intelligent Systems

EAI
RESEARCH MEETS INNOVATION

Springer

Editors
Luis Carlos Méndez-González
Industrial Engineering and Manufacturing
Universidad Autónoma de Ciudad Juárez
Ciudad Juárez, Chihuahua, México

Luis Alberto Rodríguez-Picón
Industrial Engineering and Manufacturing
Universidad Autónoma de Ciudad Juárez
Ciudad Juárez, Chihuahua, México

Iván Juan Carlos Pérez Olguín
Industrial Engineering and Manufacturing
Universidad Autónoma de Ciudad Juárez
Ciudad Juárez, Chihuahua, México

ISSN 2522-8595 ISSN 2522-8609 (electronic)
EAI/Springer Innovations in Communication and Computing
ISBN 978-3-031-29774-8 ISBN 978-3-031-29775-5 (eBook)
https://doi.org/10.1007/978-3-031-29775-5

This Springer imprint is published by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Contents

Part I Artificial Intelligence and Fuzzy Techniques Applications in the Industry 4.0

Machine Learning and Edge Computing for Industry 4.0 Applications: Concepts and Extensive Review 3
Leonardo Barboni

Failure Detection System Controlled by a Mixed Reality Interface 21
Alan Yamir Rodríguez Gallegos, Luis Carlos Méndez-González, Alan Iván Hernández Holguín, and Luis Alberto Rodríguez-Picón

Industry 4.0 in the Health Sector: System for Melanoma Detection 43
Verónica Angelica Villalobos Romo, Soledad Vianey Torres Arguelles, Jose David Diaz Roman, Jesus Martin Silva Aceves, Salvador Noriega Morales, and Claudia Georgina Nava Dino

Assistive Device for the Visually Impaired Based on Computer Vision 71
Alan Iván Hernández Holguín, Luis Carlos Méndez-González, Luis Alberto Rodríguez-Picón, Iván Juan Carlos Pérez Olguin, Abel Euardo Quezada Carreón, and Luis Gonzalo Guillén Anaya

Part II Analytical Strategies for Productive Processes Based on Industry 4.0

Development and Evaluation of a Machine Learning Model for the Prediction of Failures in an Injection Moulding Process 101
A. Rojas-Rodríguez, F. S. Chiwo, H. Arcos-Gutiérrez, C. Ovando-Vázquez, and I. E. Garduño

An Approach to Select an Open Source ERP for SMEs Based on Industry 4.0 and Digitization Considering the SHERPA and WASPAS Methods 123
Juan Vicente Barraza de la Paz, Luis Alberto Rodríguez-Picón, Iván Juan Carlos Pérez-Olguín, and Luis Carlos Méndez-González

The Technological Role of Steepest Ascent Optimization in Industry 4.0 Modeling 145
Paulo Eduardo García-Nava, Luis Alberto Rodríguez-Picón, Luis Carlos Méndez-González, Iván Juan Carlos Pérez-Olguín, and Roberto Romero-López

The Role of Industry 4.0 Technologies in the Energy Transition: Conceptual Design of Intelligent Battery Management System Based on Electrochemical Impedance Spectroscopy Analysis 175
W. J. Pech-Rodríguez, Enrique Rocha-Rangel, Eddie N. Armendáriz-Mireles, Gladis G. Suarez-Velázquez, and L. C. Ordóñez

Performance Analysis of Eight-Channel WDM Optical Network with Different Optical Amplifiers for Industry 4.0 197
Anuj Kumar Gupta, Raju Sharma, Digvijay Pandey, Vinay Kumar Nassa, Binay Kumar Pandey, A. Shaji George, and Pankaj Dadheech

Part III Soft Computing Application in the Industry 4.0

Traffic Signs Configuration with a Geo-simulation Approach 215
Ariadna C. Moreno Román and Mailyn Moreno Espino

Emotional Diagnosis for Employees Within the Framework of Industry 4.0: A Case Study in Ciudad Juarez 243
Florencio Abraham Roldan-Castellanos, Ivan Juan Carlos Pérez Olguín, Luis Carlos Méndez-González, and Luis Ricardo Vidal-Portilla

Architecture for Initial States Algorithm for Blockchain Scalability in Local OnPrem IIoT Environments 275
Alfonso José Barroso-Barajas, Jesús Andrés Hernández-Gómez, Roberto Antonio Contreras-Masse, and Salvador A. Noriega-Morales

Distribution Route Optimization Using Floyd-Warshall Weighted Graph Analysis Algorithm with Google Maps Integration in Industry 4.0 Context 287
Uriel Ángel Gómez Rivera, Iván Juan Carlos Pérez Olguín, Luis Asunción Pérez Domínguez, Luis Alberto Rodríguez-Picón, and Luis Carlos Méndez-González

Feature Selection in Electroencephalographic Signals Using a Multicriteria Decision Analysis Method 307
Alexis Edmundo Gallegos Acosta, María Dolores Torres Soto, Aurora Torres Soto, Eunice Esther Ponce de León Sentí, and Carlos Alberto Ochoa Ortiz Zezzatti

Index 337

Assistive Device for the Visually Impaired Based on Computer Vision

Alan Iván Hernández Holguín, Luis Carlos Méndez-González, Luis Alberto Rodríguez-Picón, Iván Juan Carlos Pérez Olguin, Abel Euardo Quezada Carreón, and Luis Gonzalo Guillén Anaya

1 Introduction

The artificial intelligence field is one of the focuses of industry 4.0. It has been used for varied applications like robotic implementations [1], object recognition [2], and maintenance tasks [3]. Classification and object recognition are subjects of interest for developing novelty devices to provide assistive technology for visual impairment.

Vision problems correspond to all those ailments and physical problems related to the gradual or sudden decrease in the sense of vision that prevails in a good part of the population and generates a certain degree of blindness. According to the WHO [4], in 2019, there were about 2.2 billion people with some vision problems, including cases of blindness, which corresponds to a state in which those affected have such impaired vision that they cannot distinguish essential characteristics of objects, such as colors, shapes, and spatial location.

Due to the variety of cases and the difficulty in comparing certain conditions and degrees of blindness, these belong to a spectrum of vision problems in which affected users may experience particular situations in each case. Many cases of blindness may be unique and may not match the traditional definition of blindness,

A. I. H. Holguín (✉) · L. C. Méndez-González · L. A. Rodríguez-Picón · I. J. C. Pérez Olguin
Department of Industrial Engineering and Manufacturing, Institute of Engineering and Technology, Universidad Autónoma de Ciudad Juárez, Chihuahua, México
e-mail: al216737@alumnos.uacj.mx; luis.mendez@uacj.mx; luis.picon@uacj.mx; ivan.perez@uacj.mx

A. E. Q. Carreón · L. G. G. Anaya
Department of Electrical and Computation, Institute of Engineering and Technology, Universidad Autónoma de Ciudad Juárez, Chihuahua, México
e-mail: abquezad@uacj.mx; lguillen@uacj.mx

L. C. Méndez-González et al. (eds.), *Innovation and Competitiveness in Industry 4.0 Based on Intelligent Systems*, EAI/Springer Innovations in Communication and Computing, https://doi.org/10.1007/978-3-031-29775-5_4

where a person does not have a sense of vision. There have been documented cases of people who are legally blind but can distinguish specific shapes or patterns with difficulty [5].

The proposed device focuses on implementing a light object classification model based on a Support Vector Machine (SVM) multi-class implementation and a detection model based on YOLO v4 to classify and detect objects in pictures and video inputs provided by the camera of a portable device. The user can interact with the recognition system through an Android or web app.

In literature, there are several approaches to image classification. In this paper, the SVM algorithm was selected because this approach produces lightweight models and can be easily adapted to high dimension problems with multiple features. It can be applied to image classification tasks.

1.1 Background and State of the Art

Some technologies implemented in assistive devices for visually impaired people are artificial intelligence, focused on computer vision, the Internet of Things, and cloud computing. Also, these techs integrate the core fields of industry 4.0.

Computer vision encompasses several techniques that involve the use of software tools for the design of algorithms and involves the use of software tools for the design of algorithms involving the application of machine learning techniques (such as neural networks) in order to train systems with information stored in databases to recognize patterns in certain databases so that they can recognize patterns in specific characteristics of a product, such as the color and shape of particular objects. These techniques, in conjunction with image processing and facial and object recognition, attempt to emulate the human sense of vision [6].

In computer vision, focusing on image processing, recognition systems have been designed for color identification tasks [7]. Accessibility tools have been adapted for blind people in systems that use audio feedback to inform the user of the shape or color of a particular object [8]. Object recognition and location systems have been developed in the environment of a house with support for the Internet of Things and cloud services [9]. Pattern analysis techniques have been implemented to feedback intelligent assistance systems to monitor and detect particularities in a user's daily routines [10].

Accessibility consists of providing the means to adjust an environment to the user's needs and preferences [11]. It is based on the implementation of systems, devices, and methods of assistance to facilitate people's daily activities, with a special focus on those with disabilities. In particular, blind people may have problems performing their daily activities, so alternatives such as the Braille system to allow writing and reading and pets and canes to facilitate navigation have been developed. Assistive devices based on technologies such as artificial intelligence, computer vision, and the Internet of Things have been implemented to provide alternative solutions [12].

In the field of accessibility systems applications, home automation systems focused on accessibility for blind people have been implemented, like implementations that allow the user to access information about their environment through a smartphone with an interface with descriptive audio [13]. A connectivity environment between smart devices has been developed that allows the integration of external systems such as canes and smartphones to facilitate the interaction between the user and their environment [14]. A wearable device robust to external disturbances, such as lighting and noise, has been designed to provide an accessibility interface to facilitate navigation, obstacle detection, and facial recognition [15]. Regarding rehabilitation, the learning curve for blind people to acquire the necessary skills to operate assistive devices for navigation, orientation, and Braille writing has been analyzed [16].

Cloud computing provides access to scalable and virtualized resources utilizing a service scheme through the Internet [17]. The Internet of Things (IoT) consists of an interconnected system of computing devices, machines, smart devices, and objects that can transfer information over a network.

In the development of the cloud computing and IoT field, these technologies have been implemented in home automation projects to propose improved systems focused on low power consumption and better connectivity between devices with cloud services [18]. Monitoring and control systems have been established for smart lamps [19]. An analysis of the implementation of the Internet of Things and computer vision has been developed to monitor the behavior of customers and users in given environments, such as businesses, to train intelligent systems to provide assistance and obtain information on their interests [20]. Due to the versatility of IoT, implementations have also focused on blind people, such as intelligent door systems with connectivity to cloud services and other smart devices [21]. The implementation of edge computing was used to reduce the bandwidth load and improve the response of IoT-based systems [22].

Some approaches were found in literature in the application of assistive devices focused on the visually impaired, which can vary according to the focus of the aid provided, portability, and software or hardware capabilities [23]. These assistive devices focus on three main tasks: navigation, detection, and classification.

Assistive systems that integrate navigation contain tools that allow greater ease of movement in outdoor or indoor environments; sensors can be incorporated into the navigation system, which can sometimes represent a high battery consumption. These systems also contain algorithms that allow predicting the position of objects concerning time and help to predict possible future obstacles in real time; in this case, in the literature, it is found that these systems may have a low accuracy [24]. It was also found in the literature that a dual system conformed by integrated lenses and shoes was developed to obtain better obstacle detection responses. However, this system is limited to detecting objects embedded in the ground [25].

Vision assistive systems have also been developed to detect various objects through a camera in real time. Patel et al. [26] proposed the inclusion of sensors through which it is possible to obtain other types of information, such as the distance and size of the detected objects; however, this system requires long execution times.

Other systems found in the literature focused on detection accuracy, for which they integrated algorithms based on neural networks, but they have the disadvantage of high consumption of computational resources, while to ensure object registration, GPS has been implemented to guarantee the detection of objects that have been detected with low accuracy [27].

On the other hand, classification-focused assistance systems are responsible for labeling images according to certain defined features, such as color and shape [7, 28].

The systems reported in the literature show a significant advance in detecting objects and their various characteristics, such as color, shape, and distance. In some cases, this detection is performed in real time. However, those that ensure accuracy require a long time for the identification or a high computational cost, and those that are fast reported low accuracy.

1.2 Proposed System

The system is a mobile device connected to an external camera to take pictures of objects, which will be classified using an SVM multi-class implementation executed through an Android app. A conceptual representation of the system is presented in Fig. 1.

The system process includes four stages:

1. **Voice recognition**: The users use voice commands to activate the app functions and can request the acquisition of an image through the camera.
2. **Image acquisition**: A photo is taken and loaded into the classification model. A simple image processing is applied to re-scale the image and represent its data.

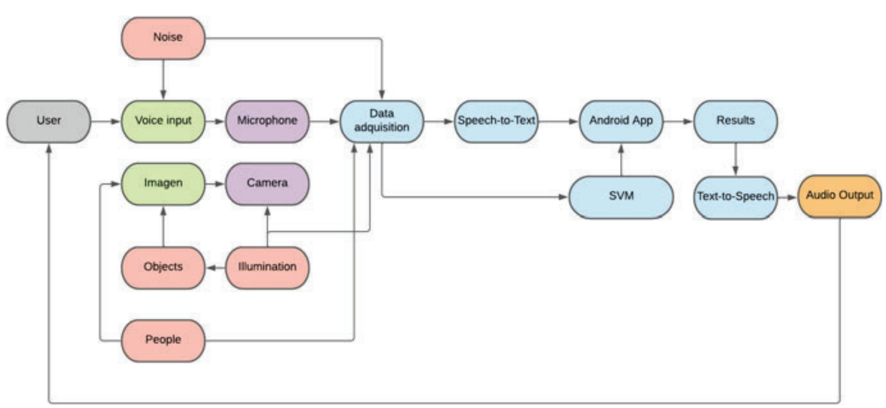


Fig. 1 Diagram of the proposed system

3. **Classification**: The users can choose which algorithm instance to be executed. If they choose the local version, the response will be faster but with reduced accuracy. If they choose the cloud version, the response will be more accurate and fast, but it requires a stable Internet connection.
4. **Results and feedback**: The model predicts the image's label, and the result is processed by a voice synthesis tool to provide the user with audio feedback on the detected class in the image.

2 Methodology

The SVM was developed in three stages: configuration, which includes the algorithm setup, dataset creation, training, and validation; classifications test, which uses new image data to verify the algorithm's performance; and implementation.

The configuration process (shown in Fig. 2 includes the following steps:

1. **Classes and parameters setup**: It consists of classes and images dataset selection, features of the Support Vector Machine, hyperparameters configuration, and cost function definitions.
2. **Training and validation set creation**: The image dataset is obtained from a repository, and the collection of images is segmented into two smaller subsets: the training set and the validation set. The former is the data the algorithm uses to train the model for predictions over new data, and the latter allows for validating the model's performance.

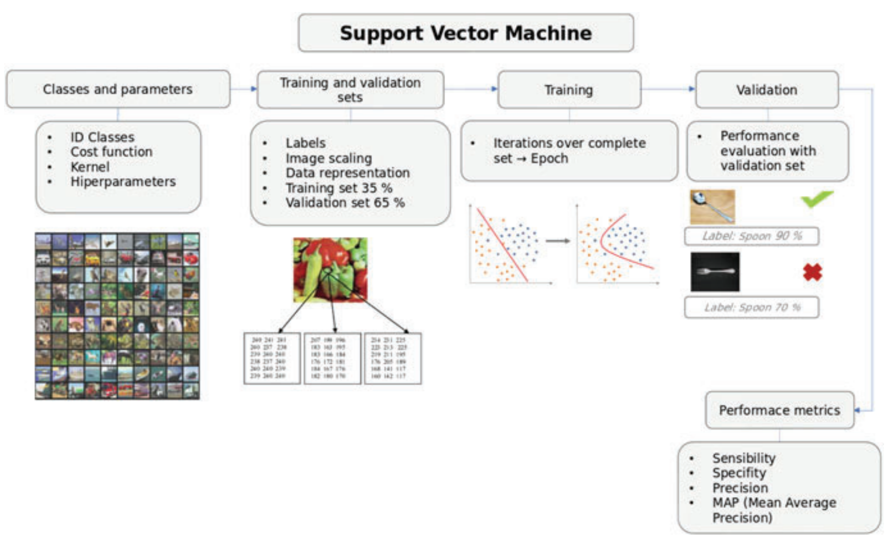


Fig. 2 SVM configuration process

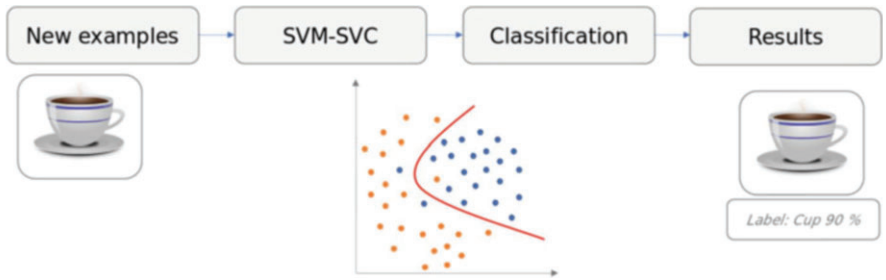


Fig. 3 SVM classification process

3. **Validation**: The model's performance is tested with the validation set (data not included in the training set) to know the model's reliability in making predictions over new data.
4. **Performance metrics and model correction**: The performance metrics are obtained after the model's validation, and, if required, the necessary adjustments or corrections to the model are made to improve his prediction ability.

The classification test and implementation stages require the deployment of the algorithm through an Android application (light model) and a cloud environment (robust model) to use the device's capabilities and make predictions over new images taken by the camera. The classification process (shown in Fig. 3 has four steps:

1. **Acquisition of new image data**: The device is used to take photos of objects. The images generated are loaded into the algorithm to start the classification.
2. **SVM-Support Vector Classifier**: The weight values are loaded into the model, and the input images are transformed to adjust to a default resolution.
3. **Classification**: The model generates a solution vector that can be converted in an output label by correlation. The output label represents the class predicted for the object identified in the input image.
4. **Results**: The user receives the output through an audio feedback.

2.1 Algorithms Comparison

Image classification tasks can be approached through different algorithms according to the image data used and the classes defined. The most common image classification algorithms identified in the literature are described in Table 1.

In object recognition, there are two approaches for algorithm implementation: object classification and object detection. In the former category, the most common algorithms are statistical, like Naive Bayes or random forest. Later, deep learning techniques are used through convolutional neural networks and other neural network

Table 1 Comparison between algorithms for image classification

Algorithm	Focus
Support Vector Machine	Classification and detection
k-nearest neighboor	Classification
Naive Bayes	Classification
Random forest neighbor	Classification
Perceptron	Classification and detection
Convolutional neural networks	Classification and detection

architectures. One of the most remarkable differences between these two categories is the computational power required to execute deep learning algorithms, which are more power-hungry than lighter algorithms like SVM.

2.2 Objective Function

The optimization objective is to maximize the margin value between the lines generated by the support vectors. This can also be represented by Eq. 1.

$$Min\frac{1}{2}\sum_{i=1}^{n} w_i^2 \tag{1}$$

The most common loss function in this type of algorithm is the hinge loss, described by Eq. 2, in which t is the value assigned to the class (which can be -1 or 1) and y is the raw output value of the prediction.

$$l(y) = Max(0, 1 - t \cdot y) \tag{2}$$

The cost function of the algorithm is obtained by combining the optimization and loss functions. For a binary classification problem, the cost function is described by 3

$$J(w) = Min\frac{1}{2}\sum_{i=1}^{n} w_i^2 + l(y) = Max(0, 1 - t \cdot y) \tag{3}$$

For a multi-class Support Vector Machine approach, Eq. 3 is extended to more than two classes and can be implemented with many features. The multi-class SVM can be defined using the One vs. One technique, which compares each class to the rest of the data and generates a classifier for each comparison. The number of classifiers is equal to Eq. 4.

$$Number\_of\_Classifiers = \frac{n(n-1)}{2} \tag{4}$$

The SVM implementation proposed consists of a Support Vector Classifier adapted to image classification of ten classes of different objects. The classifier was deployed using four different kernel functions to compare their performance. The SVM was adapted to multi-class using the "*One vs. One*" approach, which generates 45 instances of One vs. One.

3 Materials and Methods

3.1 Dataset

The classification and detection algorithms were trained using custom versions of the Open Images v4 dataset and the Imagenette dataset (a subset of the Imagenet dataset), respectively.

The custom YOLO v4 algorithm, implemented for object detection, was pre-trained using the OpenImages v4 dataset, an image dataset for object detection algorithms [29]. It is composed of a collection of images from 600 different classes, but only 9 were extracted to train and validate the proposed classification and detection system. The classes selected are described in Table 2. Each image has an associated text file that contains the label corresponding to the image's class and the coordinates of the bounding boxes for each object in an image. The image resolution varies between all the items in the dataset. Thus, they need to be re-scaled to adjust their resolution before being implemented in the training process.

Some selected images from the Open Images v4 dataset are shown in Fig. 4. A notable feature of this collection is that the illumination, object position, and size vary significantly between all the images, which helps to avoid the over-fitting problem of using too many similar images of each class.

The multi-class SVM was trained using a subset of the ImageNet dataset [30], called Imagenette [31]. The original dataset is composed by approximately 1000 object classes and is used for competitions and benchmarking purposes, focused on object detection and classification algorithms. The Imagenette subset is a

Table 2 Classes contained in the Open Images v4 subset

ID	Class	Description
0	Person	Varied age and genre
1	Scissors	Varied styles and colors
2	Coin	Varied values and sizes
3	Mug	Varied materials and colors
4	Knife	Varied styles and materials
5	Bottle	Different materials and colors
6	Socks	Varied patterns, colors, and sizes
7	Spoon	Varied materials and sizes
8	Fork	Varied materials and sizes

Fig. 4 Sample of the OpenImages v4 dataset

Table 3 Classes contained in the ImageNet subset

ID	Class	Description
0	Cassette_player	Electronic device
1	Chainsaw	Tool
2	Church	Building
3	English_springer	Dog breed
4	French_horn	Musical instrument
5	Gas_pump	Storage and dispenser of gasoline
6	Golf_ball	Sports equipment
7	Parachute	Device used for parachuting
8	Tench	Fish species
9	Trash_truck	Utilitary vehicle

classification-oriented collection of images integrated by ten classes, presented in Table 3. Each class contains a variable number of examples, which range between 50 and 100. The training and validation sets were defined with a ratio of 40 and 60% of the original sample, respectively.

Figure 5 shows a small sample of the images that can be found in the dataset utilized for the classification algorithm.

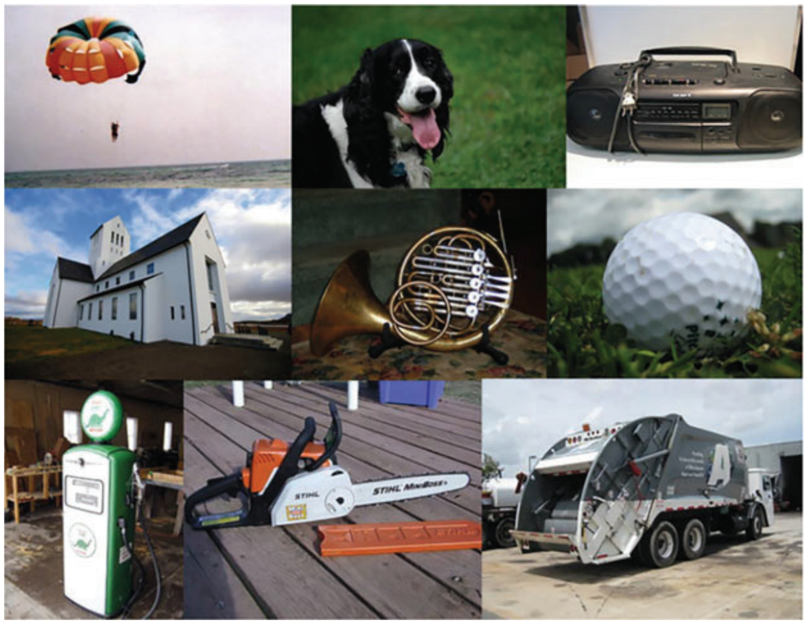


Fig. 5 Sample of images contained in the Imagenette subset

3.2 Hardware and Software

The algorithm was deployed in a mobile device. The system was constructed using the following hardware components:

- **Camera**: Integrated in the mobile device
- **Mobile device**: An Android smartphone device with an app powered by a light model of the detection algorithm

The software employed in the device is conformed by the following:

- **Android app**: Application deployed with a light detection model in the mobile device.
- **Light Support Vector Machine algorithm**: A compact version of the detection algorithm to provide fast results through the Android app.
- **Robust Support Vector Machine algorithm**: A more robust and complex version of the detection algorithm provides better prediction accuracy. This algorithm is deployed in a cloud environment connected to the Android app, so it requires a stable Internet connection.

3.3 Support Vector Machine

The SVM implementation proposed consists of a Support Vector Classifier adapted to image classification of ten classes of different objects. The classifier was deployed using four different kernel functions to compare their performance. The SVM was adapted to multi-class using the "*One vs. One*" approach, which generates 45 instances of One vs. One. The classifiers for the classes are described in Fig. 6.

For each classifier, the kernel functions used are linear kernel Eq. 5, RBF kernel (radius basis function) (Eq. 6), polynomial kernel (Eq. 7), and sigmoid kernel (Eq. 8).

$$k(x, y) = x \cdot y \tag{5}$$

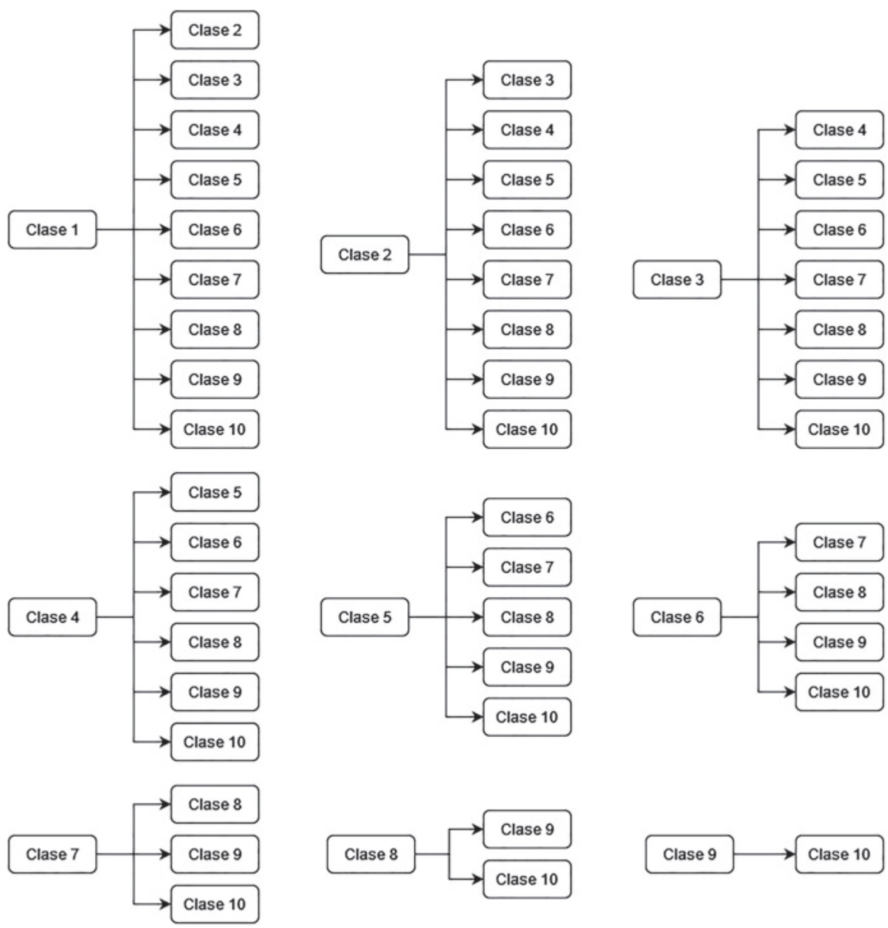


Fig. 6 Classifiers used for the model

$$k(x, y) = exp(-a||x - y||^2) \quad (6)$$

$$k(x, y) = (x \cdot y)^d \quad (7)$$

$$k(x, y) = tanh(ax \cdot y) \quad (8)$$

To create a model for the Support Vector Machine algorithm, the following tasks were completed:

- **Configuration of the algorithm**: The algorithm was characterized according to the multiple classifiers approach and using the loss function for Support Vector Machines. The environment for training, validation, and execution of the algorithm was prepared using a cloud service provided by Google through the Google Colab suite.
- **Preparation for the dataset**: The dataset was retrieved from the ImageNet repository, and ten classes were selected to create the dataset used for training and validation. This collection of images was split into two subsets to create the training set, which was used to train the algorithm, and the validation set, which was required to validate the performance of the trained model. As seen in Table 4, the training and validation sets were created with 35% and 65 & of the total images contained in the original dataset.
- **Parameter tuning**: The parameters that dictate the model's behavior were tuned. Several sets of parameters were obtained and compared with a short training performance test to obtain each set's performance metrics. The best-performing set was selected to configure the model.
- **Training**: The algorithm was trained using an iterative process that inputs the training set into the model to learn from the examples and achieve a good performance of object recognition. The trained model can receive an image as input and provide an output composed of the object class detected in the image and the model's confidence.
- **Validation**: After training, the validation set is used to test the performance in object recognition and provide performance metrics to compare the trained model with other similar implementations. A simplified overview of the training and validation process is presented in Fig. 7.
- **Implementation**: After obtaining a model with a good performance according to the metrics, it was deployed on the Android application.

Table 4 Sets for training and validation

Set	Total %	No. images
Training	35%	2800
Validation	65%	5200

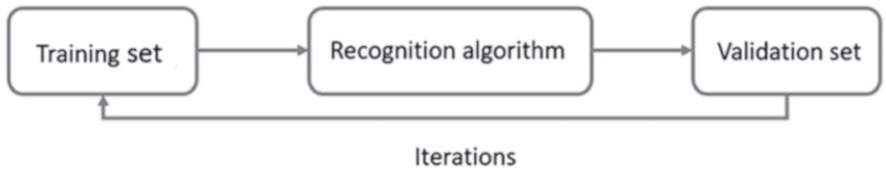
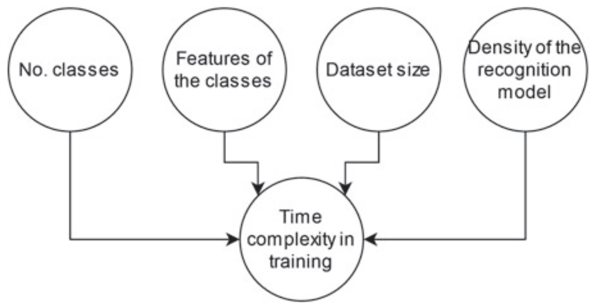


Fig. 7 Simplified training process

Fig. 8 Influence of several factors in the time complexity of the training process



3.4 Problem Characterization

The structure of the SVM model is composed of 45 classifiers, and the object recognition implementation requires a high amount of features to process the data of interest. Hence, the model has a certain degree of complexity. This situation impacts several areas, such as training, validation, and performance (as seen in Fig. 8). In a complex model, the data required can be so big that the training time can exponentially grow. The parameters can be tuned and optimized to obtain the best values to achieve good performance without incurring limited training time to reduce training problems.

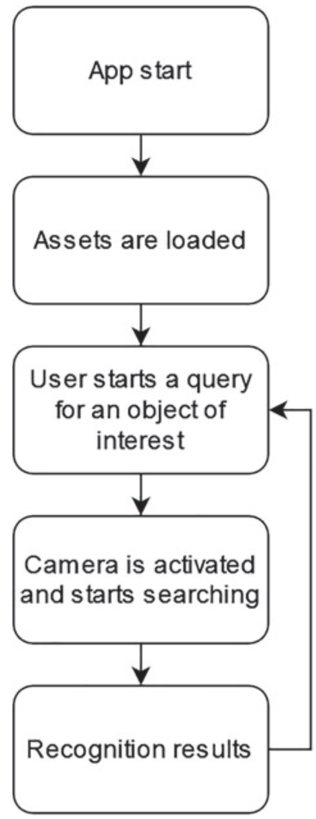
3.5 Algorithm Implementation

The classification algorithm was implemented in an Android app through a simplistic interface with voice recognition tools. The detection algorithm was based on a custom YOLO v4 [32] trained with the custom dataset labeled for object detection. The interaction model is represented in Fig. 9.

The app features Speech-to-Text and Text-to-Speech using Google built-in modules. The process of queries (Fig. 10 is composed of four stages:

1. **Stage 1**: The user starts the app, and all the assets are loaded. The audio feedback informs the user that the app is ready to start queries.
2. **Stage 2**: The user utilizes his voice to request a particular object. The system activates the camera, which the user can focus on to scan several scenes to search for the object of interest.

Fig. 9 Interaction process of the Android app



3. **Stage 3**: If the object was located, the audio feedback responded to the users indicating the status of the object. On the contrary, the audio feedback negatively responds to the user, informing that the object of interest was not found in the scenes scanned.
4. **Stage 4**: The query is finished, and the user can choose to start a new one or exit the app.

The command voice proposed for the voice command interface is enumerated in Table 5. The default languages supported by the app are English and Spanish, so any user with a specific proficiency in these languages can use the application with relative ease (as seen in Fig. 10.

3.6 Proposed Design of Experiments

The model parameters were optimized through a DOE (design of experiments) approach. First, the optimizing parameters were defined, and a DOE was formu-

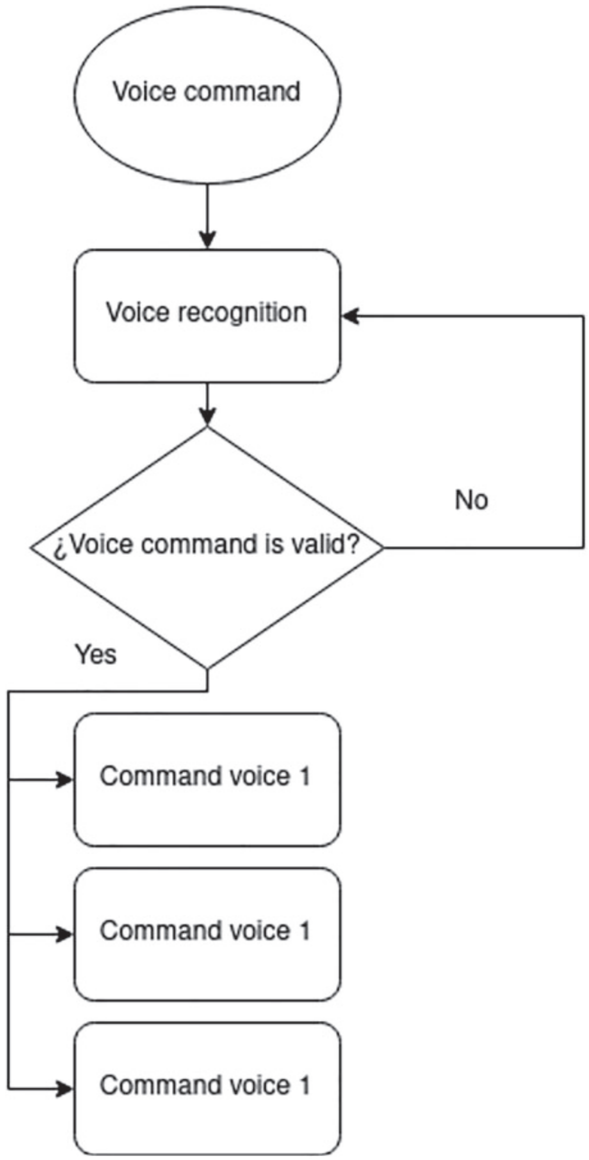


Fig. 10 Queries with voice commands

lated. Then, the parameters set were iterated using the DOE proposed, and several sets of values were obtained. The sets were compared through a performance test using the reduced-scale training process. The best-performing set was selected to train the recognition model using the images dataset. (The tuning process is shown in Fig. 11).

Table 5 Voice commands

English	Spanish	Description
Start	Inicio	Start the app and load all the assets
Search for + "object name"	Busca + "nombre de objeto"	The users select the object of interest
New query	Nueva consulta	The process returns to the initial state

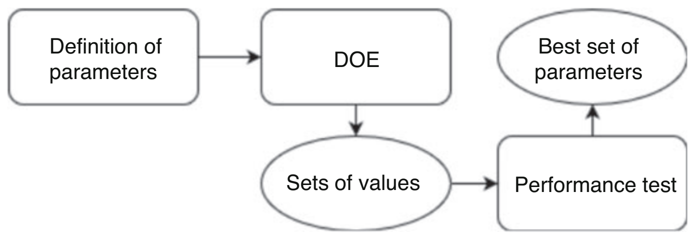


Fig. 11 Simplified tuning process for the model parameters

Table 6 Parameters for the SVM implementation

Parameter	Area of application
Standardized size of images	Dataset
No. classes	Dataset
C (Penalization for misclassification)	SVM
γ (Loss compensation)	SVM
No. epoch	Training
Learning rate	Training

According to the nature of the data required for training (the ImageNet dataset), the parameters are described in Table 6

The parameters utilized can be classified according to his area of governance in dataset parameters, SVM parameters, and training parameters.

The dataset parameters adapt the image dataset to the model altering its size, which is the resolution in pixels (width x height) and the number of classes required for the recognition model.

The SVM parameters tune the loss and cost functions to improve the classifier's performance by using the penalization and compensation parameters, which are represented by C and γ, respectively.

The training parameters can be used to manipulate the training process to reduce the like-hood of over-fitting or to achieve a good performance in a reasonable amount of time. A DOE was proposed to improve the tuning process and achieve an adequate set of parameters.

4 Results

4.1 Android App

After training and validation, the data fitting can be interpreted as the capacity of the model to make predictions about new inputs; otherwise it is the ability to classify images that were not part of the training and validation sets. In general, there are three cases of data fitting.

- **Under-fitting**: The model adjusts poorly to the data with a lot of misclassification cases. It needs more training or examples of data.
- **Acceptable fitting**: The model allows a reduced number of misclassifications and can make good predictions over new data.
- **Over-fitting**: The model adjusts very tightly to the training data, so the misclassification is greatly punished and can't make accurate predictions over new data.

A graphical user interface (GUI) in a desktop app was developed to provide a visual representation of the algorithm implementation and some performance metrics to evaluate the predictions made by the SVM model.

The GUI was developed using the Tkinter module in Python 3.9. This approach was selected because it allows to port the GUI app between the most common operative systems with access to a Python compiler. The desktop app can be executed in Windows, macOS, and most Linux distros.

The app was designed with a minimalist style to improve readability and provide a clean representation of the data obtained from the object recognition model. The basic app's GUI is shown in Fig. 12.

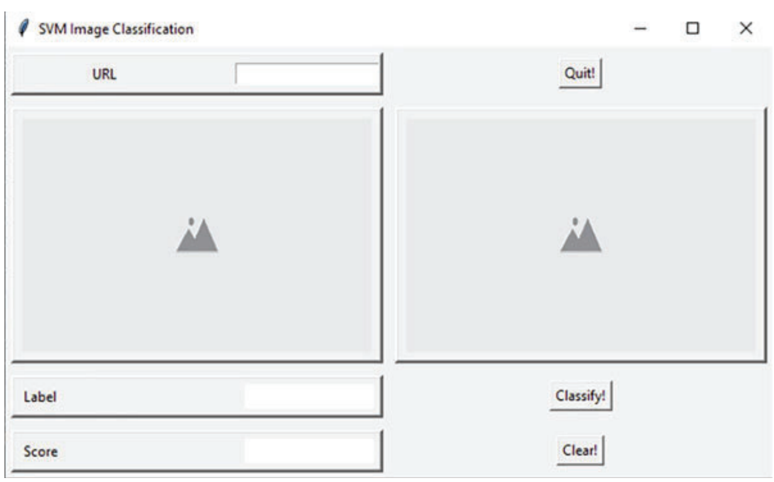


Fig. 12 GUI deployed in Windows environment

The user can interact with the app by selecting some images to test the object recognition algorithm. The results for each image are displayed in the interface using the label and confidence score generated by the model and a pie chart to visualize the score obtained by the other classes. The GUI's layout is composed of the following items:

- **Request box**: The user can indicate in a text box the parameters of the query and which image will be processed.
- **Left image box**: The image selected is displayed and scaled according to the app's window resolution.
- **Right image box**: A pie chart is displayed to provide information about the other classes detected in the image to compare the result obtained by the detected class with all the classes in the model.
- **Result box**: This box shows the class detected label corresponding to the type of object recognized in the image.
- **Score box**: Provides the accuracy score obtained in the classification process in percentage form.
- **Control buttons**: The user can use a set of buttons to control the app's behavior. The functions assigned to them are described as follows:
  - **Classify button:** If the user specifies the parameters and image to process, the model classifies the image selected and displays the results.
  - **Clear button:** Clear all the boxes in the GUI and allows to start a new classification process.
  - **Quit button:** Exits the app in secure mode without compromising the data obtained.

Several examples of the app classification process are shown in Fig. 13. The app was implemented with a general-purpose approach, so it can easily be configured with different models and classification algorithms.

The Android application was developed with a minimalist interface and implemented Text-to-Speech and Speech-to-Text functions. The interface of the app is presented in Fig. 14.

The app has two interface modes, tactile and voice commands, for less disabled users and users with a robust visual disability, respectively.

The tactile interface has the following buttons and interactions:

1. **Activation**: Toggle button that activates the classification model.
2. **Photography button**: Takes a photo using the smartphone camera and loads it into the classification model.
3. **Results button**: Initially, it is invisible until the photography button is pressed. When it is active, if the user press this button, the app will show the results obtained from the image classification process.

The voice command interface has the following interactions:

1. **Voice command "Start"**: The classification model is activated, and the app is ready for queries.

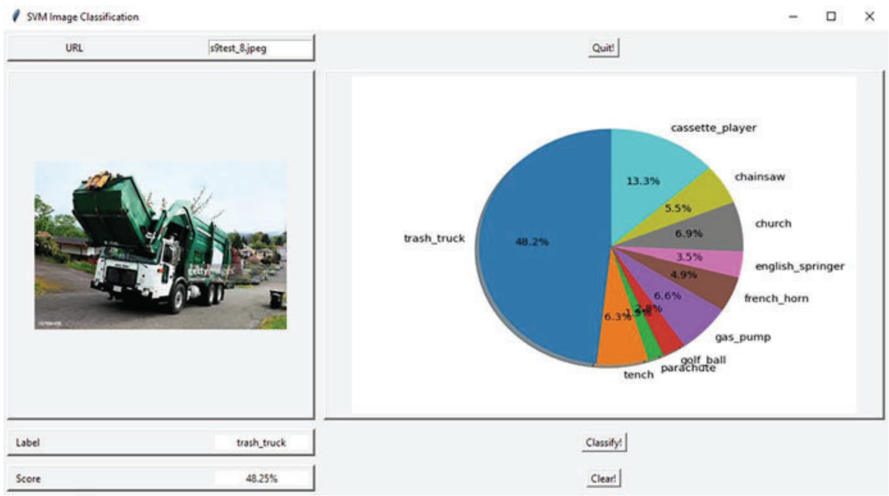


Fig. 13 Example of the classification process in the desktop app

Fig. 14 Interactions in the Android app

2. **Voice command "Search for –object_name–"**: The app will take a photo, and the classification model will process it to generate a label according to the class detected in the image. The results will be received by the user using audio feedback.
3. **Voice command "New query"**: The user can start a new query after completing the app's main process.

4.2 Classification Results

The model was tested using the validation set to measure the performance and behavior of the trained classes. The performance of the SVM classifier was measured using the following metrics:

- **Precision**: Measures how close the data dispersion between examples is.
- **Accuracy**: Measures how close are the data to a valid value.
- **Recall**: Measures the relation of the positives obtained and the expected positives.
- **Sensibility**: Is the probability for proper positive detection.
- **Mean average precision (MAP)**: This stat is calculated using each class precision value to obtain the mean for the model.

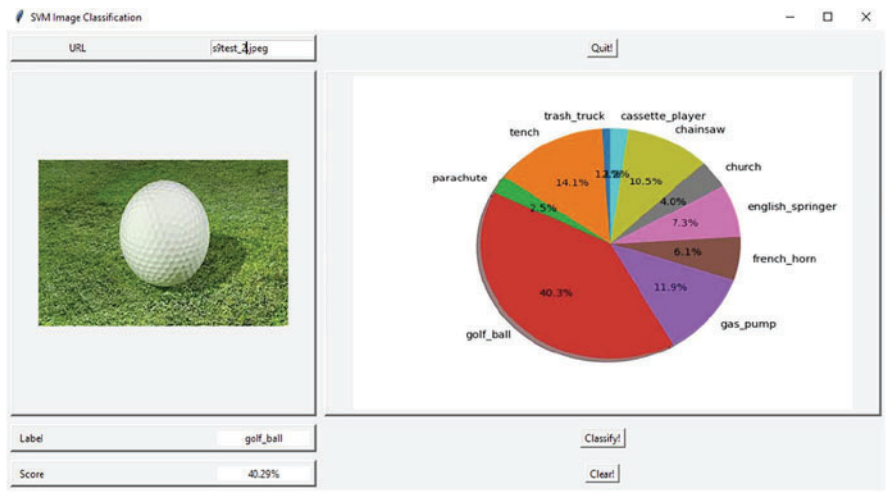
A small sample of classification tests can be seen in Fig. 15. The pie charts provide the confidence values for the other classes, and the one with the max value is the label generated by the model to classify the image.

The confusion matrix (Fig. 16) was constructed with the cases obtained for the object classes. The descriptions for each case are as follows:

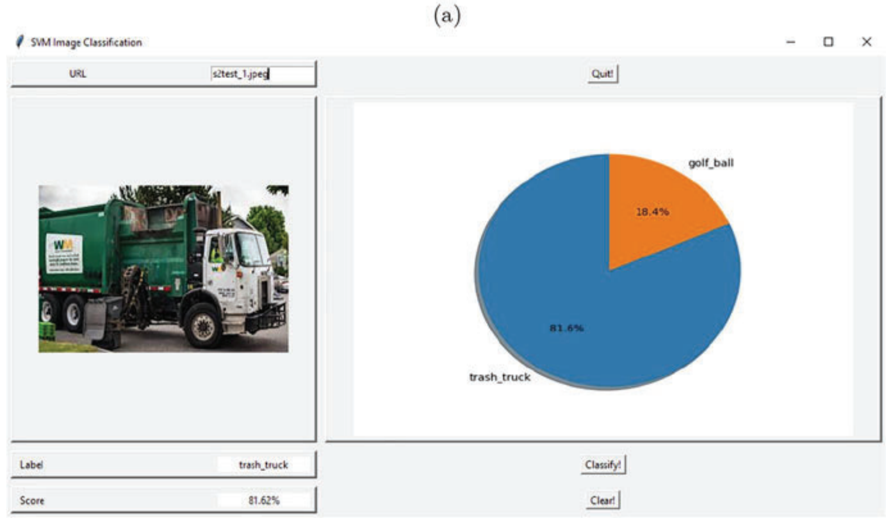
- **True positive**: The object was correctly classified, and the label coincides with his type.
- **True negative**: The object was correctly classified and is not included in the classes. No label is generated.
- **False positive**: The object was incorrectly classified, and a label was assigned to a type of object not included in the original classes.
- **False negative**: The object was incorrectly classified, and his type coincides with a label but is detected as an object not included in the classes.

For each class, their respective performance parameters were obtained. According to Tables 7 and 8, the highest performing class was Class 8, and the lowest was Class 10. The value disparity could be explained by differences in images' quality for each class in the training set and the complexity of the object represented by the class. So, the worst performing classes correspond to difficult-to-classify objects with poor variation and quality in their corresponding training images. For the other part, the classes that surpassed the 0.6 value were composed of training images of better quality for classification tasks.

The main metrics of the model are presented in Table 9. The class metrics were used to obtain the mean average precision (mAP) of the model. According to the metric values obtained, the model's performance is better than a hypothetical luck-based classifier, so it can predict the corresponding class in an image more accurately than simply guessing it. The recall of the model represents how well the model classifies positive examples compared to the total of positive cases. In this case, a recall value of 0.5 represents that the model can classify correctly half of the positive cases. The mAP obtained was low compared to the accuracy and precision values due to the low-performing classes previously explained.



(a)



(b)

Fig. 15 Example of classification tests with the validation set. (**a**) Example with 10 classes. (**b**) Example with 2 classes

The precision and mAP over the complete training process are shown in Fig. 17.

The convergence of the model was achieved at the 6000 iterations mark. At this point, the model achieved the maximum precision value. The SVM implementation achieved a moderate accuracy value (60%) and can be deployed in a lightweight model, allowing its implementation in light apps and devices without dedicated computing resources.

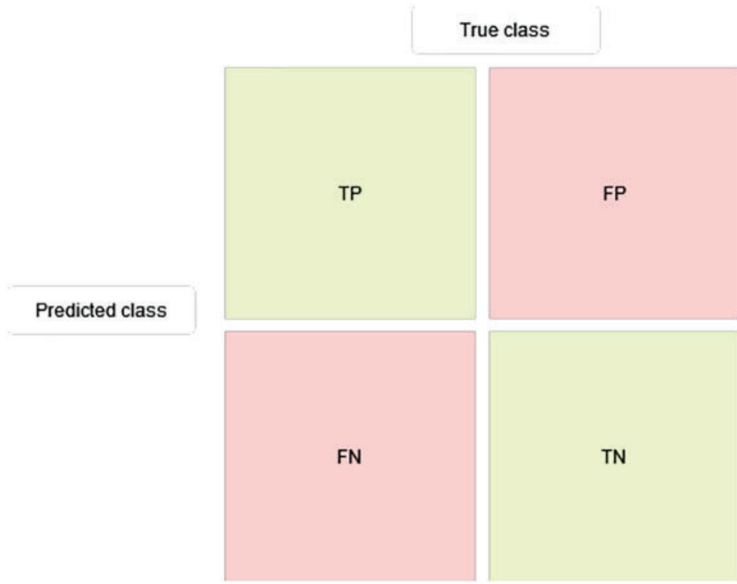


Fig. 16 Confusion matrix for the classes of the model

Table 7 Performance metrics for each class (1–5)

Metric	Class 1	Class 2	Class 3	Class 4	Class 5
Accuracy	0.56	0.43	0.7	0.68	0.52

Table 8 Performance metrics for each class (6–10)

Metric	Class 6	Class 7	Class 8	Class 9	Class 10
Accuracy	0.61	0.59	0.74	0.5	0.35

Table 9 Performance metrics for the model

Metric	Value
Accuracy	0.6
Precision	0.59
Recall	0.5
mAP	0.47

The approach's limitations include the small number of classes, the dataset, and the focus on classification-only tasks. The model can be easily scaled to process more complex data, like a significant number of classes, more populated images, and classification by features like color and shape, because the foundation of the SVM approach allows adjusting the model to a large number of features and classes.

According to the results obtained, the algorithm has a big room for improvement when combined with other machine learning techniques like deep learning. This combination can allow the implementation of classification models with the SVM classifier to achieve better accuracy and timely execution results.

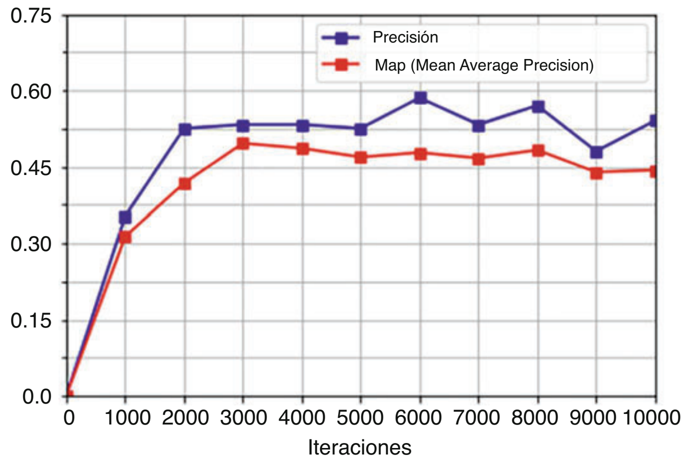


Fig. 17 mAP and precision in the training process

Table 10 Comparison between YOLO v4 classification model and the SVM implementation

Model	mAP	Accuracy	Dataset
SVM	0.47	0.6	ImageNet
YOLO v4	0.64	0.8	OpenImages

4.3 Metrics Comparison

The most representative metrics obtained for the SVM model were mean average precision (mAP) and accuracy because these measure the effectiveness of the model's predictions and the performance of the trained classes, respectively. The mAP metric is the mean of the individual precision means per class and it can tell how well the model adapts to the data.

Table 10 shows the mAP and accuracy stats for the classifier and the detection models. The stat values for the detection model (YOLO v4) are better than the classifier (SVM) due to the algorithm's robustness and its good adaptation to populated images. Nevertheless, the SVM is better suited for simple images for classification purposes.

The SVM model obtained a lower accuracy and mAP scores but a more reasonable execution time than the other model analyzed. The adobe means that the proposed model is faster than other classification algorithms but is compensated with lower accuracy.

A faster model is ideal for deployment in non-dedicated devices, like smart-phones, or in more flexible environments, like web apps and desktop apps.

Another SVM model was trained and validated to compare the impact of the hyperparameter tuning optimization. This new model was manually tuned to adjust its hyperparameters. According to Table 11, the parameter tuning optimization produced a better model than one with arbitrary parameters. The performance

Table 11 Comparison between a manually and an optimized tuned SVM model

Tuning	Accuracy	Precision	mAP	Recall
Optimized	0.6	0.59	0.57	0.5
Manual	0.47	0.43	0.45	0.39

metrics were improved with the optimization with a significant gain compared to a model without optimization. Also, the model generated with manual tuning did not surpass the 50% accuracy threshold; that means the manual tuned model is not better than a classifier based on pure luck.

5 Conclusion and Future Scope

The algorithm was designed according to the limitations and characterization made in the project's starting stage. A graphical user interface was developed to visualize the algorithm performance better, and an SVM model was trained using a subset of the ImageNet dataset. The main research product is an Android application with voice command support, classification (SVM), and detection (YOLO v4) algorithms. According to the general results, the app could perform well in a controlled indoor environment with little perturbations and noise.

The results obtained in the classification and performance tests measured the overall precision of the algorithm at 60%, which is a good score for a bare-bones classifier without using techniques like data augmentation or specialized algorithm architectures. However, comparing the SVM classifier to the state-of-the-art algorithms, the current algorithm implementation provides less accurate results. It can fail in classification tasks when the input images contain complex scenes. The performance obtained could be a product of the dataset used and the parameter tuning or due to the nature of the classifier implemented.

Implementing the SVM classifier is not recommended for complex tasks. It can only achieve good results with simple data, like images with few objects and less-polluted background (e.g., simple colors as background). For more complex images, the YOLO v4 image detector must be used. According to his performance, this implementation has room for improvement in the following areas:

- **Algorithm**: The methodology process's main challenge was the algorithm's configuration to achieve the abovementioned scores. Thus, the architecture of the algorithm could be tuned to achieve better precision with more complex images. Also, a different algorithm could be combined with the existing one to reduce the weakness of the current implementation.
- **Dataset**: The dataset used can be augmented with more examples in the training and validation sets, more classes, better image resolution, and the implementation of techniques, like data augmentation, to improve the data quality.
- **Additional features**: Other features can be implemented in the system, like other vision computer-related tasks, such as pattern recognition and object

tracking, or more complex approaches, like face recognition and detection in three dimensions. Also, the SVM algorithm can be improved by combining it with more robust classifiers, like deep learning neural networks, to allow the implementation of object detection for more complex classes.
- **App deployment**: A critical point of the deployment of the application is the supported platforms. So, a web app can be developed to use the application in a large variety of systems, like smartphones, desktop PCs, laptops, and other intelligent devices.

Due to the nature of the algorithm, it can easily be adapted to complement another object recognition algorithm, so the next major step to improve it will be to implement the SVM in a deep learning algorithm to enhance a object detection model and optimize it to achieve better performance metrics compared with state-of-the-art implementations. The following projects and implementations can be derived from the SVM approach:

- **Feature extraction**: Includes the segmentation and extraction of features of interest in images, like colors, shapes, and patterns.
- **Object tracking**: It consists of tracking an object of interest across a sequence of frames to allow the system to "remember" the object for future tasks.
- **Object localization**: Requires the detection of an object in an image, indicating the object's position according to other items in the scene.

The SVM approach can be generalized for different image classification tasks as:

- **Identification of vegetables and fruits**: Implementation based on feature extraction and object classification. Identification of classes of different types of fruits and vegetables and their state of ripeness to separate the healthy specimens from those infected by plagues or rotten.
- **Identification of plants and animals species**: Classification of animal and plant specimens by their species. This approach requires a multiple relational class hierarchy. The classes need to be segmented into subclasses and sub-subclasses to generate a robust model to process this complex task.
- **Vision systems for arm robots**: Another related work is the implementation of an object classification algorithm to provide a robot with the capability to recognize objects by their features (like color or shape) and select the target classes.
- **Medical imaging**: Image processing, feature extraction, and classification models are often used to generate automatic diagnostic tools to detect illness and health problems in medical images.
- **Identification of traffic signs**: An approach related to the scope of the current SVM implementation is the detection and recognition of traffic signs to improve visually impaired people's awareness in urban environments. This implementation can reduce the risks of outdoor navigation.

References

1. J. Ribeiro, R. Lima, T. Eckhardt, S. Paiva, Robotic process automation and artificial intelligence in industry 4.0—a literature review. Proc. Comput. Sci. **181**, 51–58 (2021). CENTERIS 2020—International Conference on ENTERprise Information Systems / ProjMAN 2020—International Conference on Project MANagement/ HCist 2020—International Conference on Health and Social Care Information Systems and Technologies 2020, CENTERIS/ProjMAN/HCist 2020
2. X. Zhou, X. Xu, W. Liang, Z. Zeng, S. Shimizu, L.T. Yang, Q. Jin, Intelligent small object detection for digital twin in smart manufacturing with industrial cyber-physical systems. IEEE Trans. Ind. Inform. **18**(2), 1377–1386 (2022)
3. Z.M. Çınar, A. Abdussalam Nuhu, Q. Zeeshan, O. Korhan, M. Asmael, B. Safaei, Machine learning in predictive maintenance towards sustainable smart manufacturing in industry 4.0. Sustainability **12**(19), 8211 (2020)
4. W.H. Organization et al., World report on vision, 2019
5. B.K. Swenor, M.J. Lee, V. Varadaraj, H.E. Whitson, P.Y. Ramulu, Aging with vision loss: a framework for assessing the impact of visual impairment on older adults. Gerontologist **60**(6), 989–995 (2020)
6. B. Zhang, Computer vision vs. human vision, in *9th IEEE International Conference on Cognitive Informatics (ICCI'10)* (2010), pp. 3–3
7. P. Mungkaruna, P. Piyawongwisal, K. Ropkhop, U. Hatthasin, The talking color identifying device for the visually impaired, in *2016 13th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology, ECTI-CON 2016* (2016)
8. P.S. Jamuni, S. Borkar, Colour and shape identification for assisting visually impaired person, in *Proceedings of the International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud), I-SMAC 2018* (2019), pp. 415–418
9. K. Srinivasan, V.R. Azhaguramyaa, Internet of Things (IoT) based object recognition technologies, in *Proceedings of the Third International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC 2019)* (2019), pp. 216–220
10. R. Phadnis, J. Mishra, S. Bendale, Objects talk—object detection and pattern tracking using tensorflow, in *Proceedings of the International Conference on Inventive Communication and Computational Technologies, ICICCT 2018* (2018), pp. 1216–1219
11. L. Valdes, *Accessibility on the Internet*, 2004
12. K. Manjari, M. Verma, G. Singal, A survey on assistive technology for visually impaired. Internet Things **11**, 100188 (2020)
13. M. Kandil, R. Albaghdadi, F. Alattar, I. Damaj, AmIE: an ambient intelligent environment for assisted living, in *2019 Advances in Science and Engineering Technology International Conferences, ASET 2019* (2019)
14. J. Connier, H. Zhou, C. De Vaulx, J.J. Li, H. Shi, P. Vaslin, K.M. Hou, Perception assistance for the visually impaired through smart objects: concept, implementation, and experiment scenario. IEEE Access **8**, 46931–46945 (2020)
15. R.A. Minhas, A. Javed, X-EYE: a bio-smart secure navigation framework for visually impaired people, in *2018 International Conference on Signal Processing and Information Security, ICSPIS 2018* (2019), pp. 2018–2021
16. D. Pawluk, N. Bourbakis, N. Giudice, V. Hayward, M. Heller, Haptic assistive technology for individuals who are visually impaired. IEEE Trans. Haptics **8**(3), 245–247 (2015)
17. L.G. Nick Antonopoulos, *Cloud Computing in a Nutshell*, vol. 15 (2011)
18. T. Chaurasia, P.K. Jain, Enhanced smart home automation system based on Internet of Things, in *Proceedings of the 3rd International Conference on I-SMAC IoT in Social, Mobile, Analytics and Cloud, I-SMAC 2019* (2019), pp. 709–713

19. A.K. Gupta, R. Johari, IOT based electrical device surveillance and control system, in *Proceedings—2019 4th International Conference on Internet of Things: Smart Innovation and Usages, IoT-SIU 2019* (2019)
20. R. Paradaa, J. Melia-seguib, A. Carrerasa, M. Morenza-cinosa, R. Pousa, Measuring user-object interactions in loT spaces, in *2015 IEEE International Conference on RFID Technology and Applications (RFID-TA) Measuring* (2015)
21. F.L. Barsha, Z. Tasneem, S. Mojib, M. Afrin, N. Jahan, M. Tasnim, U. Habiba, M.N. Islam, An IoT based automated door accessing system for visually impaired people, in *2019 5th IEEE International WIE Conference on Electrical and Computer Engineering, WIECON-ECE 2019—Proceedings* (2019), pp. 2–5
22. M.J. Junior, O.B. Maia, H. Oliveira, E. Souto, R. Barreto, Assistive technology through Internet of Things and edge computing, in *IEEE International Conference on Consumer Electronics - Berlin, ICCE-Berlin*, vol. 2019 (2019), pp. 330–332
23. R. Tapu, B. Mocanu, T. Zaharia, Wearable assistive devices for visually impaired: a state of the art survey. Pattern Recogn. Lett. **137**, 37–52 (2020). Learning and Recognition for Assistive Computer Vision
24. R. Tapu, B. Mocanu, T. Zaharia, DEEP-SEE: joint object detection, tracking and recognition with application to visually impaired navigational assistance. Sensors (Switzerland) **17**(11), 2473 (2017)
25. A. Pardasani, P.N. Indi, S. Banerjee, A. Kamal, V. Garg, Smart assistive navigation devices for visually impaired people, in *2019 IEEE 4th International Conference on Computer and Communication Systems, ICCCS 2019* (2019), pp. 725–729
26. C.T. Patel, V.J. Mistry, L.S. Desai, Y.K. Meghrajani, Environment for visually impaired people, in *2018 Second International Conference on Intelligent Computing and Control Systems (ICICCS)*, no. ICICCS (2018), pp. 1–4
27. I. Joe Louis Paul, S. Sasirekha, S. Mohanavalli, C. Jayashree, P. Moohana Priya, K. Monika, Smart eye for visually impaired-an aid to help the blind people, in *ICCIDS 2019—2nd International Conference on Computational Intelligence in Data Science, Proceedings* (2019)
28. A.S. Alon, R.M. Dellosa, N.U. Pilueta, H.D. Grimaldo, E.T. Manansala, EyeBill-PH: a machine vision of assistive Philippine bill recognition device for visually impaired, in *2020 11th IEEE Control and System Graduate Research Colloquium, ICSGRC 2020—Proceedings* (2020), pp. 312–317
29. A. Kuznetsova, H. Rom, N. Alldrin, J. Uijlings, I. Krasin, J. Pont-Tuset, S. Kamali, S. Popov, M. Malloci, A. Kolesnikov et al., The open images dataset v4. Int. J. Comput. Vis. **128**(7) (2020)
30. O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, L. Fei-Fei, ImageNet large scale visual recognition challenge. Int. J. Comput. Vis. **115**(3), 211–252 (2015)
31. J. Howard, imagenette. https://github.com/fastai/imagenette/
32. A. Bochkovskiy, C. Wang, H.M. Liao, Yolov4: optimal speed and accuracy of object detection (2020). CoRR, vol. abs/2004.10934

Fwd: eScripts: Decision on your submission "Assistive device based on computer vision" (ID: 7284) in Innovation and Competitiveness in Industry 4.0 Based on Intelligent Systems

Alan Hernandez

Jue 10/08/2023 21:47

Para:Luis Carlos Mendez Gonzalez <luis.mendez@uacj.mx>

1 archivos adjuntos (43 KB)

EAI-acronym_withouttext.png;

---------- Forwarded message ---------

From: **EAI Publishing** <no-reply@eai.eu>

Date: Tue, Jul 26, 2022, 7:18 PM

Subject: eScripts: Decision on your submission "Assistive device based on computer vision" (ID: 7284) in Innovation and Competitiveness in Industry 4.0 Based on Intelligent Systems

To: <alanivan.hh@gmail.com>

Cc: <ivan.perez@uacj.mx>, <abquezad@uacj.mx>, <lguillen@uacj.mx>

EAI eScripts

Dear Alan Iván Hernández Holguín,

The review process for your manuscript

Paper ID: 7284
Title: Assistive device based on computer vision
Authors: *Alan Iván Hernández Holguín, Luis Carlos Méndez-González, Luis Alberto Rodríguez-Picón, IVÁN JUAN CARLOS OLGUÍN, Abel Eduardo Quezada Carreón, Luis Gonzalo Guillén Anaya.
Publication: Innovation and Competitiveness in Industry 4.0 Based on Intelligent Systems

has been completed.

Based on the reviews and the Handling Editor evaluation below, your paper requires major revision.

If you feel you can respond to their remarks (attached below), you can submit revised version on the "My Papers" page on eScripts, or via the following direct link: https://escripts.eai.eu/paper/lists?requireRevision

Please note that upon receiving this e-mail, you will have 45 days (expires on : 10-09-2022) to submit the revised version, otherwise we will assume you are no longer interested in publishing your paper in Innovation and Competitiveness in Industry 4.0 Based on Intelligent Systems and the submission will be withdrawn from the system.

Thank you for considering Innovation and Competitiveness in Industry 4.0 Based on Intelligent Systems as an outlet for your research.

Best Regards,

Publication Department | European Alliance for Innovation
publications@eai.eu | www.eai.eu

Reviewers comments:
*** REFEREE 1 REPORT ***
==

** Below you can find comments on each criterions **
----

1 . Novelty of the contribution
I consider that the manuscript has merit in the topics it addresses. The manuscript is well written and in reasonably understandable language, although there are some spelling errors

2 . Innovation impact of the proposed concept
The social aspect of the presented project turns out to be of significant impact and importance for the readers, so I consider this to be one of the manuscript"s strengths.

3 . Technical content and correctness of the contribution
The authors clearly describe the methodology used, so that it can be replicated by any practitioner in the area of machine learning.

4 . Importance of the manuscript for the thematic area
The importance that the authors have given to the manuscript by focusing the application on a social aspect and centering it on a vulnerable sector of people is what gives it great importance for readers and can be a highly citable manuscript.

5 . Implementation potential of the proposed concept
The authors have already implemented the theoretical concepts into practice by using free and easy to collaborate systems to improve the system.

6 . Completeness of the references
The references in the manuscript are complete and current, so in this sense the manuscript does not have any setbacks.

7 . Quality of the writing
There are some spelling and grammar errors, so the authors must review the document carefully.

** General comments by referee 1 to Author**

When reviewing your manuscript, I consider that it is exciting and has many merits, but how it is written must be corrected,

that is, check the grammar and spelling of the document.

Generate a little more discussion on your results

The conclusion I consider can be improved to highlight what has been obtained as part of the methodological development

*** REFEREE 2 REPORT ***
==

** Below you can find comments on each criterions **
----

1 . Technical content and correctness of the contribution
...

2 . Quality of the writing
...

3 . Novelty of the contribution
...

4 . Importance of the manuscript for the thematic area
...

5 . Innovation impact of the proposed concept
...

6 . Implementation potential of the proposed concept
...

7 . Completeness of the references
...

** General comments by referee 2 to Author**

Dear Author:

I can make the following comments on your manuscript:
1. Improve the quality of the images since, in some cases, the information cannot be perceived, for example, figure 15.

2. It shows in your document that there is an appendix section, but nothing appears in this section. So I ask you to review this section

3. I think that you can review the bibliography a little deeper so that it includes a little more work related to the subject.

4. Perform a grammar and spelling check of the entire manuscript

eScripts

Paper ID: 7284
Title: Assistive device based on computer vision
Authors: *Alan Iván Hernández Holguín, Luis Carlos Méndez-González, Luis Alberto Rodríguez-Picón, IVÁN JUAN CARLOS OLGUÍN, Abel Eduardo Quezada Carreón, Luis Gonzalo Guillén Anaya.

Dear Alan Iván Hernández Holguín,

We are pleased to inform you that your manuscript,has been accepted for publication in Innovation and Competitiveness in Industry 4.0 Based on Intelligent Systems. The reviewers' reports have been attached at the bottom of this email.

Please upload a final, camera-ready version as approved by Editor:
https://escripts.eai.eu/paper/lists?accepted

Please make sure that your paper follows the instructions and is in the correct EAI Endorsed Transactions template. A manuscript in a different template will be sent back for correction.

After signing the copyright agreement we will proceed with the publication of your article.

Thank you for submitting your work to Innovation and Competitiveness in Industry 4.0 Based on Intelligent Systems.

For your efforts as an Author, you receive points based on the average review score of your submission as evaluated by Reviewers and supervised by the Handling Editor. These points count toward your EAI Index.

EAI Index represents a revolution in recognizing excellence in research fairly and transparently, and it is a key element in annual nominations for the member ranks of EAI Fellows, Distinguished Members, and Senior Members. You can view your current EAI Index in your user profile at any time and gain more points by volunteering in other EAI conferences and journals, or by submitting papers to these research venues:

- Bid to Review recently submitted papers in Community Review
- See open positions in conferences and journals
- Explore conferences and journals open for submission

To learn more about EAI's transparent research recognition program, please visit the EAI website.

Best Regards,

Publication Department | European Alliance for Innovation
publications@eai.eu | www.eai.eu

Reviewers comments:
*** REFEREE 2 REPORT ***
==

** Below you can find comments on each criterions **
----

1 . Novelty of the contribution
...

2 . Innovation impact of the proposed concept
...

3 . Technical content and correctness of the contribution
...

4 . Importance of the manuscript for the thematic area
...

5 . Implementation potential of the proposed concept
...

6 . Completeness of the references
...

7 . Quality of the writing
...

** General comments by referee 2 to Author**

Dear author:
I appreciate the answers sent to all my questions and suggestions, I consider that your manuscript
at this stage is ready to be considered as part of this book

*** REFEREE 1 REPORT ***

===

** Below you can find comments on each criterions **
----

1 . Novelty of the contribution
...

2 . Innovation impact of the proposed concept
...

3 . Technical content and correctness of the contribution
...

4 . Importance of the manuscript for the thematic area
...

5 . Implementation potential of the proposed concept
...

6 . Completeness of the references
...

7 . Quality of the writing
...

** General comments by referee 1 to Author**

Dear Author:
By reviewing this new version of your manuscript. I believe that your paper may be published as part of this book.