УДК 004.89
DOI: 10.31891/CSIT-2021-5-5

NATALIYA BOYKO, PAVLO SHYMANSKYI
Lviv Polytechnic National University

ANALYSIS OF ALGORITHMS FOR READING OBJECTS OF INTERFERENCE BY TELEPRESENCE ROBOT

In this paper, we propose the development of a telepresence robot for object recognition. To do this, the authors get acquainted with different reading methods, their image processing speed and accuracy of reading other things and creatures in the environment they provide, then compare and choose the most optimal algorithm for different parameters. The goal is to develop software that allows telepresence robots to read objects of possible interference. The article describes and briefly describes the algorithms for touching the primary SSD model as Fast R-CNN and YOLO. A general description of the SSD model is given. It has also been described in more detail as an SSD model. The process of image processing and the stage of learning the functional model is provided. It was also explained why a solid-state drive is the best model in terms of accuracy and speed, even if the input size of this model is much smaller than the input size of its direct competitor - the YOLO model. In addition, there was a difference in a model building between the two object recognition models. It was described in detail the stage of learning the functional model, what formulas are used in the calculations and what they affect.

Keywords: object reading, image identification, neural networks, video reading, object detectors.

НАТАЛІЯ БОЙКО, ПАВЛО ШИМАНСЬКИЙ
Національний університет “Львівська політехніка”

АНАЛІЗ АЛГОРИТМІВ ЗЧИТУВАННЯ ОБ’ЄКТІВ ПЕРЕШКОД РОБОТОМ ТЕЛЕПРИСУТНОСТІ

В даному дослідженні запропоновані алгоритми розпізнавання об’єктів робота телеприсутності. Для цього в роботі описуються різні методи зчитування об’єктів, їхні швидкості обробки зображення та точність зчитування різних предметів та істот в середовищі. В досліджені проводяться порівняння роботи алгоритмів, які впливають на вибір найбільш оптимального алгоритму по різних параметрах. Метою роботи є розробка програмного забезпечення, за допомогою якого робот телеприсутності зможе зчитувати об’єкти ймовірних перешкод. В роботі охарактеризовано та проведено короткий опис дотичних алгоритмів до представленої у роботі моделі SSD: Fast R-CNN та YOLO. Було надано загальний опис моделі SSD. Також детально описано процес побудови моделі SSD. Наведено яким чином проходить процес обробки зображень та наводяться етапи тренування функційної моделі. В досліджені надаються характеристики переваг та недоліків використання моделі SSD. Визначено, що вона є кращою у плані точності та швидкодії при тому, що її вхідний розмір є набагато меншим за вхідний розмір моделі YOLO. Крім цього наведена різниця у побудові моделей розпізнавання об’єктів. В роботі охарактеризовано процес проходження етапів тренування функційної моделі. Наводиться аналітичне пояснення процесу розпізнавання об’єктів роботом телеприсутності.

Ключові слова: зчитування об’єктів, ідентифікація образів, нейронні мережі, зчитування відеоряду, детектори об’єктів.

Introduction

Computer vision is a scientific field in the field of artificial intelligence and related methods and technologies for obtaining images that contain objects from the real environment, as well as their direct processing and obtaining various data about them, which will later be used in applied tasks [1].

Most robots that can move (independently or with the help of the user) use different methods to read information from the environment. Some read terrain and distances with the help of lidars, others with the help of a camera recognize various objects that can be potential obstacles and determine with a certain accuracy what the objects are.

To facilitate certain types of work, they also use certain mechanisms that can perform certain tasks, such as manipulating certain objects, autonomous navigation, reading objects, and others.

At the moment, there are enough analogues of robots that perform the task of recognizing various objects, but they are aimed only at a narrow specialization. Therefore, the concept of "telepresence robot" is proposed in this robot, which will not only be able to travel with the help of a user who controls it remotely, but will also be able to read various objects on its way and transmit user images to its own screen to achieve "telepresence" effect.

Today there are the following types of tasks in this field:

1. Identification.
2. Object recognition.
3. Object recognition.
4. Assessment of the situation.
5. Text recognition.
6. Generation of objects.
7. Video analysis.

Object recognition is one of the most popular topics in computer vision, so there are many methods by which

you can achieve the desired results in this topic.

Pattern recognition is a general group of non-informative data that are valuable and can be assigned to a certain class due to the selection of essential features[2].

There are many methods by which you can recognize objects, but most often use the following:

1. Use the search method to study the appearance of an object from different angles and scales.
2. With the help of the found contours of the object, its properties are investigated.
3. Use neural networks trained on a large number of examples.

A neural network is a computer system that learns different types of tasks and improves its performance by considering different correct answers, without a program designed for the task [3]. One example of such training is the recognition of images, among which there are pictures with a certain object and among which this object is not, so the network compares the results of learning and improves their results in this task.

Region Based Convolutional Neural Networks (R-CNN) is a family of machine learning models most commonly used in computer vision tasks, namely pattern recognition [4]. Briefly describing how this model works, it is worth noting the following main steps for it: first builds many regions where it is possible to find the image you are looking for, and then conducts a selective search for them to find a particular object in the region. Therefore, due to this algorithm, this model shows good results in reading objects, which is why it is also popular.

Single Shot MultiBox Detector (SSD) is one of the popular object recognition algorithms. The name of this algorithm speaks for itself: “Single Shot” means that tasks such as classification and localization are performed in one run of the neural network. “MultiBox” is the name of the limiting box regression technique. A “Detector” is an image detector that classifies objects found in images. If we briefly describe how this algorithm works, it divides the image into segments using a grid, after which each such cell will be responsible for recognizing objects in this area of the picture.

MobileNet is a fairly simplified architecture that builds lighter convolutional neural networks by separating convolutional layers, thus providing the optimal model for mobile and embedded computer vision programs [5].

Common Objects in Context (COCO) is a data set, a fairly popular database that is often used in pattern recognition tasks [6]. Due to the fact that this data set is open source - it is also used in deep learning programs. COCO contains hundreds of thousands of images, many of which already have tagged objects, which is another reason why this database is so popular.

Analysis of recent sources

Every year robotics develops more and more. The authors [1, 3] create new approaches to solving motion, localization, automation of robots. Many models achieve considerable success in solving various problems. Many technical complexes are designed for military purposes: target detection and elimination. Firefighters' robots are being created; rescuers work, able to get people out of the water from the rubble of fallen buildings. One of the many trends in robotics is the transition from telecontrolled systems, which require constant human participation to perform all the robot's actions, to autonomous systems. The operator only specifies the ultimate and intermediate goals. This is convenient for alien research, where a significant signal delay does not allow remote control [1].

Robots are created so that they can replace humans in difficult working conditions. For example, Google is developing unmanned vehicle technology. This project is led by engineer Sebastian Fran (S. Thrun), a professor at Stanford University. This car has travelled considerable distances with minimal human involvement in its management [2]. Earlier, in 2005, Sebastian Fran's Stanley project team won the DARPA Grand Challenge. The purpose of the competition was to create a fully autonomous vehicle.

To date, the problem of robot autonomy [4, 5] is very relevant. In their works, the authors [5-7] consider the robot's autonomy for the perception of the environment. Knowing the map of the area, the robot will be able to quickly determine the location of objects in space. One of the difficulties arises when the robot has no idea of the terrain and does not know its coordinates. In this case, he needs to make movements and create a map using various sensor devices and algorithms.

In this paper, the authors propose the development of object recognition robot telepresence. To do this, they learn about different methods of reading objects, their image processing speeds and the accuracy of reading other things and creatures in the environment they provide, then compare and choose the most optimal algorithm for different parameters.

The work aims to create software for recognizing objects with reliable accuracy, which will appear in the path of telepresence using machine learning methods.

To achieve this goal, the following tasks are solved:

1. Learning object recognition algorithms.

2. Implementation of an algorithm for object recognition with a sure accuracy using machine learning methods.

3. Study of the influence of parameters from the implementation of the algorithm by different methods of machine learning.

The object of research is algorithms for object recognition and methods for identifying such things.

The subject of the study is the MobileNet SSD model, which is designed for object recognition.

The scientific novelty of this study is the development of software for object recognition by telepresence of a robot using a machine file.

The theoretical basis for writing this work was foreign authors' works on the problems of automation and localization in robotics [6-9].

Presenting main material

To implement object recognition by a telepresence robot, you need to create a program that will directly detect these objects while showing good performance results and, most importantly, pattern recognition accuracy. Therefore, it was first necessary to compare and analyze different algorithms for reading things to determine which is most likely to work best on the mobile robot.

The main task of the work is to implement the most relevant robot algorithm for pattern recognition and comparative analysis, which will result in the accuracy of recognition. It will be argued that such an algorithm is most suitable for mobile devices or applications equipped with computer vision.

The implemented algorithm will be tested for efficiency with the help of video, which will present various objects, from people, transport, animals and ending with things that surround us in everyday life.

Because the program will not only recognize objects and circle them in frames but will also give assumptions about the names of these objects, as well as indicate the numerical value of this assumption, i.e. if, for example, a kitchen knife is depicted. The program will assume that it is scissors and will highlight in numbers the value of this assumption. It will be possible to understand better how wrong this assumption is.

In this way, it will be possible to correctly calculate the number of incorrect assumptions and, accordingly, correct ones, which will only improve the accuracy of comparative analysis of object recognition algorithms.

Information model [11] is a model that characterizes the features and states of an object, process or phenomenon, as well as demonstrates the relationship with the environment.

Information models are classified into the following categories:

1. Field of use.
2. Taking into account the time factor in the model.
3. The method of presenting models.
4. Language of description.
5. Implementation tools.

Based on the article's topic, it can be argued that the field of use is research because it is created to study specific characteristics.

If we turn to the category of "time factor", then for this work, the time factor is not of great importance, is it belongs to the type of static models, because over time, the result will not change.

The proposed information model refers to the formal ones where the research object is the accuracy of pattern recognition using a specific algorithm.

There is only the following limitation of the input data for this model: it will read only video because it will be a more realistic approach, as this algorithm should be used in the robot of telepresence. An example of the input video is shown in Fig. 1 a.

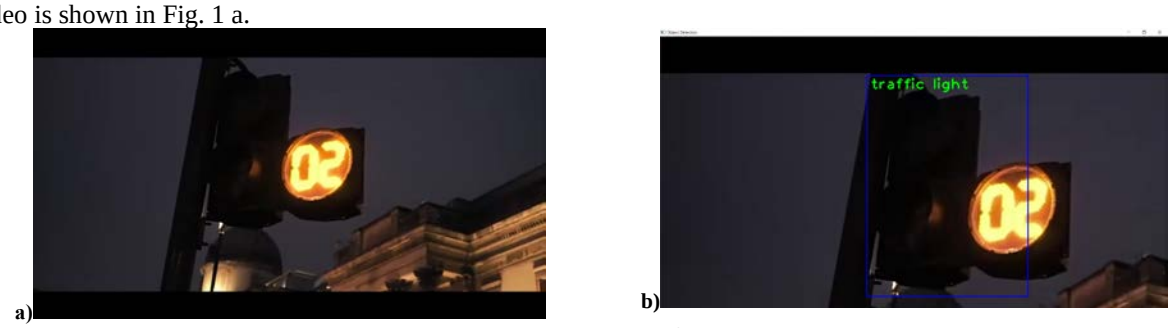

Fig. 1. a) Input video; b) Example of the output video

Figure 1 a shows a snippet from a demo video that has not yet been processed and that various objects have not yet been recognized. In Fig. 1 b shows an example of the program's output window, which clearly shows the recognition of multiple things.

The video is processed in the program, then cropped, and each of the frames of this video will be covered with a grid that divides the image into regions, which will further recognize where each such segment for the presence of different objects and then open a new window in which the processed video will be broadcast, and it will show the result of the program.

To start reading and processing the video sequence, which will later be displayed to the user as a video in which the program has already recognized the images, you must connect the OpenCV library.

Open Source Computer Vision Library (OpenCV) [12] is a library that provides tools for processing and analyzing image content. This library is sound because it allows for all the tools required to perform tasks related to image reading, text recognition or motion tracking, and more. It is written in the C ++ programming language, but it

can also be used in Python, Java, Ruby and others. It is also worth noting that it can be freely used for academic and commercial purposes, so it was chosen for this thesis.

After installing the OpenCV library, you must also download the frozen graph of conclusions for the program to work correctly.

Freezing the graph [13] identifies and stores such necessary items as scales, charts, and others in one file, which can be easily applied in the program. This process is performed to lighten the load and reduce the total number of calculations. This eliminates unnecessary metadata, gradients, and learning values by packing all this data into a single file that can be saved to disk.

A .txt file has also been created, which will store the names of all possible objects that the model can recognize. The model can recognize up to 80 different classes of COCO data set items, which can be other types of machines and household items and living objects such as people, dogs, cats and so on. In Fig. 2, you can see the names of those objects and objects that the model can recognize.

```
['person', 'bicycle', 'car', 'motorbike', 'aeroplane', 'bus', 'train', 'truck', 'boat', 'traffic light', 'fire hydrant', 'stop
sign', 'parking meter', 'bench', 'bird', 'cat', 'dog', 'horse', 'sheep', 'cow', 'elephant', 'bear', 'zebra', 'giraffe', 'backpa
ck', 'umbrella', 'handbag', 'tie', 'suitcase', 'frisbee', 'skis', 'snowboard', 'sports ball', 'kite', 'baseball bat', 'baseball
glove', 'skateboard', 'surfboard', 'tennis racket', 'bottle', 'wine glass', 'cup', 'fork', 'knife', 'spoon', 'bowl', 'banana',
'apple', 'sandwich', 'orange', 'broccoli', 'carrot', 'hot dog', 'pizza', 'donut', 'cake', 'chair', 'sofa', 'pottedplant', 'be
d', 'diningtable', 'toilet', 'tvmonitor', 'laptop', 'mouse', 'remote', 'keyboard', 'cell phone', 'microwave', 'oven', 'toaste
r', 'sink', 'refrigerator', 'book', 'clock', 'vase', 'scissors', 'teddy bear', 'hair drier', 'toothbrush']
```

Fig. 2. List all possible recognizable objects

In the COCO data set, images are taken from everyday life, a handy feature because they provide "context" to objects. This characteristic feature will make it easier to describe the environment in which a particular image was taken, as additional photos expand the general information.

In addition to the above-described feature of the COCO data set, it should also be noted that it allows for good marking and segmentation of objects in the image, making it much more intelligent to use such an image when performing a practical machine learning task, as it will improve detection efficiency and accuracy in this model.

It is worth noting that images in COCO and semantic segmentation also undergo panoptic segmentation. Panoptic segmentation is one of the image segmentation methods used for Computer Vision (CV) tasks [14]. The difference between these two segmentations is that the first (semantic) divides into regions and marks them with a specific color, but if, for example, in the image, there were several identical objects, such as three dogs. Still, different breeds, then this method is all 3- oh sketches in one color. In the case of panoptic segmentation, the main difference is that this method would draw those dogs in different colors, which gives a significant improvement in accuracy for the model that will read objects.

The functional model for this research should have the following functions:

1. Obtaining video input data.

The input video can be shot directly by the camera, or you can download a familiar video. The proposed method will use the second method to show that the program recognizes various objects.

2. Processing of the input video sequence.

The program must process the video sequence, break it into areas with a grid, and go to the next step.

3. Object recognition.

In each of the areas created by partitioning the image with a grid - to carry out the process of recognizing the site for the object's presence and if the desired thing is present, then admit it.

4. Output of work results.

When the recognition process starts, a program window will open, which will display the input video sequence, but which has already passed the recognition process. It will show how the program recognized the objects in Fig. 3. The state diagram for the graphical representation of functions of the functional model is presented.

Obtaining video input:

– By using the OpenCV library, the video will be downloaded or read by the camera, depending on the situation;

– After processing the video sequence by the OpenCV library, the algorithm will proceed to the next step;

Input video processing:

- The algorithm will split the image using a grid.
- After applying the grid, the image will be divided into regions.

Object recognition process:

- Obtained after partitioning the area undergoes the process of object recognition.
- In case of finding the necessary object - recognize it and select it.

The process of displaying the results of the program:

- A program window will open displaying the processed input video sequence.

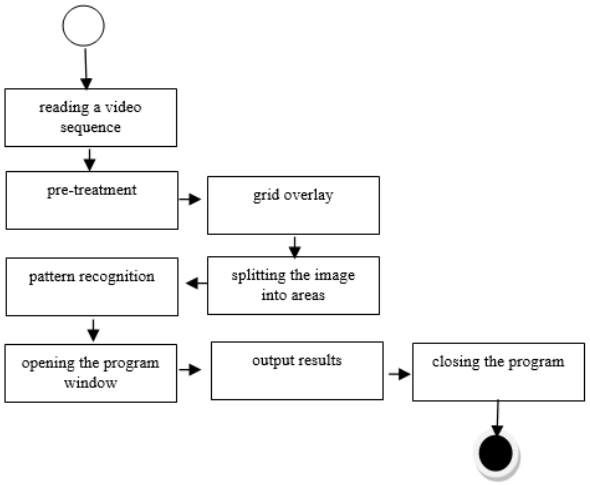

Fig. 3. State diagram

Since the SSD uses only a single detector, it provides excellent results in terms of speed, so this algorithm assumes the boundaries of fields and classes directly from the function maps in one pass. For better accuracy, this method uses small convolutional filters, which are used to predict object classes and shift to boundary fields.

Because the SSD is so fast to process, this algorithm is often used in tasks where you need to recognize objects in real-time, so this algorithm was chosen to recognize images by a telepresence robot. Although the SSD is similar to the Fast R-CNN algorithm, it speeds up this procedure much faster. When in turn, the process in Fast R-CNN runs at 7 frames per second, which is much lower than the requirements for real-time recognition.

If we describe in more detail the operation of the SSD [15], it is worth telling the steps of the recognition process. So first, VGG16 removes function maps, then detects objects with convolutional layers. Then 4 object predictions are made for each cell. Each prediction consists of a boundary window and 21 points for each class (the highest type is chosen for the constrained object). And if the SSD did not detect any objects, it reserves a class of "0" to indicate where no objects were found.

After removing the feature maps, the SSD applies 3 × 3 convolution filters to each cell to make predictions. It is worth noting that these filters work in the same way as conventional CNN filters.

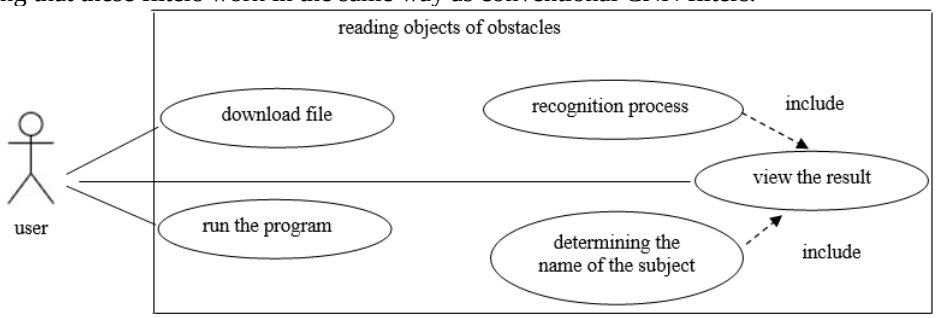

Fig. 4. Usage diagram

In Fig. 4 it is possible to observe how the user will use the program directly, which will work later in the telepresence robot.

Many different object recognition algorithms benefit from each other in various parameters and aspects. Therefore, to choose which of these methods is most suitable for the "eyes" of the telepresence robot, it was necessary to analyze them for advantages and disadvantages.

1. Fast R-CNN

It is a learning algorithm for object detection based on the method of a fast convolution network based on regions. The difference between this algorithm and similar R-CNN and SPPnet is that it considers the shortcomings of these methods and solves them in its implementation. The disadvantages of the above-mentioned parallel algorithms were in accuracy and speed - this algorithm shows much better results.

Advantages of this method:

1) Compared to plain R-CNN and SPPnet shows much better accuracy values.

2) Training is short, only one stage and uses multitasking losses.

3) All network levels can be updated through the learning process.

4) Does not use disk storage to cache functions.

2. You Look Only Once (YOLO)

This algorithm [16] is one of the most popular for solving object recognition problems. The base model

YOLO processes images in real-time at a speed of 45 frames per second, which is quite impressive. But in addition, there is an even smaller version of this network, which was able to surpass the above. Its speed is 155 frames per second. Thus, this algorithm exceeds other image detection algorithms, such as Deformable part models (DPM) and R-CNN. But it should be noted that this algorithm does not work correctly with mobile devices and robotics, so when choosing the algorithm for this thesis, SSD was chosen because it works well with mobile devices.

In fig. 5 you can see a brief description of how this algorithm works:

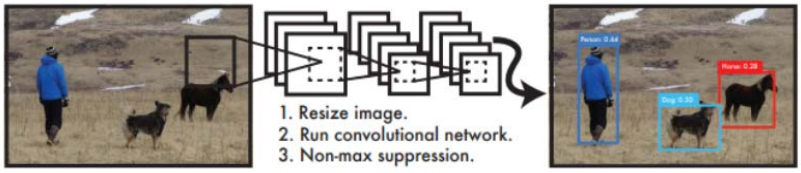

Fig. 5. Image recognition process using YOLO

First, YOLO resizes the image so that filters can be applied, after which a convolutional neural network is started, which performs object recognition. At the end of this algorithm, there is no maximum restriction.

3. SSD

The SSD only needs an input image and the correct groups of objects specified during training. Then use the convolution to estimate a small set of the same default groups. Still, they will have different aspect ratios in each location on other function maps with different scales.

These steps are well illustrated in Fig. 6:

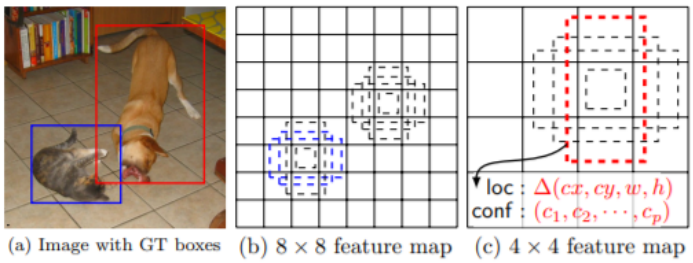

Fig.6. The process of image processing using an SSD

The algorithm then begins predicting the shape offset for each default field and the identification for all object categories. The training first combines the default fields, which contain the correct information about the objects.

It is also worth describing how this model is built. The SSD model is similar in structure to the R-CNN model, with the only difference that the first fully truncated core network was joined by folded layers, which gradually decreased in size. Thanks to this feature, the SSD model can detect and predict objects at different scales. It is also worth noting that each added functional layer can create a fixed set of detection predictions in this model. With the help of such groups and convoluted filters will perform their function. In Fig. 7 they are listed above the SSD model. For some particular layers having a size m × n with p channels, the main prediction element for such media is the potential detection parameter. This is a 3 × 3 × p core, created by either an estimate for the category or an offset of the shape relative to the standard window coordinates. Thus, the initial value is obtained in those places where the kernel will be used for each of them × n places.

In Fig. 7 you can see what the SSD model looks like. This figure clearly shows where exactly several end layers are added to the end of the leading network.

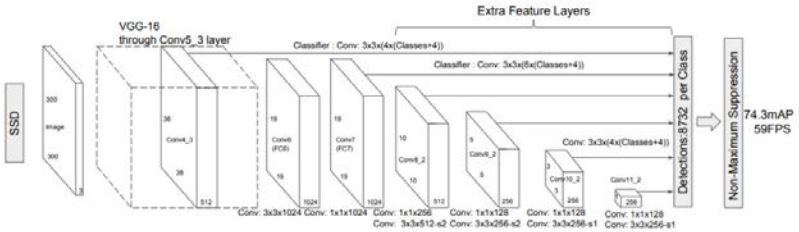

Fig. 7. SSD model

Layers that have been added to the end of the core network involve shifting to fields of different scales and sizes and associated predictive confidence.

By default, a set of bounding boxes is combined with each cell of the object map at the top of the model. By default, fields enclose a map of objects convolutely. Due to this, the position of each area becomes fixed according to the cells. For each such cell, an offset is provided relative to the standard window shapes in the cell. In addition, class evaluations also indicate that the desired example of a class is present in such fields. In Fig. 6 you can see the images

of the bounding boxes, they are similar to R-CNN, but they are applied to several function maps with different resolutions. Due to the permission to use other forms of function maps, this permission allows you to discretize the space of possible output boxes effectively.

It is also worth briefly showing the results of the comparison between the SSD and the YOLO model and, in the end, make sure that the SSD model provides the best accuracy results. In fig. 8 you can see a comparison of these models.

Although the SSD model has an input size of 300 × 300, while the YOLO model has an input size of 448 × 448, the SSD exceeds its opponent inaccuracy and speed. In fig. 8 shows that the end of the Single Shot Detector produces 59 frames per second, and YOLO only 45 frames per second. These measurements were taken from the VOC2007 test [17].

Training in this model also has its feature. The SSD assigns accurate information to specific outputs in a fixed set of detector outputs. When the model has coped with the last step, the assignment of loss functions follows, the method of error backpropagation is performed at the end.

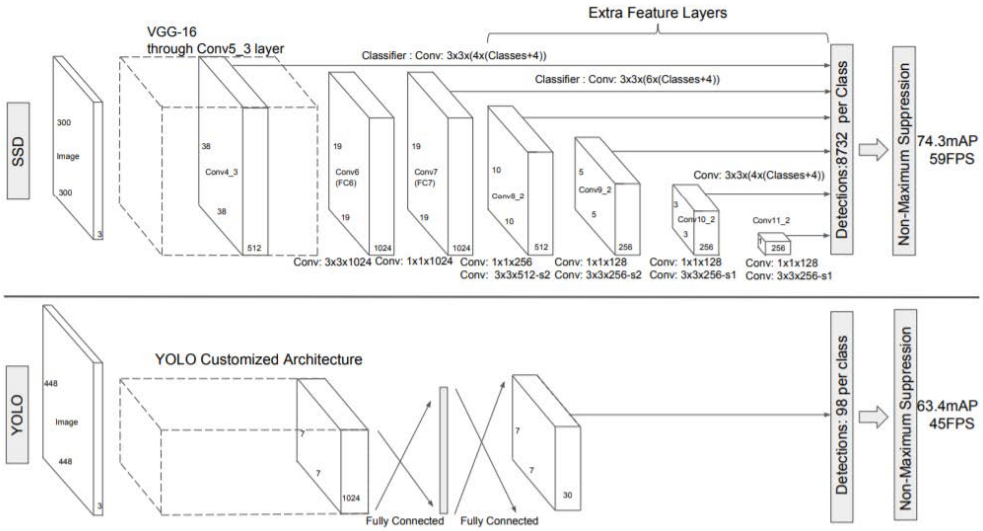

Fig. 8. Schematic comparison of SSD and YOLO models

Let $x_{ij}^{p} = \{1,0\}$ - indicator for matching the *i*-th default window from the *j*-th truth field of category *p*.

We will have the following: $\sum_i x_{ij}^{p} \geq 1$. Thus, the total loss function is the weighted sum of localization losses and loss of confidence (Formula 1).

$$L(x, c, l, g) = \frac{1}{N}(L_{conf(x,c)} + \alpha L_{loc(x,l,g)}), \quad (1)$$

where *N* is the number of default fields. If *N* = 0, then the loss is also set to 0. The parameter l is the assumed block. The value of g is a parameter of the main window. Similar actions are found in the R-CNN model, where regression to the offsets for the center (*cx, cy*) of a typical constraint window and its width and height values is also performed.

Loss of confidence is the loss of "softmax" due to the confidentiality of several classes (*c*) (Formula 2).

$$L_{conf(x,c)} = -\sum_{i \in pos}^{N} x_{ij}^{p} \, log(\, c_i^{p}) - \sum_{i \in Neg}^{N} log(\, c_i^{0}), \quad (2)$$
$$c_i^{p} = \frac{exp(\, c_i^{p})}{\sum_p exp(\, c_i^{p})},$$

where by cross-confirming the value of the weight term α becomes equal to 1.

For detection, the SSD model uses both the lower and upper function map, which is designed for detection. In fig. Figure 7 shows two examples of exemplary functions, sizes 4 × 4 and 8 × 8, which are used in the framework. But in fact, use much more of these with small computational overhead. It is known that the network at different levels has completely different maps of objects. They have different empirical sizes of favourable fields. But due to the peculiarities of the model, the bounding box should not correspond to the actual susceptible areas of each layer. They first create specific function maps that have learned to respond to and recognize precise scales of objects to do this.

The Python programming language and the OpenCV library were used in this study. More details on all tools for the study are listed in Table 1.

Table 1

Development tools

Library / Tool	Using

OpenCV 3.4.1	Object reading tools
Frozen_interference_graph.pb	Scales and model configurations
COCO Dataset	Classes of readable objects
ssd_mobilenet_v3_large_coco_2020_01_14.pbtxt	SSD model architecture
Jupiter notebook 6.1.4	Writing and testing the program

The libraries and tools listed in Table 1 are quite critical for program development. Only if, for example, there is a situation when it is not possible to get a ready-made SSD model, then there will be a way to train this model again, which in turn will spend extra time.

The main requirements for the software are given in Table 2, which briefly describes the characteristics of the computer for the development of the robot program.

Table 2

Software requirements

Characteristic	Value
Operating System	Windows 10 Home
Processor type	Intel Core I5-7200U
Clock frequency	2,5 GHz
Number of cores	2 cores
Number of threads	4 threads
The amount of RAM	8 GB
Type of RAM	DDR4
Video card type	Nvidia GeForce GTX 950M
Video memory	4 GB
A programming language	Python 3.8.5

If you omit the steps with downloading SSD model architecture files, scales and configuration of this model, as well as the COCO data set, you can select the following parts of the program:

- Video processing.
- Identification and recognition of objects.
- Demonstration of the program.

During the video processing stage, the program will start its work after the user specifies the name of the video file and, accordingly, starts it. The video processing procedure itself is the division of video frames into certain areas or regions, in which the presence of objects will be checked in parallel with each other, after which the SSD model in case the object was found will "notice" it and determine it with its own trained knowledge, after which the frame of this object will be distinguished not only by a certain frame, but also by the name given to it by the SSD model.

The process of object recognition has been described in more detail above. Additionally, it is necessary to provide numerical characteristics of accuracy and quality of models as alternatives. And it is difficult to say which of the models is the best, because often for real programs make a choice to get a good balance between accuracy and speed. In Table. 3 shows the average accuracy and number of frames per second for the methods described above.

Table 3

Comparison of recognition methods

Method name	Average accuracy of pattern recognition	Number of frames per second
YOLO	73,2	7
Fast R-CNN	66,4	21
SSD	74,3	46

From table 3 you can see that the best results are obtained by the SSD method. It provides the best frame rate per second - 46, which is much higher than the YOLO or Fast R-CNN method. These values were obtained during testing of various detectors for accuracy and speed in various tests, one of which was Pascal VOC 2007 [17].

The highest value of the number of frames per second was recorded in the YOLO method and was as much as 91 frames per second, which is a very good result. In the SSD method, the highest recorded value was 59 frames per second, and the lowest 22.

It is also worth mentioning the size of the training sample, which was used to train these models. Data from the training sample can be observed in the following table 4:

Table 4

Data from the training sample

Parameters	SSD	YOLO
Training epochs	176000	5000
Batch size	From 6 to 10	From 4 to 8
Learning rate	0.001, 0.004	0.0001, 0.01

Table 4 clearly shows that the SSD model went through much more training epochs, which qualitatively later reflected on the recognition accuracy, which was shown in Table 3. The SSD model compared to the YOLO model studied with more images, from 6 to 10.

Table 5

The number of true and false predictions of SSD and YOLO models

Model	Truly positive prediction	False positive prediction	False negative prediction
SSD	168	3	115
YOLO	204	94	79

Table 5 shows the number of true and false predictions of SSD and YOLO models. It can be seen that the SSD model performs much better in false positive predictions than the YOLO model, which indicates that its feature recognition filters work correctly and rarely give incorrect predictions.

The next part of the program is a demo window, which will display the downloaded video, which was previously processed by the SSD model, resulting in the frames of this video file will show a frame of a certain size (depending on the object) around a particular object, which model SSD was able to recognize. Also, in addition to the framework, the model will be the name of the subject, the choice of this name will be based solely on knowledge of the model obtained during training, so you can not be one hundred percent sure that the prediction will be correct.

In fig. Figure 10 shows what the program demo window looks like when the program successfully processed a video file and recognized certain objects in this video sequence:

Fig. 10 Demonstration of the program

In fig. 10, the program successfully recognized the bus and highlighted it in the video.

Because the user will not work directly with this algorithm and SSD model, no user interface was provided for this. Because this implementation should work directly in the camera, the presence of telepresence is to be autonomous.

The only opportunity for the user of the final product is the presence of telepresence, to see the program's work that recognizes objects will be only when controlling the robot and watching the broadcast of its camera, in parallel with which the SSD model will work and recognize images.

The user will also need to change the video file's name, which will be specified in the program code, because, most likely, his video file will have a completely different character. After all the preparatory steps have been completed, the user must run the program code. The program will open a window in which the user's downloaded file will be displayed, but which has already been processed and has found objects that the program uses can recognize.

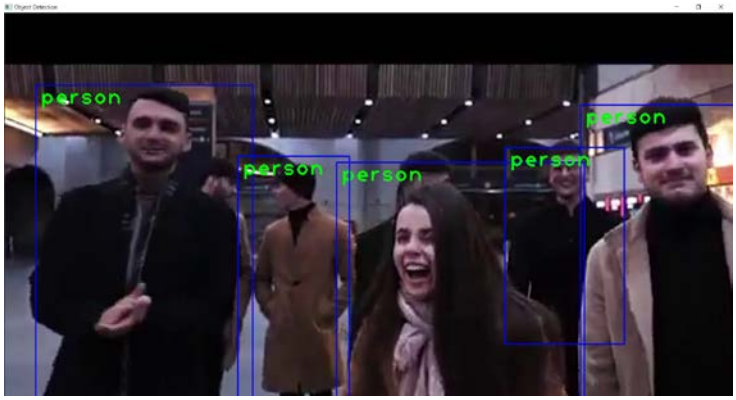

Fig. 11. Comparison of the resolution of the program demo window

When the program window opens, the user will have several options for what to do next. The first option is to watch the playback of the processed video and make sure the program works. After the timing of the video reaches its logical conclusion, the program window will close, and in the console of the program in which the code was run, all numbers of found objects will be displayed. Recognize this SSD model.

Depending on the extension of the downloaded video file with this extension, the program window will open because this program will be programmed in the future in the robot of telepresence, which will have its video camera, which will have a fixed value of the expansion of the playable video files.

Therefore, in the program at this implementation stage, the processed video using the SSD model is configured to expand the video sequence loaded into the program. In fig. 11 clearly shows how the program displays high-definition video. Thus, depending on which extension the video file will be processed, there will be a software demo window with such a resolution.

Figure 12 shows how the software demo window displays the processed low-resolution video sequence:

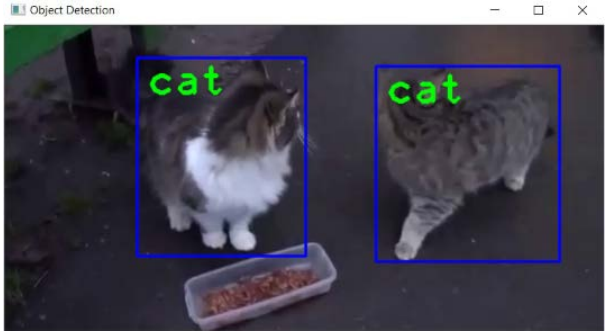

Fig. 13. Comparison of the resolution of the demo window of the program

The second option for the user is to temporarily close the program window in case the video viewing is sufficient. To do this, while displaying the processed video, press the "E" key on the keyboard. After that, the demo window of the program will close, and the user will be able to change the name of the video file in the program code or start the program again.

Conclusions

Thus, the work characterised the subject environment, and the main task for this research was formulated. This task was the implementation of an algorithm that will later be used in the robot of telepresence. The purpose of this algorithm was to identify objects that could be potential obstacles to the robot. The implementation of this algorithm was found using the SSD model, the operation of which was described in detail in the second section.

In the presented research, the analysis of the subject area in which it is described in what field of use, the language of the description and a way of representation of models was carried out. The design of data structures was also carried out, where it was described in more detail and explained how the program reads and receives data.

In addition, a functional model was designed, described how the main stages of this software implementation, and for a better understanding of the UML state diagrams, which provides a general view of the program. In addition, the mathematical and algorithmic support of this study was explained, similar detectors of objects were given, the basic principle of their operation was described and why the SSD method was chosen.

In addition, an analysis of software implementation was conducted. Describes how the processes of reading, processing and output of the boot video. The program guide was also provided. This guide explains how the user can interact with the program, namely run it, download the video file that the program should process, how the demo window will open and how to close the window.

A comparison was made between three detectors of objects considered to be quite good. It was determined which methods show the best results of accuracy and speed, and the corresponding data are presented in tables and figures.

REFERENCES

1. What You Need To Know About Telepresence Robots: What They Are and Use Cases // [Electronic resource] OhmniLabs Writer. – 2021. - Access mode: https://ohmnilabs.com/content/what-to-know-about-remote-telepresence-robot/
2. Hancock E. Pattern Recognition // [Electronic resource] Journal Pre-proof, Vol. 123 – 2021 - Access mode: http://csitjournal.khmnu.edu.ua/
3. Hwang S., Wug Oh S., Kim S. J. Single-shot Path Integrated Panoptic Segmentation // [Electronic resource] Computer Vision and Pattern Recognition. – 2020. – Access mode: https://arxiv.org/abs/2012.01632
4. He K., Gkioxari G., Dollar P., Girshick R. Mask R-CNN // [Electronic resource] .- 2018. – pp. 1-17. - Access mode: https://arxiv.org/pdf/1703.06870.pdf
5. Girshick R. Fast R-CNN // [Electronic resource] arXiv e-prints. – 2015. – Access mode: https://arxiv.org/pdf/1504.08083.pdf
6. Min Read J. S. An Introduction to the COCO Dataset // [Electronic resource] Roboflow Blog. - 2020. - P. 17. – Access mode: https://blog.roboflow.com/coco-dataset/
7. Amazon.com: Brookstone Rover 2.0 App-Controlled Wireless Spy Tank: Toys & Games // [Electronic resource] Amazon.com. – 2020. - P. 1. – Access mode: https://www.amazon.com/Brookstone-Rover-App-Controlled-Wireless-Tank/dp/B0093285XK
8. Double Robotics - Telepresence Robot for Telecommuters // [Electronic resource] Double Robotics – 2021. - P. 2. – Access mode: https://www.doublerobotics.com/double2.html
9. Beam // [Electronic resource] Beam. – 2021. - P. 1. – Access mode : https://suitabletech.com/products/beam.
10. Amazon.com: Appbot Riley Home Safety Movable Camera Robot: Camera & Photo // [Electronic resource] Amazon.com – 2021. - P. 1. – Access mode: https://www.amazon.com/Appbot-Riley-Controlled-Movable-Safety/dp/B01LWXF28H.

11. Gandhi R. R-CNN, Fast R-CNN, Faster R-CNN, YOLO — Object Detection Algorithms // [Electronic resource] Toward data sience. – 2018. – Access mode: https://towardsdatascience.com/r-cnn-fast-r-cnn-faster-r-cnn-yolo-object-detection-algorithms-36d53571365e

12. Redmon J., Divvala S., Girshick R., Farhadi A. You Only Look Once: Unified, Real-Time Object Detection // [Electronic resource] arXiv e-prints. – 2016. – Access mode: https://arxiv.org/pdf/1506.02640v5.pdf

13. Freeze Tensorflow models and serve on web // [Electronic resource] CV-Tricks.com – 2017. - P. 1. – Access mode : https://cv-tricks.com/how-to/freeze-tensorflow-models/ .

14. Shiledarbaxi N. Guide to Panoptic Segmentation +A Semantic + Instance Segmentation Approach // [Electronic resource] Analytics India Magazine – 2021. – Access mode: https://analyticsindiamag.com/guide-to-panoptic-segmentation-a-semantic-instance-segmentation-approach/ .

15. Hui J. SSD object detection: Single Shot MultiBox Detector for real-time processing // [Electronic resource] Medium – 2020. - P. 1. – Access mode: https://jonathan-hui.medium.com/ssd-object-detection-single-shot-multibox-detector-for-real-time-processing-9bd8deac0e06.

16. Choudhury A. Top 8 Algorithms For Object Detection One Must Know // [Electronic resource] Analytics India Magazine – 2020. - P. 1. – Access mode: https://analyticsindiamag.com/top-8-algorithms-for-object-detection/.

17. Hui J. Object detection: speed and accuracy comparison (Faster R-CNN, R-FCN, SSD, FPN, RetinaNet and…) // [Electronic resource] Medium – 2020. P. 1. – Access mode: https://jonathan-hui.medium.com/object-detection-speed-and-accuracy-comparison-faster-r-cnn-r-fcn-ssd-and-yolo-5425656ae359.

Nataliya Boyko **Наталія Бойко**	Ph.D., Associated Professor at the Department of Artificial Intelligence, Lviv Polytechnic National University, Lviv, Ukraine e-mail: Nataliya.I.Boyko@lpnu.ua https://orcid.org/0000-0002-6962-9363 Scopus ID: 57191967462	Доцент кафедри Системи штучного інтелекту Національного університету “Львівська політехніка”
Pavlo Shymanskyi **Павло Шиманський**	Student at the Department of Artificial Intelligence, Lviv Polytechnic National University, Lviv, Ukraine e-mail: pavlo.shymanskyi.kn.2017@lpnu.ua	Студент кафедри Системи штучного інтелекту Національного університету “Львівська політехніка”