

SAVONIA

University of Applied Sciences

THESIS – BACHELOR'S DEGREE PROGRAMME
TECHNOLOGY, COMMUNICATION AND TRANSPORT

OCCUPANCY MONITORING SYSTEM

Using object detection

AUTHOR/S Arina Kondrateva

Field of Study Technology, Communication and Transport	
Degree Programme Degree Programme in Information Technology, Internet of Things	
Author(s) Arina Kondrateva	
Title of Thesis Occupancy monitoring System	
Date 31 May 2022	Pages/Number of appendices 40 / 2
Client Organisation /Partners "Peremena", Sortavala, Russia	
<p>Abstract</p> <p>This thesis is based on creating an algorithm for counting visitors to a public place using an object detection, as well as conducting related research. The basic motivation for the thesis is medical reasons, namely an algorithm capable of reducing the spread of respiratory viral diseases indoor. During a pandemic, there is a high risk of contracting respiratory viral diseases. Rooms that are crowded with people present a particularly high risk. In such places it is impossible to keep enough distance to be considered safe. Thus control of the occupancy of public places can help in the prevention of illnesses.</p> <p>This goal was achieved by controlling the allowable number of people in a certain area. Thus, the thesis presents the work done on the creation of an algorithm that reads video from cameras. The counting of the number of people was carried out using neural networks that allow to detect a person in the frame for further tracking of movement. When creating the algorithm, an already trained YOLOv3 neural network was applied. Also, all the necessary research on the topic has been done and can be found in the thesis.</p> <p>The results obtained during the research and development of the program are satisfactory. They demonstrate the ability to track a person's movement by using cameras and neural networks. As well as the possibility of further analysis for counting people. The thesis presents the tests carried out on the algorithm as well as the analysis of the results.</p>	
<p>Keywords</p> <p>Iot, Object Detection, Neural Network, Image analysis, It</p>	

CONTENTS

1	INTRODUCTION	6
1.1	Structure	6
1.2	Goals... ..	7
1.3	Methods	7
2	PROJECT RELEVANCE	9
2.1	General use	9
2.2	Comparison with other methods of control	9
2.2.1	Human eye control	9
2.2.2	Physical control with turnstiles.....	9
2.2.3	Physical control with infrared ray	10
2.2.4	Administration control	10
2.3	Motivation	10
2.3.1	Medical use	10
2.3.2	Business and marketing use	12
2.3.3	Use for social research purposes	12
2.4	Specific example	12
3	BACKGROUND INFORMATION	14
3.1	Neural Network.....	14
3.1.1	Neural network capabilities	14
3.1.2	Using a neural network for current purpose	14
3.1.3	How neural network works.....	14
3.1.4	Python and neural network	16
3.2	Object detection	16
3.2.1	How object detection works	17
3.2.2	Object detection in current project	20
4	IMPLEMENTATION.....	21
4.1	OpenCV.....	21
4.2	YOLOv3-tiny	21
4.3	Occupancy monitoring algorithm.....	22
5	TESTING AND DISCUSSIONS.....	27

5.1 Tests	27
5.2 Upgrade opportunities.....	30
5.3 Integration into existing systems	30
5.4 Risk analysis.....	30
6 CONCLUSION	32
6.1 Key points	32
6.2 Evaluation of results.....	32
REFERENCES.....	33
APPENDIX 1: SCHEMATIC OF THE ALGORITHM	35
APPENDIX 2: PROGRAM.....	36

LIST OF FIGURES

FIGURE 1. Demonstration of factors influencing the spread of respiratory viral diseases. (Bazanta, and Bushb 2020.).....	11
FIGURE 2. Schematic representation of a case where an automatic people counting system can be used. ...	13
FIGURE 3. Similarities between neural network processes and human intelligence. (Tariq 2016.).....	15
FIGURE 4. Diagram of the architecture of neurons in the human brain. (Tariq 2016.).....	15
FIGURE 5. Neural network architecture diagram showing layers and connections. (Tariq 2016.)	16
FIGURE 6. Two-level neural network (Murthy, Hashmi, Bokde, and Geem, 2020.).....	17
FIGURE 7. Two concepts of architectural object detection (Iyer, Bhensdadiya and Ringe 2021.)	18
FIGURE 8. How Yolo works. (Kajabad and Ivanov 2019.).....	19
FIGURE 9. NMS (non-max suppression) technique. (Joshi, Nama, Shah, and Vijayakumar 2020.).....	19
FIGURE 10. Human detection on the image	20
FIGURE 11. Yolo's demonstration (Yolo 2022.).....	21
FIGURE 12. Schematic algorithm.....	22
FIGURE 13. Getting a video on the input and loading Yolo.....	22
FIGURE 14. Loop processing input video	23
FIGURE 15. Get output from Yolo.....	23
FIGURE 16. Go through each detection done by Yolo	24
FIGURE 17. Finding center point	24
FIGURE 18. Find closest track to new point.....	25
FIGURE 19. Check tracks to be deleted	25
FIGURE 20. Check person's direction	26
FIGURE 21. Main concepts of algorithm.....	26

FIGURE 22. One human detection	27
FIGURE 23. Several humans' detection error	28
FIGURE 24. Several humans' detection success	28
FIGURE 25. Cross-moving detection	29
FIGURE 26. Human with hat detection	29

1 INTRODUCTION

The work on the thesis is based on the creation of the program for occupancy monitoring with object detection, as well as the analysis and collection of information necessary for its full functioning and achievement of the established goals. In the form of a project, a request was received for the creation of an Automatic system for controlling people in a public place. The main idea is to develop an algorithm that analyzes video images received from the camera. During this analysis it is necessary to detect the presence of a person and the direction of movement, so that based on this data the system would determine the number of people on the public premises by counting those who entered and left. It is assumed that the video image will cover only a minor area at the entrance, which will be enough to count visitors entering and leaving the premises. Moreover, the algorithm must be able to integrate with already existing systems in public places. Therefore, if the location at present has a suitable camera to receive an image from it, there is no need to install a new one, which makes the system more flexible and saves money.

In this thesis a public space means any indoor area that wants to keep track of the number of visitors. In turn, occupancy monitoring system can help to adjust the situation during a pandemic and viral diseases. Thence, for example, a museum may wish to use this application during the period of coronavirus exacerbation to control the number of people in a certain area and reduce the spread of the disease. Moreover, such a system can be operated by cafés, restaurants, shops, and much more. (Finnish institute for health and welfare 2022.)

More details about the principles of operation and usefulness of the thesis are described in this paper.

1.1 Structure

The structure of the thesis follows the work on these areas. The thesis is divided into six chapters that go in a clear sequence. These chapters describe three stages of work. They can be roughly described as follows: research before implementation, implementation, and data analysis after it.

- Pre-implementation (Chapters 1, 2, 3)
The first part in the structure describes the introduction, the collection of data that will serve as a basis for the thesis and personal brainstorming on various issues before the implementation of the work on the project algorithm. This is preparation for the core part of the thesis.
- Implementation (Chapter 4)
This part describes the essential product obtained during the thesis work - the algorithm and the program. And shows which way it works and how it was implemented.
- Post- Implementation (Chapters 5, 6)
This is the final part in which the work on the data obtained after the implementation was carried out and demonstrated. It includes a description of the tests of the algorithm and

their analysis, as well as a variety of discussions on the results of the project and evaluation.

1.2 Goals

The key goal of the thesis is to develop an algorithm with usage of a neural network capable of counting visitors to a public place. In addition to this, there are several general goals in the thesis work, like the collection of information and data analysis.

Functionality Goals

- Achieve the definition of a person in the image.
- Calculate the direction of movement of a person in a video.
- Using the collected data on the direction and identification of the person in the video, count the visitors who entered and exited.

Programming Goals

- Create code with Python and OpenCV.
- Make code easy to read and optimized.

Testing Goals

- Test code using different materials.
- Achieve satisfactory test results.
- Analyze test results.

Research Goals

- Collect information about neural networks, object tracking and other necessary for analysis software .
- Conduct an analysis of all information, including data obtained before the start of the thesis (for example, background), data during the work on the thesis (for example, an algorithm) and data at the end of the thesis (for example, a test).

Additionally

- Write a report and all additional necessary papers.
- Present the thesis.

1.3 Methods

The work on the thesis included many methods of obtaining and analyzing information. There are three main methods of obtaining material and working with it during the project. All approaches are important and have delivered a significant contribution to the result.

- Literature review

This method expressed in work with multiple resources. Information obtained from various articles and books is reviewed , from which the entire base of the thesis follows. This is what created the background for further work. The key action, in this case, represents the selection of significant points in the present literature and, by means of some transformations and unification, to derive the necessary knowledge.

- Experimental Analysis

The experimental way helped in writing the algorithm. So, some coefficients and functions in the program were selected by trying different options. In this way, among many different variables, it was possible to identify those that produce the most accurate results. The essence of the method lies in the active intervention and testing of different values.

- Modulation

This method was used especially in the part of testing. Modeling of various situations for analysis was created to collect data on the results of work, as well as for corrections. The modulation of various human movements and situations on the camera served as the basis for tests and analyzes of the results.

2 PROJECT RELEVANCE

2.1 General use

The initial purpose of implementing the project is to control the number of visitors in a public space. The use of the project is flexible and can be established in any place and for any task where it is necessary to count the people circulating in the space covered by the camera. The main data obtained by operating the program is the number of humans who entered and exited in and out of a certain room located conditionally on one side of the image being processed. Therefore, the most effective way is to use the project to control the count of people currently in the territory of a certain public space, as well as the ability to count the total number of visitors per day. Moreover, it does not matter whether the public area is large or small, has it one or more rooms and floors. The essential condition is only the presence of one entrance to the public space. However, the possibility of upgrading the project for spaces with multiple entrances is also available. Any area or event that does not wish to use another method of occupancy monitoring can act as a public territory.

2.2 Comparison with other methods of control

To confirm the importance of the project, it can be considered as part of a comparison with other methods of occupancy monitoring that do not use cameras and neural networks for these purposes.

2.2.1 Human eye control

The first such competitors are the use of cameras and human power to replace the computer architecture. The principle is almost the same, but the analysis is carried out not by a computer, but by a human. While the video comes from the cameras in both cases, or a person can even personally be present at the entrance and count the number of people only using their eyes. This method, however, is the most inefficient and expensive, since it involves hiring a permanent staff member or introducing new responsibilities to those already hired. The effectiveness of this method is not higher.

2.2.2 Physical control with turnstiles

The following option for controlling the number of visitors is not to use the analysis of the received image, but physical control. This includes, for example, turnstiles. Their accuracy will be higher than the method proposed by this thesis, due to the smaller number of possible operation errors during the calculation. When going through a turnstile, the person will be counted almost without faults, on the other hand the use of neural networks, implies the same percentage of false positives and not counting several guests. In this case, the choice of method is based on how accurate the calculation should be, and often it is possible to ignore few errors by creating a slight artificial underestimation of the maximum number of visitors when using the proposed project. But the comfort of the visitors themselves will be a plus since the presence of horizontal bars can create queues and be inconvenient. (Mairs turnstile 2022.)

2.2.3 Physical control with infrared ray

It is also possible to refer to physical control sensors with an infrared ray and similar. The fundamental idea of such sensors is like turnstiles, and a person is counted when he cuts off the ray. However, such detectors have a serious minus, in contrast to the use of cameras - the difficulty of determining the direction of movement of a human, so by interrupting the stream it is possible to confirm only the fact that a human crosses a conditional passage, but not the track of his action. Although it is possible to fiddle with setting up two sensors to emit two beams and determine the direction of the person based on which ray was crossed first. But this method does not expect an unusual human movement. And such sensors do not distinguish who specifically crossed the line of the beam, for example, a cart can be perceived as a person. (Amila 2019.)

2.2.4 Administration control

In addition, one of the effective ways of occupancy monitoring can rightfully be considered the creation of appointment and the closure of the possibility of free visits. This is not the same method that is meant by ordinary human control. Here, counting is not carried out on the spot and the number of visitors is limited in advance by scheduling. However, if such a way can be actively operated, for example, by medical institutions, salon, and so on, for other public spaces, this method of control is very difficult, unusual, and poorly introduced. So, it is badly suited for a café, as an example.

2.3 Motivation

2.3.1 Medical use

From a medical point of view, the project is relevant for tracking the spread of respiratory viral diseases like Covid-19. The air in a closed space becomes a risky environment when pathogens of respiratory viral diseases enter, by controlling and limiting the number of visitors at a time, the spread of such illnesses can be slowed down. The viruses transmitted by airborne droplets poses the greatest danger precisely on the premises. Infected people can exhale disease-causing particles when they breathe out. These pieces are extracted and begin to circulate in the air, so particles enter other people and transmit infection to them. The closed space makes this process much riskier since the air circulates in one room and is not updated. Therefore, it is possible to identify the following factors that increase the risk of infection associated with in-doors:

- Poor ventilation of the room and insufficient replacement of air from the street.
- The actions of people that increase the amount of release of virus particles. For example, coughing, sneezing, and even playing sports and speaking loudly.
- Large congestion of people in the room at the same time.

By highlighting these key problems, it is possible deal with them. Paying attention to the last point, the solutions proposed in the thesis help to solve the issue of crowds of people in the premises. Controlling the number of visitors can decrease the risks of infection, and in combination with other methods of combating the spread, even reduce them to a mini-mum. At this time, the first problem would be solved by enhanced ventilation and air purification system, and the second problem by wearing masks, which will reduce the increase of particles from a person. (Bazanta & Bushb 2020.)

As a more convincing demonstration of the effect of the number of people in a room on the spread of respiratory viral diseases, it is possible to use the calculations made by Bazanta and Bushb (2020, 7-8).

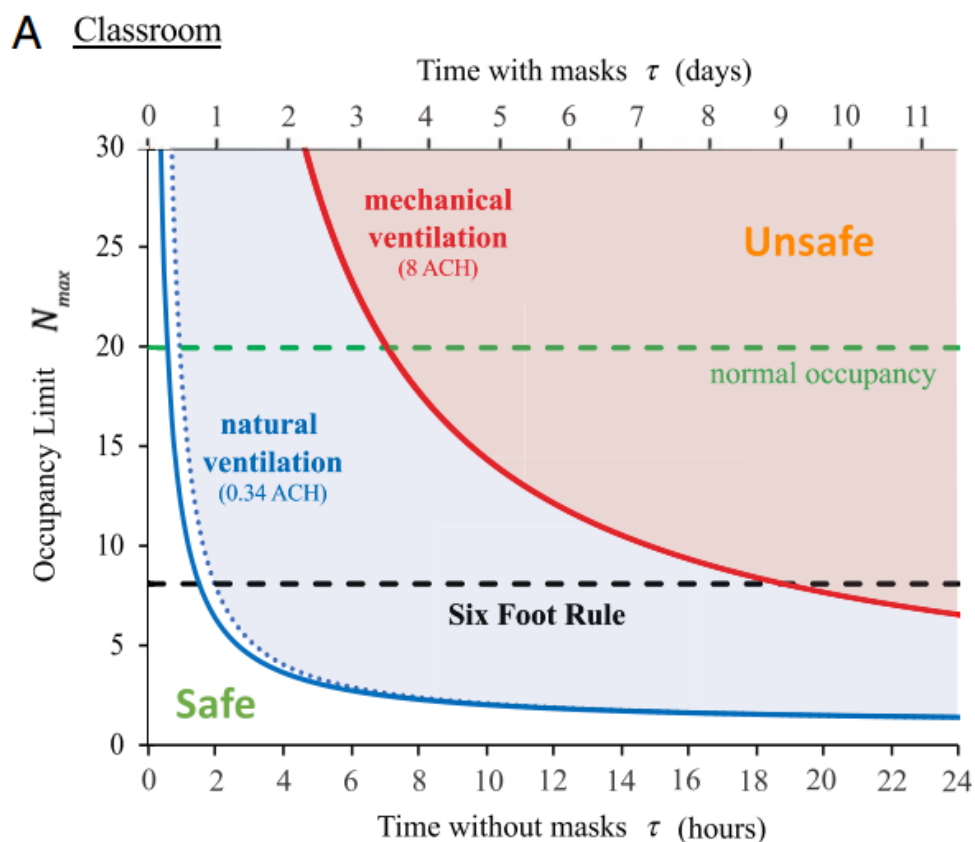


FIGURE 1. Demonstration of factors influencing the spread of respiratory viral diseases. (Bazanta, and Bushb 2020.)

It is possible to consider the diagram the authors provided, obtained during the research, which shows graphs of the danger of infection depending on various indicators (see FIGURE 1.). The scheme was realized following the formula derived during the study and demonstrates an example with a school class, however, the patterns obtained from this case are partially valid for many other. It is worth noting the study indicates that scheme is valid if people in the room are unengaged in high respiratory activity and occupy a low-risk group due to young age.

Looking at the graph, the following data can be distinguished: In the lower left corner, there is the maximum protected zone where the risk of infection is minimal, and in the upper right corner, the contrary, the most dangerous area with the highest possibility of the virus spreading. Furthermore, the graph shows the indicators affecting the chance of the getting virus. Hence, two different airing curves can be seen - the blue one represents natural ventilation and the red one is mechanical. The total hours without wearing a mask, at the bottom, on the x-axis. And the number of people in the room on the y-axis. It is also worth noting the safe number of humans depends directly on the size of the room, the six-foot line is typically obtained from calculations in which took the average area of the class and calculated how many individuals can be in this sector with the condition of sufficient

free space. These are significant calculations. They should be used to determine the optimal allowable sum of people in the territory in each case.

Research made by Bazanta and Bushb (2020) suggests that attendance control, among other factors, can indeed reduce the risk of exposure to airborne viral diseases. The graph clearly shows that the increase in the number of people in the room (considering the area of the receipts itself) directly affects getting into a less protected zone.

2.3.2 Business and marketing use

The medical method represents the essential justification of the idea of this thesis, and so the software was adapted primarily for that use. However, the uses of the algorithm do not end there. The project is flexible and suitable for those who want to consider statistics on visiting a store or other establishments. The purpose of collecting such data may be to analyze this information and build a further marketing system on them. So, assuming that many store visitors do not actually purchase the goods presented in it, the data collected with the help of this thesis can compile statistics of amount buyers for the business. These statistics can serve for further analysis and business development.

2.3.3 Use for social research purposes

Additionally, in rare cases, algorithm and this thesis can help in other scientific studies and research. Therefore, the collection of such data may be needed for social and marketing research and much more. The use of the project is different due to its flexibility.

2.4 Specific example

To clearly discover the use of the indoor people counting system, it is reasonable to describe a specific example of the intended usage of the project. Consequently, possible to consider the scheme of a tourist park in which there is both external and internal activity, a schematic example can be seen below (see FIGURE 2). The territory is a park with different tourist routes and a closed pavilion with an exhibition. There is access control on the area with tickets, but it is carried out only at the entrance to the whole park. It is senseless to limit the number of visitors in an open zone, as there is low risk of spreading diseases. However, this is not true in the closed pavilion on the territory, which leads to the fact that the entrance to that must be controlled separately. (Ruskeala 2018.)

The project tries to replace the usage of another type of visitor control in the pavilion and maintain an automated system for counting guests inside. This method is the most convenient and low-cost therein particular case.

As follows, a system with a camera can be installed at the entrance to the pavilion along with a screen where information about the level of risk of infection is shown. It is equally possible to integrate such a notification into existing display, for example, to an information monitor. In that way, park visitors will be informed about how safe is for them to explore the pavilion now and, based on data, they can decide whether should enter indoor or if they want to continue the outdoor activity for the moment and possibly return later.

It is also recommended to equip the pavilion with extra methods of cutting spreading viral diseases transmitted by airborne droplets, like establishing good ventilation in the room and other measures to prevent viruses extending. In the end, the total number of visitors in a particular pavilion is calculated based on data on the area of the room, its ventilation, and more. (Canadian center of occupational health and safety 2020.)

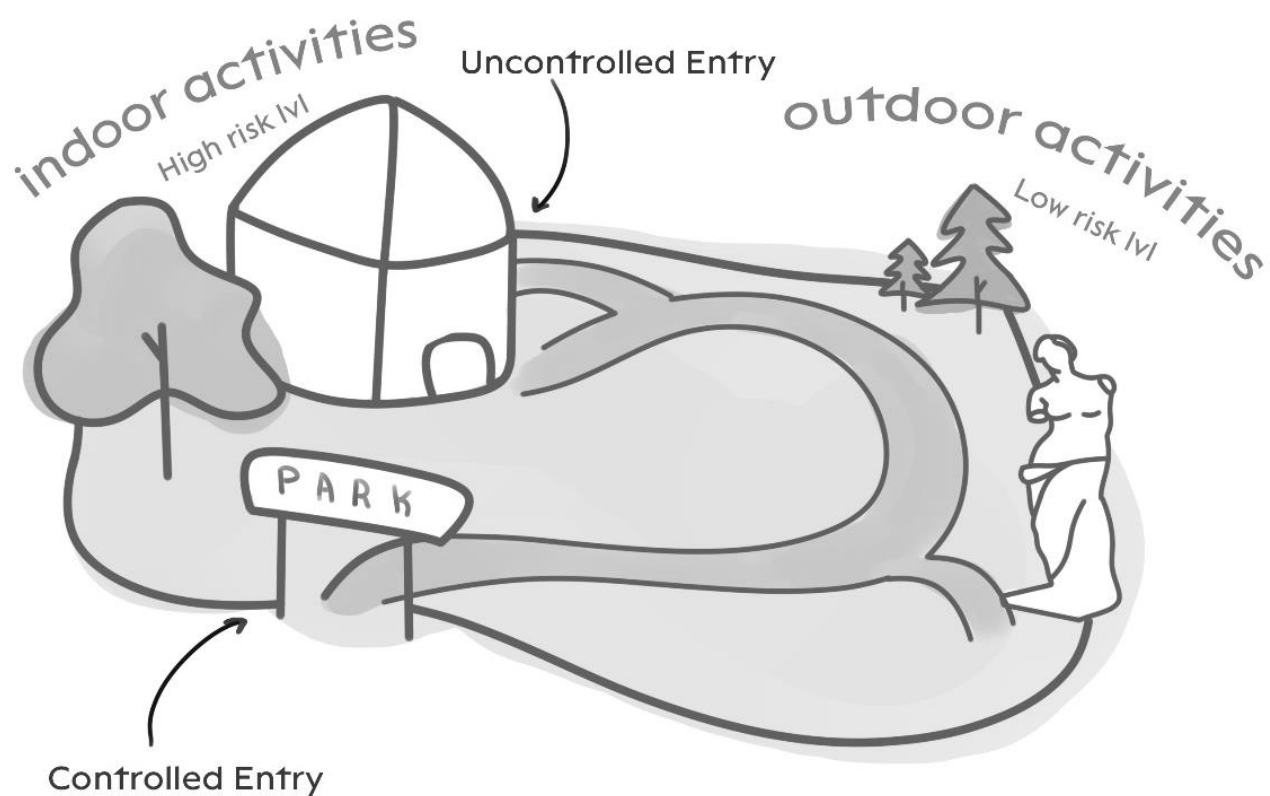


FIGURE 2. Schematic representation of a case where an automatic people counting system can be used.

3 BACKGROUND INFORMATION

3.1 Neural Network

3.1.1 Neural network capabilities

The creation of the occupancy monitoring algorithm uses neural network technologies to achieve basic functions. The use of artificial intelligence in the thesis and work on the analysis of its capabilities was carried out because these technologies have numerous advantages. So, among the main areas of application of neural networks are forecasting, decision-making, pattern recognition, optimization, and data analysis. (Nduati 2020.)

The idea of neural networks is to simulate the work of the human brain - namely, its ability to learn and correct errors. Therefore, possible to say that this are main feature of any artificial intelligence. This ability is great for realizing the main goals of the thesis.

3.1.2 Using a neural network for current purpose

The fundamental area of neural networks involved in the thesis obtains systems for recognizing and classifying objects in images. It is for this function that artificial intelligence is involved in the project, with the help of which a person is recognized in the image for his further analyses in the visitor counting system.

One of the key advantages of neural networks, in contrast to traditional algorithms, is the possibility of self-learning, which allows the creation of a more accurate and diverse program. In the learning process, the neural network can identify complex relationships between inputs and outputs, as well as perform generalization. Without this, the definition and classification of objects in the image seem almost impossible, which forces the use of artificial intelligence in such project.

3.1.3 How neural network works

In this thesis, a pre-trained neural network from Yolo (More about Yolo itself is also described in the project below.) is used, so work on training the artificial intelligence was not carried out, however, assuming the principles of the machine learning are necessary to process on the project. (Redmon, Divvala, Girshick and Farhadi 2016.)

Artificial intelligence is similar to a brain, the biological counterpart. The similarity is noticeable in the schematic representation below (see FIGURE 3). In a network, a neuron performs computational actions. It receives information, processes data, and passes on. This action imitates what happens in human brain: person is asked a question, they think about it and give an answer. Similarly, a neural network receives an input, processes data, and issues an output.

Moreover, the computer does not only imitate the sequence of the process, but also the structure of the human brain. Accordingly, biological brain consists of many neurons united by connections, which allows humans to carry out the mental activity (see FIGURE 4). The neural network imitates this principle with its own neurons and connections between them.

Human brain:



Neural network:

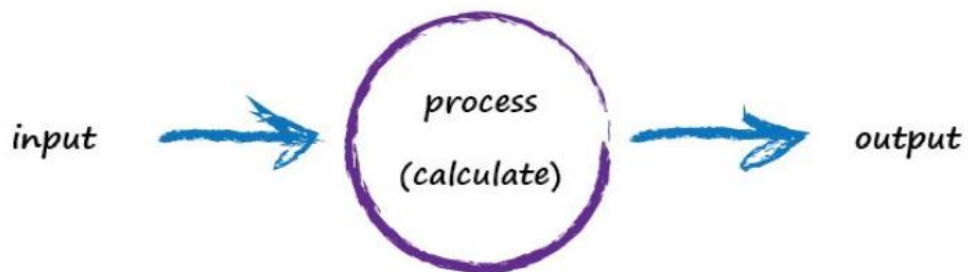


FIGURE 3. Similarities between neural network processes and human intelligence. (Tariq 2016.)

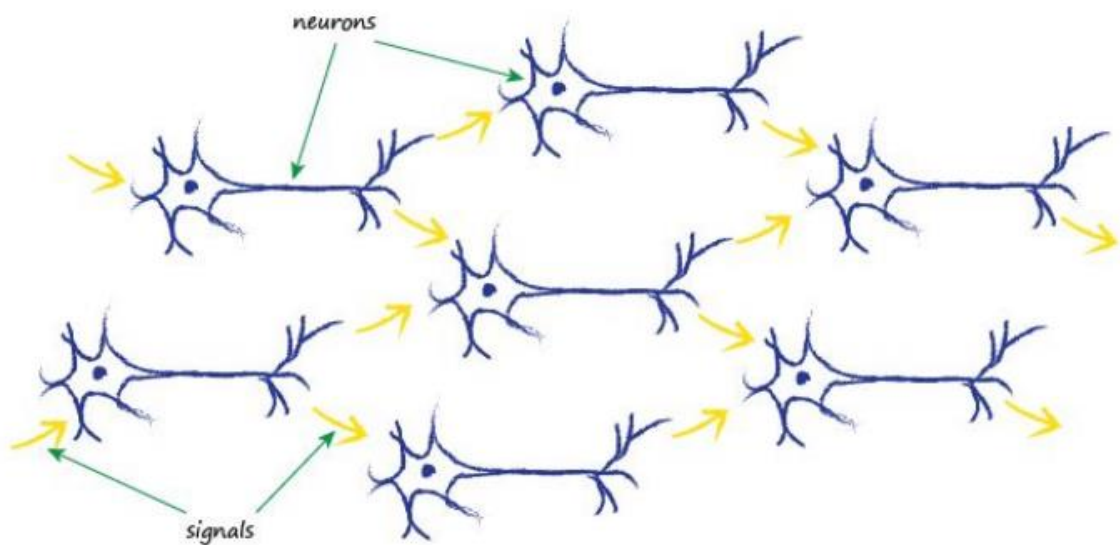


FIGURE 18. Diagram of the architecture of neurons in the human brain. (Tariq 2016.)

All neurons can be divided into various types: input, hidden, and output. Often there are many neurons, and they form layers (input, hidden, and output, respectively) as shown in the diagram below (see). Each of the neurons has two parameters: input and output data. Neurons communicate, and these connectors also have their own weight, which allows information to change as it passes along the chain. Thanks to all this, the input information takes the form of the result, and the desired effect is achieved in this case, the definition of a person in the video. (Tariq 2016, 12-54.)

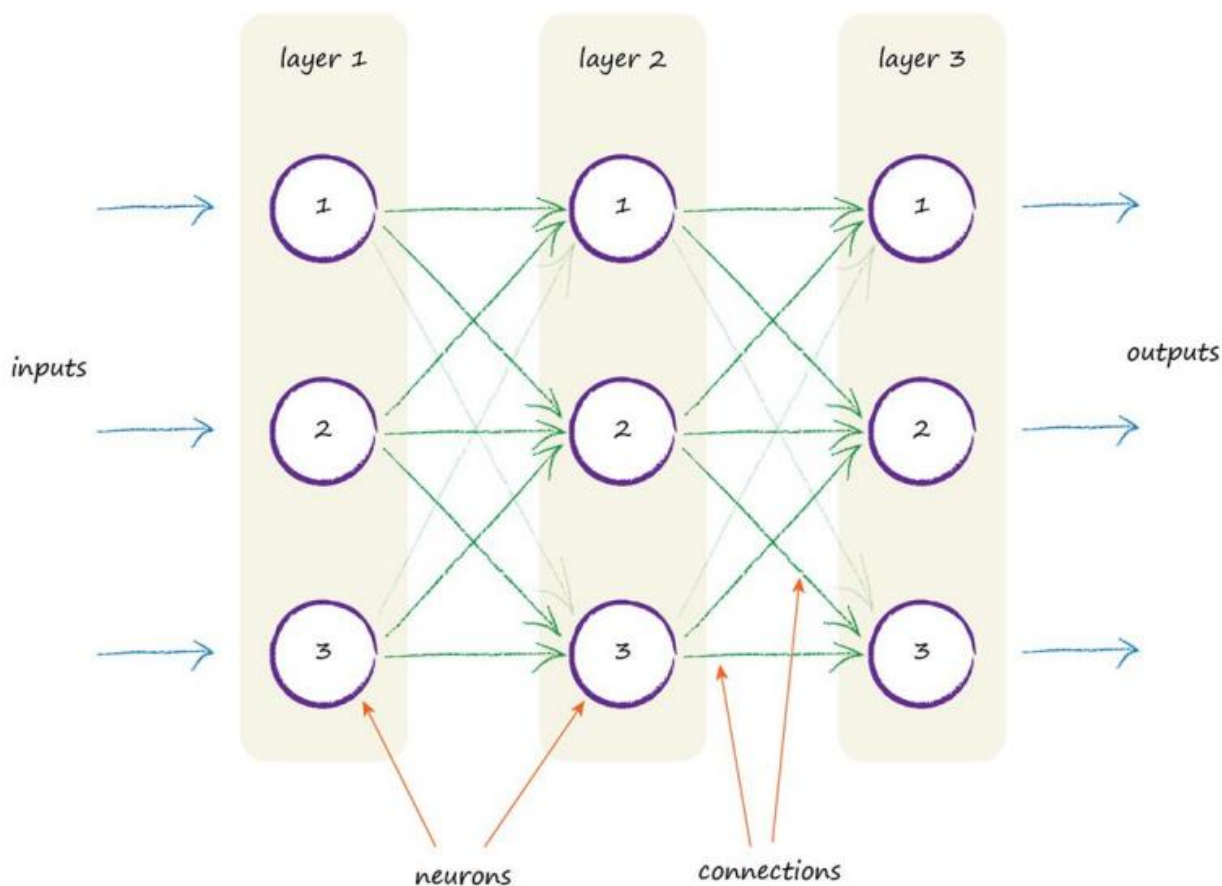


FIGURE 5. Neural network architecture diagram showing layers and connections. (Tariq 2016.)

3.1.4 Python and neural network

This thesis uses Python for programming. It is also suitable for writing a neural network and satisfies all the tasks set in the project. Artificial intelligence will not present a problem to implement if know how each neuron is modeled and the connections between them. To work on a thesis to create a counter of visitors, as well as similar ones, it is convenient to use Python, since there are many libraries for neural networks, guides and books written by specialists, such as Parker (2021). This concise and flexible language is an excellent choice for working on this thesis.

3.2 Object detection

Object detection remain essentially a method that allows the computer to determine where and what is in the image. Such object identification and location can be used for many projects, like detecting people in an image and counting them.

3.2.1 How object detection works

To begin with, it should be noted that there are several architectures of neural networks created to define objects. These are two-level and one-level architectures. In this project, Yolo is applied, which belongs to the latter.

Describing in short, a two-level system uses the so-called regions in the picture to decide if a certain object is in this area. A picture is fed to the input of the neural network, it is run through CNN (Convolutional Neural Networks) to form feature maps of the expected objects. Then Separate artificial intelligence determines the areas in which the target is most likely to be located - so the picture is divided into regions, and there are initially a very large number of such regions. These areas are compressed using RoIPool (regions of interest pooling), based on the principle of areas with similar features. And so the data is fed into the neural network, which selects the object class in the previously selected regions. These steps can be seen schematically in the picture below (see). This kind of CNN models have two drawbacks: they are quite slow and look only at certain parts and not at the whole picture. An example of such a neural network is RCNN(Region Convolutional Neural Networks). RCNN analysis was presented by Girshick, Donahue, Malik (2013, 1-2).

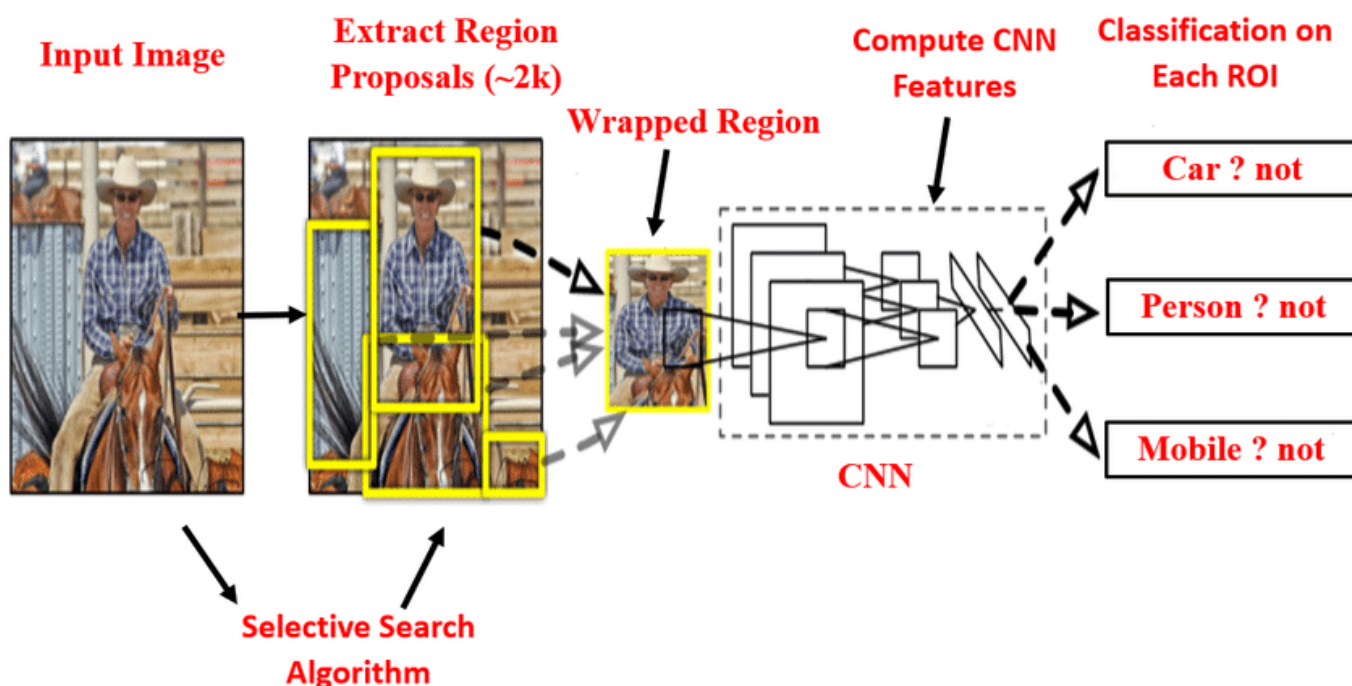


FIGURE 6. Two-level neural network. (Murthy, Hashmi, Bokde, and Geem, 2020.)

The principle of operation of one-level systems copies the first phases of the two-level architecture. One- and two-level architectures are compared in the picture below (see). Therefore, in them the first block sends a picture to the input and then feature maps are created using CNN (in the case of Yolo, its own CNN called Darknet-53 is used). Further, in contrast to two-level architecture, the

feature maps themselves are analyzed, giving out the positions and sizes of the bounding boxes as well as their classes. (Redmon, Divvala, Girshick and Farhadi 2016.)

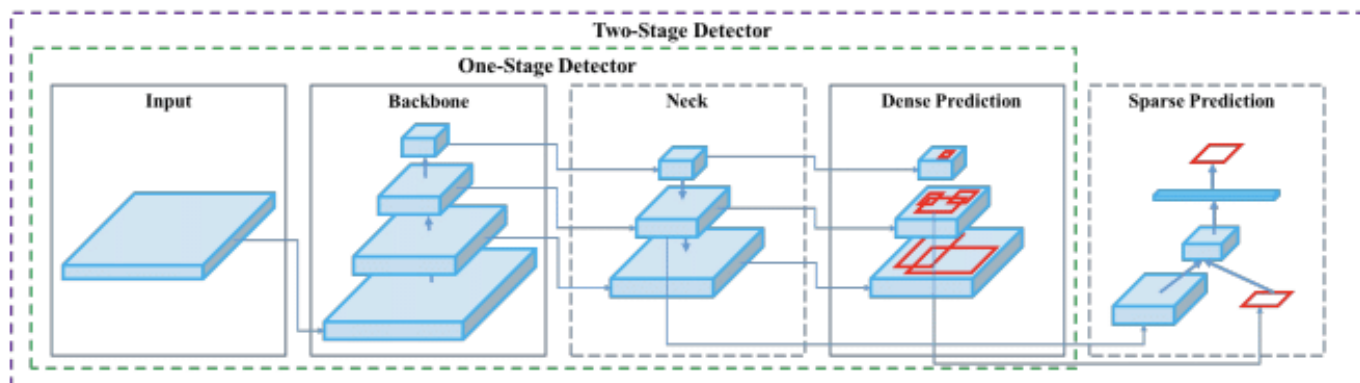


FIGURE 7. Two concepts of architectural object detection. (Iyer, Bhensdadiya and Ringe 2021.)

Neck is the third block in the algorithm; this block aggregates information from separate layers from previous steps. As follows, the accuracy is increased.

The contiguous block is called Dense Prediction. This is a characteristic feature of Yolo that allows it to be of such high quality. This step works as follows.

- The picture is divided into $S \times S$ segments. Usually, it is 13×13 but there are other scales too.
- The parts obtained from the division of the picture are called grid cells, and bounding boxes are attached to each such unit. Moreover, several bounding boxes are drawn at once, and usually these are three pieces, they have different shapes, since it is not yet clear what shape the object is and how much space it occupies exactly. Their positions and sizes are calculated relative to the center of the cell. It is also worth noting the proportions of the drawn rectangles can be set by the programmer or employ the existing ones in the data.
- At the same time, each cell is responsible for predicting class probabilities. This does not mean that some parts contain some object, it is just a probability. That is, if an object is found in that cell, it will be designated by the class assigned to this cell.
- Then the data obtained from both previous processes are combined.

These steps are well demonstrated in the picture below (see).

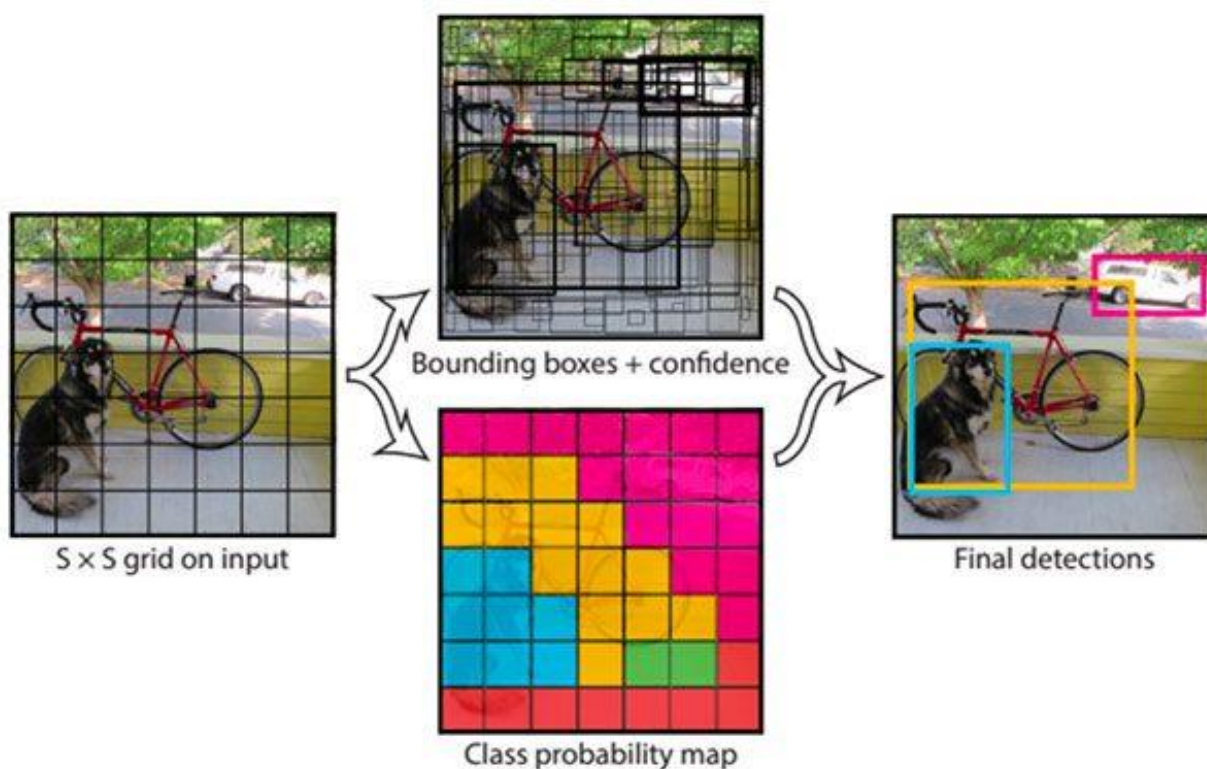


FIGURE 8. How Yolo works. (Kajabad and Ivanov 2019.)

Therefore, on each cell, the following two parameters should be determined: which of the drawn bounding boxes of the anchor (cell it is connected to) is the most suitable and which object by class is inside the anchor boxes. In fact, further, the task of the neural network is to predict five parameters for each anchor for a certain cell as accurately as possible and then quickly determine the object. In the end, only the predicted bounding boxes with a high confidence score remain. Next, use the NMS (non-max suppression) technique to filter out the bounding box, leaving only one. Demonstration of NMS can be seen below (see). (Redmon, Divvala, Girshick and Farhadi 2016.)

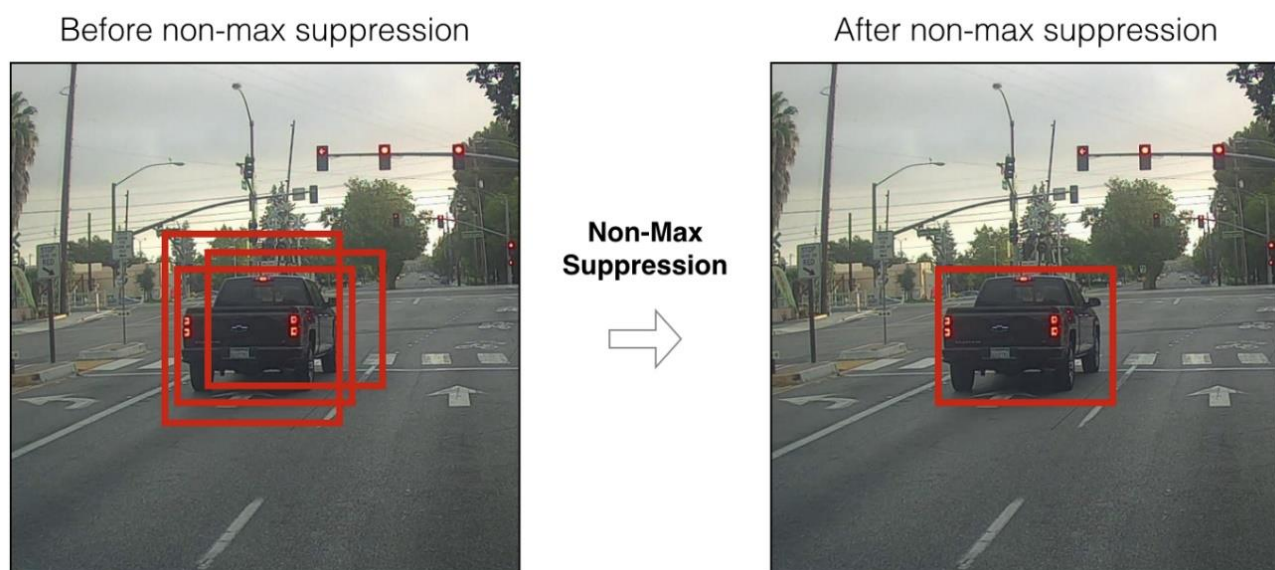


FIGURE 9. NMS (non-max suppression) technique. (Joshi, Nama, Shah, and Vijayakumar 2020.)

3.2.2 Object detection in current project

In this project, the result obtained on the output layer of Yolo is actively used to detect objects. The data obtained at the output of the neural network allows to determine the location of the object and its class on the video, in the case of the thesis it is a person, this data becomes the basis for determining its direction of movement. An example of detecting a human object using Yolo in this thesis project can be seen below (see FIGURE 5). Here the bounding box of a person and his probability are clearly visible.

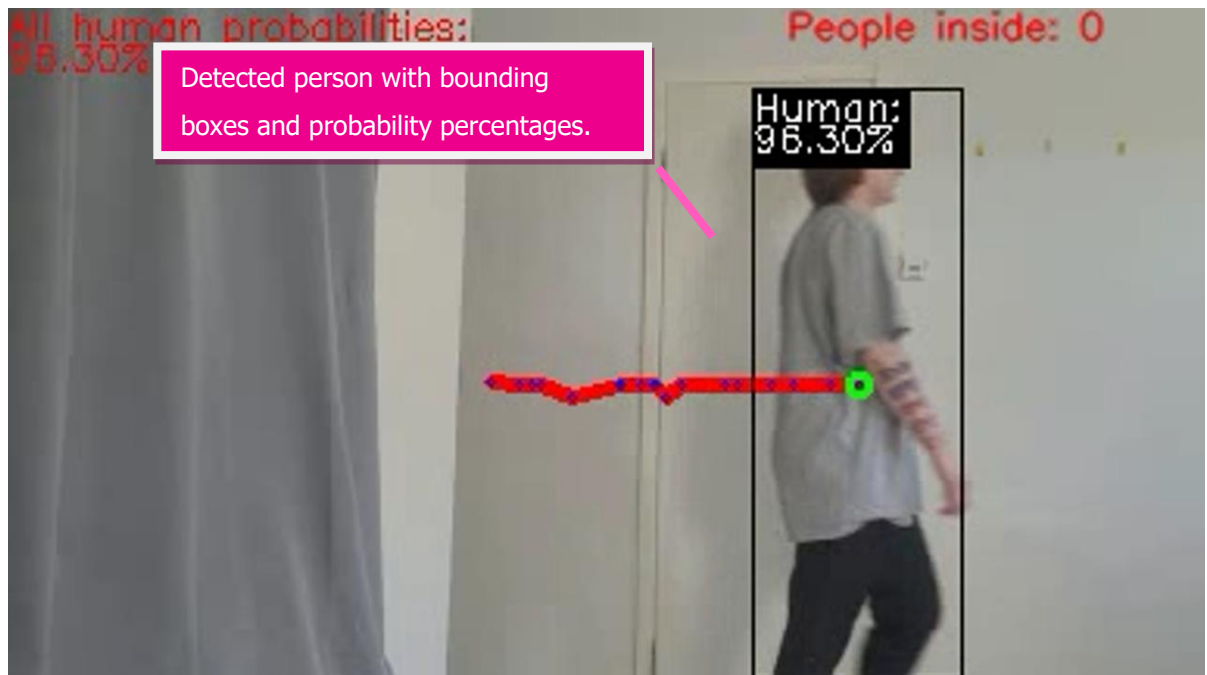


FIGURE 10. Human detection on the image.

4 IMPLEMENTATION

4.1 OpenCV

The project in this thesis is carried out in the form of a program written in Python. It is significant that the project uses OpenCV. This is an open-source library whose main area is computer vision, and it is frequently used for analysis, classification, and image processing. OpenCV additionally provides the ability to read and write images, capture, and save videos. The library was initially developed in C++. But in addition, Python and Java bindings have been released. As follows, it is a valuable tool to help achieve the goals of the thesis. (OpenCV 2022.)

4.2 YOLOv3-tiny

Yolo (you only look once) is a convenient algorithm for detecting objects in an image. This is one of the best options for these purposes and has already been trained. An example of Yolo's work can be seen in figure (see). Thereon, Yolo neural network detects horses in the picture. This algorithm features high speed and accuracy of object detection. In this project, the tiny version of the network is operated since it is barely demanding on the power of the computer, although less accurate at the same time. So, a tiny version of Yolo consists of fewer layers and especially, only two output layers, while the network weights are large and take up little memory. In the future, it is possible to replace the tiny version with a full one. (Yolo 2022.)

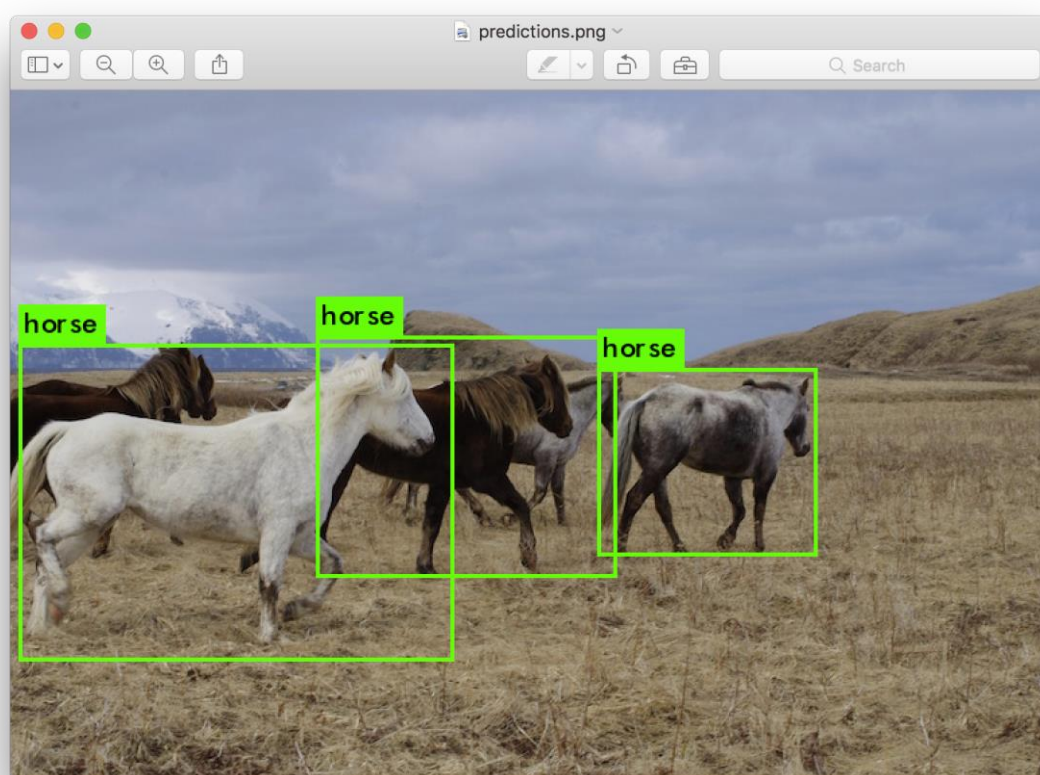


FIGURE 11. Yolo's demonstration. (Yolo 2022.)

4.3 Occupancy monitoring algorithm

The occupancy monitoring algorithm created in this thesis can be described in the following diagram (see FIGURE 6). This picture can also be found in the appendix in a larger format (see APPENDIX 1).

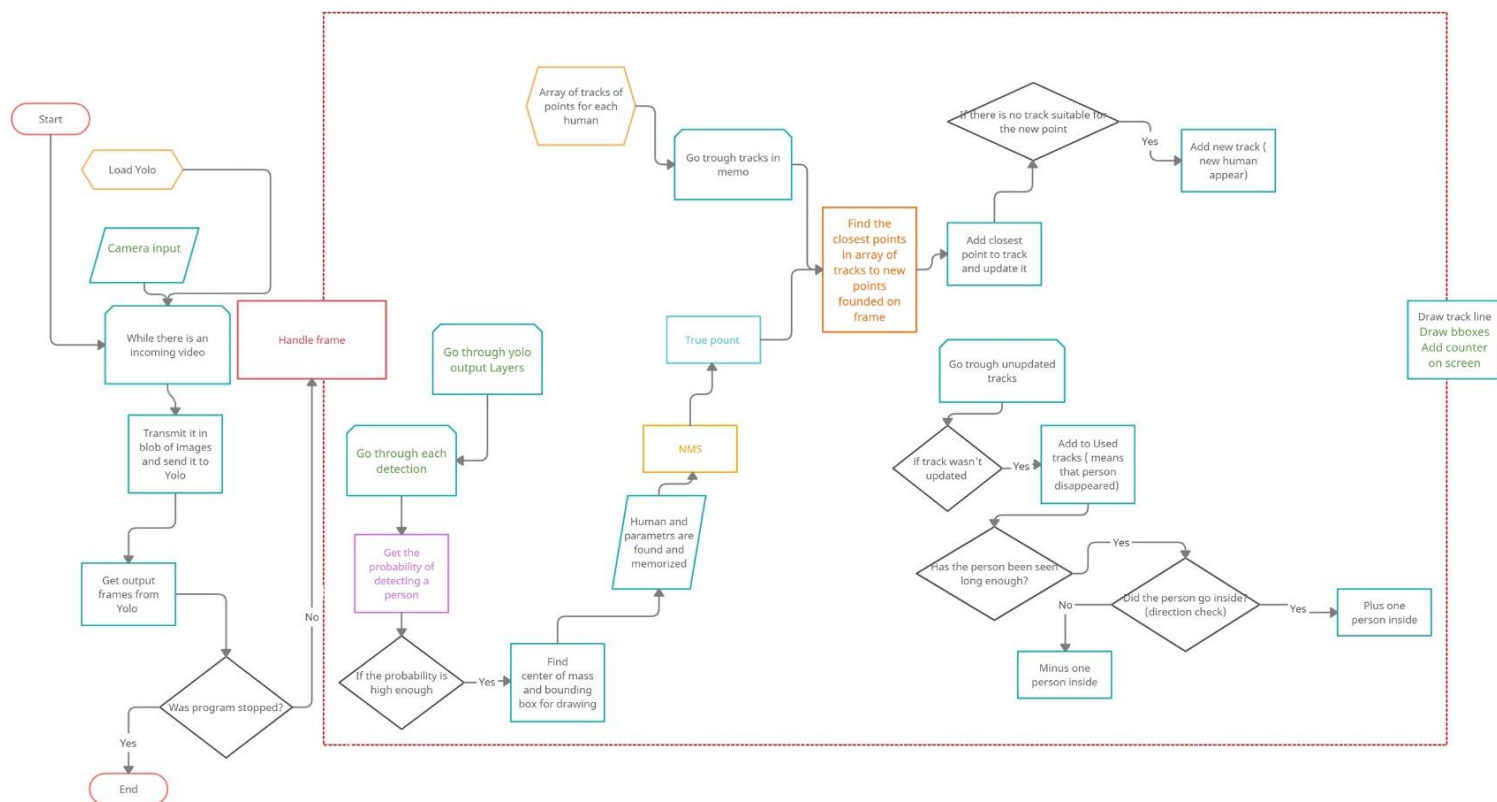


FIGURE 12. Schematic algorithm.

For a clearer understanding of the principle of the project, let's consider the presented scheme in detail and in parts. Let's look at the steps in programming with additions of screenshots to better demonstrate the part of the code in question. The entire code can also be found in the appendix (see APPENDIX 2). The program algorithm is as follows:

- At the start of the program, Yolo is loaded, the necessary variables are set, and the video Input is defined (see FIGURE 7).

```

main.py X
C: > Users > rinak > Desktop > Thesis > WorkingProgram(Updated2) > main.py > ...
1  from cv2 import VideoCapture
2  import numpy as np
3  import cv2
4
5  #Camera input
6  cap = VideoCapture(0)
7
8  #Load a pretrained tiny yolo network
9  yolo = cv2.dnn.readNetFromDarknet("ycfg.cfg", "yolov3-tiny.weights")
10 yolo.setPreferableBackend(cv2.dnn.DNN_BACKEND_OPENCV)
11 yolo.setPreferableTarget(cv2.dnn.DNN_TARGET_CPU)

```

FIGURE 13. Getting a video on the input and loading Yolo.

- Next, the program goes into a loop that is valid if the file is open, and the user has not exited the program. The loop checks if the video stream is still running, if not it displays a notification about it (see FIGURE 8). If there is a video, it is converted into an image blob (in other words an array or set of variables) and sent to Yolo.

```

while cap.isOpened():
    #Read a new frame
    ret, frame = cap.read()
    #Check if file hasn't ended
    if not ret:
        print("Can't receive frame (stream end?). Exiting ...")
        break

    scale_percent = 10 # percent of original size

    #Get scaled width and height values
    width = int(frame.shape[1] * scale_percent / 100)
    height = int(frame.shape[0] * scale_percent / 100)
    #Make a tuple of dimensions
    dim = (width, height)
    #Resize image using new dimensions
    frame = cv2.resize(frame, dim, interpolation = cv2.INTER_AREA)

    #Get a blob of an image
    blob = cv2.dnn.blobFromImage(frame, 1/255, (width, height), [0,0,0], 1, crop=False)
    #Feed the blob to the neural network
    yolo.setInput(blob)

```

FIGURE 14. Loop processing input video.

- Then the index of the output Yolo layer and the data from these layers are determined (see FIGURE 9).

```

#Indexes of output layers
layerNames = yolo.getLayerNames()
ounames = []

#Get the names of the yolo's output layers
for i in yolo.getUnconnectedOutLayers():
    ounames.append(layerNames[i-1])

#Run forward the neural network and get the outputs
out = yolo.forward(ounames)

#75 bboxes, 85 parameters
#print(out[0].shape)
#300 bboxes, 85 parameters
#print(out[1].shape)

#Process the frame
handleFrame(out, frame)

```

FIGURE 15. Get output from Yolo.

- Frames received from Yolo are passed to the Handle Frame function.

- The function goes through the Yolo output layers and then through each detection on each frame (see FIGURE 10). Then the neural network gives the percentage of the probability of finding a person on the frame, if this percentage is large enough, then the person is found.
- For each found person, the centers of mass and the bounding boxes are calculated on each

```
#Go through 2 yolo outputs
for layerOut in out:
    #Go through each detection(75 and 300)
    for detection in layerOut:
        #Probability that detection is a person
        personProbability = detection[5]
        #If sure enough
        if personProbability > personThereshold:
            #Set that we have found a human
            humanDetected = True
```

FIGURE 16. Go through each detection done by Yolo.

frame. These parameters will be remembered in memory.

- An NMS (non-max suppression) is carried out and the final true central point is determined from the data obtained from bounding box (see FIGURE 11).

```
#Get rid of subsqunt boxes
truePersonIndexes = cv2.dnn.NMSBoxes(bboxes, personConfider

#Store center points after nms
trueCenterPoints = []
#Inflate true center points array
for i in truePersonIndexes:
    trueCenterPoints.append(centerPoints[i])
```

FIGURE 17. Finding center point.

- The program runs through an array of human motion tracks. It is worth noting that it is the movement tracks for each selected person that are saved, in such tracks the last saved point is the most important.

All the last points of each track are compared with the new received point from the processes described above the frame to find among them the closest point to the just received one (see FIGURE 12). At the same time, there is a maximum possible distance range of a point from a track, beyond which it is not assigned to a certain track. Thus, if the point is too far from all existing tracks, it will not be assigned to any of them.


```

#Go through each of the tracks in memory
for trackInd in range(len(memory)):
    #Find the closest new point
    minDifference = 99999
    minDiffPointIndex = -1
    for truePointInd in range(len(trueCenterPoints)):
        #Remember that everything is a point, so we need to also specify that we are interested in x([0])
        #Every value is a point
        #x is in [0]

        if trackEndings[trackInd] != None and trueCenterPoints[truePointInd] != None:
            diff = abs(trackEndings[trackInd][0] - trueCenterPoints[truePointInd][0])
            #Change if distance is smaller
            if diff < minDifference and diff < maxDiffAllowed:
                minDifference = diff

```

FIGURE 18. Find closest track to new point.

- The new point is added to the nearest found track if one exists and the track is updated.
- The next step of the program goes through all the tracks.
- Here the program checks whether the track has been updated or not. If the track has not been updated, then it has not received a new point, that is, the person to whom the track was attached has left the camera coverage area (see FIGURE 13). This track goes to used tracks that will be deleted.

```

#Delete tracks that were not updated
tracksToBeDeleted = []
for trackInd in notUpdatedTrackIndexes:
    if trackInd == None:
        continue
    tracksToBeDeleted.append(trackInd)
    #Check if the person got in
    #If person was seen for long enough frames
    if len(memory[trackInd]) > 10:

```

FIGURE 19. Check tracks to be deleted.

- In addition to this, the program also checks whether the person on the obsolete track was visible for a long time, and if so, then this person must be counted as either entered or exited. The direction of the person is checked by the program. This is determined by comparing the start and end points (see FIGURE 14). And adds the corresponding value to the counter of visitors.
- The program also displays this counter and additionally draws a track of a person and boxes.

```

#Check if the person got in
#If person was seen for long enough frames
if len(memory[trackInd]) > 10:
    if memory[trackInd][0][0] < memory[trackInd][len(memory[trackInd])-1][0]:
        peopleIn += 1
    else:
        peopleOut += 1

```

FIGURE 20. Check person's direction.

Thus, the principle of the program is that with the help of Yolo, human detection is possible. Furthermore, its central point is marked, and it can now be considered as point on the coordinate field. Many points of a person, thanks to the detection on different frames, are combined into a track and updated along the way of this person. And when a person leaves the frame, his direction is determined by the track and the memory of this will be deleted. This principle and the main important concepts are shown in the figure below (see FIGURE 15).



FIGURE 21. Main concepts of algorithm.

5 TESTING AND DISCUSSIONS

5.1 Tests

To check and confirm the operability of the system, tests were carried out on various videos. For the demonstration, screenshots from a video clip made in a well-lit space and with a neutral background were used to better distinguish all important aspects of the algorithm. Tests were conducted using images of people of varying body build, height, clothing, and movement habits. Most of the attempts ended with a satisfactory result, except for a few failures. The accuracy of the program varies on different test materials, but on the video used in this thesis, it is 92.5% accurate. So, in the analysis of the results, a video is used where 14 people enter and 13 people exit, while the program produced 2 mistakes.

It is also possible to analyze some cases in more detail for a more proper understanding of the correct and false detections of the system. The following represents an analysis of some examples of various movements encountered in life and how the program identifies them.

- One human

One person is determined precisely, all movements are tracked without errors. The calculator works without problems. A screenshot with one person can be seen below in the picture (see).

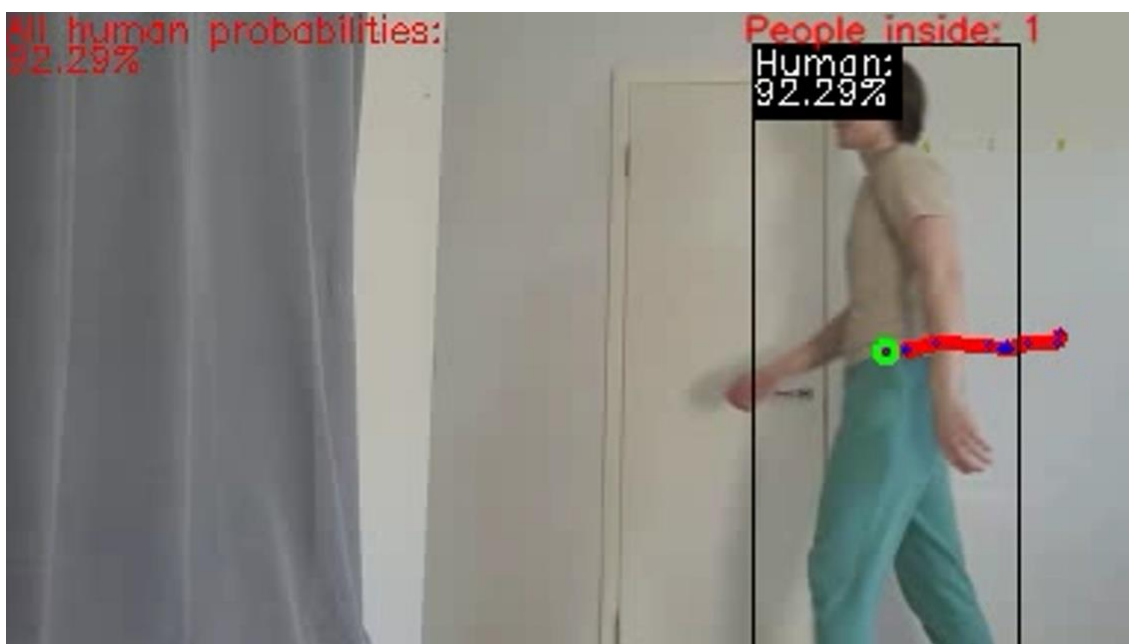


FIGURE 22. One human detection.

- Several humans

Multiple humans can cause an error if one or more people overlap another person for the entire path captured by the camera. Such an example can be seen in the picture below (see FIGURE 16.). This is due to the lack of opportunity for Yolo to identify the human in the frame. At the same time, it is worth noting that if a person is unblocked for the entire segment of the path, his presence will be recorded, such an example can be seen in the picture (see FIGURE 17) . It is noticeable the track of one person is shorter than the other,

since at first body was overlapped. Otherwise, the definition of several individuals does not cause problems, and the counter works



FIGURE 23. Several humans' detection error.

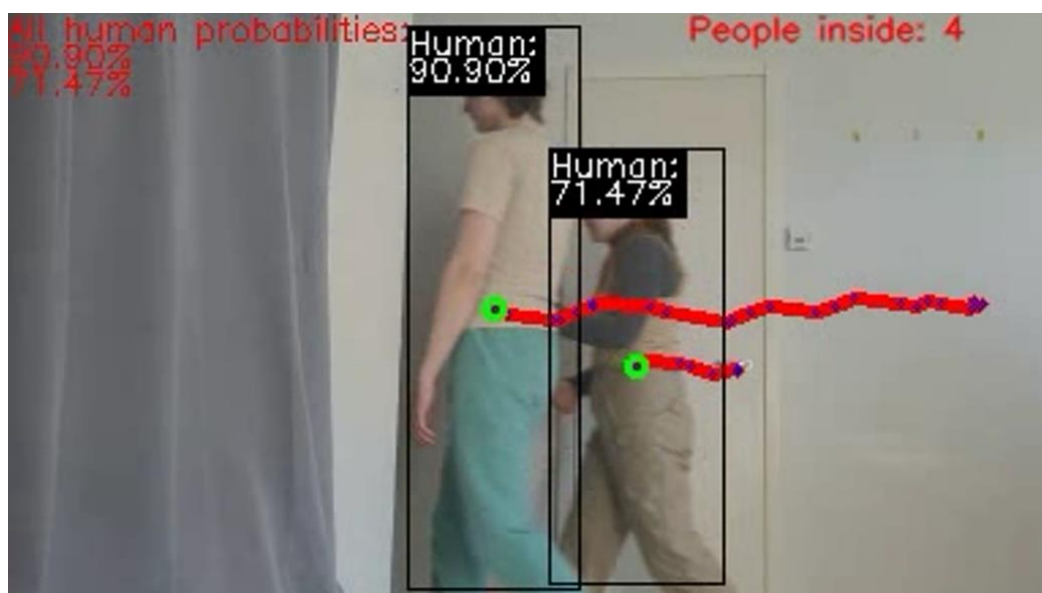


FIGURE 24. Several humans' detection success.

- Cross-moving

Cross traffic, when one or more people walk towards each other across movement lines, can at first glance cause the same overlap problem. In fact, the situation here is not so problematic. The point is that even if a person is blocked by oncoming humans crossing his movement, he will not be out of sight all the time of his action. This is possible only in places with a large circulation of people. Otherwise, the individual will leave the closed area, and his movement will be recorded. This can be noted in the picture below (see). On the left, in the camera's field of view, a person will be counted.

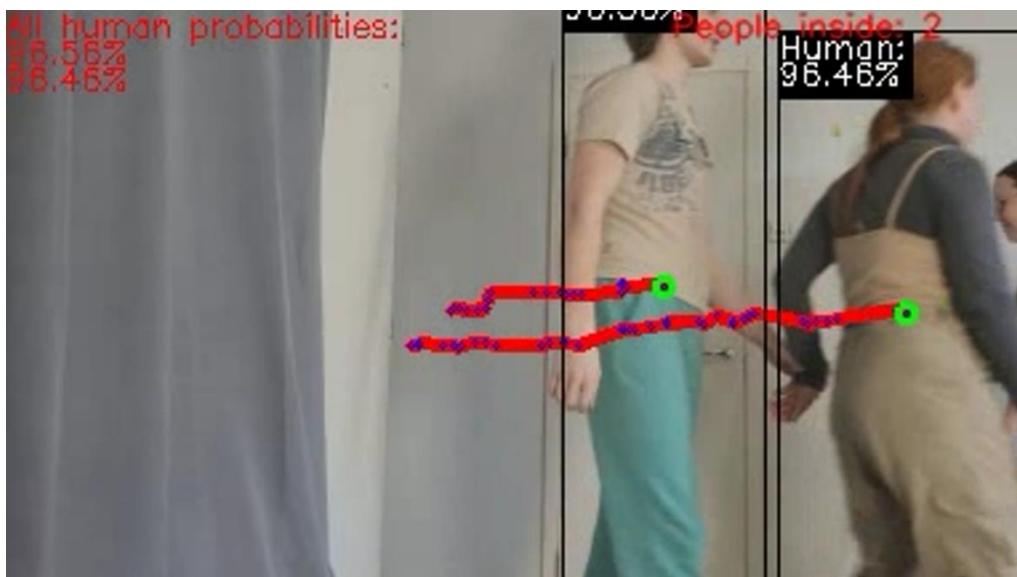


FIGURE 25. Cross-moving detection.

- Hats and clothes

Tests show that clothes, including hats, glasses, and so on, do not affect the result of the detection. However, it can be noted that the probability that a human is identified becomes lower but is still sufficient. It is possible to recognize that it in the picture below (see). A person passes in a headdress that completely covers his face.



FIGURE 26. Human with hat detection.

- Unusual moving

Unusual types of movement are equally not a problem. As long as there is a movement of a person along the x-axis on the received frame and a human is detected on it, the system will be able to compose his track, which means program will work correctly. Like so, it can be assumed that most unpredictable situations will not cause errors.

5.2 Upgrade opportunities

Having considered all the results obtained during the tests, it is possible to say that the system works satisfactorily and performs the key functions. However, it obtains many opportunities to improve and increase the accuracy of operation.

The most significant thing to hold in mind is that the program presented as a demonstration in the thesis uses a simplified version of a neural network for detecting objects. Developer Yolo provides the ability to apply the tiny version. As follows, it is possible to operate the neural network in projects on less powerful computers, but at the same time sacrificing the accuracy of the definition. This happens because the artificial intelligence is weaker, but its weights take up less computer memory. However, from all this it follows that the occupancy monitoring algorithm can be upgraded at least when using a more effective neural network, giving a more accurate and faster result for determining objects.

5.3 Integration into existing systems

As stated in the objectives of this project, the product must be flexible for use. The algorithm obtained during the work only needs to have a video image received from the cameras, while the cameras must transmit the movement of people along the x-axis. At the same time, camera position adds more variables to the system: for example, at what angle it is located and at what distance. The accuracy of the result is tuned in the code by changing the coefficients responsible for the minimum allowable track size, that is, how many humans were followed on the video, as well as the maximum suitable distance from the newly found point to the track closest to it, that is, after how much the neural network detected the person from his previous detection. However, the system does not have perfect flexibility. The location of the cameras will still affect the result, even if the degree of influence is acceptable. The best video input for the code would be where the people's tracks are clearly visible on the x-axis, the camera is pointing directly at them perpendicularly, the lighting is adequate, and the background is relatively neutral.

5.4 Risk analysis

Based on all the data received, after completing the work on the code and testing, risk analysis can be derived to better understand the possible weaknesses of the project and options for correcting them. It is immediately possible to single out the fundamental risk of the program's work: not counting a person. This problem can occur for the following reason: a human is not identified on enough video frames to compile his track. That is, the neural network could not determine the person for such a period that was set in the program with a variable coefficient, usually it simply means that the person was not visible throughout his entire track. The solution to this issue may be to create a gap when calculating the number of visitors per area of a public place. Gap means that the allowed number of people in the program is deliberately lower than possible, so as to leave space for errors. That is, if a person is not counted by the system, the real number of visitors will still be within acceptable limits. This gap will serve as a safety net in case some individual was uncounted. See also additional risk analysis below.

Risk: Person not counted due to overlap.

Risk level: High in places with a large flow of visitors and medium with a normal flow.

Solution: Creating a gap for supposed missed positives.

Risk: The difference in proportions on the image due to different angles and distances of installing the camera cause incorrect operation of the system.

Risk level: High

Solution: An algorithm containing several easily replaceable coefficients that must be changed in each individual case for the system to work. These coefficients include: the minimum track length for scoring and the maximum allowable distance from the new detected point to the track expected to be detected.

Risk: Insufficient performance of the intended hardware to establish the system.

Risk level: Low

Solution: In fact, the tested systems are already quite loyal to the performance of the hardware. On the contrary, with the presence of stronger capabilities, the system can be upgraded and increase its accuracy. Thus, it is possible to strike a balance between the available computer performance and the quality of the data received.

6 CONCLUSION

6.1 Key points

This thesis includes the key aspects of creating a project, searching, and analyzing the necessary information for it, motivation for its creation, as well as testing and discussing the results. It fully reveals the work done and describes it as specifically as possible. To sum up from what has been stated in the thesis, the main key points of the work can be singled out and described as follows.

- The thesis is based on the creation of a visitor counting project with object detection.
- The work on the thesis covered various aspects: the collection and analysis of information by various methods, the implementation of the project, and further analysis of the result.
- Algorithm developed in this thesis is versatile, and it can be used by any public place that wants to track the number of visitors by camera and a neural network.
- The most important motivation for occupancy monitoring is medical. Visitor control can help reduce the spread of respiratory viral diseases such as Covid-19.
- Neural network and object detection is the best way to implement visitor counting in places where there is no other way to control.
- Yolo is a very efficient and well-suited trained neural network for this application.
- The principle of operation of the algorithm described in the thesis is based on identifying a person on each frame and tracking his movement track, which makes it possible to count the entered and exited visitors.
- Multiple tests were performed on the algorithm demonstrating its performance. These tests are described in the thesis.
- An analysis was carried out on the results of the algorithm, the results of which were also described.

6.2 Evaluation of results

The goals of the thesis were achieved. The result of the work done is a working algorithm and analysis performed before and after its creation.

The algorithm for counting people can be difficult to use due to the following factors: in many places there are more convenient alternative methods of use, the algorithm has a problem with triggering when people overlap. In addition, another problem is the ethics of using cameras in public places and tracking people.

On the other hand, the thesis and the algorithm show many positive results: the research carried out and the results can be used in other studies and projects. Moreover, the thesis demonstrates a suitable way to occupancy monitoring in many cases. The algorithm is quite accurate and efficient, in addition, it is upgradable and very flexible.

REFERENCES

- Amila, Buddhika 2019. Investigation of Infrared Sensors for Robot Navigation. IOSR Journal of Engineering 09(9):11-17. https://www.researchgate.net/publication/335961720_Investigation_of_Infrared_Sensors_for_Robot_Navigation. Accessed 23.05.2022.
- Bazanta, Martin and Bushb, John 2020. A guideline to limit indoor airborne transmission of COVID-19. In Zhang, Renyi (Ed.). Cambridge, Massachusetts Institute of Technology, Department of Chemical Engineering. <https://www.pnas.org/doi/10.1073/pnas.2018995118>. Accessed 10.05.2022.
- Canadian center of occupational health and safety 2020. Indoor Ventilation: Guidance During the COVID-19 Pandemic. <https://www.ccohs.ca/>. Accessed 27.05.2022.
- Finnish institute for health and welfare 2022. Coronavirus COVID-19 – Latest Updates. Transmission and protection. Updated 20 April 2022. <https://thl.fi/en/web/infectious-diseases-and-vaccinations/what-s-new/coronavirus-covid-19-latest-updates/transmission-and-protection-coronavirus>. Accessed 27.05.2022.
- Girshick, Ross, Donahue, Jeff, Darrell, Trevor and Malik, Jitendra 2013. Rich feature hierarchies for accurate object detection and semantic segmentation. <https://arxiv.org/abs/1311.2524>. Accessed 23.05.2022.
- Iyer, Rakkshab, Bhensdadiya, Kevin Prabhulal and Ringe, Priyansh Shashikant 2021. Comparison of YOLOv3, YOLOv5s and MobileNet-SSD V2 for Real-Time Mask Detection. International Journal of Research in Engineering and Technology. https://www.researchgate.net/publication/353211011_Comparison_of_YOLOv3_YOLOv5s_and_MobileNet-SSD_V2_for_Real-Time_Mask_Detection. Accessed 23.05.2022.
- Joshi, Kartik, Nama, Tushar, Shah, Rahul Hemal and Vijayakumar P. 2020. Two-Level Authentication System for Access Gate Entry in a Localized Campus Network. International Journal of Engineering and Advanced Technology (IJEAT): 9 pp. <https://zenodo.org/record/5593906#.Yo6Zx6hByUk>. Accessed 23.05.2022.
- Kajabad, Ebrahim Najafi and Ivanov, Sergey V. 2021. People Detection and Finding Attractive Areas by the use of Movement Detection Analysis and Deep Learning Approach. Procedia Computer Science. https://www.researchgate.net/publication/336082028_People_Detection_and_Finding_Attractive_Areas_by_the_use_of_Movement_Detection_Analysis_and_Deep_Learning_Approach. Accessed 23.05.2022.
- Mairs turnstile 2022. What is Turnstile Mechanism. Updated 28 April 2022. <https://mairsturnstile.com/turnstile-mechanism.html>. Accessed 27.05.2022.
- Magalhães, Rafael 2019. Object Recognition Using Convolutional Neural Networks. https://www.researchgate.net/publication/337146307_Object_Recognition_Using_Convolutional_Neural_Networks. Accessed 23.05.2022.
- Mishra, Shanti Bhushan and Alok, Shashi 2017. HANDBOOK OF RESEARCH METHODOLOGY. Education. https://www.researchgate.net/publication/319207471_HANDBOOK_OF_RESEARCH_METHODODOLOGY. Accessed 10.05.2022.
- Mohan, Sachin 2020. 6 Different Types of Object Detection Algorithms in Nutshell. MLK (Machine Learning Knowledge). Updated June 30, 2020. <https://machinelearningknowledge.ai/different-types-of-object-detection-algorithms/>. Accessed 23.05.2022.
- Murthy, Balaram, Hashmi, Mohammad Farukh, Bokde, Neeraj Dhanraj and Geem, Zong Woo 2020. Investigations of Object Detection in Images/Videos Using Various Deep Learning Techniques and

Embedded Platforms. Applied Sciences. https://www.researchgate.net/publication/341099304_Investigations_of_Object_Detection_in_ImagesVideos_Using_Various_Deep_Learning_Techniques_and_Embedded_Platforms-A_Comprehensive_Review. Accessed 23.05.2022.

Nduati, Judy 2020. Introduction to Neural Networks. Section's Engineering Education. <https://www.section.io/engineering-education/introduction-to-neural-networks/#neural-networks-overview>. Accessed 27.05.2022.

OpenCv 2022. <https://opencv.org/>. Accessed 27.05.2022.

Parker, James R. 2021. Python : An Introduction to Programming. Mercury Learning & Information

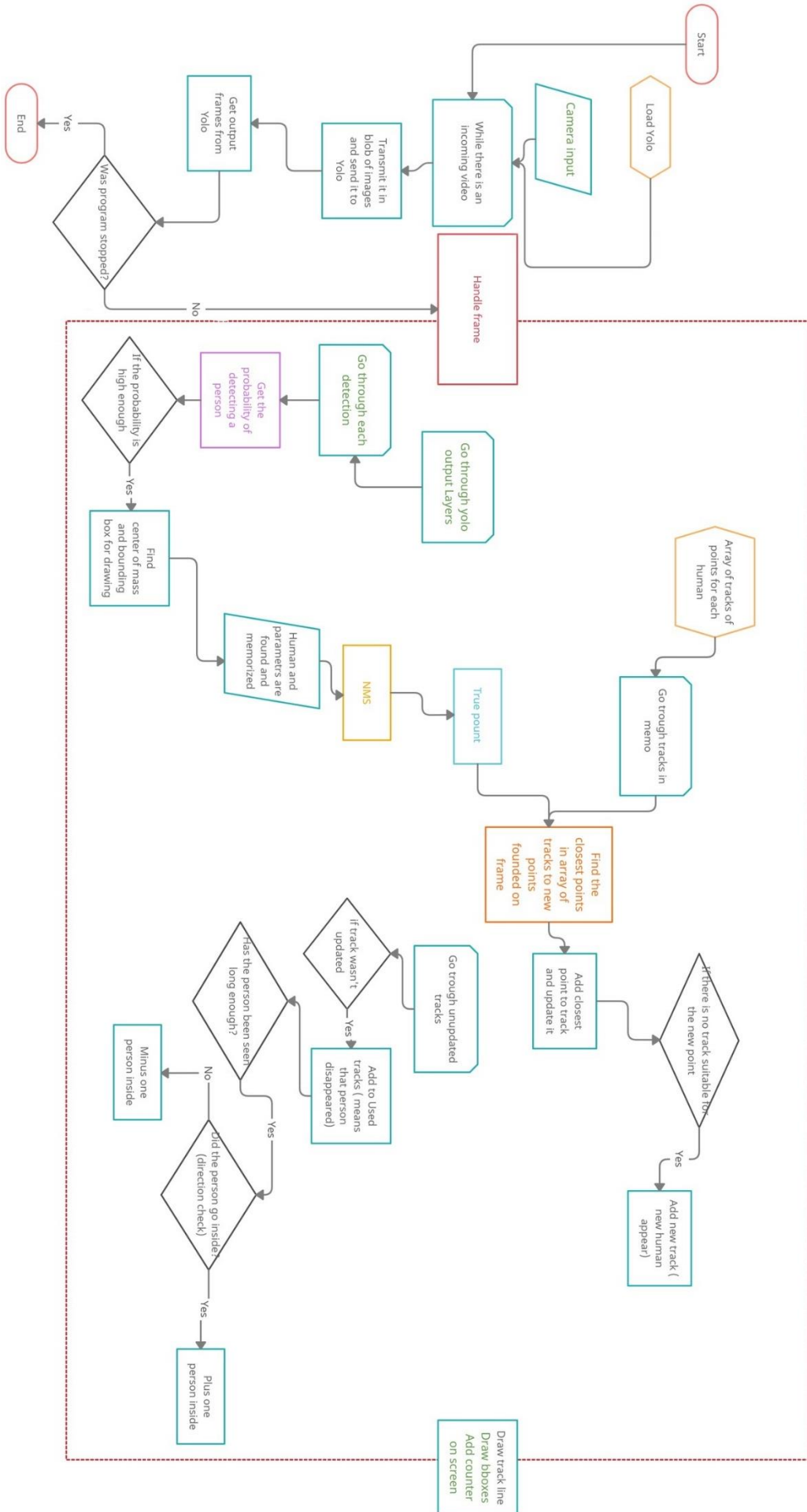
Rashid, Tarig 2016. Make Your Own Neural Network 1st Edition. CreateSpace Independent Publishing Platform

Redmon, Joseph, Divvala, Santosh, Girshick, Ross and Farhadi, Ali 2016. You Only Look Once: Unified, Real-Time Object Detection. <https://arxiv.org/abs/1506.02640>. Accessed 23.05.2022.

Ruskeala 2018. <https://ruskeala.ru/>. Accessed 27.05.2022.

Yolo 2022. YOLO: Real-Time Object Detection. <https://pjreddie.com/darknet/yolo/>. Accessed 23.05.2022.

APPENDIX 1: SCHEMATIC OF THE ALGORITHM



APPENDIX 2: PROGRAM

```
from cv2 import VideoCapture
import numpy as np
import cv2

#Camera input
cap = VideoCapture(0)

#Load a pretrained tiny yolo network
yolo = cv2.dnn.readNetFromDarknet("ycfg.cfg", "yolov3-tiny.weights")
yolo.setPreferableBackend(cv2.dnn.DNN_BACKEND_OPENCV)
yolo.setPreferableTarget(cv2.dnn.DNN_TARGET_CPU)

#Max difference in pixels for new center point position to be connected to the frame.
maxDiffAllowed = 30 #px

#Array of x values of detected human. 1 value/frame
#Array of TRACKS
memory = []

#Counters of people
peopleIn = 0
peopleOut = 0

#Thresholds
personThreshold = 0.5
NMSThreshold = 0.2

def handleFrame(out, image):
    #Memorize original shape
    height, width, channels = image.shape
    #Preset unseen human
    humanDetected = False

    #Load global variables
    global peopleIn
    global peopleOut
    global memory

    #Array of found bounding boxes
    bboxes = []
    #Array of found center points
    centerPoints = []
    #Array of detection confidences
    personConfidences = []

    #Go through 2 yolo outputs
    for layerOut in out:
        #Go through each detection(75 and 300)
        for detection in layerOut:
            #Probability that detection is a person
```

```

    personProbability = detection[5]
    #If sure enough
    if personProbability > personThereshold:
        #Set that we have found a human
        humanDetected = True
        #Find new center of box
        #Scale them(values are in 0-1 format, need to multiply by original w and
h)
        cx = int(detection[0] * width)
        cy = int(detection[1] * height)
        #Get scaled width and height of box
        w = int(detection[2] * width)
        h = int(detection[3] * height)
        #Get top left corner
        x = int(cx - w/2)
        y = int(cy - h/2)

        #Memorize centers
        centerPoints.append((cx, cy))
        #Memorize bounding box for drawing
        bboxes.append([x,y,w,h])
        #Memorize the confidence of this detection
        personConfidences.append(personProbability)
    #Get rid of subsquent boxes
    truePersonIndexes = cv2.dnn.NMSBoxes(bboxes, personConfidences, personThereshold,
NMSThreshold)

    #Store center points after nms
    trueCenterPoints = []
    #Inflate true center points array
    for i in truePersonIndexes:
        trueCenterPoints.append(centerPoints[i])

    trackEndings = []
    #Create an array of last tracks points
    for track in memory:
        #Append last point
        trackEndings.append(track[len(track)-1])

    #Create an array of memory indexes which are not changed
    notUpdatedTrackIndexes = [*range(len(memory))]

    #Go through each of the tracks in memory
    for trackInd in range(len(memory)):
        #Find the closest new point
        minDifference = 99999
        minDiffPointIndex = -1
        for truePointInd in range(len(trueCenterPoints)):
            #Remember that everything is a point, so we need to also specify that we are
interested in x([0])
            #Every value is a point
            #x is in [0]

```

```

        #I want to change to null used values, so no repetitions

        if trackEndings[trackInd] != None and trueCenterPoints[truePointInd] !=
None:
            diff = abs(trackEndings[trackInd][0] - trueCenterPoints[true-
PointInd][0])
            #Change if distance is smaller
            if diff < minDifference and diff < maxDiffAllowed:
                minDifference = diff
                minDiffPointIndex = truePointInd

        #Add new point and delete used values
        if minDiffPointIndex != -1:
            memory[trackInd].append(trueCenterPoints[minDiffPointIndex])
            trueCenterPoints[minDiffPointIndex] = None
            trackEndings[trackInd] = None
            notUpdatedTrackIndexes[trackInd] = None

        #All the points are added

        #Delete tracks that were not updated
        tracksToBeDeleted = []
        for trackInd in notUpdatedTrackIndexes:
            if trackInd == None:
                continue
            tracksToBeDeleted.append(trackInd)
            #Check if the person got in
            #If person was seen for long enough frames
            if len(memory[trackInd]) > 30:
                if memory[trackInd][0][0] < memory[trackInd][len(memory[trackInd])-1][0]:
                    peopleIn += 1
                else:
                    peopleOut += 1

        #Delete discontinued tracks
        memTempo = []
        for a in range(len(memory)):
            if a in tracksToBeDeleted:
                continue
            else:
                memTempo.append(memory[a])
        memory = memTempo

        #Add new tracks if new people present
        for newHumanPoint in trueCenterPoints:
            #Remember that it is a track, so we need to send it as list
            memory.append([newHumanPoint])

        #Draw trailing line
        for track in memory:
            #Transform memory to array
            points = np.array(track)
            if len(points) >= 2:

```

```

    #Flatten the array
    points = points.reshape((-1, 1, 2))
    #Write tracking line
    cv2.polylines(image, [points], False, (0,0,255), 3)
    #Go through each frame's center
    for point in track:
        #Write the circle for ease of tracking
        cv2.circle(image, point, radius=1, color=(255, 0, 0), thickness=1)

#Draw bboxes and human probabilities
#Height of a character = 10px
y_padding = 20
cv2.putText(image, "All human probabilities:", (0, 10), cv2.FONT_HERSHEY_SIMPLEX,
0.4, (0,0,255), 1)
for i in truePersonIndexes:
    #Get the bounding box
    box = bboxes[i]
    #Draw the center point
    cv2.circle(image, centerPoints[i], radius=3, color=(0, 255, 0), thickness=2)
    #Draw the bounding box
    cv2.rectangle(image, (box[0], box[1]),(box[0] + box[2], box[1] + box[3]), 2 )
    #Draw the tag for bounding box
    cv2.rectangle(image, (box[0], box[1]), (box[0]+50, box[1]+25), (0,0,0), -1)
    cv2.putText(image, "Human:", (box[0], box[1]+10), cv2.FONT_HERSHEY_SIMPLEX, 0.4,
(255,255,255), 1)
    #Get confidence in %
    label = str(format(personConfidences[i] * 100, '.2f')) + '%'
    cv2.putText(image, label, (box[0], box[1]+20), cv2.FONT_HERSHEY_SIMPLEX, 0.4,
(255,255,255), 1)
    #Write human to the list of humans
    cv2.putText(image, label, (0, y_padding), cv2.FONT_HERSHEY_SIMPLEX, 0.4,
(0,0,255), 1)
    y_padding += 10

#Counting string to be printed on the frame
pIn = "People inside: " + str(peopleIn - peopleOut)

#Add people counter to the frame
cv2.putText(image, pIn, (250, 10), cv2.FONT_HERSHEY_SIMPLEX, 0.4, (0,0,255), 1,
cv2.LINE_AA)

cap = cv2.VideoCapture('futureCam.mov')
fourcc = cv2.VideoWriter_fourcc(*'XVID')
vWriter = cv2.VideoWriter(filename='output.avi', fourcc=fourcc, fps=60.0, frame-
Size=(384, 216))

while cap.isOpened():
    #Read a new frame
    ret, frame = cap.read()
    #Check if file hasn't ended
    if not ret:
        print("Can't receive frame (stream end?). Exiting ...")
        break

```

```

scale_percent = 10 # percent of original size

#Get scaled width and height values
width = int(frame.shape[1] * scale_percent / 100)
height = int(frame.shape[0] * scale_percent / 100)
#Make a tuple of dimensions
dim = (width, height)
#Resize image using new dimensions
frame = cv2.resize(frame, dim, interpolation = cv2.INTER_AREA)

#Get a blob of an image
blob = cv2.dnn.blobFromImage(frame, 1/255, (width, height), [0,0,0], 1, crop=False)
#Feed the blob to the yolo network
#Feed the blob to the neural network
yolo.setInput(blob)

#Indexes of output layers
layerNames = yolo.getLayerNames()
ounames = []

#Get the names of the yolo's output layers
for i in yolo.getUnconnectedOutLayers():
    ounames.append(layerNames[i-1])

#Run forward the neural network and get the outputs
out = yolo.forward(ounames)

#75 bboxes, 85 parameters
#print(out[0].shape)
#300 bboxes, 85 parameters
#print(out[1].shape)

#Process the frame
handleFrame(out, frame)

#Write the frame to the output file
vWriter.write(frame)
cv2.imshow('CameraOut: ', frame)

#Check if q was pressed
if cv2.waitKey(1) == ord('q'):
    break

#Finish the job
cap.release()
vWriter.release()
cv2.destroyAllWindows()

```