Baltic J. Modern Computing, Vol. 10 (2022), No. 3, 526-544
https://doi.org/10.22364/bjmc.2022.10.3.24

Human's Behavior Tracking in a Store Using Multiple Security Cameras

Gintaras NARVILAS, Valdas URBONAS, Eglė BUTKEVIČIŪTĖ

Faculty of Informatics, Kaunas University of Technology,
Studentu 50, Kaunas, LT-51368, Lithuania

gintaras.narvilas@ktu.edu, valdas.urbonas@ktu.edu,
egle.butkeviciute@ktu.lt

Abstract. Recently, various Internet of Things (IoT) platforms offer cloud-based AI using their services like Machine Learning and Neural Networks that operate in their powerful computer engines and provide functionality to IoT devices. Artificial intelligence is widely used in the topics of computer vision for developing smart technologies and intelligent systems. The aim of this research is to create an automatic tool for tracking human's behavior in stores to identify the most popular places. Based on this information, more efficient product or commercial arrangements could be applied that are based on the latest tendencies. In this paper all humans as objects were localized by identifying human features using Gaussian mixture probability density model. After all images were collected and humans were labeled, the real-time human detection and tracking methods were realized. Several NN architectures were analyzed and compared. It appears that the pre-trained Custom Vision model reaches 80% accuracy in 1 hour of training at Microsoft Azure platform. The proposed technique allows not only to track people's behavior but also create a heatmap of the store providing the most visited places, where customers stop and pay more attention.

Keywords: object recognition, artificial intelligence, convolutional neural networks, Gaussian mixture model, ResNet model, Custom Vision.

1. Introduction

A growing demand for various systems to be able to automatically adapt to their surrounding environment with limited human assistance stimulates the development of intelligent systems. Neural Networks (NN) and other Machine Learning (ML) techniques are commonly used in various tasks where it is hard or even not possible to specify sets of procedural rules (Serban et al., 2020). Deep convolutional neural networks (DCNN) became a very popular method for image classification (Krizhevsky et al., 2017). It is based on low/mid/high-level features integration where those levels can be enriched by the number of stacked layers (depth) (He et al., 2016). Increasing depth by adding more convolutional layers with small filters in all layers may lead to significantly more accurate DCNN architectures (Simonyan and Zisserman, 2015).

Artificial intelligence (AI) is included in many intelligent systems from healthcare (Kodan and Chauhan, 2020) to space mission planning (Cesta et al., 2007). In port transportation channels an automatic image recognition is used to avoid accidents or traffic issues (Jiang et al., 2018). Intelligent driving systems also cover image stitching, image enhancement, and DCNN-based object detection to prevent accidents even in foggy weather (Kumari et al., 2022). Image classification and especially object recognition in images is very useful in medicine. Using magnetic resonance images, a temporal lobe epilepsy (Drane et al., 2015), early stage of Alzheimer's disease (Jacobs et al., 2015) or even cancer (Alom et al., 2019) can be recognized.

In topics that are related to computer vision, the human as an object detection is one of the most active research. In this process all objects in the image that are humans are localized by identifying human features (Bertozzi et al., 2003). It may become a challenging task because different objects have various features that are used for object recognition. Furthermore, human detection algorithms usually challenge to detect humans from images or videos when positions dynamically change, other objects are moving in the background or the camera itself is constantly moving. In this research a human tracking technique is presented that uses multiple cameras at the same time and records humans positions in the store together with the time spent in each section.

The aim of this research is to create an automatic tool for tracking human's behavior in the store and identify the most popular places. This would allow shop managers to track the most favorite sections or products in the store. Based on this information, more efficient product or commercial arrangements could be applied that are based on the latest tendencies. The proposed system allows not only to track people's behavior but also create a heatmap of the store providing the most visited places, where customers stop and pay more attention, and where are so called "black spots" or the areas through which customers do not pass.

The paper presents a new concept of how to prepare and apply different NN models to the human behavior tracking in the store. Also, the existing solutions are compared, and the results are discussed. This paper is organized as follows: in Section 2 the related works are presented. Section 3 introduces the data gathering system architecture and applied methods. In Section 4 the multiple object detection and tracking techniques are presented. Finally, Section 5 includes results of proposed methods, the comparison of different techniques, and examples.

2. Related work

This research mainly focuses on the automatic process for the human detection and tracking in a store using multiple security cameras. The final algorithm detects the most popular areas (sections) to understand the customers behavior in a store.

In production there are not many solutions that fulfill the customers behavior tracking, heatmap appliance and multiple cameras technology. A similar technology is "Neuratum X-MAP" that is compatible with any IP camera (https://neuratum.com/). It uses Artificial intelligence and Artificial Vision techniques for object tracking. Another technology related to this research is "Mirame Retail Heatmap" that allows to organize and manage the most popular areas visually (https://www.mirame.net/). Both products are created for commercial use and do not provide their algorithms specification or any other information that is related to AI technologies. For this reason, algorithms were

compared to similar object recognition, tracking methodologies that could be found in other similar researches.

In the object recognition task, the functional object properties can be extracted from visual information. There are many researchers who investigate object recognition. Some of them propose to learn objects' functional properties from video sequences (Kjellström et al., 2011). Others use techniques such as a "two images" method that eliminates the effect of the background color (Alçiçek and Balaban, 2012) or perform image edge detection together with image gray histogram to enhance the foreign object image processing (Jiang et al., 2018). Sometimes a background is removed from the image to improve the performance of object recognition. In literature a technique called DOG was found for this task. It is based on the background segmentation recognition (Fang et al., 2019).

Feature extraction and analysis is very important in humans as separate objects in image detection. In general, a human detection algorithm can be considered as a process where two or more processes can run concurrently. The algorithm may be designed to detect a human, his motion and behavior at the same time (Al-Nawashi et al., 2017). In literature different techniques could be found for human recognition in images or videos. A geometrical-based approach was proposed by (Al-Hazaimeh et al., 2019). The main idea of this algorithm is to classify the detected object based on some shape features that are extracted from the object upper portion. Another research for human motion image detection is based on Gaussian mixture model and CAMSHIFT where clear noise and object detection are applied to the human body model (Liu, 2021). Some other research uses hierarchical image composition (Kulikajevas et al., 2021), grid-based constructions (Dalal and Triggs, 2005) and many other techniques (Nguyen et al., 2016).

One of the most popular background subtraction algorithms is based on a Gaussian mixture model that uses a mixture of K Gaussian distributions for modeling each pixel using recent history $\{X_1, \ldots, X_t\}$. This is an unsupervised ML method. The probability to observe the current pixel t value can be defined as a sum of Gaussian distributions with different weights $w_{i,t}$ (see eq. (1)).

$$P(X_t) = \sum_{i=1}^{K} w_{i,t} \cdot f\left(X_t, \mu_{i,t}\right) \tag{1}$$

where K is the number of distributions and $\mu_{i,t}$ – mean value of the i^{th} Gaussian value at time t (Fradi and Dugelay, 2012). The sum of weights $w_{i,t}$ equals to one ($\sum_{i=1}^{K} w_{i,t} = 1$). The prior weights of the K distributions at time t are defined in eq. (2).

$$w_{i,t} = (1 - a)w_{i,t-1} + a\left(M_{i,t}\right) \tag{2}$$

where a corresponds to the learning rate; $M_{i,t}$ is equal to 1 if the model matches and equal to 0 for the rest models. The function $f\left(X_t, \mu_{i,t}\right)$ is also called a probability density function and is expressed in this form:

$$f\left(X_t, \mu_{i,t}\right) = \frac{1}{(2\pi)^{\frac{\pi}{2}}\left|\Sigma_{i,t}\right|^{\frac{1}{2}}} e^{-\frac{1}{2}\left(X_t - \mu_{i,t}\right)^T \Sigma_{i,t}^{-1}\left(X_t - \mu_{i,t}\right)} \tag{3}$$

where $\Sigma_{i,t}$ is the covariance matrix of the i^{th} density (Power and Schoonees, 2002). In the image processing and object recognition (such as humans) in images or videos the

training computational load is very important to compensate for the reduced prediction computational load. The Gaussian Mixture Model is suitable for feature selection in the context of large amounts of data because it increases the computational efficiency with respect to standard implementations (Lagrange et al., 2017). An improvement for Gaussian mixture model was created by (Zivkovic, 2004) where recursive equations were used to constantly update the parameters together with simultaneous selection of appropriate number of components for each pixel. This method is also called Gaussian mixture probability density model (MOG2). This algorithm is fast and can adapt to different backgrounds because it selects the needed number of components per pixel. In this paper the MOG2 algorithm was selected and applied to the data gathering process and the automatic bounding boxes detection. However, it is not accurate enough for the real time image processing. More complex models need to be considered.

In researches about human action recognition or activity recognition significant results have been reached by (Ehatisham-Ul-Haq et al., 2019). Authors proposed a technique that is based on computationally efficient feature extraction from the data obtained from RGB-D video cameras together with inertial body sensors. K-nearest neighbors and support vector machine classifiers were used for the training and testing. However, this technique requires sensorial data that is impossible to get from all customers in the store.

For the object detection in images the CNN models are commonly used due to their capability to adopt to the real-work practice like blurring, motion, shadows, etc. These models mostly rely on local features and tend to lose global structures features. This causes high inaccuracies in the classification process. (Kalake et al., 2022) proposed a combination of CNN and a histogram of oriented gradient (HOG) descriptor. It reduced the number of false positive predictions. However, this technique requires background elimination from the training and testing processes that is hard to apply in a constantly moving environment like a shop.

Another model architecture that is based on CNN model for the automatic human detection is the Residual Networks (ResNet) that incorporates residual block architecture (Alam et al., 2020). A residual block combines convolutional operations (2D convolutional layer) and a skip connection (ReLu) into an output. Rectified linear unit (ReLU) is a commonly used activation function in neural networks.

The main disadvantage of the ResNet models is that their architecture is designed to detect only one object in each image. For the multiple objects tracking the You Only Look Once (YOLO) - unified, real-time object detection approach could be used (Redmon et al., 2016). It is a fast algorithm that learns very general representation of objects. The authors state that it outperforms methods like DPM or R-CNN when generalizing information from natural images to other domains. YOLO (V2) is often used in real time object detection because the model is able to accurately classify and localize the forged and non-forged objects (Raskar and Shah, 2021). The disadvantage of this method is that it often makes false positive mistakes (detects more objects than it should). Also, it is not capable to detect two objects that are very close to each other due to the limitation in bounding boxes.

Recently, various Internet of Things (IoT) platforms offer cloud-based AI using their services like Machine Learning and Neural Networks that operate in their powerful computer engines and provide functionality to IoT devices. An example of IoT appliance is the concept model of an edge-centric service-oriented system together with model driven local video data processing (Bazhenov and Korzun, 2019). It considers a video service of human recognition around the production equipment. However, these

networks have downsides because the IoT devices always rely on cloud-connectivity to perform each task (Ali and Ishak, 2020).

In this research CNN, ResNet and Custom Vision architecture models were analyzed and compared for the real time object tracking in a store. The proposed solution includes multiple object recognition, tracking their movements and behavior by gathering information about changes in each object. It should be noted that the multiple objects detection model is trained using a pre-trained Custom Vision Machine Learning (ML) model at Microsoft Azure platform. However, due to limitations (cost and time) the real-time object tracking is performed on the inner shop server. The proposed tracking technique includes the rate of change in X and Y axis and height, and width of each object.

3. Data collection process

3.1. Data collection system architecture

At the beginning of this research the correct data needs to be collected where each image is labelled with the coordinates of human position. One of the solutions is to use publicly available image datasets (like https://neptune.ai/). Even though these databases are a great solution in various research, it is not suitable for this particular problem. Images from the prepared public databases usually have good lighting and the object or objects are most likely in the middle of the picture. Furthermore, these images are not collected using fisheye lenses that are common in the stores. That is why a separate section of this research is to properly collect and prepare the data for the further analysis.

The system architecture is presented for the data collection part. It consists of two main blocks: Google for Google Colaboratory where image processing is being made and Network Video Recorder (NVR) system. The deployment diagram (see Figure 1) shows how data collection and model creation systems are connected and what parts each of them include.

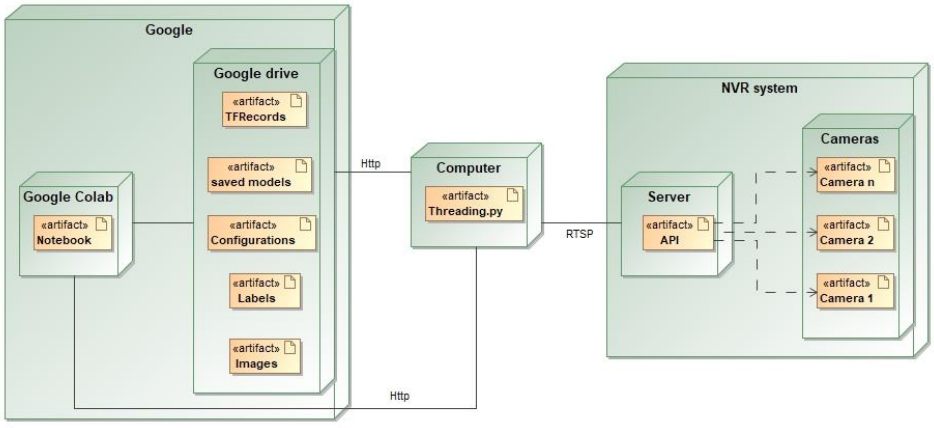

Figure 1. A deployment diagram of the data collection system.

In the deployment diagram (see Figure 1) the "Computer" block is presented that includes the device on which data collecting is being made by using the artifact

"Threading.py" (python script). This script communicates for the interaction with the "NVR system" block over Real Time Streaming Protocol (RTSP) to collect data from NVR. The collected data (images and labels) is transferred to Google Drive through Hypertext Transfer Protocol (HTTP) for further data preparation and processing in the Google Colaboratory. The TensorFlow bibliography of Python was chosen for the data recordings and artifacts of "TFRecords" generalization in the Google Colaboratory. The TFRecord format is a simple format for storing a sequence of binary records. The advantages of this protocol are a cross-platform, cross-language library for efficient serialization of structured data. The collected data and estimated features are later saved in the Google Drive.

3.2. Data collection process and methods

At the beginning of the data gathering process several different models were applied for the human recognition in images. It appears that the MOG2 algorithm is a good tool for human detection. In this research each camera may record several or even a group of people at the same time. The workflow of the data collection process and scripts for each thread are represented in Figure 2. Even though this workflow diagram corresponds to the process of a single thread, five security cameras were included, and the same steps were performed for each camera in parallel.

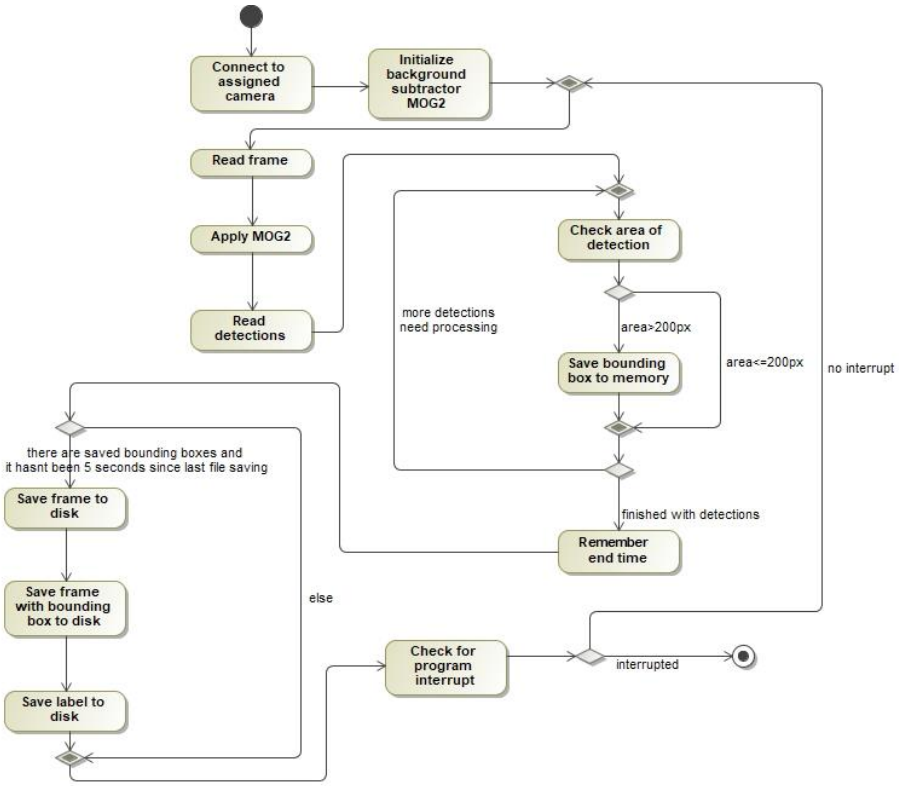

Figure 2. The workflow of the data collection process.

Every single thread (process initiated by the program in the computer) connects to its assigned camera and initializes the background subtraction algorithm MOG2. After connecting to the camera and initializing the algorithm, the loop of the main part of the process begins where the camera's frame is being marked as red and MOG2 is applied. After the MOG2 appliance, areas of people as objects are detected.

Finally, images with bounding boxes and original images together with coordinates of bounding box as labels were saved for the further filtering. If all detected areas of pixels are red and if the area is greater than 200 pixels, the bounding box is depicted, and the cycle continues until there are no more detected areas to check. When all detections are being verified and saved in memory, each thread is being checked if any additional frame is saved in the last 5 seconds. If there are no more frames in the disk in the last 5 seconds, then each thread saves the data in the disk. Finally, if there is no program interruption the main program loop repeats until it is being interrupted. Using this method 4 983 images were collected in 6 hours.

Figure 3. The image of the HIKVISION security camera (https://www.hikvision.com/en/).

Data collection was made on five “HIKVISION” IP cameras (see Figure 3) with similar angles and image size of 352 x 288 pixels. To collect images from those cameras the Python programming language and computer vision library “OpenCV” version '4.5.5' was chosen together with the Python threading library to collect data from multiple cameras at once. To collect data (images and object annotations) the Gaussian Mixture-based Background/Foreground Segmentation Algorithm (MOG2) was used. One important feature of this algorithm is that it selects the appropriate number of gaussian distributions for each pixel. It provides better adaptability to varying scenes due illumination changes.

3.3. Examples and main issues for collected images

An automatic data (images) collecting process was applied for five security cameras in parallel. The MOG2 algorithm was able to properly detect changes in frames and put bounding boxes on those detected areas. To ensure that detected movements were caused by humans the size of detected object contour area was evaluated. If the area was bigger than 200 pixels the detection was attributed as human, and a bounding box was designated to this area. However, analyzing various scenarios, it was noticed that saving

bounding boxes that are bigger than 200 pixels does not ensure that only humans were detected (see Figure 5). Also, smaller areas led to inadequate results (see Figure 4). The video signal from cameras was cropped in 5 seconds intervals. A 5 second gap between images with bounding boxes was important to increase the possible variety of images, instead of gathering every single frame of a single person. This is essential for the next part of this research where human tracking models were applied to ensure that the model does not overfit the training dataset.

To ensure the quality of the object tracking model some images were removed from the dataset because of the faulty human detections of moving areas. At this point images were filtered manually to avoid possible misinterpretations. The main criteria for images were that it must have detected most of a human's body (from knees up to head, along with maximum width that was visible).

Furthermore, it was quickly noticed that in one security camera a relatively large part of detections were a TV screen that was showing various advertisements and invalidating those images, human recognition process and labels (see Figure 6 (left)). Another camera was located in the corner of the shop and recorded every movement outside the window. These images were also removed from the object tracking model training dataset (see Figure 6 (right)).

Figure 4. An example of image processing when movement is not detected.

Figure 5. An example of moving object detection using MOG2 algorithm.

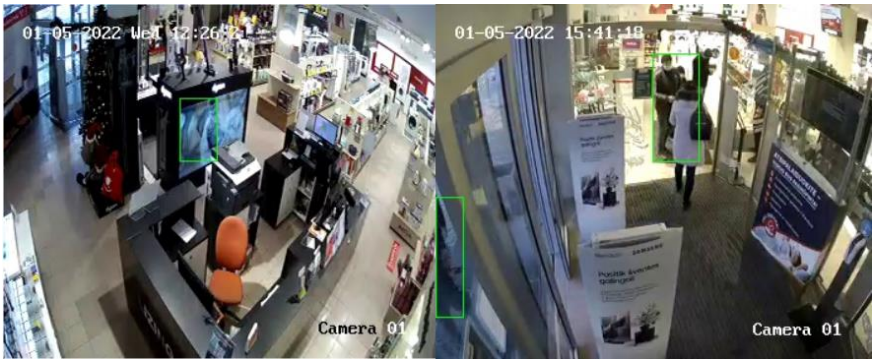

Figure 6. An example of the TV screen detection (left); An example of the human recognition outside the store (right).

After filtering the dataset was reduced from 8 000 images to 2 000 images, indicating about 75 % failure rate. Obtained labels were combined into .csv files, split to training and testing datasets, and then converted to TensorFlow record format. Even though the accuracy seems to be low, the automated data collection process is much faster than manual tracking, saving pictures and marking objects (humans).

4. Multiple objects tracking methods

Image labels were saved in the Pascal VOC XML Annotation format together with the class assigned to object detection, image itself and coordinates information of human location in the image. Several models were compared in the multiple object detection part that are based on different architectures:

- CNN models
- ResNet models
- Pretrained Custom Vision ML models

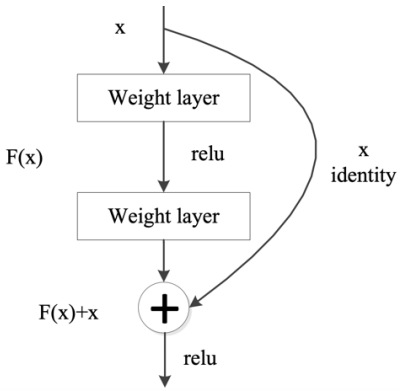

Figure 7. ResNet's module with ReLu activation function (Guo et al., 2019).

In this paper a represented ResNet model for human detection consists of three main blocks: 2D convolutional layers, batch normalization block and ReLU blocks. A skip connection is applied before the ReLU activation function that allows information to flow from earlier model layers to following layers. A schematic block of the ResNet model with the ReLU activation is represented in Figure 7. The skip connection of the residual block allows the model to achieve very deep structures without a gradient problem. A residual block is also called the identity block.

Another important block of the ResNet model is a Batch normalization that accelerates a Deep Network training by reducing internal covariate shift (Ioffe and Szegedy, 2015). It allows the use of much higher learning rates and simplifies the initialization. Also, the Batch normalization sometimes eliminates the need for dropouts.

However, the ResNet architecture models are capable of only one object detection at a time. For the multiple object detection, a mixture model was implemented that combines the ResNet model together with YOLO technique. The loss function is defined as *G* and its output is a one-dimension list defined as (4).

$$G(t, C, B) = t \cdot t(C + B \cdot 5) \tag{4}$$

where t corresponds to a particular cell of the grid where a fixed number of bounding boxes are being predicted; *C* is the number of object classes; *B* represents the number of coordinates in each cell that allows overlapping objects.

In the object tracking process several algorithms were considered like Kernel correlation filter (KCF), Multiple instance learning (MIL), Boosting (López-Sastre et al., 2019). These and similar methods are implemented in the OpenCV library. However, they are not suitable for tracking multiple objects at the same time. That is why the DeepSORT algorithm (Veeramani et al., 2018) was considered. It extracts object features using a model feature map to specify object tracking without knowing the full object trajectory. This method makes predictions and corrects results according to real object observations. The changes are made with respect to the rate of change in X and Y axis and height, and width of each object. This way the partial object tracking is implemented when detected objects do not appear in each frame. This technique was only applicable on CNN and ResNet-YOLO architecture models because the Custom Vision library does not allow export feature maps of the model.

Even though the Custom Vision model may be applied to this multiple object detection task, it is not capable to track objects (humans) almost chaotic behavior. For example, while looking in the video recordings, it was noticed that some people may cross each other's path and their bounding boxes overlap. Furthermore, it is important to be able to recognize which detected object corresponds to which object that is being tracked. The track of each object needs to be saved and recognized especially when several objects are being tracked at the same time. Since it was not possible to apply DeepSORT technique to this architecture, the intersection over union (IOU) method was realized (Bao et al., 2021). If two objects are on top of each other and have the same overlap area values, then the IOU value is equal to 1 for both objects. If objects have no overlapping areas, then the IOU value is equal to zero.

For the model comparison a mean average precision *mAP* values were estimated (eq. (5)).

$$mAP = \frac{1}{N} \sum_{i=1}^{N} \frac{\sum_{j=1}^{I} P_{ij}}{I} \tag{5}$$

where N is the number of object classes that are being predicted; I corresponds to the number of IOU boundaries; variable P_{ij} is an accuracy of the object class in the j^{th} boundary.

The proposed method starts object tracking when the first time it is being detected by generating a new track. The track variable saves a position, size, acceleration of the X and Y axis, width, and height of the object. At the initial state this track has an undefined phase. After a certain number of correct predictions, the track is being confirmed and defined. If the track is invisible in a certain number of frames, it needs to be cancelled. If an object is not detected in a frame, the track position value is updated according to the recent acceleration values. When objects are detected in the image they are referred to tracks with respect to their present positions. The equivalence is estimated using IOU value that helps to identify which track corresponds to which object.

5. Models' comparison and results

All analyzed models were trained using 2000 and 3000 images that were collected from five security cameras. For the model comparison part, a separate testing data set was prepared that includes 466 pictures (300 pictures with one or multiple people and 166 pictures without any moving object). In total there were 799 objects.

Several different CNN models were trained and compared. However, all of them had high time consumption and not sufficient accuracy. The final model architecture consists of several 2D convolutional layers, pooling layers, flattening and dense layers (in the Table 1 the final CNN model is named *Model1_CNN*).

ResNet architecture models use identification and convolutional blocks. They consist of 2D convolutional layers with Batch Normalization function and ReLu activation function between identity block groups. To avoid overfitting the automated callbacks were applied. At first, some checkpoints (results obtained during training, between epochs) were saved and if several epochs did not improve the validation loss value, the model training stopped. Also, it was noticed that using high filter count layers early resulted in better performance than later high filter counts. All analyzed models were created like a ResNet 50-layer network. It should be noted that not all models were included in the comparison list because their training process took more than 126 hour and were not accurate enough. Other models in 60 hours training still could not reach higher than 10% confidence level. These results led to inaccuracies like only few objects were detected in all images or too many objects were assigned as humans. It appears that the best ResNet model is 50-layer network with 7x7 kernel size and additional convolution block with 2 identity blocks together with another identity block in the previous identity block group. For the ResNet model the YOLO output structure was used to identify several objects in one image (in Table 1 it is named *Model2_final*).

Finally, other models that were compared in this paper are based on Custom Vision Machine Learning. In total four models were trained and tested:

- *Model3_quick*: only the last layer of the whole Custom Vision architecture was trained at Microsoft Azure platform. Training time is not defined and depends on the loss function minimization (from 5 to 15 min approximately).
- *Model4_advanced*: the last three layers of the Custom Vision network were trained at Microsoft Azure platform.

- *Model5_advanced*: the last three layers of the Custom Vision network were trained at Microsoft Azure platform (one hour longer than *Model4_advanced* model).
- *Model6_grey*: the last three layers of the Custom Vision network were trained at Microsoft Azure platform. This model was trained with filtered images where RGB spectrum was transformed into grayscale.

The *mAP* value was selected as the loss function to evaluate the performance of different models. Table 1 summarizes results of different models that were trained using 2000 images. CNN and ResNet models are not accurate enough and in 60 hours of training the *mAP* value reached only 0,12. Due to high computational complexity those models were stopped earlier and could not be trained properly. Meanwhile pretrained Custom Vision models are more accurate (*mAP* value reaches 0.72).

Table 1. Models testing results using different architectures (using 2000 images for training).

Model	Architecture	Training time	Testing time, sec.	Num. of recognized objects	*mAP*
Model1_CNN	CNN	60 hours	253.96	466	0.12
Model2_final	ResNet+YOLO	60 hours	37.70	2767	0.03
Model3_quick	Custom Vision	5 – 15 min	12.34	845	0.70
Model4_advanced	Custom Vision	1 hour	18.86	770	0.67
Model5_advanced	Custom Vision	2 hours	18.56	813	0.72
Model6_grey	Custom Vision	2 hours	25.35	786	0.66

Further analysis showed that model accuracies reduce in time. It was noticed that the primary dataset for training (2000 images) consists of only one season (winter) data where people are dressed up with warm and bloated jackets. Meanwhile, in spring more and more people came to the store looking thinner. To increase *mAP* values additional data set for training was prepared and the number of images increased from 2000 to 3000. The testing dataset was not changed because it already contained images from both seasons. Custom Vision models were trained longer to reach similar *mAP* with a bigger dataset. However, in 10 or 11 hours of training, testing accuracies barely increased (see Table 2). Also, the *mAP* values did not change in CNN and ResNet models. According to these results the best model is based on Custom Vision architecture where the last three layers are trained at Microsoft Azure platform.

Table 2. Models testing results using different architectures (using 3000 images for training).

Model	Architecture	Training time	Testing time, sec.	Num. of recognized objects	*mAP*
Model1_CNN	CNN	60 hours	1007.49	466	0.12
Model2_final	ResNet+YOLO	60 hours	142.55	2767	0.03
Model3_quick	Custom Vision	5 – 15 min	20.85	894	0.71
Model4_advanced	Custom Vision	10 hours	19.08	810	0.73
Model5_advanced	Custom Vision	11 hours	19.53	810	0.73
Model6_grey	Custom Vision	10 hours	20.21	682	0.66

To better understand trained Custom Vision models behaviours the Receiver Operating Characteristic (ROC) and Precision-Recall (P-R) curves are presented (see Figure 8 and Figure 9 respectively). To avoid 10 or more hours training time duration only *Model3_quick* and *Model4_advanced* models were compared. It could be seen that all three models have similar P-R curves that overlap in some points. However, the best performance is reached when *Model3_quick* is trained using 3000 images (see Figure 8). Even though accuracy is not very high and reaches only 80%, the trained Custom Vision model is sufficient for human's behavior tracking in a store.

As it was mentioned before, the Custom Vision network is confidential, and it is not allowed to export feature maps of the trained model. That is why the object tracking was realized using the IOU method. The average precision values according to different IOU thresholds are shown in Figure 10. It could be noticed that the average precision value of each model reduces when the IOU threshold increases. If the IOU value is close to zero, the precision reaches approximately 0.7. The best IOU threshold is considered to be 0.5 with approx. 0.4 precision and 0.7 recall values (see Figure 9 and Figure 10).

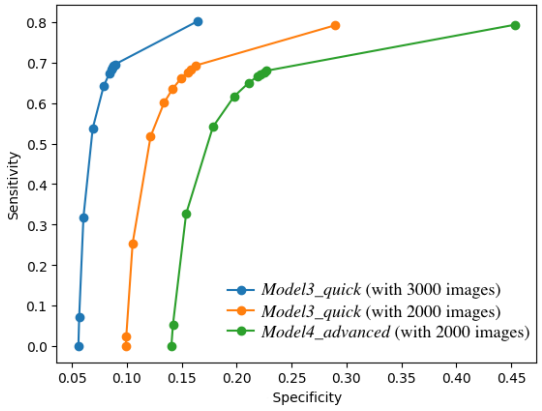

Figure 8. ROC curve.

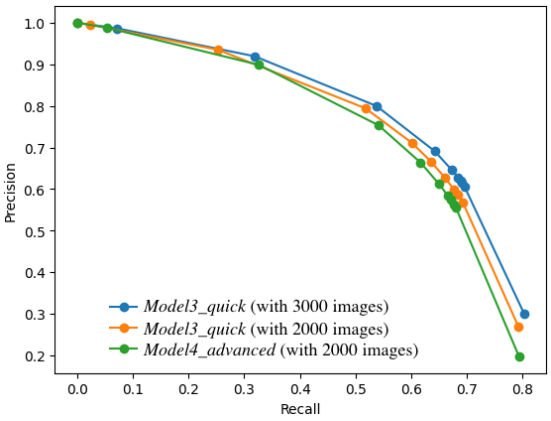

Figure 9. P-R curve.

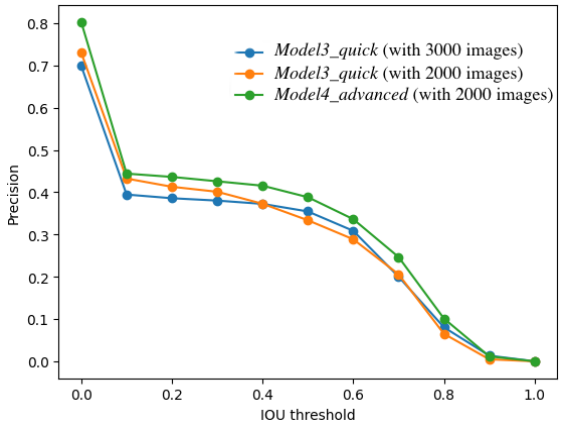

Figure 10. Averaged precision values with different IOU thresholds.

Figure 11. A heatmap of visited places in the store (averaged values in 4 hours recordings).

The proposed technique that consists of a trained Custom vision model together with IOU methods was applied for human tracking in the store. The most popular sections are recognized and the Heatmap is created (see Figure 11). In this example the Heatmap shows the average number (in 4 hours) of visitors in each section. Green areas represent less popular places and red color shows where customers are most likely appearing. Figure 12 shows an example of how sections are split into separate segments in the real-time video recording. If the bottom of a bounding box that represents human crosses a segment (particular rectangle) in a store, the area is marked as visited. To overcome the possible faulty detections, the additional rule was applied: if the detected object did not move, the tracker would not send the data to the system and this detection is not considered as human. This allows to record only moving objects in a store.

Several examples of human recognition and tracking in a store are presented in Figure 13. As It could be seen that the proposed technique works sufficiently well. Humans are detected in every image where rectangles (blue and red) mark each moving object. However, it could be noticed that some people are left without noticing, especially if they are very close to each other. This implies that the model can detect a human but with some inaccuracies in coordinates of the detected area. Even though some errors appear in the detection process, the most visited areas are properly identified.

Figure 12. An example of separate section in the real time recording.

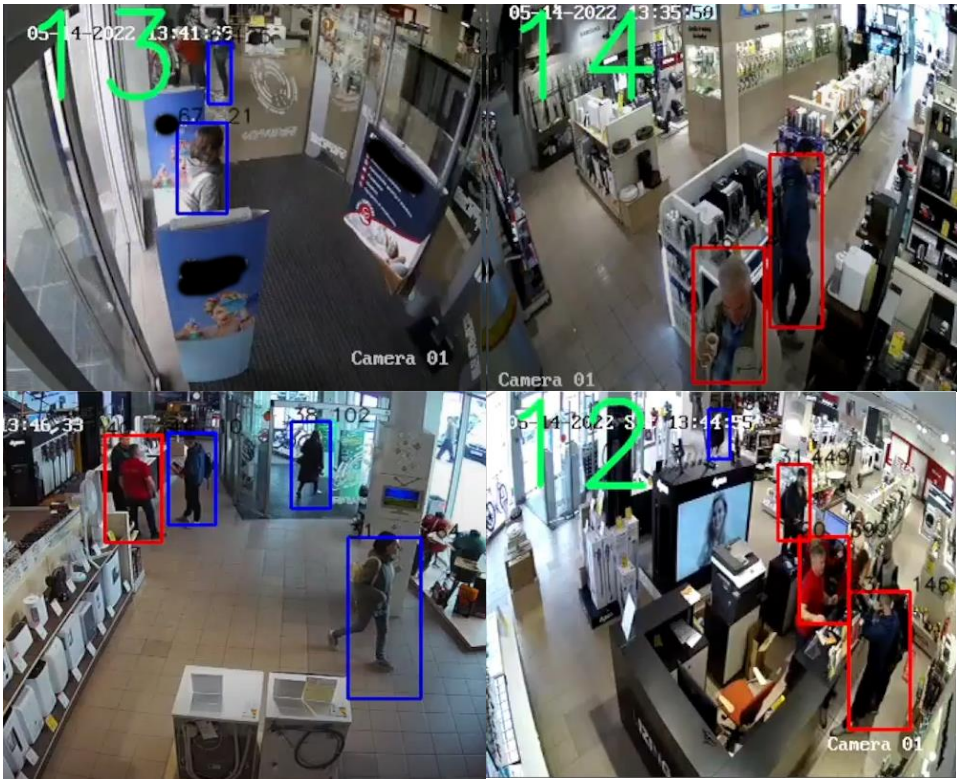

Figure 13. Examples of live object tracking the store.

6. Conclusions

For the data gathering process an automatic tool was designed that includes five security cameras and the MOG2 algorithm for human detection and labeling in each image. The main issue in collected data appeared to be false detections (bounding boxes placed in the wrong image parts) such as TV screens or humans behind store's glass or even not precise (too big or too small) area of human body. After filtering the dataset was reduced from 8000 to 2000 images that represents 75% failure rate.

Three different architectures were compared in the human detection and tracking part: CNN, ResNet and Custom Vision. It appears that due to time limitations and low dataset for the training process, it is very hard to train CNN and ResNet models. Meanwhile, the pre-trained Custom Vision models may reach 80% accuracy in 1 hour of training at Microsoft Azure platform. However, it is not possible to export feature maps of these trained models. To apply the real-time object tracking the trained Custom Vision model was used together with the IOU method to measure the possible object overlapping and track customers behavior in a store.

The proposed technique was applied to the most popular sections detection in a store creating a heatmap for visualization of collected data. It was defined that if the bottom of a bounding box that represents human crosses a particular segment in a store, the area is marked as visited. To overcome the possible faulty detections, the additional rule was

applied: if the detected object did not move, the tracker would not send the data to the system and this detection is not considered as human. This allows to record only moving objects in a store.

Future research should include gathering bigger datasets to train the Custom Vision model and reach better performance. Images from various seasons and different stores should also be included in the future analysis.

References

Al-Hazaimeh, O. M., Al-Nawashi, M., Saraee, M. (2019). Geometrical-based approach for robust human image detection. *Multimedia Tools and Applications*, *78*(6), 7029–7053. https://doi.org/10.1007/s11042-018-6401-y

Al-Nawashi, M., Al-Hazaimeh, O. M., Saraee, M. (2017). A novel framework for intelligent surveillance system based on abnormal human activity detection in academic environments. *Neural Computing and Applications*, 565–572. https://doi.org/10.1007/s00521-016-2363-z

Alam, M., Samad, M. D., Vidyaratne, L., Glandon, A., Iftekharuddin, K. M. (2020). Survey on Deep Neural Networks in Speech and Vision Systems. *Neurocomputing*, *417*, 302–321. https://doi.org/10.1016/j.neucom.2020.07.053

Alçiçek, Z., Balaban, M. Ö. (2012). Development and application of "the Two Image" method for accurate object recognition and color analysis. *Journal of Food Engineering*, *111*(1), 46–51. https://doi.org/10.1016/j.jfoodeng.2012.01.031

Ali, O., Ishak, M. K. (2020). Bringing intelligence to IoT Edge: Machine Learning based Smart City Image Classification using Microsoft Azure IoT and Custom Vision. *Journal of Physics: Conference Series*, *1529*(4). https://doi.org/10.1088/1742-6596/1529/4/042076

Alom, M. Z., Yakopcic, C., Nasrin, M. S., Taha, T. M., Asari, V. K. (2019). Breast Cancer Classification from Histopathological Images with Inception Recurrent Residual Convolutional Neural Network. *Journal of Digital Imaging*, *32*(4), 605–617. https://doi.org/10.1007/s10278-019-00182-7

Bao, J., Wang, H., Lv, C., Luo, K., Shen, X. (2021). IOU-Guided Siamese Tracking. *Mathematical Problems in Engineering*, *2021*. https://doi.org/10.1155/2021/9127092

Bazhenov, N., Korzun, D. (2019). Event-Driven Video Services for Monitoring in Edge-Centric Internet of Things Environments. *Conference of Open Innovation Association, FRUCT*, 47–56. https://doi.org/10.23919/FRUCT48121.2019.8981505

Bertozzi, M., Broggi, A., Chapuis, R., Chausse, F., Fascioli, A., Tibaldi, A. (2003). Shape-based pedestrian detection and localization. *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC*, *1*, 328–333. https://doi.org/10.1109/ITSC.2003.1251972

Cesta, A., Cortellessa, G., Fratini, S., Oddi, A., Denis, M., Donati, A., Policella, N., Rabenau, E., Schulster, J. (2007). Mexar2: Al solves mission planner problems. *IEEE Intelligent Systems*, *22*(4), 12–19. https://doi.org/10.1109/MIS.2007.75

Dalal, N., Triggs, B. (2005). Histograms of oriented gradients for human detection. *Proceedings - 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2005*, *I*, 886–893. https://doi.org/10.1109/CVPR.2005.177

Drane, D. L., Loring, D. W., Voets, N. L., Price, M., Ojemann, J. G., Willie, J. T., Saindane, A. M., Phatak, V., Ivanisevic, M., Millis, S., Helmers, S. L., Miller, J. W., Meador, K. J., Gross, R. E. (2015). Better object recognition and naming outcome with MRI-guided stereotactic laser amygdalohippocampotomy for temporal lobe epilepsy. *Epilepsia*, *56*(1), 101–113. https://doi.org/10.1111/epi.12860

Ehatisham-Ul-Haq, M., Javed, A., Azam, M. A., Malik, H. M. A., Irtaza, A., Lee, I. H., Mahmood, M. T. (2019). Robust Human Activity Recognition Using Multimodal Feature-Level Fusion. *IEEE Access*, *7*, 60736–60751. https://doi.org/10.1109/ACCESS.2019.2913393

Fang, W., Ding, Y., Zhang, F., Sheng, V. S. (2019). DOG: A new background removal for object recognition from images. *Neurocomputing*, *361*, 85–91. https://doi.org/10.1016/j.neucom.2019.05.095

Fradi, H., Dugelay, J. L. (2012). Robust foreground segmentation using improved Gaussian mixture model and optical flow. *2012 International Conference on Informatics, Electronics and Vision, ICIEV 2012*, 248–253. https://doi.org/10.1109/ICIEV.2012.6317376

Guo, Q., Yu, X., Ruan, G. (2019). LPI radar waveform recognition based on deep convolutional neural network transfer learning. *Symmetry*, *11*(4), 1–14. https://doi.org/10.3390/sym11040540

He, K., Zhang, X., Ren, S., Sun, J. (2016). Deep residual learning for image recognition. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, *2016-Decem*, 770–778. https://doi.org/10.1109/CVPR.2016.90

Ioffe, S., Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. *32nd International Conference on Machine Learning, ICML 2015*, *1*, 448–456.

Jacobs, H. I. L., Gronenschild, H. B. M., Evers, E. A. T., Ramakers, I. H. G. B., Hofman, P. A. M., Backes, W. H., Jolles, J., Verhey, F. R. J., Van Boxtel, M. P. J. (2015). Visuospatial processing in early Alzheimer's disease: Amultimodal neuroimaging study. *Cortex*, *64*, 394–406. https://doi.org/10.1016/j.cortex.2012.01.005

Jiang, L., Peng, G., Xu, B., Lu, Y., Wang, W. (2018). Foreign object recognition technology for port transportation channel based on automatic image recognition. *Eurasip Journal on Image and Video Processing*, *1*. https://doi.org/10.1186/s13640-018-0390-7

Kalake, L., Dong, Y., Wan, W., Hou, L. (2022). Enhancing Detection Quality Rate with a Combined HOG and CNN for Real-Time Multiple Object Tracking across Non-Overlapping Multiple Cameras. *Sensors*, *22*(6). https://doi.org/10.3390/s22062123

Kjellström, H., Romero, J., Kragić, D. (2011). Visual object-action recognition: Inferring object affordances from human demonstration. *Computer Vision and Image Understanding*, *115*(1), 81–90. https://doi.org/10.1016/j.cviu.2010.08.002

Kodan, A., Chauhan, A. (2020). Reply to article "Inequality and the future of healthcare": Embracing AI for primary healthcare physicians! *Journal of Family Medicine and Primary Care*, *9*, 5397.

Krizhevsky, A., Sutskever, I., Hinton, G. E. (2017). ImageNet classification with deep convolutional neural networks. *Communications of the ACM*, *60*(6), 84–90. https://doi.org/10.1145/3065386

Kulikajevas, A., Maskeliunas, R., Damaševičius, R. (2021). Detection of sitting posture using hierarchical image composition and deep learning. *PeerJ Computer Science*, *7*, 1–20. https://doi.org/10.7717/peerj-cs.442

Kumari, S., Choudhary, M., Kumari, K., Kumar, V., Chowdhury, A., Chaulya, S. K., Prasad, G. M., Mandal, S. K. (2022). Intelligent driving system at opencast mines during foggy weather. *International Journal of Mining, Reclamation and Environment*, *36*(3), 196–217. https://doi.org/10.1080/17480930.2021.2009724

Lagrange, A., Fauvel, M., Grizonnet, M. (2017). Large-Scale Feature Selection With Gaussian Mixture Models for the Classification of High Dimensional Remote Sensing Images. *IEEE Transactions on Computational Imaging*, *3*(2), 230–242. https://doi.org/10.1109/tci.2017.2666551

Liu, Y. (2021). Human motion image detection and tracking method based on Gaussian mixture model and CAMSHIFT. *Microprocessors and Microsystems*, *82*(December 2020). https://doi.org/10.1016/j.micpro.2021.103843

López-Sastre, R. J., Herranz-Perdiguero, C., Guerrero-Gómez-olmedo, R., Oñoro-Rubio, D., Maldonado-Bascón, S. (2019). Boosting multi-vehicle tracking with a joint object detection and viewpoint estimation sensor. *Sensors (Switzerland)*, *19*(19), 1–24. https://doi.org/10.3390/s19194062

Nguyen, D. T., Li, W., Ogunbona, P. O. (2016). Human detection from images and videos: A survey. *Pattern Recognition*, *51*, 148–175. https://doi.org/10.1016/j.patcog.2015.08.027

Power, P. W., Schoonees, J. a. (2002). Understanding Background Mixture Models for Foreground Segmentation Understanding Background Mixture Models for Foreground Segmentation. *Image and Vision Computing*, 267–271.

Raskar, P. S., Shah, S. K. (2021). Real time object-based video forgery detection using YOLO (V2). *Forensic Science International*, *327*(180), 110979. https://doi.org/10.1016/j.forsciint.2021.110979

Redmon, J., Divvala, S., Girshick, R., Farhadi, A. (2016). You only look once: Unified, real-time object detection. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, *2016-Decem*, 779–788. https://doi.org/10.1109/CVPR.2016.91

Serban, A., Poll, E., Visser, J. (2020). Adversarial Examples on Object Recognition: A Comprehensive Survey. *ACM Computing Surveys*, *53*(3). https://doi.org/10.1145/3398394

Simonyan, K., Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, 1–14.

Veeramani, B., Raymond, J. W., Chanda, P. (2018). DeepSort: Deep convolutional networks for sorting haploid maize seeds. *BMC Bioinformatics*, *19*(Suppl 9), 1–9. https://doi.org/10.1186/s12859-018-2267-2

Zivkovic, Z. (2004). Improved adaptive Gaussian mixture model for background subtraction. *Proceedings - International Conference on Pattern Recognition*, *2*, 28–31. https://doi.org/10.1109/icpr.2004.1333992

Received February 28, 2022, accepted August 20, 2022