Seeing without sight – An automatic cognition system dedicated to blind and visually impaired people

Bogdan Mocanu[1,2], Ruxandra Tapu[1,2] and Titus Zaharia[1]
[1]ARTEMIS Department, Institute Mines - Télécom/Télécom SudParis, UMR CNRS 8145 - MAP5 and 5157 SAMOVAR, Evry, France
[2]Department of Telecommunications, Faculty of ETTI, University "Politehnica" of Bucharest
e-mail: {bogdan.mocanu, ruxandra.tapu, titus.zaharia}@telecom-sudparis.eu

Abstract

In this paper we present an automatic cognition system, based on computer vision algorithms and deep convolutional neural networks, designed to assist the visually impaired (VI) users during navigation in highly dynamic urban scenes. A first feature concerns the real-time detection of various types of objects existent in the outdoor environment relevant from the perspective of a VI person. The objects are followed between successive frames using a novel tracker, which exploits an offline trained neural-network and is able to track generic objects using motion patterns and visual attention models. The system is able to handle occlusions, sudden camera/object movements, rotation or various complex changes. Finally, an object classification module is proposed that exploits the YOLO algorithm and extends it with new categories specific to assistive devices applications. The feedback to VI users is transmitted as a set of acoustic warning messages through bone conducting headphones. The experimental evaluation, performed on the VOT 2016 dataset and on a set of videos acquired with the help of VI users, demonstrates the effectiveness and efficiency of the proposed method.

1. Introduction

Recent statistics, relative to people with visual disabilities published by the World Health Organization (WHO) [1] in August 2014, show that more than 0.5% of the total population suffers from visual impairments (VI). Among these, 39 million people are completely blind. Unfortunately, by the year 2020 worldwide the number of individuals with VI is estimated to double [2].

Regular activities, commonly performed by normal humans, such as: safe navigation in a novel indoor/outdoor environment, independent shopping or simply reaching a desired destination become highly challenging for VI people [3]. In order to infer additional cognition over the surroundings, the VI users rely on traditional assistive elements. Most often, they concern trained dogs or white canes. Although such elements are quite popular, they show quickly their limitations when confronted to the high dynamics of a real outdoor scene. Today, the white cane always represents the simplest and most affordable travel aid available. However, it requires an actual contact with the obstacle. In addition, it cannot offer information about the object type, its degree of danger, time to collision, and it cannot detect overhanging obstacles.

Within this context, the elaboration of an assistive device dedicated to blind and visually impaired people that can improve cognition over the environment and facilitate the safe, autonomous navigation in novel outdoor spaces is a crucial challenge.

In this paper, we propose an assistive device that combines computer vision techniques and deep convolutional neural networks in order to detect, track and recognize objects encountered during the outdoor navigation. The major contributions proposed concern: (1) a novel object tracking algorithm that uses a regression-based approach to learn offline relationships between the object appearances and its associated motion patterns; (2) a visual attention model able to handle object occlusions, sudden camera and object movements, while minimizing the drift; (3) an object recognition methodology that exploits the YOLO [4] approach and extends it with new categories specific to VI–dedicated assistive devices; (4) a cognition system able to understand the recognized objects and launch acoustic warnings only for relevant obstacles depending on their degree of danger.

At the hardware level, the proposed system is composed of a regular video camera, a processing unit (an ultra book computer equipped with an nVidia (GTX 1050) graphical board and bone conduction headphones.

The rest of the paper is organized as follows: in Section 2, we briefly review the state of the art. The focus is put on assistive systems, based on computer vision methods, dedicated to the VI users. Section 3 presents the proposed cognition system that involves two major stages: obstacle detection and tracking. The experimental results, conducted on the VOT 2016 [5] dataset as well as on a video corpus acquired in real life scenarios are presented in Section 4. Finally, Section 5 concludes the paper and opens new directions for further work.

2. Related work

In the last years, due to the proliferation of computer vision algorithms, various systems dedicated to blind and VI users exploiting various artificial intelligence paradigms have been proposed [6].

The SmartVision system introduced in [7] is designed to detect sidewalks boarders and objects situated in front of the VI user. The safe walking path is determined based on the Canny edge detection algorithm and relevant edge selection in an adapted Hough space. The obstacles are identified using a zero crossing approach and texture masks. However, the system is highly sensitive in the presence of multiple edges in the scene (*e.g.,* street intersection or crossroads).

The Mobile Vision framework proposed in [8] is completely integrated on a mobile device. The prototype is designed to detect landmarks in the environment and to guide the VI person towards such landmarks. The system identifies objects of interest using an image color histogram representation and an edge detection algorithm. However, because the smartphone needs to be handheld, the method is considered as intrusive.

In [9] and [10] a real-time obstacle detection and classification system integrated on a smartphone device is proposed. The detection algorithm is based on interest point extraction and tracking, camera motion estimation and moving object identification based on motion vectors. The recognition process exploits a BoW / VLAD image representation used within a SVM training/prediction process. Even though the system returns overall good results, it cannot detect large, flat structures or correctly estimate the distance between the VI user and an obstruction.

In [11], an embedded 6DOF SLAM dedicated to VI users is proposed. The system performs ego-motion estimation by integrating 3D/2D object appearance and scene global rectification using entropy minimization. The system works in near real-time. However, at this point the framework has a reduced applicability in the context of the VI users and needs to be further extended in order to incorporate additional semantic information.

A head-mounted, stereo vision navigation assistant for VI is proposed in [12]. In order to extract and maintain the orientation information, the authors incorporate visual odometry and feature-based metric topological SLAM. A map of the user surrounding environment is constructed from the dense 3D data. From the user's perspective, the system is considered as invasive because it needs to be mounted on the head. In addition, it requires a powerful processing unit that needs to be carried during navigation.

An aerial obstacle detection algorithm embedded on a 3D mobile device is proposed in [13]. The system performs scene reconstruction using depth maps, while the obstacles are detected using the distance histogram extracted from the 3D data. The algorithm has been tested by actual VI users and proves to be effective. However, the approach is highly sensitive to sudden camera movements and changes in the light intensity.

In [14], a RGB-D assistive device is designed to detect humans and recognize objects. As indicated by the authors, the system properly functions only in indoor scenarios. In addition, the use of regular headphones is inappropriate in the context of VI users.

The Kinect Cane system introduced in [15] is designed to recognize objects from depth data using the Kinect sensor. The method detects different types of objects and informs the VI user about the object's type such as: chairs or upwards stairs. The feedback is transmitted to the VI user through vibrations.

Similarly, in [16], a Microsoft Kinect system is proposed to perceive the environment and to identify nearby structures. Both methods introduced in [15] and [16] are dedicated to indoor navigation scenarios. In addition, they prove to be highly sensitive to the training phase.

In [17], the authors propose a complete system that performs simultaneously moving object detection and tracking, subject localization and map extraction using a RGB-D camera. The system works in real-time and proves to be robust to ego-motion and noise. However, it is highly sensitive to changes in the illumination conditions and require a dedicated processing unit.

The analysis of the state of the art shows that each method has its own advantages and limitations over the others. However, for the moment, the VI users cannot be completely confident about the robustness, reliability or overall performance of the existing prototypes.

3. Proposed approach

Fig. 1 presents the proposed framework, with the main steps involved: object detection and recognition, object tracking and acoustic feedback.

3.1. Object detection module

In recent years, the tracking-by-detection approaches have become increasingly popular for solving the problem of robust object localization in subsequent video frames, despite important object motion, changes in view-point or other acquisition-related variations.

The initial object detection is performed by applying the YOLO [4] algorithm on the first frame of the video stream. YOLO treats the object detection problem as a regression mechanism for spatially separated bounding boxes and their associated class probabilities. We decided to use YOLO due to the real-time processing capabilities and its reduced number of false positives. In addition, the detector can be used to predict candidate location for novel objects in video frames where such action is required.

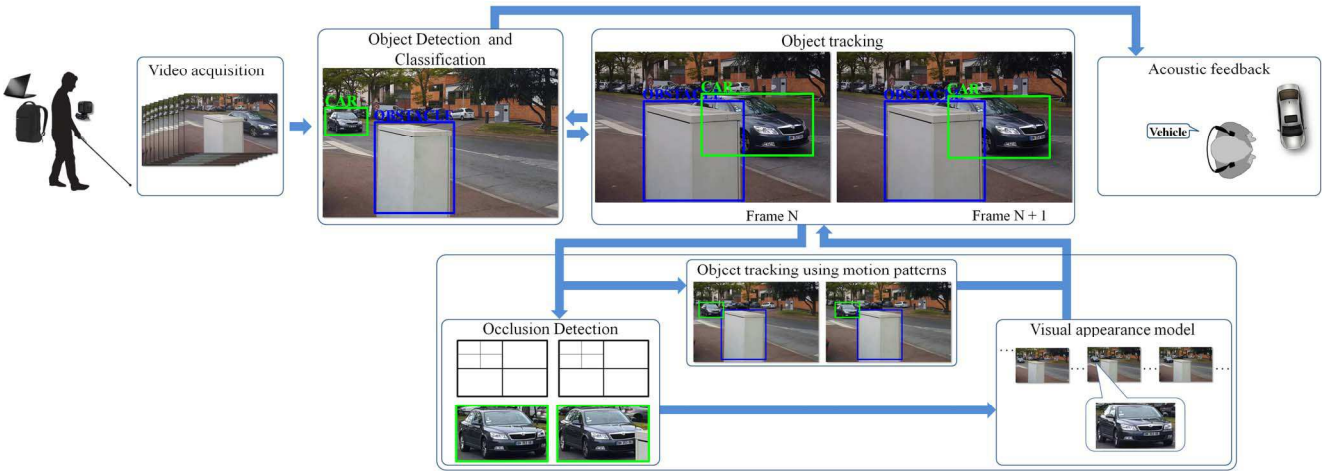

Figure 1: The global architecture of the proposed approach.

Each detected object is tracked in the subsequent video frames, as described in the following section.

3.2. Object tracking

The proposed approach is a generic object tracker based on two convolutional neural networks trained offline. The key principle consists of alternating between tracking using motion information and predicting the object location in time based on visual similarity.

Initialization phase. As for the GOTURN approach [18], our tracker uses a regression-based technique to learn offline generic relationships between the object appearances and its associated motion patterns. The training of the neural network is performed offline with moving objects instances taken from the real world. When tracking novel objects, the sets of weights characterizing the neural network remain unchanged. In this way, no online fine tuning is performed.

We have adopted a similar network architecture as the one proposed by GOTURN [18]. The network receives as input the target object as well as the associated search regions. The output is a set of high level image features that are applied as input to the fully connected layers. The role of the fully connected layers is to compare the feature from the target object in the current frame and to estimate the novel location of the object of interest in the following frame.

The tracking system based on motion patterns proves to be very fast (30 fps), robust and accurate (*i.e.*, even when tracking objects that undergo important scale and appearance changes). However, tracking based solely on motion information suffers from several limitations such as: high sensitivity to sudden/large camera movement, incapacity to handle long-term occlusions, inability to deal with multiple moving objects located in the same vicinity. In addition, we argue that the object estimated position together with its associated context area is insufficient to reliably determine if the new location of the bounding box actually contains the object of interest.

In order to overcome such limitations, we propose to integrate in the process rich visual cues, established from the object previous positions and appearances. After obtaining the initial candidate location, we introduce a refinement strategy that aims to adaptively modify the bounding box position and shape in order to avoid incorrect/false object tracks due to the background clutter. The refinement process includes two stages, which are occlusion detection and object appearance modeling.

Occlusion identification and processing. The process of occlusion detection and handling is illustrated in Fig. 2.

We apply a quadtree decomposition algorithm in order to divide the candidate object location and its reference bounding box into a set of non-overlapping image patches. The partition process is repeated until the third level of decomposition. We decided to use only three levels of decomposition in order to ensure a "reasonable" degree of descriptiveness of the similarity measure. The image patches, at the initial resolution and from all levels of decomposition, are compared against the correspondent one in the reference frame.

The similarity degree between the image patches is obtained using the DeepCompare [19] algorithm. The comparison technique is a CNN-based model trained to take into account a wide variety of changes into the image appearance. The system does not require any manually tuned features and is able to learn, directly from the training data, a general similarity function that serves to compare patches.

The image patches are processed by using a 2-channel network architecture that offers the best trade-off between the computational speed and the system accuracy. The two patches being compared are considered as a 2-channel image that is directly applied to the first convolutional layer of the neural network.

Figure 2: Occlusion detection using quad-tree decomposition.

The bottom of the CNN is composed of a series of convolutional, ReLU and max-pooling layers. The top module is a fully connected, linear decision layer. The system has great flexibility and is fast to train.

In order to reduce the processing time we have adopted the following strategy that helps us to speed up the image patch comparison process. As in [20], we propose to divide the convolutional layers into smaller 3x3 kernels separated by ReLU activations. The similarity scores returned by the DeepCompare [19] algorithm can range between [-1.1, +10], where -1.1 signify the lowest visual similarity, while a value of +10 is returned for highly similar image patches.

The similarity score (S_{score}) between image patches is further analyzed in order to identify the beginning of an occlusion for the tracked object. We consider the object as being in an occluded state if the associated S_{score} for the image patches situated on the second and third level of decomposition return negative values (Fig. 2). Also, objects of interest characterized by larger bounding boxes show a similar behavior (*i.e.,* negative similarity scores on the 2nd and 3rd level of decomposition when compared with the reference object). If such parasite information is not eliminated, the tracking process can be biased and, at long term, the object of interest can be completely lost.

To overcome such limitations, we propose to update the size of the bounding box and adjust its shape/size in order to eliminate such undesired information. The system can perform the following cutting operations in four different directions: left (L), right (R), up (U) and bottom (B). The process consists in reducing the size of the bounding box with 1/8 of the initial size. The selection of 1/8 of the initial size makes it possible to avoid too brutal shrinkage of the bounding box. However, when a more powerful trimming operation is required, the process is applied recursively until the global similarity score with respect to the reference patch stops increasing. The object bounding box cutting direction is determined based on the visual similarity scores obtained after performing the quadtree decomposition. In the case of the example illustrated in Fig. 2, because on the second and third level of decomposition all similarity scores are negatives two cutting operations are evaluated on the bottom (B) and on the right (R) side (Fig. 3).

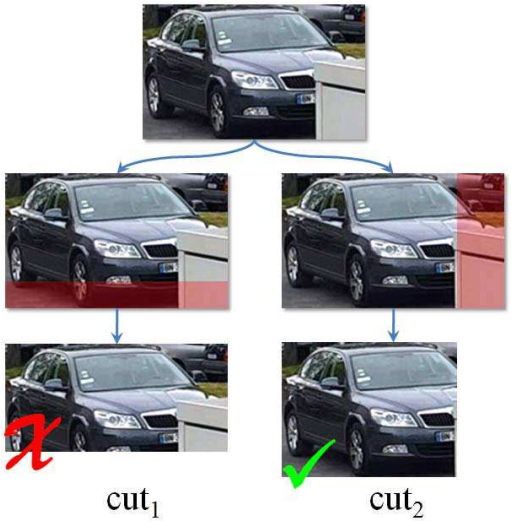

Figure 3: Object bounding box adjustment based on the maximum similarity score.

Let us denote by $S_{score_{cut_1}}$ and $S_{score_{cut_2}}$ the similarity scores (*w.r.t.* the reference patch) obtained by the two trimmed image patches. The system selects as the optimal cut the one that maximizes the similarity score (MS_{score}) presented in equation (1):

$$MS_{score} = \max\left\{S_{score_{cut_1}}; S_{score_{cut_2}}\right\}; \qquad (1)$$

In order to validate the cut, we impose MS_{score} to be superior to the original S_{score} computed between the image patches at the first level of decomposition.

In addition, no cut is allowed for image patches with less than 5 pixels on the third level of decomposition. We impose this constraint since small patches have a reduced descriptive power and DeepCompare cannot perform relevant evaluations.

A final stage in our refinement process concerns the construction and use of an adaptive object appearance model.

Adaptive object appearance model. In order to handle obstacles characterized by long term occlusion or large movements we propose to extend the tracker with an adaptive visual attention model. The proposed tracker is considerably more effective than a regular tracker based solely on a strong motion model. The key principle of the proposed approach consists in alternating between tracking using motion information and predicting the object location in time based on visual similarity.

Various trackers, based on visual features [21] construct

appearance models for both, the interest objects and the background information. Due to the real-time constraint imposed on our application, we decided to develop a model solely for the tracked objects. The major difficulty that needs to be addressed and solved is related to the extraction of reliable and representative object instances that can serve to effectively update the appearance model.

Commonly, most trackers use a single/fixed appearance model selected from the first frame of the video stream. However, such an approach shows quickly its limitations when confronted to the high dynamics of real urban scenes. A single model is insufficient to cope with important changes in obstacles shape, pose or features. In order to overcome this limitation, a continuous update of the object appearance model is required. In the state of the art, various authors consider as a positive example the tracker's current location and attempt to predict the object novel position within a neighborhood search area, by exploiting the object's trajectory information [22].

Even though this approach shows promising results, it suffers from several drawbacks. Thus, if the tracker is not sufficiently precise when estimating the novel object location, the object appearance model tends to be updated with sub-optimal positive examples (Fig. 4a). Over the time, the accumulation of such false positives can significantly degrade the model and determine the tracker's drift. In contrast, if multiple positive examples are selected from nearby locations, the object model is constantly updated and the current appearance can incorporate too much contextual information and thus become confusing (Fig. 4b).

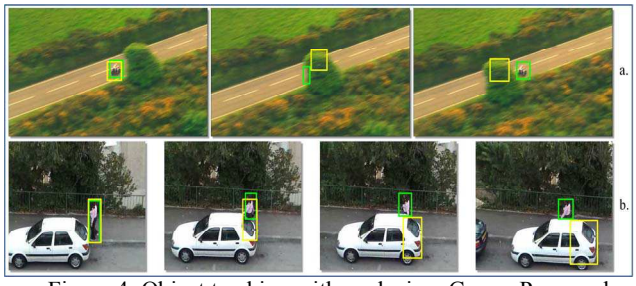

Figure 4: Object tracking with occlusion. Green: Proposed approach, Yellow: GOTURN algorithm; (a). Total occlusion; (b). Partial occlusion that degenerates the appearance model to incorporate false instances.

In our work, we have adopted a tracking-by-detection approach that continuously updates the object appearance model with novel instances whenever such an action is required. In contrast with other state of the art techniques [21], because of the real-time constraint imposed by the targeted application, no learning process is performed in the online stage.

In order to determine the novel location of the interest object a multiple patch matching strategy is proposed. The objective is to estimate, with high accuracy, the new position of the object bounding box in the adjacent frame.

The input is the candidate location returned by the motion-based tracking algorithm (*cf.* Section 3.2). The predicted object position, together with its associated context region is further analyzed for a more accurate object location estimation. The context region is subsequently used as a search area in order to determine, independently, the best location for each instance in the object appearance model. At each stage, the similarity score provided by the DeepCompare algorithm is computed.

In order to reduce the processing burden instead of a brute force search, we have adopted a hierarchical approach, similar to the block-based motion estimation in method used in MPEG-4 [23]. The location that yields the highest DeepCompare similarity score is retained as correct for the current object appearance model. The final object location in the adjacent frame corresponds to the instance that provides the maximal value of similarity.

To validate the object location, we impose the maximum similarity score to be superior to the average score obtained within the temporal sliding window that incorporates the last N video frames processed. In our experiments, we selected N equal with 10 frames that corresponds to a temporal interval slightly inferior to half a second.

The object appearance model is constantly updated with novel elements if the visual similarity scores of the current instance with *all* the frames being analyzed (located in the temporal sliding window) satisfy the similarity condition. In the same time, the most ancient object instance in the model is discarded. In this way, we ensure that the object appearance is not updated with sub-optimal instances (*i.e.*, occluded versions of the object).

3.3. Obstacle classification

The image patches are classified using a modified version of the YOLO [4] algorithm. We extended the system with additional training classes specific to a wearable assistive device dedicated to VI users.

The object class is predicted by performing a global reasoning about the entire video frame. Unlike traditional classification system our framework encodes during training and testing the contextual information about the object class and its appearance. Then, as for YOLO, our system learns generalizable representations of objects and is less likely to return false alarms or missed detection when applied to novel/unexpected video instances input.

In the context of VI user application we retained as relevant the following object classes: car, bicycle and humans. In addition, we constructed a novel global class called generic static obstructions with its associated subclasses: garbage cans, overhanging branches, fences, pylons, edge of pavements and stairs.

3.4. Acoustic feedback module

After the objects are tracked and classified we need to determine their degree of danger relative to the VI. We observed that not all obstacles presented in the scene represent a potential risk for the blind. We propose to use two proximity areas, situated in the near surrounding of a user, both with a trapezoidal shape: one situated on the persons walking path and the other at the head level (Fig. 5). The system will launch acoustic warnings only for the object situated in the areas delimitated by the trapeziums of interest.

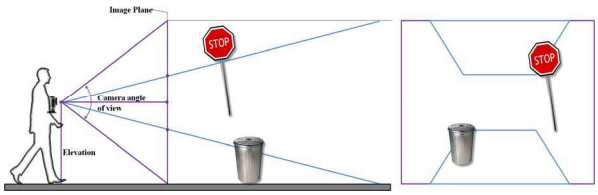

Figure 5: Visual impaired user proximity areas.

The classified objects are analyzed and prioritized depending on their potential level of danger. A detected object is marked as urgent (*U*) if it is situated within the trapeziums of interest (user proximity region), otherwise the obstacles is categorized as normal (*N*) or non-urgent. By employing two areas of proximity we can prevent the system to launch acoustic warning messages for all the detected objects existent in the scene. We adopted this strategy in order to overwhelm the user with too much information.

In order to keep the acoustic feedback intuitive only the following set off alarms will be generated by the system, in descending order of priority: "vehicle", "bicycle", "human" and "obstruction".

Finally, in order to infer to the VI people information regarding the relative position of the detected object the acoustic warning messages are encoded in stereo using the right, the left and both channels.

In addition, not to confuse the VI user with too much information, the warning messages are sent with a frequency rate inferior to two seconds, regardless of the scene dynamics. The sound patterns are transmitted to the VI person through bone conduction headphones.

4. Experimental evaluation

Datasets and baseline systems: The proposed tracker is evaluated using the VOT2016 dataset. The video dataset contains 60 high challenging image sequences with the following visual attributes: illumination change, motion and size change, occlusion and various camera motions. All the sequences were annotated by human observers. We compared the proposed method with four state of the art algorithm denoted: GOTURN [18], C-COT [24], Stapler [25] and TCNN [26].

The entire system, integrating all modules (*i.e.*, obstacle detection, tracking and object classification) was evaluated on the same video dataset as in [10]. The database includes 20 video sequences recoded with the help of VI users. The videos are acquired at a resolution of 320 x 240 pixels, are trembled and cluttered.

Implementation details: For each tracker, the default parameters and the source codes provided by the authors are used in all the evaluations. All the experiments were performed on a portable processing unit (regular ultrabook computer, with Visual Studio C++/Matlab on an Intel Core Kaby Lake i7-7700HQ, 32GB RAM and NVIDIA GeForce GTX 1050 GPU). Because the processing unit needs to be carried by the VI user in the backpack we decided to use an ultrabook due to its lightweight (inferior to 1 kg).

Evaluation measures: In the state of the art, various techniques use as evaluation metric the center prediction error. However, as indicated in [27] this measure is highly sensitive and dependent on the annotation. In addition, the measure completely ignores the interest object size and does not take into account the apparent tracking failure. We evaluated the proposed tracker using the quantitative measures, as described below.

As indicated in [28], the quantitative evaluation of the proposed approach is determined using the region overlap measure (Φ). This measure is computed as the overlap between the predicted target region (R_t^T) and the ground truth annotation data (R_t^G).

$$\Phi = \{\phi_t\}_{t=1}^{N}, \quad \phi_t = \frac{R_t^G \cap R_t^T}{R_t^G \cup R_t^T}; \tag{2}$$

where R_t denotes the region of the object at time *t,* while *N* is the total number of frames of the considered video sequence. The region overlap measure takes into account, in the same time, both the position/size of the predicted bonding box and the ground-truth data. Compared with the center-based errors, the region overlap measure Φ does not return arbitrary large errors for tracking failures. In terms of pixels classification the overlap can be interpreted as:

$$\phi_t = \frac{R_t^G \cap R_t^T}{R_t^G \cup R_t^T} = \frac{TP}{TP + FN + FP}; \tag{3}$$

where TP are true positives pixels, FN the false negatives pixels and FP the false positives pixels.

We have compared the proposed algorithm in terms of accuracy with state of the art methods as GOTURN [18], Stapler [25] and TCNN [26] and the winner of the 2016 benchmark evaluation C-COT [24].

We have performed an extensive evaluation on all the 60 sequences from VOT 2016 challenge. Our approach returns an average overlap score of 0.551 while GOTURN, Staple, TCNN and C-COT achieve 0.394, 0.49, 0.52 and 0.51 respectively (Table 1).

Table 1. Experimental evaluation of the proposed tracking system and comparison with state of the art methods using the overlap score as evaluation metric.

Sequence	GOTURN	C-COT	Staple	TCNN	OUR
Birds1	0.10	0.32	0.08	0.43	**0.48**
Bmx	**0.59**	0.13	0.30	0.17	**0.59**
Bolt1	0.43	0.46	0.51	0.51	**0.61**
Butterfly	**0.65**	0.45	0.33	0.53	0.64
Fernando	**0.45**	0.26	0.36	0.38	**0.45**
Godfather	0.05	0.32	0.51	0.31	**0.57**
Iceskater1	0.37	0.45	0.17	0.58	**0.60**
Pedestrian1	0.56	**0.67**	**0.67**	0.60	0.65
Racing	0.76	0.45	0.54	0.40	**0.77**
Shaking	0.75	0.63	0.05	0.67	**0.77**
Soldier	**0.71**	0.24	0.32	0.66	**0.71**
TOTAL	**0.39**	**0.51**	**0.49**	**0.52**	**0.55**

From the experimental results presented in Table 1 we observe that the proposed method shows a significant increase in accuracy and becomes the best-performer on a diverse set of examples. If we consider for example the *Bird1* and *Godfather* video sequences our system outperforms the traditional GOTURN method. In addition, when compared with CCOT or TCNN on *Bolt1* or *Racing* our method does not allow the system to drift due to its selective update of the appearance model.

Finally, we evaluated the entire framework in terms of precision (P), recall (R) and F1 score on the dataset of [9] that contains 20 videos with an average duration of 10 minutes recorded using VI users. The experimental results are presented in Table 2. The proposed method returns an average detection rate superior to 89% for all types of obstacles. Compared with the state of the art method [9] our system shows an improvement of more than 5%.

Table 2. Experimental evaluation the proposed framework on a set of video acquired with the help of VI users.

	Ground truth	Recall	Precision	F1 score
Vehicles	431	0.92	0.88	0.90
Pedestrians	374	0.94	0.91	0.92
Bicycles	120	0.88	0.84	0.86
Obstructions	478	0.89	0.87	0.88
TOTAL	**1403**	**0.91**	**0.87**	**0.89**

In terms of computational speed, when implementing the proposed framework on a regular ultrabook computer, running on an Nvidia GTX 1050 GPU, the average processing speed is around 15 fps.

In Fig. 6 we present some experimental results obtained by our method on the video dataset from [10] acquired in real-life scenarios with the help of actual VI. The category of each detected obstacle is also presented.

5. Conclusions and perspectives

In this paper, we have proposed a novel perception system based on computer vision methods and deep convolutional neuronal networks able to assist the visual impaired user during the outdoor navigation. In contrast to various techniques existent in the state of the art our system is able to detect, track and recognize, in real-time, all relevant object existent in the scene without any *a priori* knowledge about shape, position or dynamics. The output of our system is transformed into a set of acoustic warnings transmitted to the VI user through bone conducting headphones.

The experimental evaluation performed on a set of 60 challenging video sequences selected from the VOT 2016 and 20 video sequences recoded with the help of VI users demonstrates the effectiveness and efficiency of our method.

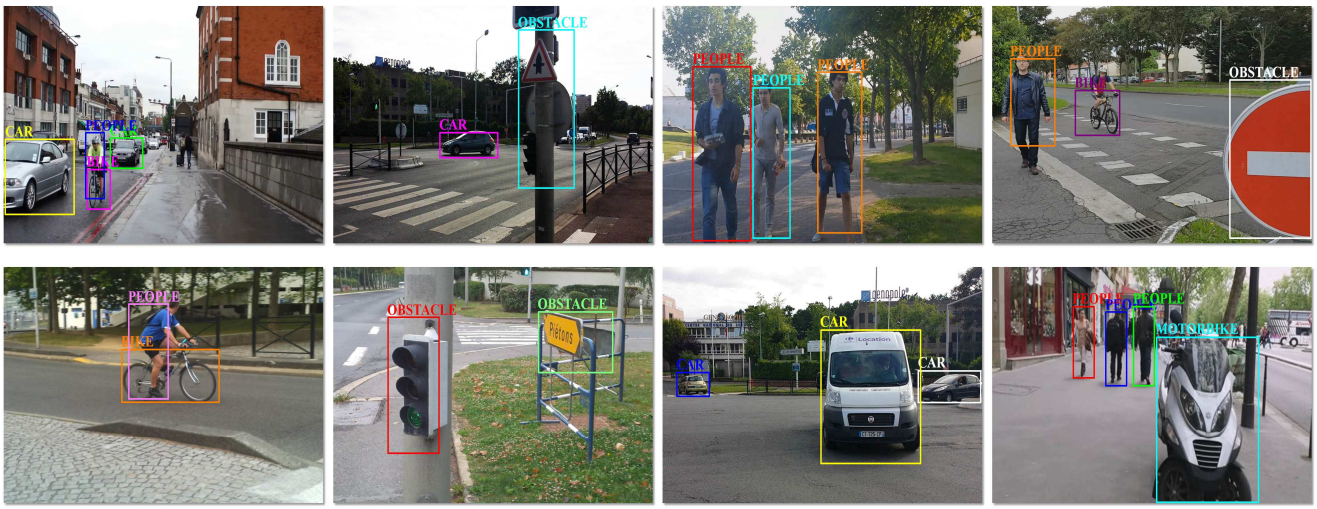

Figure 6: Experimental detection and recognition results on the video dataset acquired by VI users.

In addition, when compared with different state of the art algorithms, our approach shows an increase in performance (in terms of accuracy) of more than 5%.

For future work we propose to integrate our method in a larger framework that includes: face recognition, guided navigation and shopping assistance functions. In addition, a study with real VI users is envisaged.

Acknowledgement

This work was supported by a grant of the Romanian National Authority for Scientific Research and Innovation, CNCS - UEFISCDI, project number: PN-II-RU-TE-2014-4-0202.

This work has been funded by University Politehnica of Bucharest, through the "Excellence Research Grants" Program, UPB – GEX. Identifier: UPB–EXCELENȚĂ–2016, No. 97/26.09.2016 and UPB–EXCELENȚĂ–2017, project entitled: "Autonomous obstacle detection and recognitions system based on deep convolutional neural networks dedicated to visually impaired people".

References

[1] World Health Organization (WHO) - Visual impairment and blindness. Available online: http://www.who.int/mediacentre/factsheets/fs282/en/ (accessed on 27.06.2017).

[2] A. Rodríguez, J.J. Yebes, P.F. Alcantarilla, L. M Bergasa; J. Almazán, Assisting the Visually Impaired: Obstacle Detection and Warning System by Acoustic Feedback. Sensors 2012, 12, 17476-17496.

[3] R. Tapu, B. Mocanu and E. Tapu, "A survey on wearable devices used to assist the visual impaired user navigation in outdoor environments," 2014 11th International Symposium on Electronics and Telecommunications (ISETC), Timisoara, 2014, pp. 1-4.

[4] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In CVPR, 2015.

[5] M. Kristan, R. Pflugfelder et al. The visual object tracking VOT2016 challenge results. In ECCV Workshop, pp. 1–45, 2016.

[6] Leo, M.; Medioni, G.G.; Trivedi, M.M.; Kanade, T.; Farinella, G.M. Computer Vision for Assistive Technologies. CVIU. **2017**, 154, 1–15

[7] S. Cloix, V. Weiss, G. Bologna, T. Pun and D. Hasler, "Obstacle and planar object detection using sparse 3D information for a smart walker," International Conference on Computer Vision Theory and Applications (VISAPP), 2014, pp. 292-298.

[8] R. Manduchi, Mobile vision as assistive technology for the blind: An experimental study. In. ICCHP' 2012.

[9] R. Tapu, B. Mocanu, A. Bursuc and T. Zaharia, "A Smartphone-Based Obstacle Detection and Classification System for Assisting Visually Impaired People," In ICCV-Workshops, 2013, pp. 444-451.

[10] B. Mocanu, R. Tapu, T. Zaharia, When Ultrasonic Sensors and Computer Vision Join Forces for Efficient Obstacle Detection and Recognition. Sensors 2016, 16, 1807.

[11] J. Manuel Saez, F. Escolano and A. Penalver, "First Steps towards Stereo-based 6DOF SLAM for the Visually Impaired," In CVPR - Workshops, 2005, pp. 23-23.

[12] V. Pradeep, G. Medioni and J. Weiland, "Robot vision for the visually impaired," In CVPR - Workshops, 2010, pp. 15-22.

[13] J. M. Sáez, F. Escolano and M. A. Lozano, "Aerial Obstacle Detection With 3-D Mobile Devices," in IEEE JBHI, 19, no. 1, pp. 74-80, 2015.

[14] A. Khan, F. Moideen, J. Lopez, W. Khoo, and Z. Zhu, "KinDectect: Kinect Detecting Objects," in Computers Helping People with Special Needs, vol. 7383, 2012, pp. 588-595.

[15] H. Takizawa, S. Yamaguchi, M. Aoyagi, N. Ezaki and S. Mizuno, "Kinect cane: An assistive system for the visually impaired based on three-dimensional object recognition," IEEE ISSI, 2012, pp. 740-745.

[16] M. Brock, P. O. Kristensson, "Supporting Blind Navigation using Depth Sensing and Sonification", ACM Conference on Pervasive and Ubiquitous Computing, pp. 255-258, 2013.

[17] P, Panteleris Vision-based slam and moving objects tracking for the perceptual support of a smart walker platform. In ECCV 2014 Workshops. 2014. pp. 407-423.

[18] D. Held, S. Thrun, and S. Savarese. Learning to track at 100 fps with deep regression networks. In ECCV, 2016.

[19] S. Zagoruyko and N. Komodakis, "Learning to compare image patches via convolutional neural networks," 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 4353-4361.

[20] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. CoRR, abs/1409.1556, 2014.

[21] B. Babenko, M. H. Yang and S. Belongie, "Visual tracking with online Multiple Instance Learning," IEEE Conference on CVPR, pp. 983-990, 2009.

[22] Y. Hua, K. Alahari and C. Schmid. Occlusion and motion reasoning for long-term tracking. In Proceedings of the European Conference on Computer Vision, 2014.

[23] Y. Nie and K.K. Ma, "Adaptive rood pattern search for fast block-matching motion estimation," in *IEEE Transactions on Image Processing*, vol. 11, pp. 1442-1449, 2002.

[24] M. Danelljan, A. Robinson, F. S. Khan and M. Felsberg. Beyond correlation filters: Learning continuous convolution operators for visual tracking. In ECCV 2016.

[25] L. Bertinetto, J. Valmadre, S. Golodetz, O. Miksik, P.H.S. Torr, Staple: Complementary learners for real-time tracking. In: CVPR 2016.

[26] H. Nam, M. Baek B. Han. Modeling and propagating cnns in a tree structure for visual tracking. arXiv:1608.07242, 2016.

[27] B. Babenko, M. H. Yang and S. Belongie, "Robust Object Tracking with Online Multiple Instance Learning," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 33, no. 8, pp. 1619-1632, 2011.

[28] L.Cehovin, A. Leonardis, and M. Kristan, "Visual object tracking performance measures revisited", arXiv preprint arXiv:1502.05803, 2015.