A computer vision system supporting blind people - The supermarket case

Kostas Georgiadis[1,2], Fotis Kalaganis[1,2], Panagiotis Migkotzidis[1], Elisavet Chatzilari[1], Spiros Nikolopoulos[1], and Yiannis Kompatsiaris[1]

[1] Information Technologies Institute, Centre for Research and Technology Hellas, Thermi 57001, Greece
{kostas.georgiadis,fkalaganis,migkotzidis,ehatzi,nikolopo,ikom}@iti.gr
[2] Artificial Intelligence and Information Analysis Lab, Dpt. of Informatics, Aristotle University of Thessaloniki, Greece

Abstract. The proposed application builds on the latest advancements of computer vision with the aim to improve the autonomy of people with visual impairment at both practical and emotional level. More specifically, it is an assistive system that relies on visual information to recognise the objects and faces surrounding the user. The system is supported by a set of sensors for capturing the visual information and for transmitting the auditory messages to the users. In this paper, we present a computer vision application, e-vision, in the context of visiting the supermarket for buying groceries.

Keywords: computer vision system · assistive device · visually impaired.

1 Introduction

Although computer vision-based assistive devices for vision-impaired people have made great progress, they are confined to recognising obstacles and generic objects without taking into account the context of the activities performed by the person. This context differentiates significantly the functional requirements of an assistive device. Indeed, in the context of visiting a supermarket, a vision-impaired person would need different categories to be recognised compared to going for a walk on the beach, e.g. products and trails vs people, signs and obstacles respectively. However, existing solutions do not take into account this context and their design and structure is based on generic categories.

For example, in seeingAI[3], a mobile application that recognises what the camera sees and narrates it through the speakers, the user must select between generic categories for recognition, such as reading text, recognising barcodes, detecting people and describing colours. In a similar vein, applications also utilising the mobile phone sensors (i.e. camera and speakers) Envision[4] and Eye-D[5]

[3] https://www.microsoft.com/en-us/ai/seeing-ai

[4] https://www.letsenvision.com/

[5] https://eye-d.in/

again provide options between these generic categories, with the latter one also enabling GPS-based functionalities (e.g. where am I, what is around me).

Besides the mobile-based applications assisting the visually impaired, another approach is the systems based on glasses. Prominent examples of this category are OrCam MyEye 2[6], eSight[7], NuEyes[8] and Eyesynth[9]. OrCam MyEye 2 is a small device with a camera that is attached to the users' glasses and is able to recognise up to 100 custom objects inserted by the user (e.g. products, people), read text and recognise barcodes. On the other hand, eSight and NuEyes are glasses that work as digital magnifiers, enhancing the vision of the partially visually-impaired user. Finally, Eyesynth is a pair of glasses accompanied by a portable microcomputer communicates the surroundings of the user through the bones of the head and can be used mainly for avoiding obstacles.

On the other hand, the proposed system (e-vision) is a hybrid approach, combining the movement freedom of an external camera with the processing power and the penetration rate of mobile phones. Besides the system design, its main novelty lies in the context-aware design of the application, i.e. the structure of the application is built based on a specific context. In more detail, we showcase the supermarket context, an activity of daily living that is easy for the seeing people and currently impossible for the visually-impaired. In this context, if we rely on existing solutions, we have two options; first, one could use the text reading option, which would read out loud all the text detected in all of the products that are seen by the camera, providing an annoying experience to the user. Second, one could use the barcode recognition option, which however would mean that the user has to pick each product in the supermarket, find the angle where the barcode is visible to the camera and repeat until they find the product of interest.

In the presented system, towards creating a user friendly experience, we consider the abstraction levels of information needed to be communicated when we visit a supermarket and that is if we are looking at a trail, shelf, product or we are at the entrance/exit of the supermarket. In each abstraction level the user would need different levels of information communicated to them (i.e. in the trail level the system should be able to say you are at the drinks trail, in the shelf level that you are looking at the beers and at the product level that you are holding a specific brand of a beer). As a result, the communication of the system with the user is hassle-free providing a pleasant context-aware experience. Eventually, for the detection between the abstraction levels we rely on computer vision through deep learning, while OCR is used for detecting the text on each image and a supermarket database in combination with the detected abstraction levels is used to refine what will be communicated to the user.

[6] https://www.orcam.com/en/

[7] https://www.esighteyewear.com/int

[8] https://nueyes.com/

[9] https://eyesynth.com/?lang=en

2 Related Work

Lately, towards the new self-service supermarkets (e.g. Amazon Go), there have been a few efforts on recognising products through object detection. Early efforts relied on the combination of SIFT-alike descriptors, such as [2], where dense SIFT is combined with LLC to retrieve the best candidate classes and then a GA-based algorithm is used to create the final list of products seen in an image. In [1], SIFT-alike descriptors are combined with deep learning to generate attention maps from a combination of SIFT, BRISK and SURF features in order to achieve one-shot deep learning of products, utilising a single image per product. One-shot recognition of products is also proposed by Karlinsky et al. [5], where a probabilistic model is employed to generate bounding boxes and then deep fine-grain refinement based on the VGG-f network is applied to the coarse results. In a similar vein, in [8], a ROI detector based on Yolo_v2 for detecting the product-agnostic bounding boxes is combined with global feature matching between the database of products and the features from a fine-tuned CNN (VGG-16) of the bounding box. Finally, more closely related to e-vision is the work in [3], where an application for assistive grocery shopping is presented. More specifically, an image-based product classification scheme is proposed combining HOG features with discriminative patches and SVMs that can classify a test image between 26 coarse classes (e.g. coffee, soft drinks). On the contrary, e-vision, combining OCR and deep learning features, is a fully functional system that can provide the information that the user is looking for, including the full scale of grocery products (i.e. more than 100k products), a scalability that is offered due to the fully unsupervised nature of the proposed system.

3 The concept of the proposed supermarket application

Our objective is develop an application that can help the visually impaired in their daily living. In this direction, the proposed system comprises of 3 sensors/devices; i) a camera that can be attached to the person's head/glasses or other body part, ii) a mobile device with an accompanying application that gets the images from the camera and through the proposed methodology (Section 5 extracts meaningful information for what the camera is seeing and iii) a set of earphones that connect to the application and provide auditory feedback to the user with respect to the previously extracted information using text-to-speech technologies.

In this work we focus on one activity that is very common and frequent in our lives, a visit in a supermarket and a session of buying a set of products. In our case, we consider the Greek case, i.e. a Greek supermarket chain (Masoutis[10]), with a variety products, a combination of Greek and English labels and a Greek text-to-speech solution. If we decompose a visit to the supermarket, we can distinguish between 4 levels of abstraction. More specifically, these levels from specific to abstract are described below:

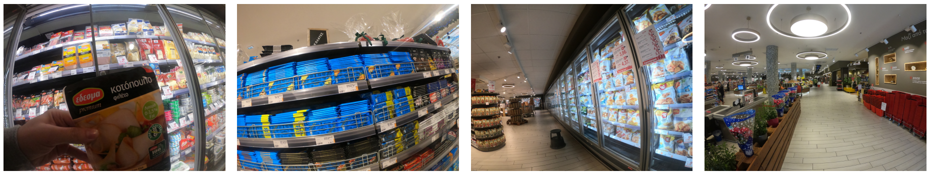
[10] https://www.masoutis.gr/

Product: The user has a specific product in their hands and they want the system to tell them what is the product (e.g. in the case of Fig. 1a, the system should say that it is *Edesma charcuterie from chicken fillet*).

Shelf: The user is in front of a shelf with a limited number of relevant products and they want the system to tell them what is the fine-grained category of the products (e.g. in the case of Fig. 1b, the system should say that you are in front of a *Shelf with chocolates*).

Trail: The user is in front of a trail with a large number of relevant products and they want the system to tell them what is the coarse-grained category of the products (e.g. in the case of Fig. 1c, the system should say that you are at the *Trail with frozen products*).

Other: The user is in the entrance/exit of the supermarket (Fig. 1d) and in this case the system informs them that they are not in front of a shelf or trail.



(a) Product (b) Shelf (c) Trail (d) Other

Fig. 1: Example images for each abstraction level

4 Onsite visits dataset and annotation

In order to evaluate our system, we performed onsite visits to two supermarkets with two individuals participating, one with partial and one with complete visual impairment, who were set up with a GoPro camera and were asked to go through a grocery shopping session. Two different stores were selected for the onsite visits, one hypermarket and one small local store so as to have images with varying lightning conditions and also diverse surroundings. The scenarios were identical in both cases and required each participant to complete a full visit in the supermarket, purchasing five distinct pre-defined products. Once the participant acquired all five products, he was instructed to proceed to the cashier where the scenario was completed. Throughout the entirety of the visit the participant had a GoPro camera attached to his chest and was instructed to use the voice command ”GoPro Take Photo” in cases where the user wanted the application to provide information to them. This resulted in capturing 39 photos in total, 16 and 23 from the first and second respectively, used for evaluating e-vision.

The acquired images were annotated in two stages: 1) abstraction level 2) category level. In the first case, images were visually inspected to provide a class label (i.e regarding the abstraction level), resulting in 14 product images, 10 shelf images, 8 trail images and 7 images labelled as other. In the second case, we utilised a full product database from the Greek supermarket chain Masoutis[11]. More specifically, the database portrays a detailed categorisation for all

[11] https://www.masoutis.gr/

the available supermarket products, including product description and product category in three generalisation levels (i.e. category level I-III), starting from the more general category (e.g. frozen product) and ending to the most specific (e.g. frozen cheese pie). Moreover, by inspecting the database, we associate the generalisation levels with the abstraction levels defined earlier in the following way; the generalisation levels I and II correspond to trail and shelf abstraction levels respectively, while the product level images were annotated by the product description level of the database.

5 Methodology

The first step towards the implementation of the supermarket application is to distinguish between the various levels of abstraction in order to identify what type of information the user is expecting as feedback. In this direction, we rely on computer vision and based on the visual features of the images we detect whether the user needs product/shelf/trail/other level of information. Next, in the cases where the images contain products, i.e. in the product/shelf/trail abstraction level, detecting text on the grocery packages and cross-referencing it with the supermarket product catalogue can provide a clear indication of what the user is looking at. Finally, in the other abstraction level, where there are no products in sight, the system informs the user that they are not in front of a trail, shelf or product.

5.1 Distinguishing between Product, Shelf, Trail and Other

The concept of differentiating among products, shelves, trails and other is of paramount importance in the context of navigation in a supermarket since it guides the level of information abstraction that needs to be communicated to the user. In order to achieve this crucial goal for the proposed system we took advantage of the increased capabilities that Deep Neural Networks (DNNs) combined with Support Vector Machines (SVMs) offer.

Three DNN architectures were compared for the purposes of feature extraction from images, namely VGG16 [6], ResNet [4] and Inception [7] pre-trained on the ImageNet dataset. From each architecture two types of features were extracted and fed to a linear-kernel SVM setting for classifying the images into one of the four categories (product, shelve, trail and other). The first type of features was formed as a concatenation of activations from the intermediate, convolutional, layers. Formally, let $M_l \in \mathbb{R}^{W \times H \times C_l}$ be the feature map of l^{th} layer (after ReLu), where $W \times H$ denotes the spatial dimensions (width and height respectively) and C_l the channels of the l^{th} layer. Hence, the first feature vector was formed as the $\mathbf{v} = max\{M(i,j,:), i \in \{1.\ldots,W\}, j \in \{1.\ldots,H\}\}$ and will be referred to as *intermediate convolutional activations*. In an equivalent aspect, we extracted the features that correspond to the output of the fully connected layers and will be referred to as *FC activations*. Let us denote by $\mathbf{W}$ the weight matrix of the penultimate layer of the employed artificial neural network architecture. Then, the second type of features corresponds to $u = ReLu(\curvearrowleft)$, where u indicates the output of the layer previous to the penultimate. Then, in order to classify the corresponding images into one of the four categories we employed linear SVMs. The cost parameter C was discovered using cross validation, while we followed the common practice of all-vs-all scheme to achieve multi-class classification.

5.2 Detecting text in food packages

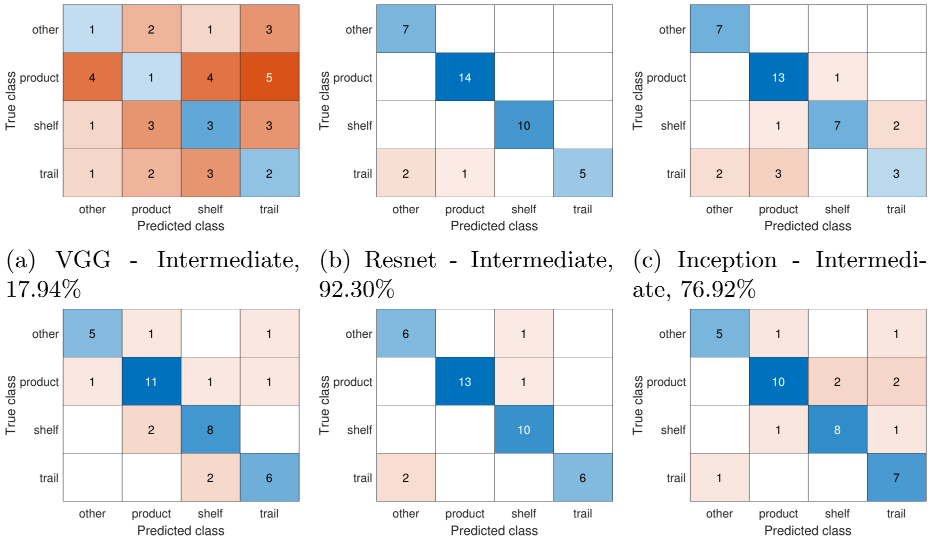
Once the image is categorised based on the abstraction level, an optical character recognition (OCR) mechanism, identifying characters in both English and Greek, is enabled to extract the available text from the product packages (e.g. brand, product description). Depending on abstraction level, two different algorithmic approaches are followed. In the case, the user wants to identify a specific product (i.e. abstraction level: Product) the bounding box with the largest dimensions and the text it encapsulates is only examined, as it will be the one closest to the camera. A search for the identified word(s) in the Masoutis database determines the product description. In the case the user wants to be informed regarding the trail or shelf he is currently standing (i.e. abstraction level: Trail, Shelf) the entirety of the text detected in the whole image by the OCR algorithm is utilised, resulting in an array tabulating each recognised word. A shrinkage of the array is performed by removing the stopwords, any invalid words that cannot be traced in Masoutis database (e.g. frazen) and a number of words that emerge in the database in high frequency but with no discriminative power (e.g. gift, offer), formulating an array that includes only valid words. A search for each valid word is then performed in the product description entries aiming to identify all registrations that encapsulate the selected word and by performing a process equivalent to data binning, associate them with the generalisation level I and II descriptions (depending on the abstraction level). The corresponding unique generalisation level descriptions are encountered as votes and are fed to a majority voting (MV) protocol to determine which shelf or trail is illustrated in the selected image. MV is a decision making protocol that provides a decision (a label in our scenario) when it receives more than half of the votes. More specifically, if the class label corresponds to “Trail”, the aforementioned MV protocol will be applied only in the generalisation level I descriptions, while if the image is identified as “Shelf” MV will be performed only in the generalisation level II descriptions.

6 Experiments

In this section, we evalute the proposed system with respect to the accuracy of the detected content utilizing the dataset from the onsite visits (Section 4). In this direction, we present two lines of experiments; the first one aims to demonstrate the methodologies in Section 5 independently and the second one aims at evaluating the quality of the proposed supermarket application as a whole.

6.1 Distinguishing between abstraction levels

Here, we commence by quantifying the benefits of each feature extraction approach of the 6 presented in Section 5.1 regarding their classification capabilities. For classification purposes we employed a suitable machine learning procedure and measure its performance in the quaternary classification task (products, shelves, trails, other). The classifier was a standard linear kernel SVM that operated on the extracted image features. In all cases, classification performance was evaluated through the accuracy metric by means of 10-fold cross-validation. In order to produce comparable results among different feature extraction schemes, the same cross validation partition was used in every case.



(a) VGG - Intermediate, 17.94%

(b) Resnet - Intermediate, 92.30%

(c) Inception - Intermediate, 76.92%

(d) VGG - FC, 76.92%

(e) Resnet - FC, 89.74%

(f) Inception - FC, 76.92%

Fig. 2: Confusion matrices for the quaternary classification task using all the employed feature extraction schemes (Intermediate activations - top, Fully Connected activations - bottom). The percentages correspond to average accuracy across 10-fold CV.

Figure 2 illustrates the classification performances, by means of confusion matrices, for every feature extraction procedure, while the accuracy of each feature space is denoted in the caption. It is obvious that features obtained using the Resnet architecture manage to achieve significantly better classification performance in all of the four available classes. Then, among the two available Resnet-based feature types, the features obtained from the intermediate layers provide a more robust classification performance compared to the fully connected layer features (92.3% vs 89.74%). It is worth noting that the only class that does not have 100% classification accuracy is the one of Trail, which can be attributed to the variability of the trails visual appearance from the two supermarket store of the onsite visits.

6.2 Classifying trail, shelf and product images

Based on the abstraction level annotation (i.e. trail, shelf, product), a category level classification is given to each image, with accuracy serving as the evaluation metric. Examining the total accuracy for each abstraction level separately, all levels exceed 80%. The images corresponding to "Shelf" demonstrate the highest accuracy score, as only three images were miss-classified ($Acc_{Shelf} = 89.00\%$). Similar classification scores are obtained in the case of "Trail" images ($Acc_{Isle} = 87.50\%$). The lowest accuracy score is observed, in the "Product" images, with the accuracy being at $Acc_{Product} = 83.30\%$. This is expected due to the various similar products being available (e.g. beer can of 330 ml versus beer can of 500ml), which cannot be distinguished in all product angles. Finally, the system's overall accuracy is 87.0%.

6.3 Evaluating the system as a whole

The evaluation of the system as a whole, blends the predictions provided in abstraction and category level, resulting in the final version that will be employed for the application's needs. This way a two-stage classification scheme is realised with the first deciding upon the abstraction level and the second based on the initial prediction specifying the category level. The system's accuracy is formulated as the joint probability of correct category and correct abstraction level (i.e. $P(Category, Abstraction)$), while the one presented in the previous section was the conditional probability that the correct category is found given that the image was correctly classified in terms of abstraction level (i.e. $P(Category|Abstraction)$). The system produces high classification scores for both the "Product" and "Shelf" images, with the accuracy being 83.3% and 89.0% respectively. This is attributed to the infallible predictions of the first stage classifier combined with the high rates of the second (refer to $Acc_{Product}$ and Acc_{Shelf}). On the contrary, the accuracy levels for the "Trail" images is significantly lower (i.e. 54.7%) mainly due to the poor performance produced by the first stage classifier that reaches 62.5%. Overall, the system will produce a correct label with a probability of 77.15%, that is considered acceptable considering the task's complexity.

7 The implementation of the proposed supermarket application

Our final objective is the integration of the aforementioned components in a lightweight and compact mobile application. In this direction, the main tools utilised were the **Unity** game engine (Unity[12]) for the application's graphics, **Tensorflow** for extracting features from deep learning models in order to classify between trail/shelf/product/other (implementing the methodology in Section 5.1) and the **Google cloud vision** API for the OCR in order to find the

[12] https://unity.com/

descriptions of trails/shelves/products (implementing the methodology in Section 5.2).

Initially, we want to get the image from the camera of the user's device. The first step was to capture the screen of the device so that the user observations could be annotated. To achieve this, the *WebCamTexture* class offered by the Unity game engine was used, capturing the pixels of the device's selected camera and transforming them into a 2D texture. Having captured the image from the camera, the next step is to extract visual features from the intermediate layer of a pretrained ResNet model. For this, we rely on the open source Tensorflow framework, with which pretrained deep learning models can be used. For the integration of the Tensorflow API within the Unity engine we utilised the TensorFlowSharp library. Finally, the images that have been classified as Trail/Shelf/Product are given to the OCR system so as to categorise them based on which Trail/Shelf/Product they are depicting. In this direction, the captured image (texture) is given as input to the the Google Cloud Vision API enabling the Optical Character Recognition (OCR). However, because both the interface and back-end architecture where structured and developed through the Unity engine, the communication of those two platforms had to be established. Therefore, a plugin (Unity Google Cloud Vision) was employed that bridged this gap, by using a specified API key that instantly connects to the Cloud Vision API requiring an input and receiving multiple outputs. In our case, the input is the retrieved image from the camera of the device and the outputs are the responses from the Cloud Vision API plugin, which includes labels, products, text, safe search, etc. Afterwards, using the Masoutis database of products, the methodology of Section 5.2 is applied to the responses of the OCR system (i.e. text and bounding boxes). Last, after the full classification of an image is performed, the system announces the results as voice information using the **Google text-to-speech** API.

8 Conclusions

In this paper, we presented a novel context-aware application for assisting the visually-impaired in their daily living, and more specifically we showcase the activity of grocery shopping. The presented application has an overall accuracy of 77.15% in voicing exactly what information the user needs. In the future, our plan is to increase this accuracy by increasing the size of the training through either more onsite visits from more stores or by extracting more images from the accompanying video that was shot during the onsite visits. Next, we plan to increase the functionalities of the application in the following ways: first, by enhancing the system narrative in the case "Other" was detected through localising useful objects (i.e. shopping carts, employees and cashiers). As this cannot be facilitated with text detection, we will rely on visual object detection and localisation. Preliminary results with existing object localisation deep networks (e.g. RetinaNet) show promising results, but in order to include the classes of interest (e.g. shopping carts), transfer learning will be employed. As a result, for example, having the location of the shopping carts will enable the application to give directions to the user (e.g. he shopping carts are in your right). Second, detecting non-packaged products such as fruits and vegetables through fine-grained image classification is within our future plans. Finally, we also plan a cashier sub-context, where the application constantly informs the user on what money they are holding or recongises the products as the person puts them on the tray.

Acknowledgements

This work is part of project MAMEM that has received funding from the European Unions Horizon 2020 research and innovation programme under grant agreement No 644780 and project Evision that has been cofinanced by the European Regional Development Fund of the European Union and Greek national funds through the Operational Program Competitiveness, Entrepreneurship and Innovation, under the call RESEARCH CREATE INNOVATE (project code: T1EDK-02454).

References

1. Geng, W., Han, F., Lin, J., Zhu, L., Bai, J., Wang, S., He, L., Xiao, Q., Lai, Z.: Fine-grained grocery product recognition by one-shot learning. In: Proceedings of the 26th ACM International Conference on Multimedia. pp. 1706–1714. MM '18, ACM, New York, NY, USA (2018). https://doi.org/10.1145/3240508.3240522, http://doi.acm.org/10.1145/3240508.3240522
2. George, M., Floerkemeier, C.: Recognizing products: A per-exemplar multi-label image classification approach. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) Computer Vision – ECCV 2014. pp. 440–455. Springer International Publishing, Cham (2014)
3. George, M., Mircic, D., Soros, G., Floerkemeier, C., Mattern, F.: Fine-grained product class recognition for assisted shopping. In: Proceedings of the 2015 IEEE International Conference on Computer Vision Workshop (ICCVW). pp. 546–554. ICCVW '15, IEEE Computer Society, Washington, DC, USA (2015). https://doi.org/10.1109/ICCVW.2015.77
4. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
5. Leonid Karlinsky, Joseph Shtok, Y.T.A.T.: Fine-grained recognition of thousands of object categories with single-example training. CVPR (2017)
6. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
7. Szegedy, C., Ioffe, S., Vanhoucke, V., Alemi, A.A.: Inception-v4, inception-resnet and the impact of residual connections on learning. In: Thirty-First AAAI Conference on Artificial Intelligence (2017)
8. Tonioni, A., Serro, E., Di Stefano, L.: A deep learning pipeline for product recognition on store shelves. arXiv preprint arXiv:1810.01733 (2018)