Multiple-Layer Classifier with Label Correction for Semantic Segmentation

Lavinia Ferariu[1] and Simona Caraiman[2]
Faculty of Automatic Control and Computer Engineering
Gheorghe Asachi Technical University of Iasi
Iaşi, Romania
Emails: [1]lferaru@ac.tuiasi.ro, [2]sarustei@cs.tuiasi.ro

Abstract — **Semantic segmentation (SS) provides the meaning of visual scenes, thus being a key stage for navigation and environment's perception. This paper presents a solution for SS compatible with assistive wearable systems equipped with color and depth cameras. In order to ensure a compact and robust description of input color-based images, both 2D and 3D features are extracted at superpixel level, after correcting the displacements of the camera by means of adequate rectifications. Random Forests (RF) are called for solving the classification problem. In this context, this paper introduces a multilayer RF-based classifier, including a separate layer for label correction. Additional two other correcting methods are proposed for the first layers of the classifier, i.e. a fast method investigating the majority label around each object, and several customizations of the graph cut algorithm using convenient cue weights. The performance of the suggested approach is experimentally verified on diverse urban street scenes.**

Keywords — semantic segmentation; assistive wearable system; classification; 3D features; Markov Random Fields.

I. INTRODUCTION

Semantic **s**egmentation (SS) maps a top-level understanding of images to pixel level. This interpretation is a valuable asset for refining the decisions in a large variety of applications, from medical imagistic to autonomous driving. This paper discusses the SS integrated in **S**ound **o**f **V**ision System (SoV) - a wearable assistive equipment providing acoustic and tactile description of the surrounding environment [1]. SoV is meant to help the navigation of visually impaired persons with levels 3, 4 or 5 of visual deficiencies (defined by the World Health Org.), who are assisted by a white cane, a guiding dog or a guiding person.

In this context, SS is used in real time, for understanding the layout of environmental scenes, in order to map the most important visual elements into the haptic and audio representations, and send necessary alerts. Working with street scenes poses some difficulties, because the background is non-uniform and every class is instantiated by samples with a large range of colors, shapes and postures, the appearance being significantly affected by 3D rotations, weather conditions, etc. Additionally, in the case of SoV, rectification becomes very challenging, due to expected huge camera displacements caused by the movements of the head/body. Also, this application imposes critical safety requirements (i.e. any potential misclassifications should not put the visually impaired person in danger), as well as critical real-time constraints (derived from the temporal characteristics of the acquisition system, from the dynamics of the impaired person and surrounding objects, and from the computation time requested by all the processes running on SoV).

For SS, three broad current research directions have been depicted in [2]. They refer to the type of feature extractor used for solving the corresponding classification problem, and, in relation to this step, to the method used for improving the consistency of the labels. Thus, the methods can work with hand-engineered features, learned features (e.g. extracted by convolutional neural networks) or features provided via weakly supervised learning (like multiple instant learning). The accommodation of (partially) pre-trained models becomes less effective for SS in SoV, especially because the displacements of the mobile camera lead to many invalid (less confident) pixels in every frame. Therefore, SS is solved with hand-engineered features extracted at superpixel level, which could be effective for the available limited image database.

Commonly, these types of methods solve the classification by fusing 2D and 3D features, and then apply correction algorithms based on **M**arkov/Conditional **R**andom **F**ields (MRF) for improving the consistency of labeling [2]. **R**andom **F**orest (RF) is perhaps the most adopted classifier. In [3], RF works with features based on color and discrete cosine transform texture. A fusion between 2D and 3D features (like relative height from the ground, surface normal vector, nearest distance to the camera and re-projection error) is presented in [4]. Some features describe the similitude between pairs of neighbor superpixels. [5] uses a 3D feature related to the estimated normal direction and some 2D features obtained from superpixel geometry and color distribution; if the superpixel is not validated by at least five 3D points, the 3D feature is set to a predefined constant, for robustness reasons. Other 3D features are suggested in the following sections.

Classification is often solved for superpixels, because they provide a region-based compression of the image, which allows compact and robust feature-based descriptions. For SS, the superpixels must not overlay the boundaries of the objects. This requirement translates into keeping the superpixels enough small. Alternatively, this limitation could be solved by means of correction algorithms called after SS, for improving the homogeneity of the labels. Based on valuable theoretical results [6] and numerous experimental demonstrations, graph

978-1-5386-4444-7/18/$31.00 ©2018 IEEE

cut corrections are very popular. Their limitations are mainly related to the involved computational time. An interesting version of MRF is introduced in [7]; the features are fused via energy terms that describe the color and the depth conditional probabilities. Correction is also solved by aggregating multiple models corresponding to different superpixel maps [8]. In this paper, corrections at superpixel level offer adequate precision for SS, their main role being to adjust (small) incorrectly labeled objects and to validate the results of classification.

This paper proposes a layer-based classifier equipped with different label correction strategies. Each layer refines/corrects the classification done by the previous layer, thus augmenting the understanding of the image and the confidence in results. The layer-based design of the classifier is mainly motivated by the idea of separating the classification problem in simpler tasks, which usually is helpful for the overall accuracy. Additionally, the layer-wise improvement of SS is helpful for solving specific safety and real-time constraints imposed by the application. The first two layers of the classifier consider 2D and 3D hand-engineering features extracted at superpixel level. 3D features embed information from the previous frames, thus making the feature extraction more meaningful.

Three main innovative correcting strategies are introduced for supporting label consistency. One is based on a RF classifier (included in the last layer) which provides additional alternate labels with corresponding confidence scores. This relabeling is based on features extracted at object level, for the objects defined in the segmented image received from the previous layer. The other two techniques are available for the first layers of the classifier and ensure the elimination of small objects, as well as a refinement of objects' borders. One of them one uses a faster majority voting replacement. The other one is based on the graph cut applied with cue weights relying on the available confidence scores or objects' areas. As detailed in the next sections, these corrections permit adequate tuning, in compliance with safety and real-time constraints.

This paper is organized as follows. Section II describes SoV architecture, as basis for discussing the limitations induced by the hardware and the particularities of the available data. Details about the feature extractor and the suggested layer-based architecture are presented in Sections III and IV, respectively, while Section V explains the label correcting procedures. Experimental results illustrating the performance of the system on diverse outdoor scenes are presented in section V. Last section includes a few concluding remarks.

II. SoV System Architecture

The SoV system works by acquiring 3D information from the environment using color and depth sensors, together with an **I**nertial **M**easurement **U**nit (IMU) device that allows recovering the orientation of the head and cameras. The acquisition hardware is placed onto a rigid structure attached to a headgear. In order to work in both indoor and outdoor environments, and irrespective of the illumination conditions, the 3D acquisition system employs two different types of depth sensors: a stereo RGB camera with configurable baseline (LI-OV580 from Leopard Imaging) for outdoor, and a Depth-of-Field camera (Structure Sensor PS1080 from Occipital) for indoor and low light image capture. The system consumes different types of 3D streams, stereo, depth or a fusion of both, depending on the used and environment (indoor, outdoor, low light, bright sunshine). Moreover, it employs different 3D approaches to deal with the specific structure and composition of environments. The 3D processing subsystem performs a 3D reconstruction of the sensed environment and segmentation into objects of interest. Next, the system builds a custom audio and/or haptic model of the 3D scene, which is then rendered to the user by means of hear-through headphones and a custom-made haptic belt, respectively. The SoV software runs on a portable computer carried in a custom made backpack with cooling facilities.

This paper considers the 3D processing module for outdoor environments, based on the system's stereo stream. The workflow is illustrated in Fig.1. The left and right images acquired from the stereo camera are rectified and used to compute the depth based on stereo correspondence, via the Elas algorithm [9]. Next, a 3D reconstruction of the environment is built based on the depth and color. The reconstruction consists in a global 3D point cloud obtained by incrementally adding the 3D representations of the individual frames, based on camera motion estimation. A confidence measure is associated to each 3D point forming the global model. This confidence depends on the number of frames in which the points could be tracked, i.e., the point was in the sensor's field of view and the disparity computation algorithm could provide a 3D measurement for it. The estimation of the ground is solved with a fast 2D approach [10], which combines information about camera orientation from IMU and camera motion estimation.

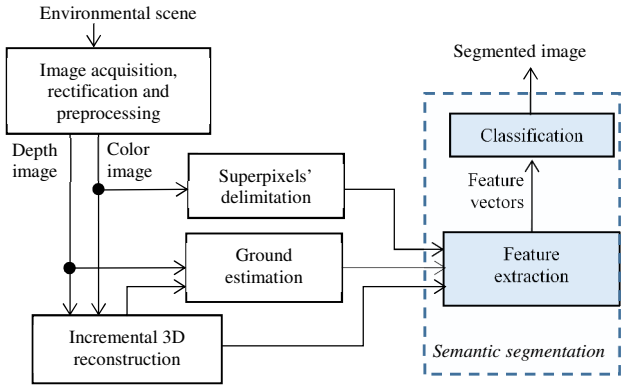

Fig. 1. The subsystems of SoV used for solving the SS.

III. Feature Extractor

The classifier is designed to work on feature vectors describing the superpixels. The superpixels are depicted in the color image converted to `Lab` map, by delimitating uniformly colored regions via SLIC algorithm [11]. This implicitly offers a compact representation of the image, which should be augmented with additional features, as long as standalone color-based features are insufficiently eloquent for outdoor/indoor scenes. In this context, the feature extractor

also considers features based on the reconstructed 3D point cloud, by only taking the confident measurements, i.e. those which could be tracked along a minimum number of frames.

Each superpixel is characterized by seven features, defined as follows: [F1] - the orientation of the superpixel relative to the ground, indicated by the cosine between the normal to ground and the normal to the closest plane approximation of the superpixel; [F2] - the average height above the ground; [F3] - the local planarity, i.e. the distance between the surface of the superpixel and its closest plane approximation; [F4] - the average neighbor planarity, i.e. the mean planarity of its neighbor superpixels; [F5] and [F6] - the average colors computed in the `Lab` map for `a` and `b` layers, respectively; [F7] - the average depth. Here, [F1] - [F4] implicitly embed some information from the previous frames, as well as information from other superpixels in the current frame, which helps for an accurate and robust SS. A robust classification is needed, as some feature values could result less precise due to the displacement of the camera.

IV. The Multi-Layer Classifier

The design of the classifier is based on a set of images relevant for the classification problem, $\mathbf{S_I} = \{\mathbf{I}_n \mid n = 1,..,N\}$, and the set of superpixels depicted from them, $\mathbf{S_P} = \{\mathbf{P}_q \mid \forall\, q = 1,..,Q,\ \exists \mathbf{I}_n \in \mathbf{S_I}\ a.i.\ \mathbf{P}_q \subset \mathbf{I}_n\}$. The training data set reflects some superpixels of $\mathbf{S_P}$, well-balanced between classes, randomly selected from several representative images of $\mathbf{S_I}$. The validation is done for all the other superpixels, i.e. for superpixels belonging both to images used and ignored during training. Some annotations indicate the desired classes for all the superpixels of $\mathbf{S_P}$, thus making possible a supervised training of the classifier, as well as the monitoring of misclassifications during the experiments. Targeted labels are stored at superpixel-level, as the corresponding segmentation precision is suitable for this application.

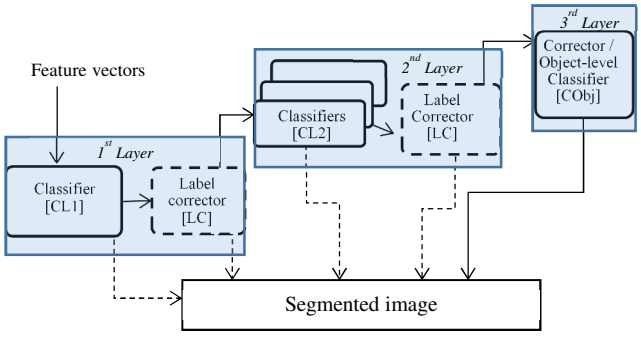

Fig. 2. The structure of the multi-layer classifier. SS can stop after each layer. Label correction in the first two layers is also optional.

Let $\mathbf{S_C} = \{\mathbf{c}_m \mid m = 1,..,M\}$ be the set of classes. Here, $c_1 = 0$ indicates the undecided class. This class could be obtained if the confidence level in any other class is very small, if the classifier has close high confidence in more than one class, or if the classification was skipped because the features of the superpixel could not be computed with an adequate confidence. As in any classification problem, one can expect that some classes are more difficult to distinguish, thus leading to frequent confusions between them and/or many undecided results. Forcing the classifier to learn all these classes in one step complicates the design and increases the risk of obtaining inconvenient overall accuracy/recall performance. The close classes could be depicted by analyzing the distribution of features in the training data set.

In this regard, a supplementary layer was introduced in the classifier. The groups of hardly separable classes are depicted and $\mathbf{S_C}$ is split in several corresponding disjoint subsets**,**

$$\mathbf{S_C} = \cup_{j=1}^{J} \boldsymbol{S}_c^j, \qquad (1)$$

with $\boldsymbol{S}_c^j \neq \emptyset,\ \forall\, j$ and $\boldsymbol{S}_c^j \cap \boldsymbol{S}_c^j = \emptyset,\ \forall i \neq j,$ with $i,\ j = 1,\dots,J$. Here, $\boldsymbol{S}_c^j$, indicates a group of hardly separable classes and $\boldsymbol{S}_c^1 = \{c_1\}$ is reserved for the undecided class. Because the classification is allowed to stop after the first layer, the classes of a group should be also semantically close, w. r. t. to the future guiding decisions.

The first two layers of the multi-layered classifier architecture (Fig. 2) result from (1). More precisely, the first layer ([CL1] in Fig. 2) assigns a label from 1 to *J*, indicating the group of classes to which the superpixel belongs. Afterwards, classification is refined in the second layer ([CL2]), where a separate classifier is designed for each group $\boldsymbol{S}_c^j$, with $j = 2,\dots J$ and $|\boldsymbol{S}_c^j| > 1$; here $|\cdot|$ specifies the size of a set. As consequence, any superpixel labeled with j >1 by the first layer calls for a supplementary classification in the second layer, if $|\boldsymbol{S}_c^j| > 1$. If $|\boldsymbol{S}_c^j| = 1$, the second layer must only map the label j into the corresponding label from $\mathbf{S_C}$

Random **f**orests (RF) are considered for [CL1] and [CL2]. Because each tree is built according to a random subset of samples and features, RF can accept a large variety of instances for each class, while being quite insensitive to outliers or irrelevant attributes. Also, RF allows a fast evaluation, in agreement with the real-time constraints of this application. The voting scores indicating the confidence in the resulted class are useful for label correction and guidance.

Two strategies working at superpixel or pixel level are proposed for correcting the labels in the first two layers ([LC] in Fig. 2). They aim improving the homogeneity of the labels. Both procedures analyze the neighbor labels in order to decide if a change is necessary. Supplementary corrections are also possible by means of the classifier suggested in the last layer, which is devoted to label correction performed at object level ([CObj]). For this classifier, the octo-connected pixels of the segmented image are separated into objects and a single label is managed for all the superpixels of an object. Unlike the correction procedures enabled in the first two layers, this classifier is not limited to change a label into another neighbor label. However, [CObj] cannot split the existing objects; hence, if some regions of the image are incorrectly merged by [CL], they remained merged after [CObj]. All the correction algorithms should have only a minor influence, as the main role in labeling must be assigned to classification.

[CObj] is an RF using the following features extracted at object level: the mean planarity, the minimum and the maximum height from ground, the minimum and the maximum depth, the mean and the difference between the

maxim and the minimum color intensity obtained in the last two layers of the LAB. In order to prevent incorrect relabeling, the object-level features are extracted only for the objects with enough valid pixels and a relabeling provided by [CObj] is accepted only if the confidence in the new label is higher than a threshold (*TH*) and the confidence shown by [CL]. The small objects ignored by [CObj] could be corrected via [LC], or simply ignored in the next decisions. By working at object-level, [CObj] implicitly analyses larger regions than [CL]. Therefore, the confidence scores provide valuable additional information for the future decisions.

Different runtime policies could be used in the multilayer classifier for preventing the violation of real- time constraints, whenever SoV faces a computational overloading: work only for the superpixels of the close objects (i.e. placed on the bottom of the acquired image); skip some intermediary frames, especially when the speed of the assisted person is very small and/or the most recent results indicate with high confidence the absence of any mobile objects or danger; work only with one or two layers of the classifier and/or disable the corrections in the first two layers.

V. Label Correction in the First Two Layers

As explained before, the label corrections in the first two layers are based on analyzing the neighbor labels found in the segmented image. LC-GC supports corrections at pixel level, via the graph cut algorithm. Mainly, graph cut searches for a trusty labeling which ensures a good homogeneity of the segmented image, by minimizing the following energy:

$$E(\mathbf{I}) = \sum_{p\in\mathbf{I}} E_c(p) + \sum_{p\in\mathbf{I}} \sum_{\boldsymbol{q}} E_{pq}(p), \qquad (2)$$

where, $E_c(p)$ specifies the lack of confidence in the class *c*, assigned to the pixel *p*, while $\sum_{\boldsymbol{q}} E_{pq}(p)$ describes the heterogeneity of the labels around *p*. Here, E_{pq} indicates the variations of the label in the q^{th} clique formed from *p*; a clique is a sequence of neighbor pixels, depicted according to a maximum admitted length (here, length 3).

In this paper, several configurations of the energy terms are proposed, defined in relation to the most desired corrections and the particularities of the RF classifier. Firstly, $E_c(p)$ is designed to use the available voting scores: $E_c(p) = 1 - Cf_c(p)$, where $Cf_c(p)$ is the confidence level for assigning the class *c* to the superpixel including *p*, i.e. the ratio of trees voting for *c*. As the ratios for all the potential classes have been already computed by RF, they can be directly used after any relabeling. Commonly, $E_c(p)$ uses the distance to the average values of the features resulted for the samples of *c*, but this involves some extra computational load for updating the averages after each replacement step.

Secondly, different spatial correction strategies are introduced via convenient settings of the cue parameters, i.e. the weights used in the second energy term of (2) for each edge of a clique. The available alternate configurations permit: i) intensifying the corrections in the neighborhood of the impaired user (where details are necessary), by increasing the cue weights on the rest of the image; ii) encouraging the use of trustier labels, $V(p) = 1 - Cf_c(p)$, where $V(p)$ indicates the cue associated to the pixel *p*; iii) encouraging the deletion of small objects: $V(p) = A_R(p)$, with $A_R(p) = k_A \cdot A(p)/\max_{p\in\mathbf{I}} A(p)$, where $A(p)$ is the area of the segmented object to which the pixel *p* belongs and k_A is a weighting parameter meant to intensify the elimination of very small objects; here $k_A = 1$ for the objects larger than average and the same $k_A < 1$ is set for all the objects smaller than average.

LC-GC is expected to provide smooth separations between objects. However this correction is suitable for the 2nd layer, because multiple labels could be assigned in a single superpixel, thus disturbing the connections with the feature vectors. Also, being NP-hard relative to the number of classes, its real-time performance is acceptable only if a few categories of objects are considered. In this context, correction is extended with a supplementary alternate procedure working at superpixel level, in full compatibility with the first two layers. LC-MV uses a majority voting for providing a fast elimination of small objects. Around any object with the area smaller than a predefined threshold, a dilated contour is drawn and all its labels are put in a list, denoted *LA*. Then, *LA* is reduced to *LB*, by excluding the labels for the undecided class and the invalid superpixels. Whenever the replacement is considered safe, the most frequent label of the (non-empty) *LB* is assigned to all the superpixels of the object. The validation of the corrections may work with or without the confidence scores given by the classifier. When the confidence scores are ignored, a new label is accepted only if the majority label is supported by a reasonable ratio of valid pixels around the object, i.e.:

$$|LB| > r_v \cdot |LA|, r_v \in (0,1) - \text{here}, r_v = 0.2. \qquad (3)$$

This validation is refined if the confidence scores are taken into account. The mean confidence of the new label, computed w. r. t. the pixels belonging to the contour of the object ($\overline{Cf}_{new}$), is compared with the mean confidence of the current label, resulted w .r. t. the pixels of the object ($\overline{Cf}_{old}$), in order to decide if the replacement is safe. As mentioned before, the confidence levels result from the voting scores obtained by RF. If the new label is trustier than the current one ($\overline{Cf}_{new} > \overline{Cf}_{old}$), the validation condition is relaxed, by enabling the replacement of any extremely small object. Also, a significantly trustier new label ($\overline{Cf}_{old} < 0.8\overline{Cf}_{new}$) is accepted for a smaller ratio in (3), i.e. for $r_v \rightarrow 0.9r_v$.

VI. Experimental Results

The experimental verifications have been done for outdoor scenes acquired with the SoV. Many frames pose difficulties to SS, because they include uneven ground, as well as a large variety of static and moving obstacles (buildings, poles, benches, fences, cars, trees etc.) - for some regions the classes being quite unclear even for a human-based annotation. The training data set was formed with valid superpixels randomly selected from 60% of the images. Given the common appearances of the street scenes, expected to include more superpixels from a few classes, the training data set is generated after analyzing the distribution of superpixels per classes. In this regard, a threshold is set to:

$$thrS = 1.5 \min_{\substack{i=1,..M, \\ Ns(i) > \frac{0.6}{M}\sum_{i=1}^{M} Ns(i)}} Ns(i) \quad (4)$$

where M denotes the number of targeted classes and $Ns(i)$ specifies the number of samples belonging to the i^{th} class. If $No(i) > thrS$, then $thrS$ random samples are selected from the class; otherwise all the samples of the class included the training images are used. The validation data set comprises the superpixels remaining in the training images from the highly populated classes, as well as superpixels from the images unused for the construction of the training data set - in this case, about 47% of the available samples.

As RF accepts any range of attributes without ill conditioning problems, the features could be kept unscaled. The correlation analysis indicates higher linear dependency between [F1] and [F2] (as, usually, the superpixels from big heights have a vertical posture) and between [F3] and [F7] (as most far objects look flat). The small correlations between [F3] and all the other features motivate the use of the average neighbor planarity as attribute.

Firstly, the two-layer classification and the correction algorithms introduced in the previous sections are separately examined, for demonstrating their role. Lastly, the overall performance is investigated on several integrating configurations. The performance of a two-layer classification is discussed vs. a single-layer one (#2-#3 vs. #1 in Table I). In this attempt, the classification is solved for $M = 6$ (i. e., undecided [1], ground [2], small static objects [3], cars [5], tall thin objects [4], tall large objects [6]). For the two-layer classification, two configurations are considered, namely V1 with four subsets – undecided, ground, small objects (static or cars), tall objects (large or thin), ($\mathbf{S_C} = \{1\} \cup \{2\} \cup \{3,5\} \cup \{4,6\}$) and V2 with three subsets - undecided, ground, objects ($\mathbf{S_C} = \{1\} \cup \{2\} \cup \{3,4,5,6\}$). Both separations allows a premature stop of the classification after the first layer, as desired by design. Also, V1 and V2 are motivated by Wilcoxon rank sum tests performed separately for each feature and any pair of classes, in order to verify the hypotehsis of equal median, while considering that [F1] and [F2] are among the most influential features in this classification problem, as shown by their quite large correlation with the targeted classes.

RFs are designed with 100 trees, based on a preliminary analysis of the misclassifications generated by a one-layer classifier with different number of trees. Because RF involves stochastic design procedures, each configuration is run for 5 trials and the best result is stored for comparison. All the experimental configurations lead to 0 errors on training. On testing, the best results were achieved by the two-layer classifiers. V2 offers slightly worse results than V1 (#3 vs. #2), perhaps because it passes significant charges to a single classifier in the 2nd layer. Wilcoxon rank sum tests indicate significant differences between the results obtained for the configurations listed in Table I, thus validating the use of a two-layer classifier. A supplementary investigation of the misclassifications showed that they do not trigger dangerous guiding decisions, because they generally refer to far objects or less significant details (e.g. branches of trees associated with the building behind, ground around very close cars associated to cars in a parking area, bottom boundaries of some buildings not very precisely delimited from an uneven ground). An example is illustrated in Fig. 3.

For a fair comparison, the correction algorithms are separately applied after a single-layer classification, using the same RF for producing the segmented images; also, one of the worst classifier is chosen from the previous tests, in order to ensure a working context for corrections as diverse as possible - with 389 wrongly labeled superpixels (from 6205 totally), i.e. 403890 pixels for the whole testing data set. Because LC-GC works at pixel-level, the results are compared by counting the number of pixels with good corrections (*NB*) and the number of pixels changed to incorrect labels (*NW*). The testing configurations are specified in Table II.

For a faster correction, LC-GC is limited to 3 iterations. As expected, LC-GC improves the homogeneity of the labels by also relabeling some boundary pixels of the objects, correctly labeled by [CL] (annotations are stored only at superpixel level). These extra changes explain the large values of *NW*, however they do not impede the future guidance. Mainly, the useful corrections ensured by LC-GC refer to the deletion of the small objects. LC-GC was tested for all the proposed configurations, with different parameters. LC-GC-i) #7 and #8 intensify the corrections of the labels from the central part of the image (outside a border of size *SB*). As the upper pixels are usually invalid, this correction ignores especially the left and right far objects. On the contrary, #9 corrects mainly these peripheral objects. For most images, the central region includes small residual objects resulted at the boundary between valid and invalid pixels, hence #8 and #7 can illustrate the undesired effect produced by too small cues weights when working on regions with many small objects. In these cases, LC-GC-i triggers many corrections, but not all of them useful for a classification defined at superpixel level. For these regions, increased cue weights are suitable. The fewest errors produced by LC-GC are obtained when the cue weights are set in relation to the confidence scores (LC-GC-ii, i.e. #10) or the area of the object (LC-GC–iii, i.e. #11-13). These configurations are also safer than the classic LC-GC #6 (using unit cue weights). The influence of k_A is marginal for LC-GC-iii, perhaps due to the distribution of objects' area.

The confidence scores are also useful in the case of LC-MV (#14-16 vs. #17-19). However, LC-MV should be limited to work on small objects, only (#14 vs. #15 and #15 vs. #17). As expected, accepting the new majority neighbor label only if many valid pixels are found around the object is useful for validating safer corrections (#14 vs. #16 and #15 vs. #18). Like LC-GC, [LC-MV] mainly eliminates the small objects.

The best results were obtained for [CObj] working with big validation thresholds. Given the design of [CObj], setting a large *TH* is a natural requirement. As long as [CObj] assigns the new labels by classification, without exploring the neighbors, any potentially incorrect labels could significantly change the meaning of the segmented image. Hence, the results of [CObj] should be validated only if a very high

confidence level is obtained (#21, 22). Some results are presented in Fig. 4 for #10, #15 and #22 - which provide the safest corrections. For these strategies the labels are mainly decided by [CL] (which is more robust by design) while accepting that in some images only a few errors are corrected.

The last experiments considered different combinations between LC-MV and [CObj], for single-layer (Table II -#24) and two-layer classifiers (Table I - #4 and #5). As shown in Table I and II, they generate fewer mistakes at recall than the configurations without corrections. The complexity order of corrections is mostly influenced by the resolution of the images (which is reasonably small, i.e. 388 x 796). In all the configurations, the RFs includes only short trees (with a maximum depth 24 for [CL] and 15 for [CObj]), thus obtaining very fast RF's evaluations.

TABLE I. EXPERIMENTAL RESULTS – CLASSIFICATION

#	*Classifier*	*Correction*	*No. of errors [superpixels]*	*Mistakes per classes [training and testing]*
1	1 layer	No	339	class 2 – 136, class 3 – 64; class 4 -99, class 5 – 18, class 6 -22
2	2 layers, with V1	No	266	class 2 -79, class 3 - 43, class 4 – 139, class 5 – 4, class 6 – 1
3	2 layers, with V2	No	274	class 2 - 127; class 3 - 44, class 4 -75, class 5 - 12, class 6 - 16
4	2 layers, with V1	[CObj]-#22	246	class 2 -71, class 3 - 43; class 4 - 112, class 5 – 18, class 6 – 2
5	2 layers, with V1	LC-MV-#15 & [CObj]-#22	235	class 2 -63, class 3 - 41; class 4 - 99, class 5 – 30, class 6 – 2

TABLE II. LABEL CORRECTION [TRAINING AND VALIDATION]

#	*Algorithm*	*Parameters*	*Corrected [NB]*	*New errors [NW]*
6	LC-GC	-	31039	16946
7	[1]LC-GC i)	*VAL*=100, *SB*=20	35665	40723
8		*VAL*=100, *SB*=40	44441	73967
9		*VAL*=0.1, *SB*=20	31342	16557
10	LC-GC ii)	-	26024	13444
11	[2]LC-GC iii)	$k_A = 100$	23171	11629
12		$k_A = 10$	23085	11672
13		$k_A = 1$	23042	11672
14	[3]LC-MV WOCF	*MinA*=1500, r_v = 1/5	36872	36563
15		*MinA*=3000, r_v = 1/5	63033	71749
16		*MinA*=1500, r_v = 1/3	31495	28849
15	[3]LC-MV – WCF	*MinA*=1500, r_v = 1/5	43414	41296
17		*MinA*=3000, r_v = 1/5	73259	79601
18		*MinA*=1500, r_v = 1/3	39556	34017
19	[4][Cobj]	*MA*=500, *TH*=0.5	27962	6716
20		*MA*=500, *TH*=0.6	27316	5701
21		*MA*=500, *TH*=0.7	23023	2919
22		*MA*=500, *TH*=0.8	13920	487
23		*MA*=750, *TH*=0.6	25558	4893
24	[CObj]-#22 & LC-MV-#15	*MA*=500, *TH*=0.8 *MinA*=1500, r_v = 1/5	33437	0

[1]LC-GC-i) is applied with cue weights equal to *VAL*, on a border of size *SB*;
[2]For LC-GC-ii),the k_A indicated here is used for the objects smaller than average;
[3] LC-MV is applied without confidence score (WOCF) or with confidence scores (WCF), for objects larger than *MinA*, by working on a contour of *WO* width;
[4][Cobj] works on objects larger than *MA*, according to the validation threshold *TH*.

VII. CONCLUSIONS

This paper proposes a multi-layer classifier with correction algorithms for semantic segmentation. The solution is built for a wearable assistive system helping the visually impaired persons. The street scene is described at superpixel level by color and 3D features which integrate information from temporal and spatial vicinities. The classification task is split into RFs organized in two layers. Optional minor relabeling is provided by means of graph cut customizations using different polices for the cue parameters and a voting correction algorithm. Additionally, correction could be done via a classification working with object-level features.

The experimental results indicate that the two-layer classification is more effective than a single-layer one. Also safe corrections could be provided by all the suggested procedures. Future research will investigate the possibility of fusing the correction algorithms via adaptive mechanism, by taking into account the layout of the segmented image and the confidence scores, in support of a robust and safe relabeling.

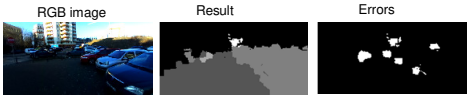

Fig. 3. Example of segmented image - testing frame (#2)

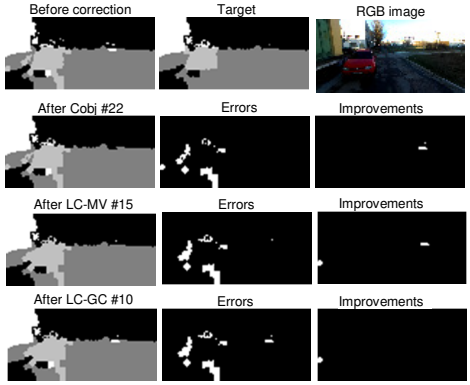

Fig. 4. Examples of corrections on a testing frame (# 22, #15, #10 - segmentation (left), improvements (right) and errors (middle) after correction)

REFERENCES

[1] http://www.soundofvision.net/

[2] H. Yu, Z. Yang, L. Tan, Y. Wang, W. Sun, M. Sun and Y. Tang, "Methods and datasets on SS: A review", Neurocomputing, vol. 304, pp. 82-103, 2018.

[3] D. Ravì, M. Bober, G. M. Farinella, M. Guarnera and S. Battiato, "SS of Images Exploiting DCT based Features and RF", Pattern Recognition, 52 (4), pp. 260-273, 2016.

[4] X. Wang, G. Yan, H. Wang, J. Fu, J. Hua, J. Wang, Y. Yang, G. Zhang and H. Bao, "Semantic Annotation for Complex Video Street Views based on 2D–3D Multifeature Fusion and Aggregated Boosting Decision Forests", Pattern Recognition, 62 (2), pp. 189-201, 2017.

[5] J. Xiao, L. Quan, "Multiple View SS for Street View Images", IEEE 12th Internat. Conf. on Computer Vision, 2009.

[6] V. Kolmogorov and R. Zabih, Member, "What Energy Functions Can Be Minimized via Graph Cuts?",IEEE Trans. on PAMI, 26 (2), 2004.

[7] I. Jebari and D. Filliat, "Color and Depth-Based Superpixels for Background and Object Segm.", Proc. Eng., 41, 1307 – 1315, 2012.

[8] L. Ladický, C. Russell, P. Kohli and P. Torr, "Associative Hierarchical Random Fields", IEEE 12th Internat. Conf. on Computer Vision, 2009.

[9] A. Geiger, M. Roser, and R. Urtasun, "Efficient Large-Scale Stereo Matching," in Asian Conf. on Computer Vision (ACCV), 2010.

[10] P. Herghelegiu, A. Burlacu and S. Caraiman, "Robust Ground Plane Detection and Tracking in Stereo Sequences Using Camera Orientation", 20th IEEE ICSTCC, Sinaia, Romania, 2016.

[11] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua and S. Süsstrunk, "SLIC Superpixels Compared to State-of-the-Art Superpixel Methods," in IEEE Trans. on PAMI, 34 (11), pp. 2274-2282, 2012.