IEEE Third International Conference on Data Stream Mining & Processing
August 21-25, 2020, Lviv, Ukraine

Comparative Analysis of Classic Computer Vision Methods and Deep Convolutional Neural Networks for Floor Segmentation

Anastasiia Skoryk
It-Jim
Kharkiv, Ukraine
seredkina.a@gmail.com

Yurii Chyrka
It-Jim
Kharkiv, Ukraine
yurii.chyrka@it-jim.com

Ievgen Gorovyi
It-Jim
Kharkiv, Ukraine
ceo@it-jim.com

Oleksiy Grechnyev
It-Jim
Kharkiv, Ukraine
shrike@it-jim.com

Pavlo Vyplavin
It-Jim
Kharkiv, Ukraine
pavlo.vyplavin@it-jim.com

***Abstract*—In the paper, we analyze the problem of automatic room floor segmentation. For this purpose, we consider several classic computer vision algorithms as well as some of the deep convolutional neural network architectures. The segmentation results are illustrated and compared. An idea for combining two groups of methods is proposed. It is demonstrated that a proper fusion provides the best segmentation quality.**

Keywords—indoor image segmentation, superpixels, graph clustering, deep learning, CNN.

I. INTRODUCTION

Image segmentation is one of the key topics in computer vision. Usually, it is interpreted as semantic segmentation, i.e. linking of each pixel in an image with a label from a particular set of classes, for example, “human”, “grass”, “road”, “floor”, “table”, etc. Segmentation appears in a wide range of applications such as scientific image analysis, robotic vision, scene understanding, augmented reality and many more [1, 2].

In the case of segmentation of surfaces with a similar texture or patterns, we can label subsets of pixels that share similar characteristics: intensities, colors, and locations. However, correct separation of different classes may be a challenge due to varying illumination, noise, occlusions, shades, light spots, reflections, and camera perspective changes.

There are many existing methods for image segmentation: from classic ones like simple thresholding [3] or superpixels [4, 5] to quite advanced deep learning-based solutions [6]. In addition, various machine learning methods are often used along with hand-crafted features [7-9].

In this paper, we analyze the problem of automatic room floor segmentation. Such a solution can be used for different purposes like mixed reality (MR) applications, interior design, and entertainment. Our goal is to analyze both classic computer vision methods as well as common deep learning (DL) based convolutional neural network (CNN) architectures. As well we propose a methodology for combination of classic and deep learning based methods in order to get the best overall result.

In Section II, we briefly describe the methods used in our experiments and show some of the intermediate image processing steps. A proposed fusion scheme of classic and DL-based branches outputs is shown in Section III. Finally, datasets and experiment results are described in Section IV.

II. METHODS OVERVIEW

A. Classical Pipeline

Among many different methods for indoor images segmentation [10-12], superpixels are the most widely used technique [4, 5].

According to the definition, a superpixel is a group of a few pixels with common properties **Error! Reference source not found.**. Representing an image as a group of superpixels allows one to get a compact representation and to retrieve the image regions sharing the same properties.

There are different variations of superpixel algorithms [4, 13]. One of the most widely used approach, the simple linear iterative clustering (SLIC), adapts a k-means clustering. The method works by clustering pixels based on their color similarity and proximity in the image plane [4]. The distance between two pixels in the combined five-dimensional LabXY [4] space is defined as follows:

$$\begin{aligned} d_{lab} &= \sqrt{(l_k - l_i)^2 + (a_k - a_i)^2 + (b_k - b_i)^2} \\ d_{xy} &= \sqrt{(x_k - x_i)^2 + (y_k - y_i)^2} \\ D_s &= d_{lab} + \frac{m}{S} d_{xy}, \end{aligned} \tag{1}$$

where S is a grid interval, m is the compactness parameter.

An example of superpixels clustering is given in Fig. 1b. Normally, two important parameters are tuned: the number of superpixels (was set to 300) and the compactness measure (was set to 7). The former corresponds to the maximum amount of superpixels to be extracted from the image, while the latter corresponds to the trade-off between proximity and

978-1-7281-3214-3/20/$31.00 ©2020 IEEE

color-similarity. Compactness controls the shape and smoothness of the superpixels' boundaries: with higher compactness they become smoother and the superpixels become more regular.

The problem is that the straightforward application of superpixels does not provide a perfectly segmented floor. In order to overcome this difficulty, we have created an additional pipeline for image processing. First of all, we group pixels of a color image (Fig. 1a) into superpixels (Fig. 1b). In parallel, we transform RGB image into HSV color space and work with the saturation channel only (Fig. 1d), because it highlights changes between the room surfaces the most. Then, we obtain an edge map of the S-channel image (Fig. 1e). From the combination of the superpixels image and the edge map

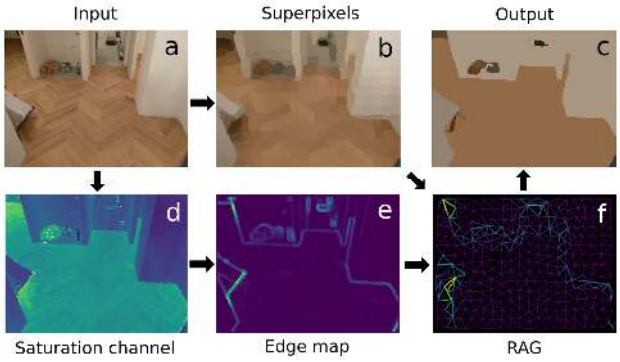


Fig. 1. The main steps of the classical pipeline. a) The input color image; b) the SLIC superpixels; c) the output clusterized image from the merged RAG; d) the saturation color channel; e) the image with highlighted edges; f) the RAG constructed from the superpixels and the edge map.

we construct a region of adjacency graph (RAG), which has greater edge weights on the borders (Fig. 1f). Finally, we cluster the superpixels into groups using graph hierarchical merging algorithm (Fig. 1c). Some details of this pipelines are given below.

RAG is an undirected weighted graph. Its vertices represent image areas (for example, superpixels), while its edges correspond to the connections between the adjacent regions [15]. RAGs give a spatial view of the images and are powerful tools for image processing if neighborhood relationships can be taken into account. In our case, images with emphasized edges (edge maps) are used to present this information. The Sobel gradient magnitude filter [16, 17] and the local binary pattern (LBP) feature map extraction [18] provides the most emphasized edges, so we used these two algorithms to construct the edge maps.

We obtain the output image regions (Fig. 1c) by performing agglomerative hierarchical clustering with mean linkage until a threshold [19, 20]. As the appropriate threshold value highly depends on the image, we estimated it from the distribution of the graph edge weights. The threshold is a value which corresponds to a specific percentile (for example 80% as shown in Fig. 2). In this case, RAGs are associated with their unique threshold while merging.

The graphs before and after the hierarchical merging are visualized in Fig. 3. The boundaries of image regions are also shown. These regions are the return segments, and one of them would be estimated as a floor. To decide which segment is the desired floor, we just take the biggest segment at the bottom of the image.

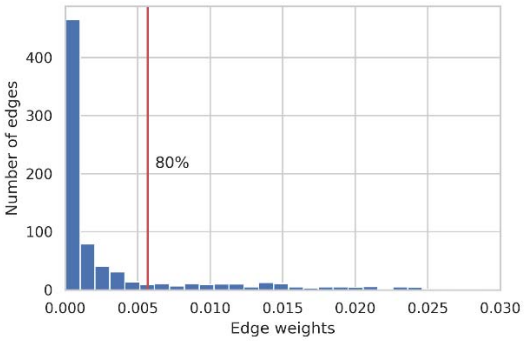


Fig. 2. The distribution of the RAG edge weights. Vertical line shows the percentile value to estimate threshold for hierarchical merging.

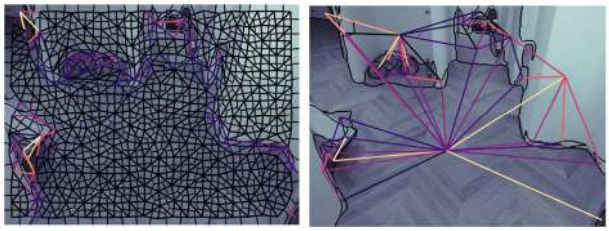


Fig. 3. The RAG before (left) and after (right) hierarchical merging. All nodes with the edge weight less than a threshold are merged together. Border of segments are shown in black.

Since the classical approach is very sensitive to parameter tuning, we have run the classical pipeline several times with different model parameters, resulting in many segmentation masks. The adjusted settings included RGB and HSV images as the SLIC inputs; Sobel filtering and LBP rotation invariant edge extraction as methods for the edge detection; 75%, 80% and 85% percentile values for the threshold estimation.

We add all binary masks together and if more than half of them are positive about a pixel, we label this pixel as true. Otherwise, we label it as false. As a result, we have a single binary segmentation mask.

B. Deep Learning Pipeine

There are many different DL architectures available for floor segmentation [6, 21-23]. We used two CNNs: light-weight RefineNet [24] (see Fig. 4a) and FastFCN [25] with a joint pyramid upsampling (JPU) (see Fig. 4b). We used both CNN architectures with minimum changes, only the output layers were transformed to predict just 2 classes: floor and not a floor.

C. *Post-processing: Texture Feature Analysis and Edge Refinement*

Masks predicted either by classical algorithms or by CNN may have Masks predicted either by classical algorithms or by CNN may have complicated boundaries, while the floor shape is usually more or less straight. Moreover, when adding many

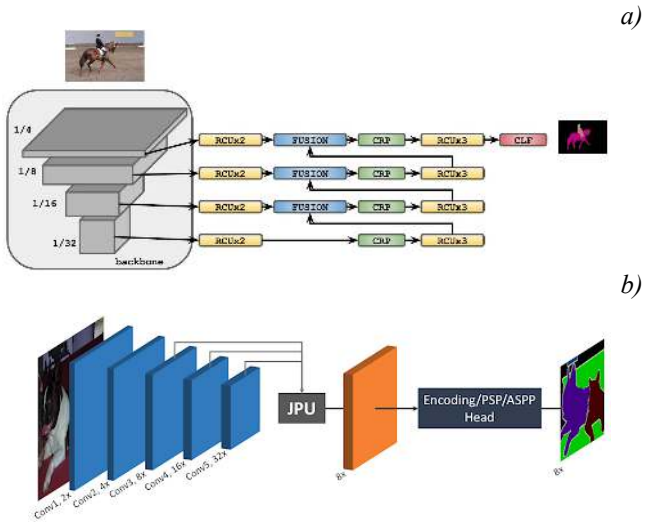


Fig. 4. The CNNs architectures used in the paper. a) The RefineNet architecture for semantic segmentation [23]. b) The FastFCN with a Joint Pyramid Upsampling (JPU) module and a multi-scale/global context module [24].

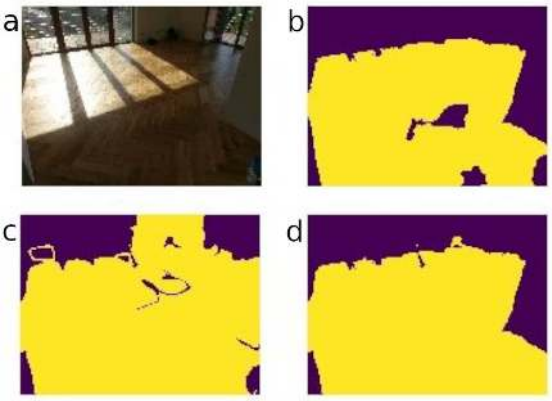


Fig. 5. Post-processing based on the texture feature analysis. a) the input image; b) the classical pipeline output; c) the mask from the deep learning pipeline; d) the mask after post-processing.

masks, instead of binary segmentation with two labels, we get determined areas (either a floor or not a floor) where pixels have the same label on each mask, and some undetermined regions where masks have opposite labels. Texture features may be really helpful for analyzing of indoor surfaces and for final classification of undetermined segments to a floor or not a floor.

The whole image or its separate segments can be represented by features such as shape differences, texture differences, color or light fluctuations. The feature extraction algorithm provides fewer but more meaningful parameters to describe an image or its parts.

A wide range of algorithms for texture features extraction including statistical-based, transform-based, graph-based approaches and many other methods and their heirs are described in the literature [26]. In this study, we use a gray level co-occurrence matrix (GLCM) [27] which determines how often different pairs of pixels appear in an image. The GLCM expresses a matrix with a shape of image bitrates. From this matrix, one can extract features like 'contrast', 'dissimilarity', 'homogeneity', 'ASM', 'energy' and 'correlation' [27].

Extracting GLCM features from different segments of the image makes it possible to calculate the Euclidean distance in multidimensional feature space from the undefined segment to the determined floor (or not a floor) segment (2). The dimensionality n of feature vectors corresponding to the number of features are taken into account.

$$d(x,y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}. \quad (2)$$

The post-processing part combines many segmentation masks obtained by different methods. The main purpose of this stage is the final classification of uncertain areas or blobs. Feature analysis resolves these uncertainties and makes more accurate prediction (see Fig.5).

Finding blobs is done by analyzing contours of the masks. All uncertain blobs are linked to one of two (a floor or not a floor) determined areas by calculating the minimum distance in the feature space (2).

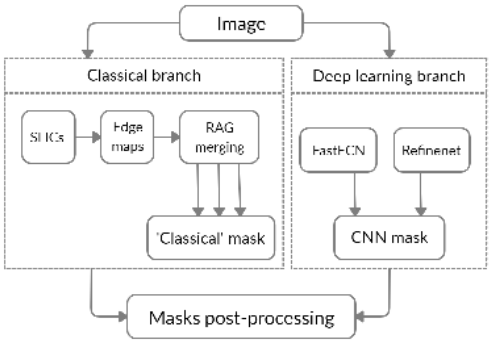


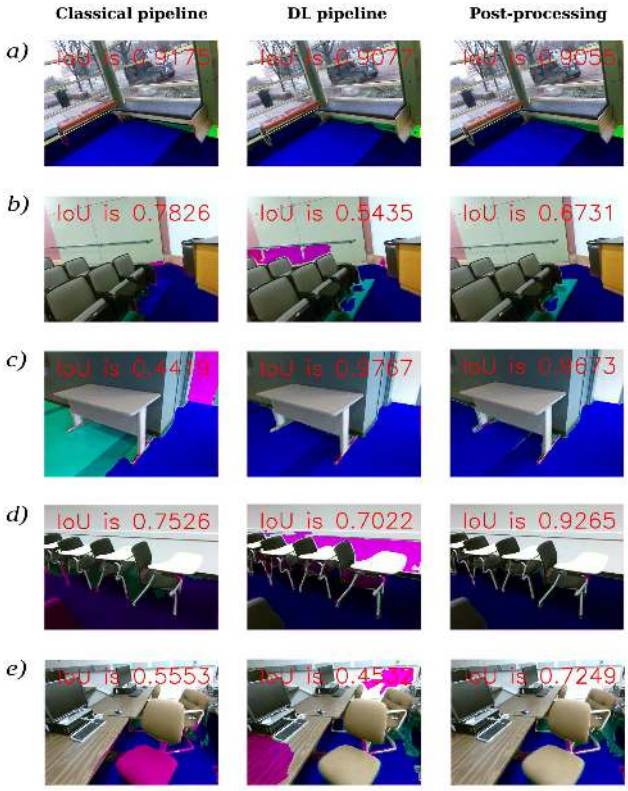
Fig. 6. Full pipeline overview. Masks from 'classical' and DL branches are combined together and post-processed.

D. *Fusion Scheme*

Both classical and DL solutions failed in some cases. In order to additionally refine the quality of segmentation maps, we decided to build a fusion scheme shown in Fig 6.

An RGB image is processed separately by the classical and the DL branches. The classical branch includes SLIC superpixeling, obtaining edge maps, RAG constructing and RAG hierarchical merging. These basic steps repeat with various parameters that provide many segmentation masks, that are summed together. The binary mask obtained by thresholding of the sum is the output mask from the classical branch.

The DL branch consists of two neural networks and independently predicts segmentation masks for the input image. They are also added together. Finally, two outputs from the both branches are combined and the post-processing



using texture feature analysis is implemented. A slight edge refinement is applied in the very end.

III. Experiments and Results

A. Datasets

We worked with several datasets. In order to train CNNs, we used 1449 images from NYUDv2 [28]; 10329 images from the SUN-RGB-D [29-31] and 8880 images from the SUN-RGB-D with NYUD removed. The target dataset was a set of 21 hand-labeled images acquired for evaluation purposes.

B. Results

To evaluate the results of segmentation we used Intersection over Union (IoU) [32]. The best result was achieved with merging of 3 masks (two from the neural networks and one summed mask from the classical pipeline) and applying the post-processing based on texture feature analysis in the end. All intermediate IoU values are shown in the table below.

TABLE I. The Result Evaluation

Mask obtained with:	***IoU***
Classical branch	0.5442
Refinenet	0.7837
FastFCN	0.7893
Deep learning branch	0.7939
Classical + deep learning branches	0.7977
Full pipeline	**0.8013**

Fig. 7. Examples of segmentation masks obtained with classical pipeline, deep learning pipeline and as a result of their combination and post-processing. a) Both classical and deep learning pipelines work well. b) Classical pipeline outperforms the deep learning approach. c) Deep learning pipeline works better than the classical one. d) Both classical and deep learning pipelines work fine and post-processing makes an improvement. e) Both classical and deep learning pipelines work bad, post-processing is used. Color legend in the figure: dark blue is true positive, magentra is false positive, cian is false negative.

Fig. 7 contains some examples of floor segmentation obtained with different setups. As expected, deep learning solution handles more challenging cases better than classical computer vision pipeline. However, for some images developed image analysis procedure provides quite competitive results or even outperforms CNN-based solution. This is explained by the size and quality of the training data, which is crucial for DL-based methods applied for typical computer vision tasks. Finally, the proposed post-processing step based on feature crafting allows refining the quality of segmentation maps.

IV. Conclusions

In this work, we analyzed the problem of room floor segmentation. We have applied both classical computer vision and deep learning techniques for this task. Firstly, we constructed a custom classical pipeline based on superpixels, region adjacency graphs, and graph hierarchical merging. Secondly, we picked two typical CNN architectures and compared their output predictions. Finally, we have built a fusion scheme to combine outputs from two branches and applied post-processing based on textural features analysis. It is clearly seen from the conducted experiments that the proper combination of computer vision methods always gives the best outcomes. In the future, we are planning to additionally improve the segmentation quality and to integrate the developed pipeline into a mobile application.

References

[1] F. Salzenstein and C. Collet, “Fuzzy Markov Random Fields versus Chains for Multispectral Image Segmentation,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 28, no. 11, pp. 1753–1767, 2006.

[2] J. Bruce, T. Balch and M. Veloso, "Fast and inexpensive color image segmentation for interactive robots," Proceedings. 2000 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2000) (Cat. No.00CH37113), Takamatsu, Japan, 2000, pp. 2061-2066 vol.3.

[3] O. J. Tobias and R. Seara, "Image segmentation by histogram thresholding using fuzzy sets," in IEEE Transactions on Image Processing, vol. 11, no. 12, pp. 1457-1465, Dec. 2002.

[4] 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 Transactions on Pattern Analysis and Machine Intelligence, vol. 34, no. 11, pp. 2274-2282, Nov. 2012.

[5] D. Stutz, A. Hermans, and B. Leibe, “Superpixels: An evaluation of the state-of-the-art,” Computer Vision and Image Understanding, vol. 166, pp. 1–27, 2018.

[6] S. Minaee, Y. Boykov, F. Porikli, A. Plaza, N. Kehtarnavaz, D. Terzopoulos, "Image Segmentation Using Deep Learning: A Survey," [online], Available: arXiv:2001.05566v2.

[7] Y. Artan, "Interactive Image Segmentation Using Machine Learning Techniques," In Canadian Conference on Computer and Robot Vision, St. Johns, NL, 2011, pp. 264-269.

[8] J. Zhang, Z. H. Tang, W. H. Gui, Q. Chen, and J. P. Liu, , “Interactive image segmentation with a regression based ensemble learning paradigm” In *Frontiers of Information Technology & Electronic Engineering*, vol. 18, no. 7, 2017, pp. 1002-1020

[9] A. Madabhushi and G. Lee, “Image analysis and machine learning in digital pathology: Challenges and opportunities,” Medical Image Analysis, vol. 33, pp. 170–175, 2016.

[10] D. C. Lee, M. Hebert, and T. Kanade. “Geometric Reasoning for Single Image Structure Recovery”. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2009.