Floor Space Optimisation And Recommendation System in 2D Space

Nakul Aggarwal, Asmita Mahajan, Snehlata Sinha, Abhilasha Lodha,
Debasmita Ghosh, Vinay Dilip Raut

Advance Technology Centers in India, Accenture, India
Email: {nakul.aggarwal, asmita.mahajan, snehlata.sinha}@accenture.com, abhilashalodha0101@gmail.com,
{debasmita.ghosh, vinay.dilip.raut}@accenture.com

***Abstract*—**

In this paper, we are proposing an open-source solution for empty space calculation in images by segmenting out the objects using computer-vision technologies. We have pre-existing solutions for 3D floor mapping but there is no end-to-end approach that can be implemented in a 2D space. Our solution takes an input image of a scene like an empty balcony or a room and provides us a 2D floor mapping of the space. Using transfer learning, a custom trained semantic segmentation model is used to identify the objects in images. The outcome of semantic segmentation model is used by our custom algorithm to determine the empty floor space in an image and segment the area into restricted and non-restricted regions. The restricted segments, like the pathway in-front of the doors, furniture etc., are the ones that should be kept empty for people to move around easily. The non-restricted floor area can be utilised for a wide number of use-cases like design recommendations, autonomous robot navigation, empty parking space identification etc. This paper mainly deals with the balcony garden design recommendation, however as mentioned the same approach can be extended to other spaces.

***Keywords*—Computer vision, 2D space, floor mapping, transfer learning, recommendation system**

1. Introduction

Efficient usage of floor space mapping and recommendation are extremely desired by house planners, interior decorators, architects, etc. to suffice functional as well as artistic purposes. Creating floor-maps is a creative process that requires imagination as well as mathematical skills. People can be trained to perform the task relatively well; however, hiring a professional workforce can be quite expensive and time-consuming. This can be best achieved by an Artificial Intelligence based recommendation system that provides a cost-effective way of obtaining the otherwise expensive space-optimization services.

Recently, number of solutions that employ floor mapping in 3D space have been implemented using Extended Reality (XR) and Augmented Reality (AR). But there are no appropriate end-to-end solutions readily available that can implement floor mapping in a 2D space. Here, we are proposing a highly optimized open-source solution which takes an input image of an empty balcony and gives out design recommendations for placement of plants in the detected empty regions. So, our solution offers an end to end approach for empty floor space calculation in 2D spaces by segmenting out the objects using computer vision based semantic segmentation algorithm.

The key benefit of using semantic segmentation over object detection is that it identifies objects present in an image at the pixel level [1]. This helps in precise localisation of the empty floor space area and thus aids in an efficient floor area calculation.The model architecture used in developing the solution is Mask R-CNN. Using transfer learning, this model is trained on custom dataset to segment six classes namely wall, door, window, ceiling, railing and furniture present inside the image. A custom logic is developed to determine the empty floor space in the image using the output of semantic segmentation model. The calculated empty floor space is then segmented into restricted and non-restricted regions.

The restricted area is the one that is kept empty for people to move around easily, like pathway in front of doors, furniture etc. The non-restricted floor area can be effectively utilised for wide number of applications like indoor design recommendations, autonomous robot navigation, empty parking space identification etc. Here we have demonstrated a similar use case for balcony garden design recommendation. For every identified non restricted area, the number of plant rows are calculated and using Image Blending technique, the plant images get seamlessly blended into the non-restricted segments, thereby giving a beautiful balcony garden design recommendation.

2. Literature Review

Related Architectures:

Cheng et al. [2], proposed the current cutting edge semantic segmentation approach based on DeepLabV3+, who enhanced DeepLabV3 [3] into a encoder-decoder model. The original FCN (2015) had a mean IOU of 67.2% and DeepLabV3+ (2018) has IOU of 67.2% on the PASCAL VOC 2012 dataset. DeepLabV3+ outperforms many SOTA approaches like ResNet-38 and PSPNet with an IOU of 87.83% on the PASCAL VOC 2012 dataset with DeepLabV3+ (Xception) method and an IOU of 89.00% with DeepLabV3+ (Xception-JFT) method as it works effectively in detecting the object boundaries. However, architectural properties can affect the robustness of a model significantly. The generalization capability of DeepLabv3+ model, using a ResNet-backbone, depends strongly on the type of image corruption.

Panqu wang et al. [1] have utilised deep Convolutional Neural Networks (CNNs) to get a significant improvement over previous semantic segmentation systems outperforming FCN, DilatedNet, and DeepLabv2. They have shown how to improve pixel-wise semantic segmentation by manipulating convolution related operations. They have tested their solution on Cityscapes, KITTI road estimation and PASCAL VOC2012 dataset. Their model, ResNet-DUC-HDC, have been designed with a new dense upsampling convolution (DUC) operation to enable pixel-level prediction on feature maps, and hybrid dilated convolution (HDC) to solve the gridding problem, effectively enlarging the receptive fields of the network. However, there are SOTA models like DeepLabV3+ and Mask-RCNN that has outperformed ResNet-DUC-HDC in certain scenarios.

Ross Girshick [4] proposed a Fast Region-based Convolutional Network method (Fast R-CNN) for object detection. Fast R-CNN builds on previous work to efficiently classify object proposals using deep convolutional networks. Compared to previous work, Fast R-CNN employs several innovations to improve training and testing speed while also increasing detection accuracy. Then Shaoqing Ren et al. [5] merged Region Proposal Network (RPN) and Fast R-CNN into a single network by sharing their convolutional features using the recently popular terminology of neural networks with 'attention' mechanisms. They were able to achieve state-of-the-art object detection accuracy on PASCAL VOC 2007, 2012, and MS COCO datasets with only 300 proposals per image. Further Kaiming He et al. [6] proposed an approach for efficient detection of objects in an image while simultaneously generating a high-quality segmentation mask for each instance. The method is called Mask R-CNN, it extends Faster R-CNN by adding a branch for predicting an object mask in parallel with the existing branch for bounding box recognition. Mask R-CNN outperforms all the existing, single-model entries on every task, including the COCO 2016 challenge winners. Waleed Abdulla [7] keeping [6] as baseline he has utilised Mask R-CNN for object detection and segmentation. His model generates bounding boxes and segmentation masks for each instance of an object in the image. It's based on Feature Pyramid Network (FPN) and ResNet101, trained on coco dataset [8]

Related Datasets:

Dodge et al. [9] have used FCN for segmentation of walls, Faster R-CNN for detection of rooms and Googles character recognition API to detect the size of the room. They have introduced R-FP [9], a dataset of the floor plan which labels wall segmentation. They have observed IOU metrics for the wall segmentation which concludes that models trained over one dataset give poor results when tested on other datasets. Liu et al. [10] converted bitmap images into vectors using a pipeline that consists of semantic segmentation, junction prediction, integer programming and vector format processing. They have used 870 self-labelled images, with a training-test split of 88.5% - 11.5%. Liu et al. have reported recall and accuracy metrics calculated from IOU for rooms and objects, and euclidean distances for the walls. This paper achieves a significant performance boost with a large corpus of floorplan vector data and popup 3D models, opening potentials for a new family of big-data Computer Vision research.

Kalervo et al. [11], used a similar pipeline as Liu et al. but they have presented a novel image dataset called CubiCasa5K, a large-scale floorplan image dataset containing 5000 samples annotated into over 80 floorplan object categories. They have reported overall and mean accuracy, mean IOU for the segmentation map, and accuracy and recall for the junctions, rooms and objects. Their implementations on the novel dataset have significantly improved the research on automatic floorplan image analysis as it presents a more elaborate means for examining the problem in a comprehensive manner.

Chen Liu et al. [12] have proposed FloorNet, a novel deep neural architecture for automatically reconstructing a floorplan in indoor space. This paper has set a benchmark for floorplan reconstruction by acquiring RGBD video streams for 155 residential houses or apartments and annotating complete floorplan information.

3. Methodology

Our solution offers an end to end flow for empty floor space optimization and recommendation system for balcony garden in 2D space. Figure 1 illustrates the pipeline for the same. The image uploaded is first passed for quality check, where if the standards are not met, the user is asked to upload the image again. The quality check approved image is then passed to custom trained semantic segmentation model to segment 6 classes namely wall, door, window, ceiling, railing and furniture.

The output from the semantic segmentation model is then used to calculate empty floor space in the image by our custom algorithm. The identified floor area is further segmented to restricted and non-restricted areas marking them as red and green zones respectively. Using the non-restricted floor area, the number of rows of plants to be placed is calculated. The plant images are seamlessly blended with the uploaded image background using our image blending module to give a beautiful balcony design recommendation. All these individual modules are packed together to create the end to end pipeline.

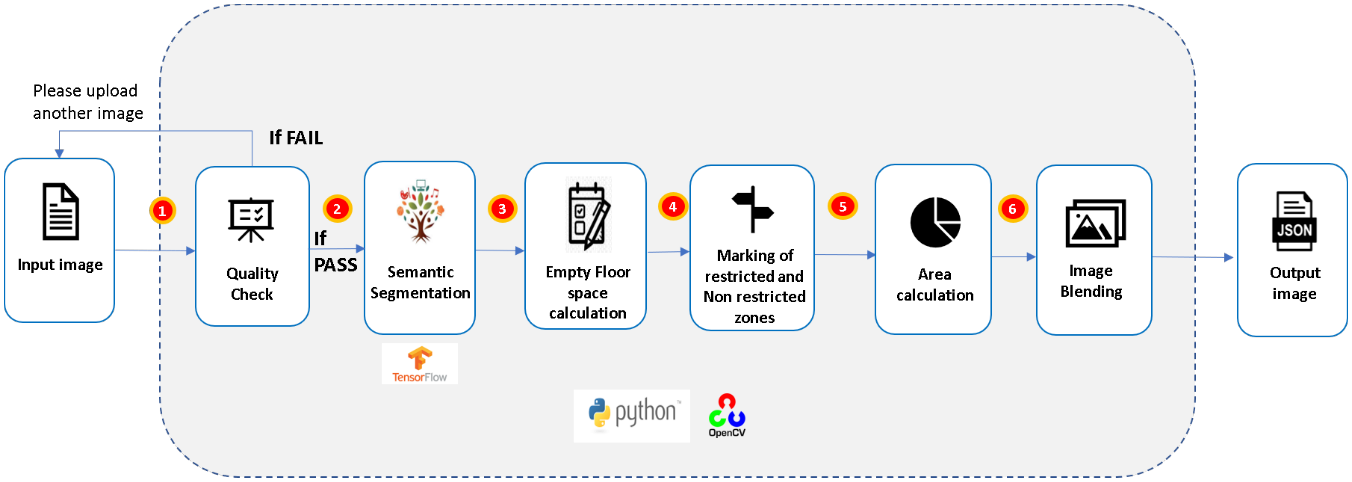

Figure 1: Floor space optimisation and recommendation system pipeline

3.1. Data Acquisition

As we proceed with our algorithm to extract empty floor space, we come across various indoor and outdoor space datasets that suit our purpose. Our research here is primarily focused on dataset consisting of different balcony images. The input images are in .jpg, .png and .jpeg format. The primary source of our dataset are Adobe Photoshop, internet and custom balcony images with varied angles, backgrounds and quality captured by us using different phone cameras as shown in Figure 7.

(a) (b) (c) (d)

Figure 2: Dataset sourced from Adobe photoshop, internet and custom.

The balcony dataset images are further annotated for six classes namely wall, door, window, ceiling, railing and furniture for segmentation. An open-source VIG annotation tool is used and JSON files are created for each class mentioned above.The entire dataset is split into 80% train, 10% test and 10% validation set.

3.2. Quality check

The input image is first passed to the quality check module that checks the quality of the input image to be processed further. The image quality is checked based on two parameters- brightness and blurriness.

For brightness we calculate the mean of all the pixels in each channel using the Pillow library. This returns the average pixel brightness of three channels. We use a threshold value and if the average is less than the threshold, then the image is less bright. Blurriness is related to the sharpness of an image and a blurry image does not have clear edges. So here, we have used the specific edge detection filter which is the Laplacian filter. It returns the average variance of the edges in an image. The higher the number, the sharper the edge is. We use a threshold value and if laplacian variance is less than the threshold, we can state that the image is blurry.

If the image passes the set threshold for both these parameters, it is passed on for further processing which is to go through the semantic segmentation module for empty floor space detection in the image. If the threshold parameters are not met and the standard quality check fails, the user is asked to upload a new image with better quality.

3.3. Semantic segmentation model

Once the image has passed the quality check module mentioned in section 3.2, it then goes through the semantic segmentation model to identify the six annotated object classes in the image. A semantic segmentation model [13] is trained on our custom data set mentioned in section 3.1. Semantic segmentation identifies objects present in an image at the pixel level and aids in precise localisation of the empty floor space area. There are four available architectures to perform semantic segmentation namely R-CNN, Fast R-CNN, Faster R-CNN and Mask R-CNN. Mask R-CNN is an extension to Faster R-CNN and detects multiple objects, objects of different scales, and overlapping objects in an image. This improves the speed and efficiency for object identification and that's the reason we have used Mask R-CNN pre-trained on coco dataset. [8].

Architecture and working of Mask R-CNN:

Mask R-CNN uses Resnet 101 architecture to extract features from the image which act as an input for the next layer. These features are then applied to region proposal network (RPN). Here, we get the regions which model predicts contain some object. These regions are then passed to fully connected network to predict class labels and bounding boxes. Additionally, Mask R-CNN generates masks for each class. For all predicted Regions of Interest (RoI), Intersection over Union (IoU) is computed. If IoU is greater than the set threshold (0.5), we consider that RoI, else neglect it. Once we have our ROIs, mask branch is added to the architecture. This returns segmentation mask of size 28 x 28 for each region that contains an object, which is then scaled up for inference.

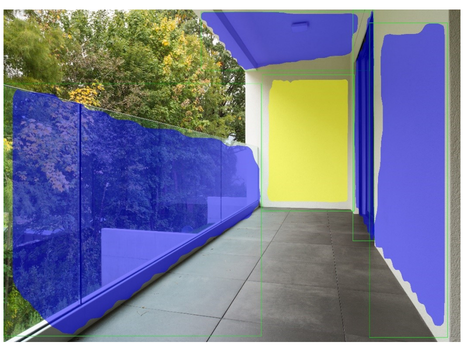

Figure 3: Semantic segmentation model output

The hyper-parameters are fine tuned and the model is trained with the parameters mentioned in table 1. Continuous learning is also integrated with this module. Therefore the model will keep on learning as new data is fed to the model over time. This will enhance the model performance and accuracy. With continuous learning, the semantic segmentation model can also be utilised for other applications like parking space identification, indoor design recommendation etc. The output of the semantic segmentation model (Mask R-CNN) gives the bounding box coordinates of the regions identified in the image. A custom algorithm is developed to determine the empty floor space in the image using the output of semantic segmentation model.

TABLE 1: Hyper-parameters used in model training

Epochs	Steps per epoch	Learning rate	Detection confidence
10000	1000	0.001	0.9

3.4. Empty floor space calculation

The output of the semantic segmentation model helps us identify the objects present in the input image as shown in the Figure 3. Using this, we try to identify the empty space present in the image. A custom logic is built to identify the four boundary points of the empty floor space using the coordinates of the identified objects.

These coordinates are then used to find the intersection points between them. For example, in the Figure 3, the bottom-right corner of railing is intersecting with bottom-left corner of the identified wall. After getting intersection points, distance between these points is calculated and a polygon is drawn as shown in the Figure 4 below. This polygon depicts the identified empty region.

Figure 4: Floor space mapping

3.5. Marking of restricted and non-restricted regions

Once the empty space is identified as explained in the section 3.4, we have to mark the specific region which can be actually utilized further. For this, the entire space is divided into restricted and non-restricted regions. These restricted and non restricted regions vary depending on the use case. We have demonstrated the same for our particular use case, efficient and aesthetic plant placement.

Restricted or Red zones are defined as those regions where no plants should be placed so that they do not block any pathway like in front of the door or where the furniture lays. To mark the restricted region (red zone), a proportion of empty space area in front of the door is identified and marked as red as shown in the Fig 5. Non-Restricted or green zones are defined as those regions where we can place our plants, like in front of railing, walls etc. Using this, a proportion of non-restricted empty space is identified in front of railings and walls and marked as green as shown in Figure 5.

3.6. Area calculation and utilisation for plant placement

The custom algorithm mentioned in section 3.4 determines the total empty floor space area. The logic presented

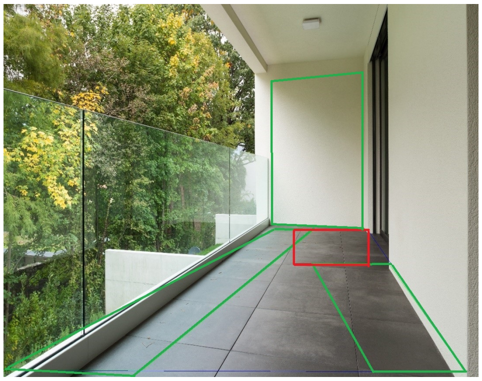

Figure 5: Red and green zone identification

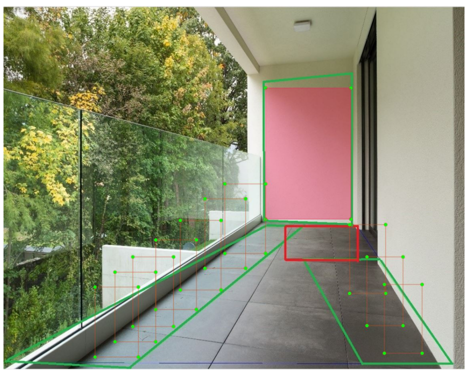

Figure 6: Bounding box creation in the Green Zone

in section 3.5 helps in the marking of red and green zones. In our algorithm, initially the area for all green zones is computed using built-in OpenCV function.

Any line can be represented as c = ax + by. Considering two points P and Q satisfy the given line equation, we can set the values so that this equation holds true. For identification of points in green zone shown in Figure 5 where plants should be placed, a line equation, 1 is written using the two vertices (P and Q) of identified rectangle as mentioned below:

$$a = Q[y] - P[y]$$

$$b = P[x] - Q[x]$$

$$c = a * (P[x]) + b * (P[y]) \tag{1}$$

where, P(x,y) and Q(x,y) are the two endpoints of the railing and a and b are variables of linear equation 1.

If b is less than 0, equation 2 represents the line passing through points P and Q else, equation 3 represents the same.

$$ax - by = c \tag{2}$$

$$ax + by = c \tag{3}$$

Using the above line equation, we get the bounding box coordinates for the first plant placement. To place the next bounding box we have kept spacing of 200 pixels. The space between two adjacent boxes is not considered while placing plants so that none of the images overlap. These steps are iterated till the bounding box coordinates exceed the green zone and thus, the total number of plants that can be placed in one row is calculated, refer Fig. 6.

Afterwards, plant images are selected from the repository and blended inside each bounding box. Thus, using area calculation, we determine an efficient and coherent way of utilising empty floor space area for the placement of plants.

3.7. Image Blending

Here, we are using the mapped empty floor space for placing plants and generating balcony garden design recommendation. We have a repository of plant images from which our algorithm randomly selects the foreground images to put in the identified empty regions. We can add the plant images directly to the empty regions using the coordinates from semantic segmentation model but then both the images would not be seamlessly blended as they have some background associated to them. To make the final image more authentic and real, foreground plant images are randomly picked and merged with the background using our image blending module purely based on OpenCV [14].

Blending images of varying sizes involves three major stages: selecting, masking and overlaying. To blend images of different sizes, the module first selects a Region Of Interest (RoI) in the background image which in our case is the identified empty region. The next step is to replace the values in that particular region with the values of the foreground image and then add them back in. However, here we plan to transfer a specific region and do not want the background from the foreground image. To achieve this we performed multiple masking and bit-wise operations.

The foreground image is first resized to fit the coordinates of the identified regions (RoI) detected in the background image. The resized image is gray scaled and thresholded to create a binary image (mask). In order to bring back the original colors, we apply image processing to inverse the mask using bit-wise operations. The final blended portion is then generated by adding our foreground and RoI, which is then overlaid over the original background image.

4. Result and Conclusion

Our approach offers an end to end solution for floor mapping and empty floor space detection which is a key requirement across many domains and multiple use cases. There are already implemented approaches available for floor mapping in the 3D space but there's no such implementation for 2D space. Because of the third dimension i.e. depth, present in 3D, the remoteness of objects in images is easily determined. This aids in accurate identification and segmentation of objects in the image in 3D space.

However, identifying objects in a 2D space is a bit challenging due to the lack of depth parameter. Here, we

(a) (b)

(c) (d)

Figure 7: Output image with 1 or 2 row of plant

only have two parameters to work around, and yet we are able to achieve strikingly accurate results. Using the segmentation approach, we are able to implement empty floor space identification efficiently for plant placement.

Test images are manually annotated for six classes (wall, door, window, ceiling, railing and furniture) to yield ground truth masks. The predicted masks we get from our model are compared with the ground truth. The performance metrics that we have used here is Intersection Over Union (IoU) which is one of the most efficient accuracy metrics applied for semantic segmentation based use cases. IoU is defined as the area of overlap between the predicted segmentation and the ground truth divided by the area of union between the predicted segmentation and the ground truth, as shown in Figure 10.

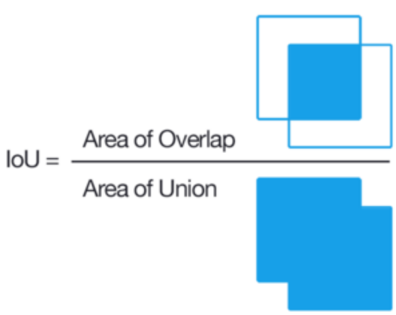

Figure 8: IoU calculation visualized. Source: Wikipedia

This metric ranges from 0–1 (0–100%) with 0 signifying no overlap and 1 signifying perfectly overlapping segmentation. The IoU that we are able to achieve on our dataset is 0.964 (96.4%) for a threshold of 0.5.

The train, test and validation accuracies achieved for our use case are as follows -

$$\text{Train Accuracy} - 98\%$$

$$\text{Test Accuracy} - 96\%$$

$$\text{Validation Accuracy} - 95\%$$

From the above results, we can see that our approach is capable of effectively identifying empty floor spaces and classifying them into restricted and non-restricted areas even when the images are taken from different camera angles.

For our future work, we plan to extend the proposed technique to other use cases like indoor design recommendations, autonomous robot navigation, empty parking space identification, and compare them with other state-of-the-art methods.

References

[1] P. Wang, P. Chen, Y. Yuan, D. Liu, Z. Huang, X. Hou, and G. Cottrell, "Understanding convolution for semantic segmentation," in *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2018, pp. 1451–1460.

[2] L. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," *CoRR*, vol. abs/1802.02611, 2018. [Online]. Available: http://arxiv.org/abs/1802.02611

[3] L. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking atrous convolution for semantic image segmentation," *CoRR*, vol. abs/1706.05587, 2017. [Online]. Available: http://arxiv.org/abs/1706.05587

[4] R. Girshick, "Fast r-cnn," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, December 2015.

[5] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," 2016.

[6] K. He, G. Gkioxari, P. Dollár, and R. B. Girshick, "Mask R-CNN," *CoRR*, vol. abs/1703.06870, 2017. [Online]. Available: http://arxiv.org/abs/1703.06870

[7] W. Abdulla, "Mask r-cnn for object detection and instance segmentation on keras and tensorflow," https://github.com/matterport/Mask_RCNN, 2017.

[8] T. Lin, M. Maire, S. J. Belongie, L. D. Bourdev, R. B. Girshick, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: common objects in context," *CoRR*, vol. abs/1405.0312, 2014. [Online]. Available: http://arxiv.org/abs/1405.0312

[9] R. I. of Technology, "Floor plan from rakuten real estate and pixel-wise wall label," in *https://rit.rakuten.co.jp/data release/,*, 2019-04-23.

[10] C. Liu, J. Wu, P. Kohli, and Y. Furukawa, "Raster-to-vector: Revisiting floorplan transformation," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[11] A. Kalervo, J. Ylioinas, M. Häikiö, A. Karhu, and J. Kannala, "Cubicasa5k: A dataset and an improved multi-task model for floorplan image analysis," *CoRR*, vol. abs/1904.01920, 2019. [Online]. Available: http://arxiv.org/abs/1904.01920

[12] C. Liu, J. Wu, and Y. Furukawa, "Floornet: A unified framework for floorplan reconstruction from 3d scans," *CoRR*, vol. abs/1804.00090, 2018. [Online]. Available: http://arxiv.org/abs/1804.00090

[13] K. He, G. Gkioxari, P. Dollar, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[14] I. Culjak, D. Abram, T. Pribanic, H. Dzapo, and M. Cifrek, "A brief introduction to opencv," in *2012 Proceedings of the 35th International Convention MIPRO*, 2012, pp. 1725–1730.