Supporting information for

Computer vision for recognition of materials and vessels in chemistry lab settings and the Vector-LabPics dataset

Sagi Eppel[*,#,1,2], Haoping Xu[#,2,3], Mor Bismuth[4], Alan Aspuru-Guzik[*,1,2,3,5]

[1]Department of Chemistry, [2]Vector Institute, [3]Department of Computer Science

[5]CIFAR Lebovic Fellow, [1,2,3,5]University of Toronto

[4]Department of Cognitive science, Open University of Israel

sagieppel@gmail.com, haoping.xu@mail.utoronto.ca, morbismut@gmail.com, alan@aspuru.com

Table Of Content

* Corresponding authors # Equal contribution

Resources Available for download

Supporting information includes additional details on the nets training, code, trained models, and videos of nets results.
The Vector-LabPics dataset available in these URLs:
https://www.cs.toronto.edu/chemselfies/
https://zenodo.org/record/3697452
The codes and trained models for all the nets used for the work available from these repositories:
https://github.com/aspuru-guzik-group/Computer-vision-for-the-chemistry-lab
https://zenodo.org/record/3697767
Videos of chemical processes that were annotated by the nets are available as supporting material and on youtube.

Training Mask-RCNN

For the Mask R-CNN network[1] backbone, we used a ResNet50 [2] FPN model pretrained on the COCO dataset. The RPN, box head, mask head, and subclass predictor are all trained from scratch using the Vector-LabPics dataset. We use two slightly different separations of superclasses to train two different models. The first model only separates instances into vessels and materials superclasses based on their subclass label. The other model has a more fine-grain separation, where instances are classified into three superclasses, vessel, liquid, and solid. Additionally, only instances under material superclass are considered for subclass classification. For both cases, SGD optimizer is used with an initial learning rate of 0.0025, a momentum of 0.9, and a weight decay of 0.0001. Additionally, the learning rate is stepped down by a factor of 10 for every 10 epochs. Both models were trained using a single Nvidia P100 for 200 epochs.

Unified GES net architecture

The unified GES net[4] (Figure 1c) is built on a ResNet101 encoder[2] that receives the image, a pointer point, and an ROI mask. Both the pointer point and ROI mask are represented as binary masks (Figure 1c) in which the value of the pixels corresponding to the pointer point or ROI have a value of one while the rest of the pixels have a value of zero. The ROI and Pointer map was processed using a single convolutional layer before being merged with the feature map of the first layer of the encoder, using element-wise multiplication and addition, respectively (Figure 1c) this was done similar to previous works.[4,5] The final convolutional layer of the Resnet is connected to three different heads: the segmentation head consists of a standard PSP[3] layer followed by three upsampling layers connected to two skip connections (from the Resnet layers 3,4). The classification head is similar to the ResNet

standard classification head, it follows the ResNet final convolutional layer, and involves two fully connected layers. The evaluation head is similar to the classification head but with a single output channel corresponding to the predicted segment IOU. All these heads were similar to those used in GES net[4].

Training the GES net

The training was done by picking random instance masks from the dataset. For each instance segment, a single random point was selected inside the mask and used as an input to the net. The ROI for the material prediction net (Figure 1a) was chosen as the region of the vessel containing the material. The ROI for the vessel GES net[4] (Figure 1b) was chosen as the sum of the regions of all the vessels in the image. For both nets, the ROI was chosen as covering the entire image in 60% of the train iterations. The segment region was predicted as a binary mask, and the loss was the standard cross-entropy with the GT segment. The evaluator head output predicts the segment score as a single number, and the loss is the square of the difference between the predicted IOU and the real IOU. The classification head predicts the probability for each class as a binary softmax prediction. The loss is the standard cross-entropy, averaged for all the classes with equal weight. The nets backbone is a pointer net pre-trained on the COCO panoptic dataset.[4,5] Due to the small size of the dataset, significant augmentation was used, including deforming, noise adding, cropping, and decoloring. Each of the nets was trained on a single Titan XP GPU for about 200 epochs. The training was done using, and Adam optimizer with a decreasing learning rate (from 1e-5 to 1-6), the learning rate was decreased by 10% every 10,000 steps. When reached 1e-6, the learning rate was increased to the previous max minus 10%. Hence, in the first cycle, the learning rate starts at 1e-5 and decreases to 1e-6. In the second cycle, the learning rate started at 9e-6 and decreased to 1e-6. In the third cycle, the learning rate started at 8e-6 and decreased to 1e-6, and so forth. This approach combines the rigidity of the cyclic learning rate with the convergence of the decaying learning rate approach.

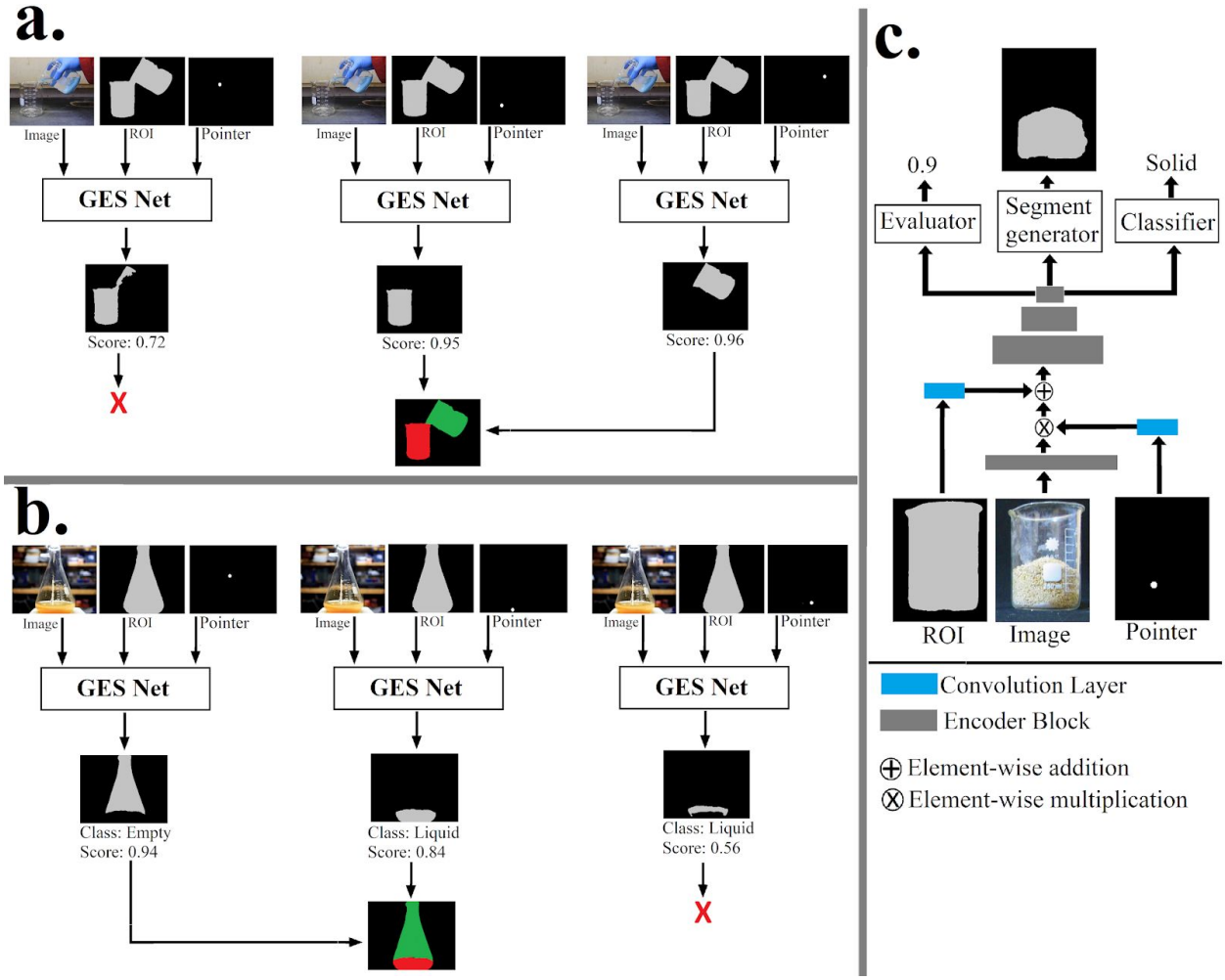

Figure 1: (a,b) GES net for vessel and material instance segmentation. The net receives an image, an ROI mask, and a pointer point in the image, and outputs the mask instance containing the point within the ROI, for (a) the vessel and (b) the material. The net also outputs the confidence score, which is an estimation of how well the predicted region matches the real region in terms of the IOU. In the case of the material (b), the net also predicts the material class. The predictions with the highest scores are merged into the final segmentation map (a,b). (c) Unified GES net architecture.

Training Fully convolutional net PSP for semantic segmentation.

Semantic segmentation was applied using a fully convolutional neural net with PSP architecture[3] and Resnet101 Encoder.[2] This was done similar to previous works.[3] The final convolutional layer of the Resnet101 is connected to a standard PSP[3] layer followed by three upsampling layers connected to 2 skip connections (from the Resnet layers 3,4). The multi-class prediction was achieved by predicting an independent binary map (Two-channels map) for each class. For each pixel, this map predicts whether or

not it belongs to the specific class. Creating this map is done by simply splitting the final layer of the FCN (which has a depth of 2N layers) into N binary two-channels maps (where N is the number of classes). The training loss for this net was the sum of the losses of all of the class predictions. The training learning rate, solver, augmentation was done the same as in the GES net training (Above section). The training was done on a single Nvidia Titanx XP for 200 Epochs.

1. Reference

[1] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. *Mask r-cnn*. In Proceedings of the IEEE international conference on computer vision, pages 2961–2969, 2017.

[2] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. *Deep residual learning for image recognition.* In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 770–778, 2016.

[3] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. *Pyramid scene parsing network.* In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 2881–2890, 2017.

[4] Sagi Eppel and Alan Aspuru-Guzik. *Generator evaluator-selector net: a modular approach for panoptic segmentation*. arXiv preprint arXiv:1908.09108, 2019.

[5] Eppel, Sagi. *"Class-independent sequential full image segmentation, using a convolutional net that finds a segment within an attention region, given a pointer pixel within this segment." arXiv preprint arXiv:1902.07810 (2019).*