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]

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

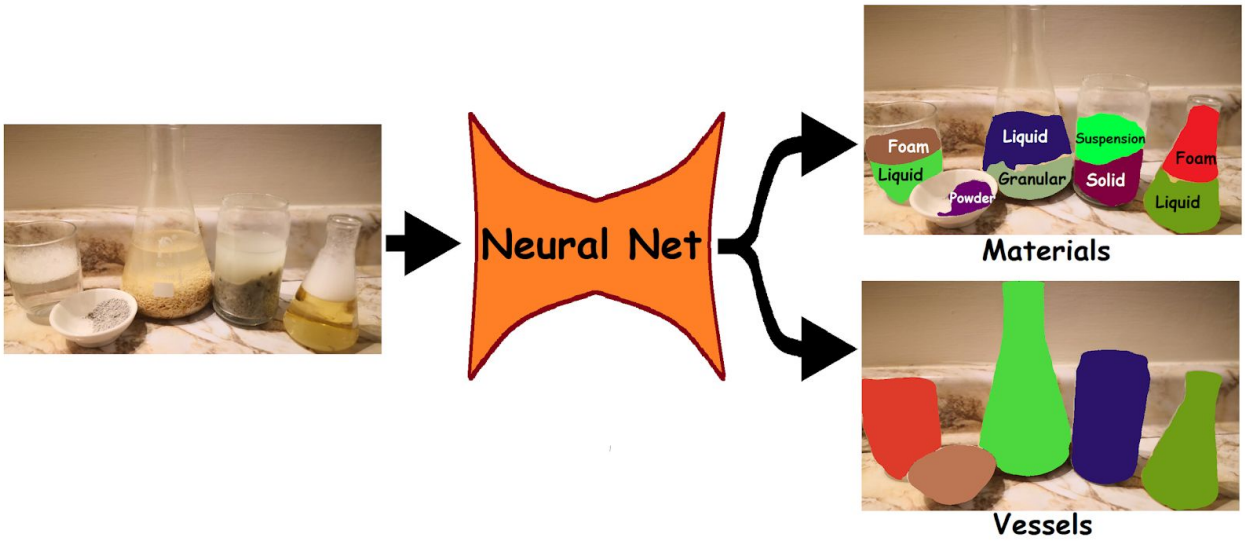
[1]Department of Chemistry, [2]Vector Institute, [3]Department of Computer Science

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

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

Abstract

This work presents a machine learning approach for computer vision-based recognition of materials inside vessels in the chemistry lab and other settings. In addition, we release a dataset associated with the training of the model for further model development. The task to learn is finding the region, boundaries, and category for each material phase and vessel in an image. Handling materials inside mostly transparent containers is the main activity performed by human and robotic chemists in the laboratory. Visual recognition of vessels and their content is essential for performing this task. Modern machine vision methods learn recognition tasks by using datasets containing a large number of annotated images. This work presents the Vector-LabPics dataset, which consists of 2187 images of materials within mostly transparent vessels in a chemistry lab and other general settings. The images are annotated for both the vessels and the individual material phases inside them, and each instance is assigned one or more classes (liquid, solid, foam, suspension, powder, …). The fill level, labels, corks, and parts of the vessel are also annotated. Several convolutional nets for semantic and instance segmentation were trained on this dataset.[@] The trained neural networks achieved good accuracy in detecting and segmenting vessels and material phases, and in classifying liquids and solids, but relatively low accuracy in segmenting multiphase systems such as phase-separating liquids.



* Equal Contributions. # Corresponding author. @ The dataset and models used for this work are available here.

Introduction

Experimental chemistry consists largely of the handling of materials in vessels[1]. Whether it involves moving and mixing liquids, dissolving or precipitating solids, or extraction and distillation, these manipulations almost always consist of handling materials within transparent containers and depend heavily on visual recognition. For chemists in the lab, it is crucial not only to be able to identify the vessel and the fill level of the material inside it but also be able to accurately identify the region and phase boundaries of each individual material phase as well as its type (liquid, solid, foam, suspension, powder, etc.). For example, when a chemist is trying to create a reaction in a solution, it is important to ensure that all materials have been fully dissolved into a single liquid phase. A chemist attempting to separate the components of a mixture will often use phase separation for liquid-liquid extraction or selective precipitation; these and many other tasks depend heavily on the visual recognition of materials in vessels.[2-6] Creating a machine-vision system that can achieve this is essential for developing robotic lab systems that can perform the full range of operations used for chemical synthesis.[7-14] The main challenge in creating an image recognition system that can achieve this is that material phases can have a wide range of textures and shapes, and these may vary significantly even for the same type of material. Classical computer vision algorithms have hitherto mostly relied on edges or colors in order to identify objects and materials.[15,16] While these methods can achieve good results in simple conditions and controlled environments, they fail in complex real-world scenarios.[17-23] In recent years, convolutional nets (CNN) has revolutionized the field of computer vision, leading to the development of a wide range of new applications from self-driving cars to medical imaging.[24] When CNN's are trained with large numbers of examples of a specific task, they can achieve almost human-level recognition of objects and scenes under challenging conditions.[25] Training such methods effectively requires a large number of annotated examples.[26,27] For our purpose, this means a large number of images of materials in vessels, where the region and the type of each individual material and vessel are annotated.[28] This work presents a new dataset dedicated to materials and vessels with a focus on chemistry lab experiments. The dataset, called Vector-LabPics, contains 2187 images of chemical experiments with materials within mostly transparent vessels in various laboratory settings and in everyday conditions such as beverage handling. Each image in the dataset has an annotation of the region of each material phase and its type. In addition, the region of each vessel and its labels, parts, and corks are also marked. Three different neural nets were trained on this task: a Mask R-CNN[29] and a generator-evaluator-selector (GES) net[30] were trained on a task requiring instance-aware segmentation,[31] which involved finding the region and boundaries of each material phase and vessel in the image, while a fully convolutional neural net (FCN)[32] was trained for semantic segmentation, which involved splitting the image into regions based on their class.[33]

The Vector-LabPics dataset

Creating a large annotated dataset is a crucial part of training a deep neural net for a specific task. For deep learning applications, large datasets dedicated to specific tasks such as ImageNet[26] and COCO[31] act as a basis on which all methods in a given field are trained and evaluated. An important aspect of the dataset for image recognition is the diversity of the images, which should reflect as many different scenarios as possible. The more diverse the dataset, the more likely it is that a neural net trained on this dataset will be able to recognize new scenarios that were not part of the dataset. The goal for the

Vector-LabPics dataset and the method described here is to be able to work under a wide range of conditions as possible. Some of the most important sources of images for this dataset include YouTube, Instagram, and Twitter channels dedicated to chemistry experiments. The list of contributors that enable this work is given in the acknowledgment section. Another source is images taken by the authors in various everyday settings. In total, the dataset contains 2187 annotated images. The annotation was done manually using the VGG image annotator (VIA).[34] Each individual vessel and material phase were annotated, as were the labels, corks, and other parts of the vessels (valves, etc..). Each instance segment received one or more classes from those shown in Table 1. The dataset has two representations. The non-exclusive presentation is based on overlapping instances (Figure 1a). In this mode, the different segments can overlap; for example, when a solid phase is immersed in a liquid phase, the solid and liquid phases will overlap (Figure 1a). In a case of overlap, a priority (front/back) was added to each segment in the overlap region. For example, if a solid is immersed in a liquid, priority will be given to the solid (Figure 1a).

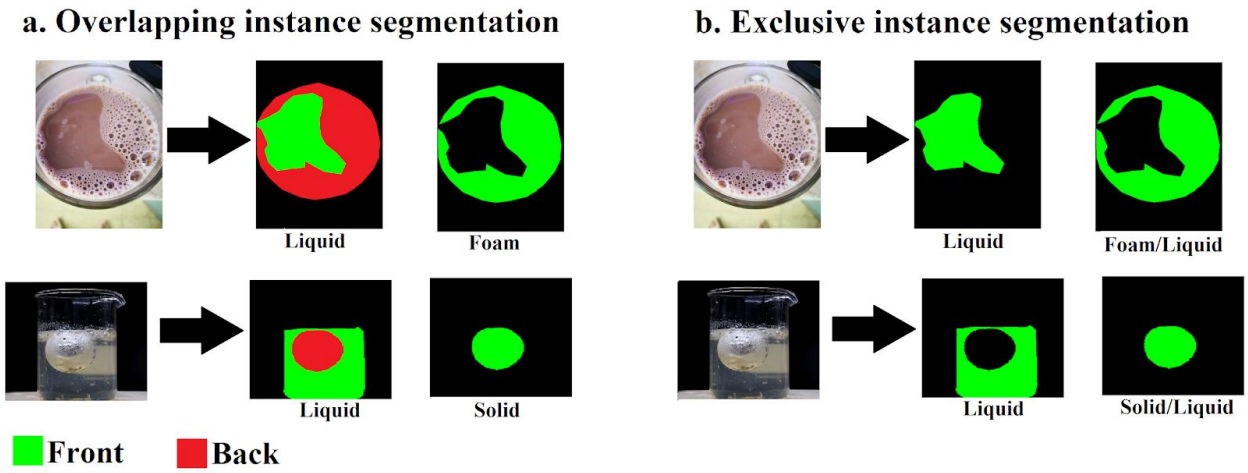


Figure 1: (a) Instance segmentation with overlapping segments. In the case of overlap, the overlapping region is marked as either the front (green) or back (red); (b) non-overlapping (simple) instance segmentation: each pixel can belong to only one segment, and only the front region of the segment is used. Each segment can have several classes.

The dataset also contains a simple version with non-overlapping instances. In this case, each pixel can correspond to only one vessel instance and one material instance. If there is overlap, the front instance with higher priority is used, and the back instance is ignored, in the overlapping region (Figure 1b). In addition, the pixel can be assigned one label/part instance (Figure 2). Altogether, the simple representation has three channels: (i) the vessel instance; (ii) the material instance; (iii) the label/cork/vessel part instance. Instances from the same channel cannot overlap; this means that two material instances may not overlap, but that the material and vessel instance may overlap. Another approach is a semantic representation in which each pixel is assigned several classes but no instance (Figure 3). More accurately, each class has a binary map containing all the pixels in the image belonging to this class. This semantic representation is not instance-aware and does not allow us to separate different instances of the same class, such as adjacent vessels or phase-separating liquids (Figure 3).

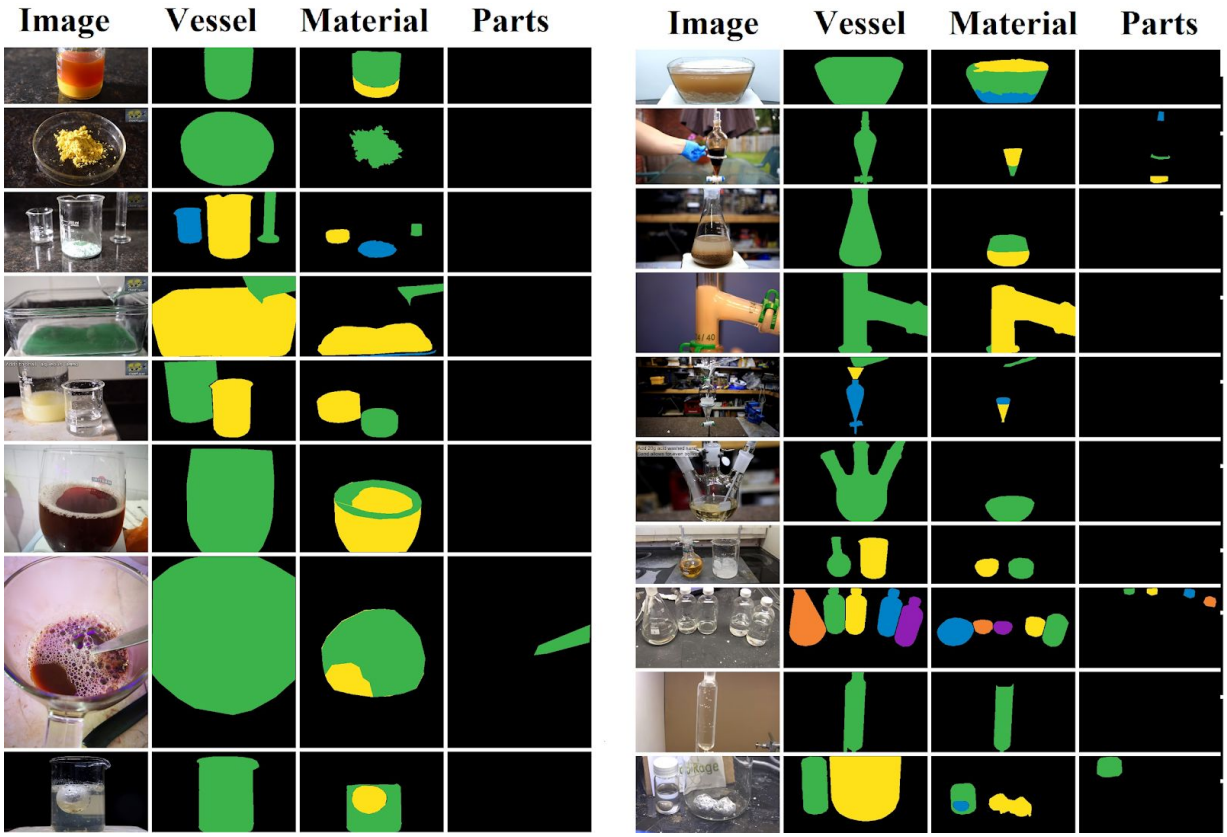


Figure 2: Exclusive instance segmentation map from the Vector-LabPics dataset. The segmentation is composed of three channels: the vessel, the material phases, and the vessel parts. Segments from the same channel cannot overlap.

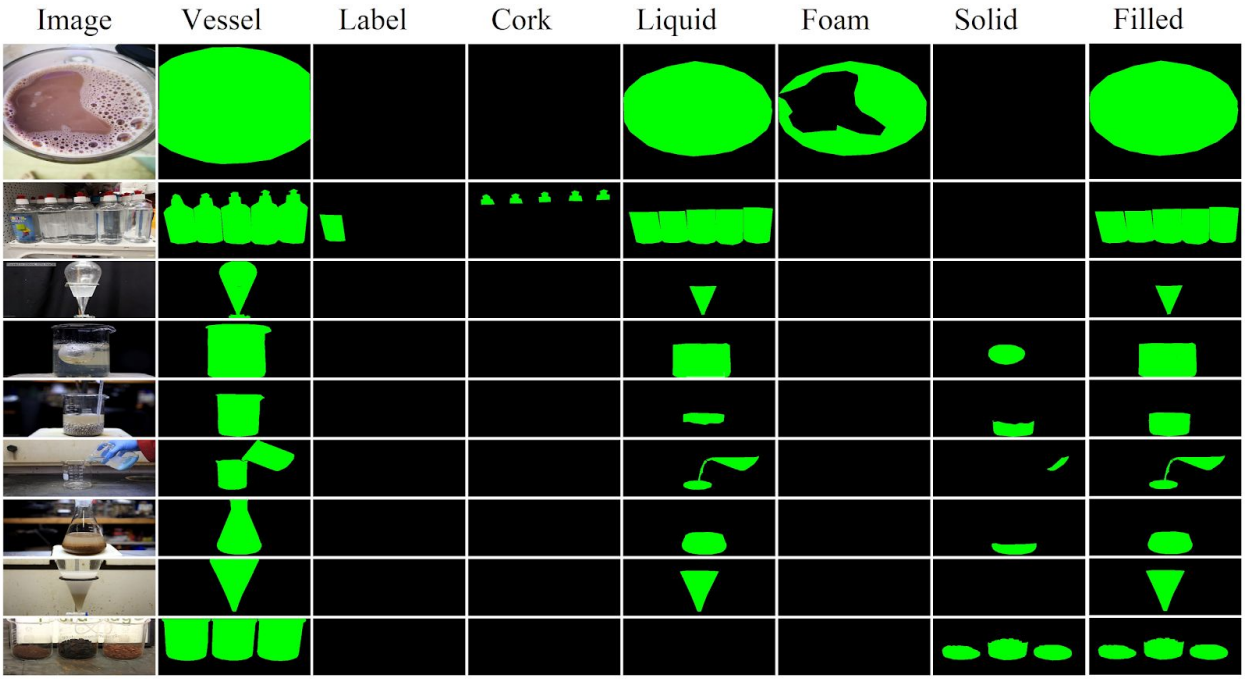


Figure 3: Examples of semantic segmentation maps from the Vector-LabPics dataset. Each class has a binary segmentation map that covers all the pixels belonging to the class. Not all classes are shown.

Results and discussion

Semantic and instance segmentation

Finding the region and class of each vessel and material phase in the image can be done using either semantic or instance segmentation. Instance-aware segmentation involves splitting the image into regions corresponding to different objects or material phases.[31] This method can detect and separate phases of materials of the same class, such as phase-separating liquids or adjacent vessels (Figure 2). In addition, the segmentation and classification stages in this method can be separated, allowing for the segmentation of unfamiliar materials (i.e., materials classes not in the training set). Two types of convolutional nets were studied for instance-aware segmentation: a Mask R-CNN[29] and a GES net.[30] Mask R-CNN is the leading method for instance-segmentation according to almost all the major benchmarks.[31] GES net is another method for both instance and panoptic segmentation and is designed to work in a hierarchical manner. Another approach is semantic segmentation,[32] which predicts for each class a binary map of the region in the image corresponding to that class (Figure 3). The main limitation of this class-based segmentation method is that it cannot separate different phases or object instances from the same class, such as phase-separating liquids or adjacent vessels (Figure 3). In addition, if the class of the material is not clear, or if it did not appear in the training set, the net will not be able to segment the material region. The only advantage of the semantic segmentation approach is that neural nets for such tasks are much easier and faster to train and run. For this task, we use the standard fully convolutional neural net (FCN) using the pyramid scene parsing (PSP) architecture.[35]

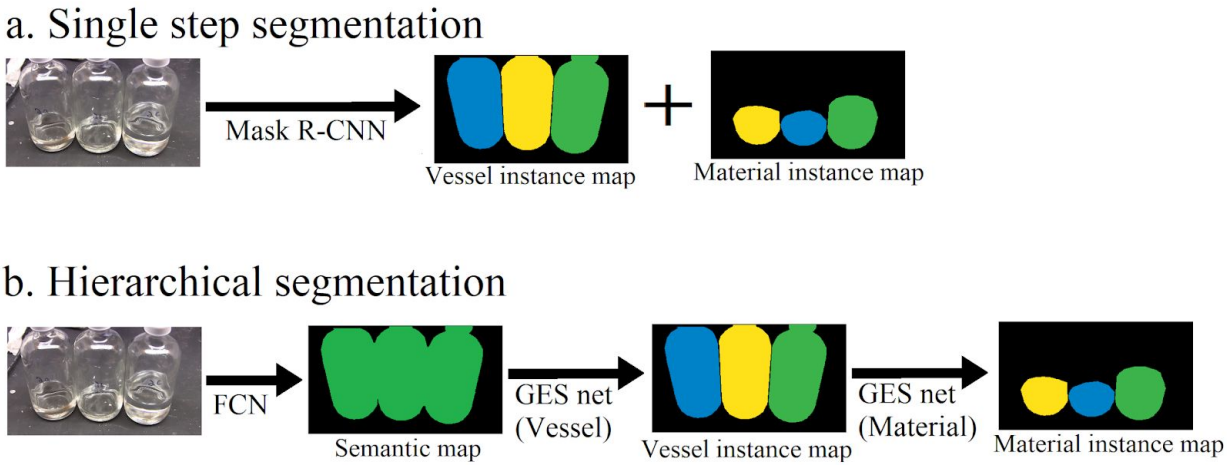


Figure 4: (a) Single-step segmentation using Mask R-CNN to find both the vessel and material instances simultaneously; (b) Hierarchical segmentation, in which an FCN finds the general region of the vessels in the image and this region is then transferred to a GES net for vessel instance detection. Each vessel instance segment is transferred to a second GES net for the segmentation of the material instance.

Hierarchical versus single-step segmentation

The problem of recognition of materials in the vessel can be solved either by finding the vessel and materials in a single step (Figure 4a) or hierarchically, by first finding the vessel using one system and then the materials inside the vessel using a second system (Figure 4b).[2,19,28,36] The single-step approach

was applied using Mask R-CNN and FCN, which traces both vessels and materials simultaneously (Figure 4a). The alternative hierarchical image segmentation approach involves three steps (Figure 4b): (1) finding the general region of all vessels using FCN for semantic segmentation; (2) splitting the vessel region (found in Step 1) into individual vessel instances using a GES net for vessel instance segmentation; and (3) splitting each vessel region (found in Step 2) into specific material phases using another GES net for material instance segmentation (Figure 4b).

Evaluation metrics

In this work, we employed two standard metrics for evaluating segmentation quality. The intersection over union (IOU) is the main metric for the evaluation of semantic segmentation and is calculated separately for each class.[32] The intersection is the sum of the pixels that belong to the class, according to both the net prediction and the ground truth (GT), while the union is the sum of pixels that belong to the class according to either the net prediction or the GT. IOU is the intersection divided by the union. The recall is the union divided by the sum of all pixels belonging to the class according to the GT annotation. Precision is the union divided by the sum of all pixels belonging to the class according to the net prediction. For instance-aware segmentation, we choose to use the standard metric of *Panoptic quality* (*PQ*).[37] *PQ* consists of a combination of recognition quality (*RQ*) and segmentation quality (*SQ*), where a segment is defined as the region of each individual object instance in the image. *RQ* is used to measure the detection rate of instances and is given by $RQ = \frac{TP}{TP+(FP+FN)\times 0.5}$, where *TP* (true positive) is the number of predicted segments that match a ground truth segment; *FN* (false negative) is the number of segments in the ground truth annotation that do not match any of the predicted segments, and *FP* (false positive) is the number of predicted segments with no matched segment in the *GT* annotation. Matching is defined as an IOU of 50% or more between predicted and ground truth segments of the same class. *SQ* is simply the average IOU of matching segments. PQ is calculated as $PQ = RQ \times SQ$.

Class-agnostic PQ metric

The Standard *PQ* metric is calculated by considering only those segments that were correctly classified. This means that if a predicted segment overlaps with a ground truth segment but has a different class, it will be considered mismatched. The problem with this approach is that it does not measure the accuracy of segmentation without classification; a net that predicts the segment region perfectly but with the wrong class will have a *PQ* value of zero. One method to overcome this problem is to pretend that all segments have the same class, in this case, the *PQ* quality will depend only on the region of the predicted segment. However, given the class imbalance, this will increase the weight of the more common classes, and will not accurately measure the segmentation accuracy across all classes. To measure class-agnostic segmentation in a way that will equally represent different classes, we use a modified *PQ* metric. The *PQ*, *RQ*, and *SQ* values for the class-agnostic method are calculated as in the standard case, while the definitions of *TP, FP*, and *FN* are modified. The *TP* for a given class is the number of GT instances of this class that match predicted instances with $IOU>0.5$ (regardless of the predicted instance class). The *FN* for a given class is the number of GT instances of this class that does not match any predicted segment (regardless of the predicted segment class). If an instance has more than one class, it will be counted for each class separately. The *FP* for a given class is the fraction of GT segments that belong to this class multiple by the total number of class agnostic *FP* segments. The total number of class agnostic *FP* (false positive) is the number of predicted segments that do not match any ground truth segments

regardless of class (matching means *IOU*>0.5 between segments regardless of class). For example, if 20% of the GT instances belong to the solid class, and there are 1200 predicted segments that do not match any GT segments, the *FP* for the solid class would be: $1200 \times 0.2 = 240$. In other words, to avoid using the predicted class for the *FP* calculation, we split the total *FP* among all classes according to the class ratio in the GT annotation.

Semantic segmentation results

The results of the semantic segmentation net are shown in Table 1 and Figure 5. It can be seen that the net achieved good accuracy *(IOU* > 0.8, Table 1) for segmentation of the vessel region, fill region, and liquid regions in the image, and a medium accuracy for solid segmentation (*IOU*=0.65). As can be seen from Figure 5, these results are consistent across a wide range of materials, vessels, angles, and environments, suggesting that the net was able to achieve a high level of generalization when learning to recognize these classes. For the remaining subclasses, the net achieved low accuracy (*IOU*<0.5, Table 1). The more common subclasses, such as suspension, foam, and powder, were recognized in some cases, while the more rare subclasses (gel, vapor) were completely ignored (Table 1). It should be noted that some of these subclasses have very few occurrences in the evaluation set, meaning that their statistics are unreliable. However, the low detection accuracy is consistent across all of the subclasses. This can be attributed to the small number of training examples for these subclasses, as well as the high visual similarity between different subclasses.

Instance segmentation results

The results of instance-aware segmentation are shown in Table 2 and Figure 6. It can be seen from Table 2 that the nets achieve good performance in terms of recognition and segmentation for most types of materials in the class-agnostic case (*PQ*>0.5). This is true even for relatively rare classes such as vapor and granular phases, implying that the net is able to generalize the recognition and segmentation process such that it does not depend on the specific type of material. The nets achieve low performance in the recognition of foams and chunks of solid, which usually contain small instances with wide variability in terms of shape. For class-dependent *PQ*, the main classes of vessels, liquids, and solids were detected and classified with fair accuracy (*PQ*>0.4, Table 2). However, almost all subclasses gave low *PQ* values; the only subclass that was classified with reasonable quality was Suspension, which had a relatively large number of training examples (Table 2). For multiphase systems containing one or more separate phases of materials in the same vessel, the quality of recognition was significantly lower than that of one-phase systems (Table 2). In multiphase systems, there is a tendency to miss one of the phases: for a solid immersed in liquid, the tendency is to miss the solid (Figure 6). One reason for this is that the phase boundaries between materials and air tend to be easier to see than those between liquids and materials. Another reason is that instances in multiphase systems tend to be smaller than instances in single-phase systems. The quality of recognition strongly depends on the segment size, and the larger the segment, the higher the quality (Table 2). This is true for both single-phase and multiphase systems (Table 2). It can also be seen from Table 2 that the hierarchical segmentation approach (using the GES net) gave better results than single-step segmentation using Mask-RCNN, although this was at the cost of a much longer running time of around three seconds per image compared to 0.2 seconds for the single-step approach

Results on videos: The nets were demonstrated by running them on videos containing processes like phase separation, precipitation, freezing, melting, and foaming. The annotated videos are available as

supporting materials. It can be seen from these videos that the nets can detect processes like precipitation freezing and melting by detecting the appearance of new phases like suspension, solids, and liquids. Also, processes such as phase separation and pouring can be detected by detecting the new phases and the change in the liquid level.

Conclusion

In this paper, we introduce a set of new computer vision methods tailored to chemical matter and the Vector-LabPics dataset. These were designed for the recognition and segmentation of materials and vessels in images, with an emphasis on a chemistry lab environment. Several convolutional neural nets were trained on this dataset. The nets achieve good accuracy for the segmentation and classification of vessels as well as liquid and solid materials in a wide range of systems. However, the nets ability to classify materials into more fine-grained material subclasses such as suspension, powder, and foam was relatively low. In addition, the segmentation of materials in multiphase systems, such as phase-separating liquids, had limited accuracy. The major limitation on increasing the accuracy of the net is the relatively small size of the dataset. Major datasets for image segmentation such as COCO[31] and Mapillary consist of tens of thousands of images, while the Vector-LabPics dataset contains only 2,197 images thus far. We also prioritized the creation of a general system that operates under a broad range of conditions over a system with higher accuracy that works only under a narrow set of conditions. It is clear that in order to achieve high accuracy under general conditions, the size of the dataset needs to be significantly increased. Alternatively, it is well established that a net that gives medium accuracy in a general setting can achieve a high level of accuracy under specific conditions by fine-tuning it on a small set of images containing these conditions. To conclude, image recognition of vessels and material phases is an essential part of chemistry laboratory work. Developing a machine vision system that can achieve this is likely to play an important role in automating lab systems. While the accuracy of our system is still below what is required for a truly autonomous lab, this work demonstrates many of the key image-recognition aspects needed for such a system. Increasing the dataset size and the system accuracy, as well as integrating it with robotic systems, is our next goal.

Table 1: Results for the semantic segmentation net

Class	IOU	Precision	Recall	N Eval[1]	N Train[2]
Vessel	**0.93**	0.96	0.97	497	1669
Filled	**0.85**	0.92	0.92	497	1660
Liquid	**0.81**	0.89	0.90	452	1419
Solid	**0.65**	0.82	0.75	108	512
Suspension	0.46	0.68	0.59	132	519
Foam	0.26	0.47	0.37	31	283
Powder	0.18	0.28	0.35	46	269
Granular	0.26	0.72	0.29	21	74
Gel	0.00	0.00	0.00	1	49
Vapor	0.00	0.00	0.00	4	29
Large chunks (solid)	0.00	0.00	0.00	7	38
Cork	0.20	0.33	0.35	15	329
Label	0.07	0.09	0.33	12	227
Vessel parts	0.15	0.23	0.31	112	536

1. Number of images, in the evaluation set, that contain the class.
2. Number of training images that contain the class.

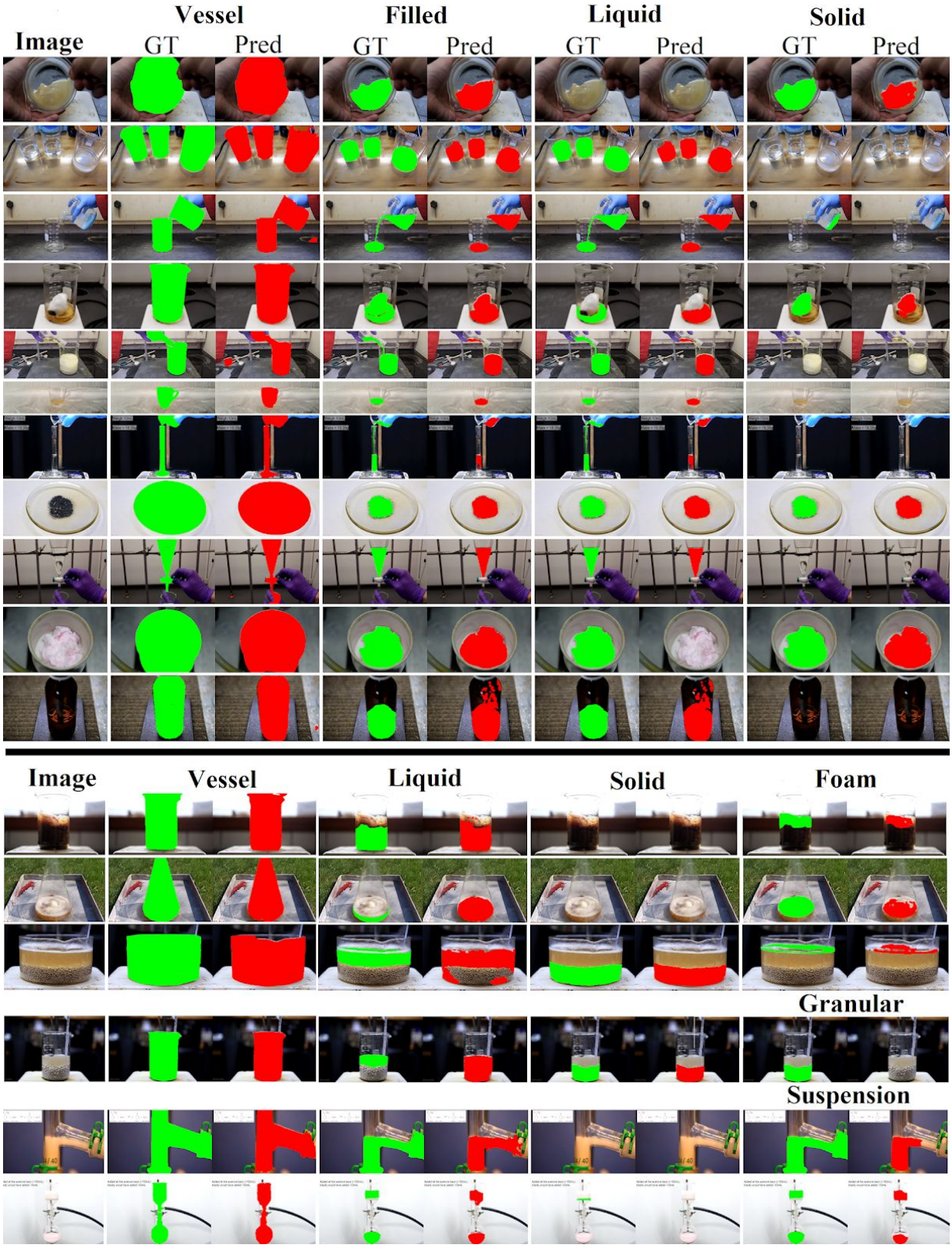


Figure 5: Results of semantic segmentation. The prediction (Pred) results are marked in red while the ground truth (GT) annotations are marked in green. Not all classes are shown. Images were taken from the Nile Red YouTube channel.

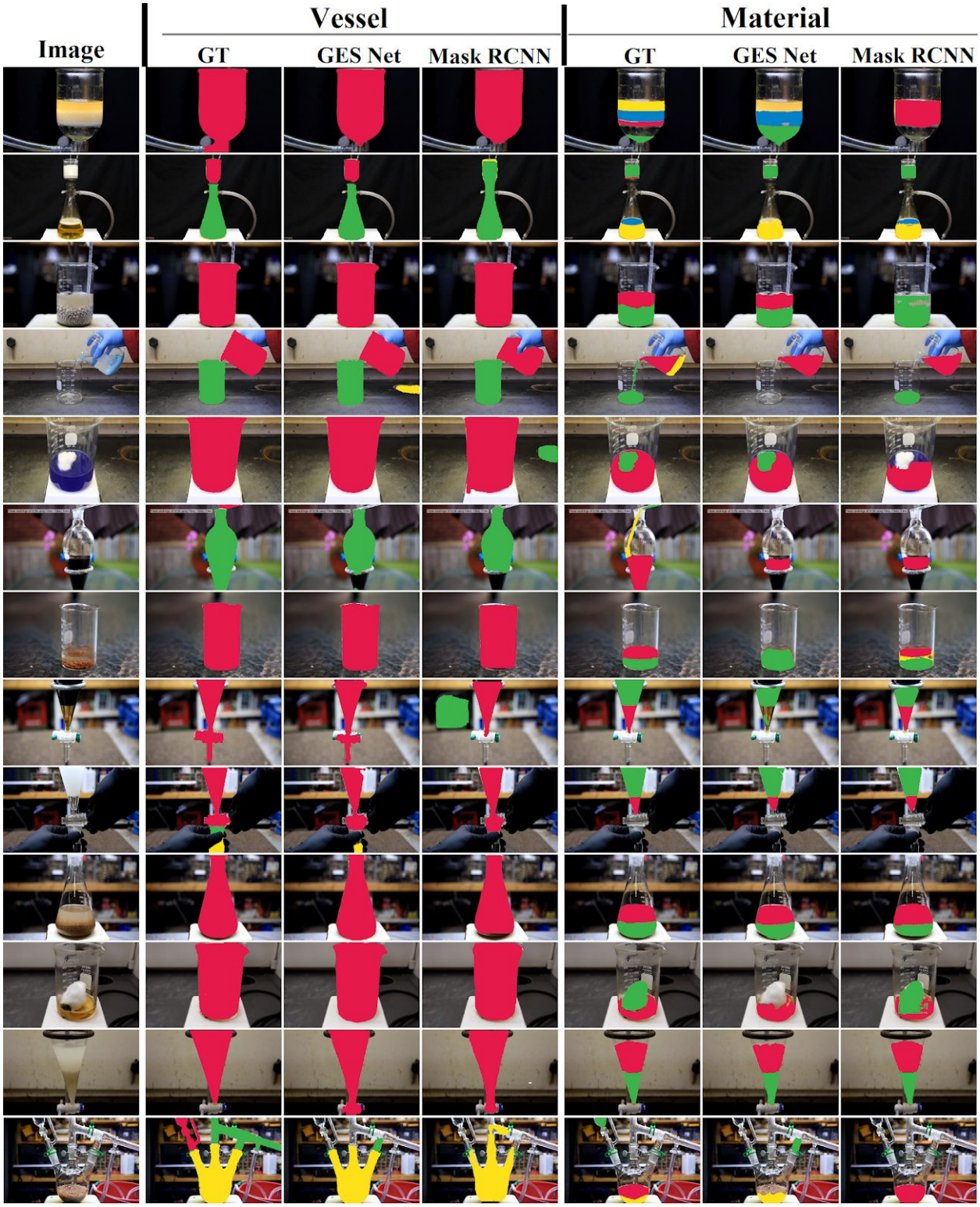


Figure 6: Ground truth and predicted results for instance segmentation.

Table 2: Instance-segmentation per class

	Hierarchical segmentation (GES net)						Single step segmentation (Mask RCNN)						Number of instances with class	
	Class agnostic[1]			Class dependent[2]			Class agnostic[1]			Class dependent[2]				
Class	PQ	RQ	SQ	PQ	RQ	SQ	PQ	RQ	SQ	PQ	RQ	SQ	Test set	Train set
Vessel	84	93	91	84	93	91	76	85	89	76	85	89	629	2696
Liquid	56	69	81	54	67	81	45	54	82	42	51	82	658	2347
Solid	48	57	83	44	52	85	31	38	80	13	15	84	100	758
Suspension	63	74	84	43	50	87	52	61	85	10	12	82	136	708
Foam	24	28	86	17	19	88	14	18	80	03	04	69	28	317
Powder	40	48	83	30	36	83	27	35	77	13	15	84	42	344
Granular	77	90	86	16	17	91	38	47	81	08	09	93	21	97
Large chunks	23	37	62	00	00	-	27	34	80	00	00	-	7	44
Vapor	71	81	88	62	67	93	67	75	88	00	00	-	4	39
Gel	84	95	89	00	00	-	00	00	-	00	00	-	1	115
Mean all subclasses	54	65	82	24	27	88	32	38	82	05	06	82		
Single Phase[3]	70	82	86	64	74	87	57	66	86	46	54	86	377	1551
Multiphase[4]	39	53	74	35	47	76	30	40	76	20	27	75	440	1810
For instance size larger than 5000 pixels														
Single Phase[3]	75	86	87	68	78	88	60	69	87	50	57	87	323	1286
Multiphase[4]	47	63	75	44	57	77	37	48	77	26	34	76	289	1440
For instance size larger than 10000 pixels														
Single Phase[3]	77	87	88	70	79	89	62	71	87	50	58	87	269	1126
Multiphase[4]	51	68	75	45	59	76	41	54	76	30	40	75	206	1240

1. Matching between GT and predicted segments depends only on segments overlap and not on class.
2. Standard metrics, i.e., matching GT and predicted segments must have the same class.
3. Single-phase system: only one material phase in a given vessel.
4. A multiphase system: more than one separate material phase in the vessel.

Methods

Training and evaluation sets

The Vector-LabPics dataset was split into training and evaluation sets by selecting 497 images for the evaluation set and leaving 1691 images for the training set. The images for both sets were taken from completely different sources so that there would be no overlap in terms of the conditions, settings, or locations between the evaluation and training images. This was done in order to ensure that the results from the neural net for the evaluation set will represent the accuracy that is likely to be achieved for an image taken in a completely unfamiliar setting.

Training with additional datasets

Training on related tasks is a way to increase the robustness and accuracy of a net. Reasoning about liquids for tasks like pouring and volume estimation by robots has been explored for problems relating to robotic kitchens and can be viewed as a related problem.[2-6] However, the available datasets for these tasks consist mostly of 3D models, volumes, and bounding boxes[2] and do not fit for the semantic and instance segmentation training used here. Containers in everyday settings appear in several general datasets, although the content of these vessels is not annotated. The COCO dataset[31] is the largest and most general image segmentation dataset and contains several subclasses of vessels, such as cups, jars, and bottles. We speculated that training with Vector-LabPics and related vessel classes from the COCO dataset could improve the accuracy of our nets. The nets were co-trained with subclasses of vessels from the COCO panoptic dataset and gave the same accuracy as the nets trained on Vector-LabPics alone, implying that the addition of new data did not improve or degrade the performance. However, it should be noted that for most images in the Vector-LabPics dataset, the vessel is the main or only object in the image. A net co-trained on the COCO dataset has an advantage in more complex environments where it is necessary to separate vessels from various other objects in the image.

Semantic segmentation using FCN

The semantic segmentation task involves finding the class for each pixel in the image (Figure 3). The standard approach is the fully convolutional neural net (FCN). We have implemented this approach using the PSP architecture.[32,35] Most semantic segmentation tasks involve finding a single exclusive class for each pixel; however, in the case of Vector-LabPics, a single-pixel may belong to several different classes simultaneously (Figure 3). The multi-class prediction was achieved by predicting an independent binary map for each class. For each pixel, this map predicts whether or not it belongs to the specific class (Figure 3). The training loss for this net was the sum of the losses of all of the classes predictions. Other than this, the training process and architecture were those of the standard approach used in previous works.[32,35]

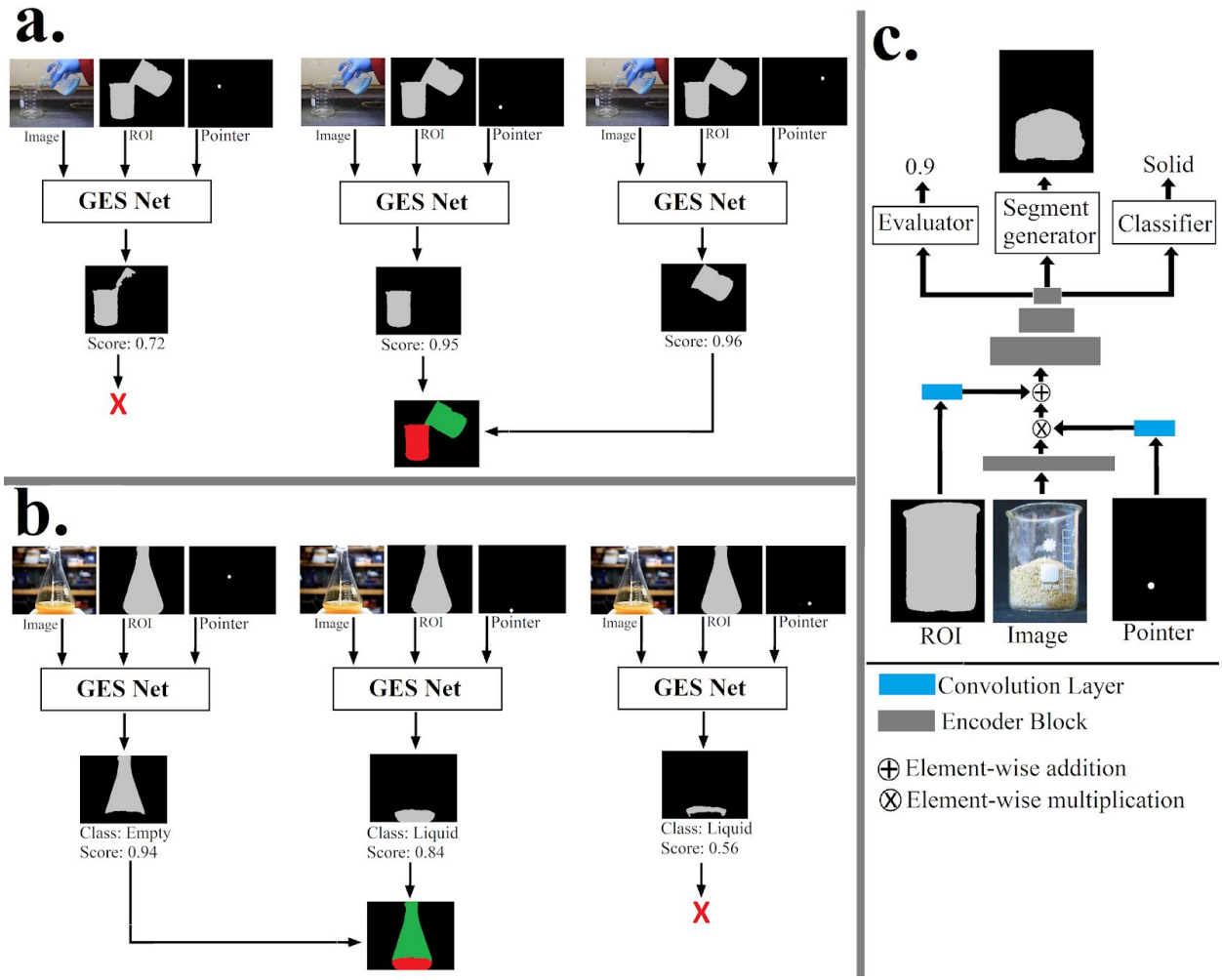


Figure 7: (a,b) GES net for vessel and material instance segmentation. The net receives an image, an ROI mask, and a pointer point in the image, and outputs the mask instance containing the point within the ROI, for (a) the vessel and (b) the material. The net also outputs the confidence score, which 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.

Hierarchical instance segmentation using a unified GES net

The generator evaluator[30] approach for image segmentation is based on two main modules: (1) a generator that proposes various regions corresponding to different segments of vessels or materials in the image; and (2) an evaluator that estimates how well the proposed segment matches the real region in the image and selects the best segments to be merged into the final segmentation map. Although previous studies have used different nets for the generator and evaluator,[30] this work uses a single unified net for both, i.e., one net that outputs the segment region (generator), its confidence score (evaluator), and it's class (Figure 7). In this case, the generator net consists of a convolutional net that, given a point in the image, finds the segment containing that point (Figure 7).[38,30,39] Picking different points in the image will lead the net to predict different segments. This can occur even if the point is in the same segment. Another input for the net is the region of interest (ROI) mask (Figure 7), which limits the region of the image in which the

output segment may be found.[28,38] Hence, the output segment must be contained within the ROI mask. In addition to the output segment and its class, the GES net also predicts the confidence score for this segment, which simply represents how well the prediction fits the real segment in the image in terms of the intersection over union (IOU). The net was run by picking several random points inside the ROI region and selecting the output segments with the highest scores (Figure 7a,b). Two different nets for vessel segmentation and material segmentation were trained separately and used hierarchically (Figure 4b). Hence, the region of all vessels was first found using FCN for semantic segmentation and was then transferred to a GES net for vessel instance segmentation (as an ROI input), to identify the regions of individual vessels (Figure 7a). The region of each vessel was transferred (as an ROI mask) to another network that finds the region and class of each material phase inside the vessel (Figure 7b).

Single-step Instance Segmentation using Mask-RCNN

The Mask R-CNN model was used to predict both the vessel and the materials instances in a single step.[29] Following the previous work, the model uses ResNet as a backbone,[40] followed by a Region Proposal Network (RPN) which provides a list of candidate instances. Given such candidates, both the instance binding box and class are predicted by the box head. In addition, masks for both vessel and material classes are generated. As the mask loss only considers the prediction corresponds to the class label, it enables the model to predict highly overlapped masks correctly as inter-class competition is avoided. This is especially important for our case, as most of the material instances are stored inside a vessel, which leads to almost complete overlap between the vessels and materials. Since an instance in the Vector-LabPics dataset could belong to multiple subclasses, the original Mask-RCNN is modified to handle multi-label classification. Such a function is enabled via an additional subclass predictor; it takes the same ROI feature generated by the box head, and output label powerset as the multi-label subclass prediction. This predictor takes the same feature vector from the box head and uses a single fully connected layer to do the classification. The subclass loss is defined as a binary cross-entropy loss. As Mask-RCNN is designed to do instance segmentation, the results of the net need to be merged into the panoptic segmentation map. Two separate panoptic segmentation maps are created for the material and vessels. Proposed instances for each map are filtered by removing low confidence instances. After that, all the remaining instances are overlaid on the corresponding segmentation map. In the case of overlapping masks, the mask with the higher confidence will cover the one with the lower confidence.

Supporting Information:

1. The codes and trained models for all the nets used for this work are available from this URL:

   https://github.com/aspuru-guzik-group/Computer-vision-for-the-chemistry-lab

2. The full Vector-LabPics dataset is available from this URL:

   https://zenodo.org/record/3697452

3. Videos of chemical processes annotated by the net can be viewed here:

   https://www.youtube.com/playlist?list=PLRiTwBVzSM3B6MirlFl6fW0YQR4TtQmtJ

   https://zenodo.org/record/3697693

Acknowledgment

We like to thank the sources of the images used for creating this dataset without them this work was not possible. These sources include Nessa Carson (@SuperScienceGrl Twitter), *Chemical and Engineering Science chemistry in pictures*, YouTube channels dedicated to chemistry experiments: *NurdRage*, *NileRed*, *DougsLab*, *ChemPlayer*, *and Koen2All*. Additional sources for images include Instagram channels chemistrylover_(Joana Kulizic),*Chemistry.shz* (Dr.Shakerizadeh-shirazi), *MinistryOfChemistry*, *Chemistry And Me*, *ChemistryLifeStyle*, *vacuum_distillation*, *and Organic_Chemistry_Lab*. We are grateful to the Defense Advanced Research Projects Agency (DARPA) for funding this project under award number W911NF-18-2-0036 from the Molecular Informatics program. A.A.-G. Thanks Anders G. Frøseth for his generous support.

Reference

[1] James W Zubrick. T*he organic chem lab survival manual: a student's guide to techniques*. John Wiley & Sons, 2016.

[2] Roozbeh Mottaghi, Connor Schenck, Dieter Fox, and Ali Farhadi. *See the glass half full: Reasoning about liquid containers, their volume and content.* In Proceedings of the IEEE International Conference on Computer Vision , pages 1871–1880, 2017.

[3] Monroe Kennedy, Kendall Queen, Dinesh Thakur, Kostas Daniilidis, and Vijay Kumar. *Precise dispensing of liquids using visual feedback.* In 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 1260–1266. IEEE, 2017.

[4] Connor Schenck and Dieter Fox. *Detection and tracking of liquids with fully convolutional Networks.* arXiv preprint arXiv:1606.06266, 2016.9

[5] Tz-Ying Wu, Juan-Ting Lin, Tsun-Hsuang Wang, Chan-Wei Hu, Juan Carlos Niebles, and Min Sun. *Liquid pouring monitoring via rich sensory inputs.* In Proceedings of the European Conference on Computer Vision (ECCV), pages 335–351, 2018.

[6] Connor Schenck and Dieter Fox. *Perceiving and reasoning about liquids using fully convolutional networks.* The International Journal of Robotics Research, 37(4-5):452–471, 2018.

[7] Steven V Ley, Richard J Ingham, Matthew O'Brien, and Duncan L Browne. *Camera-enabled techniques for organic synthesis.* Beilstein journal of organic chemistry, 9(1):1051–1072, 2013.

[8] Sebastian Steiner, Jakob Wolf, Stefan Glatzel, Anna Andreou, Jarosław M Granda, Graham Keenan, Trevor Hinkley, Gerardo Aragon-Camarasa, Philip J Kitson, Davide Angelone, et al. *Organic synthesis in*

a modular robotic system driven by a chemical programming language. Science, 363(6423):eaav2211, 2019.

[9] Connor W Coley, Dale A Thomas, Justin AM Lummiss, Jonathan N Jaworski, Christopher P Breen, Victor Schultz, Travis Hart, Joshua S Fishman, Luke Rogers, Hanyu Gao, et al. *A robotic platform for flow synthesis of organic compounds informed by ai planning.* Science, 365(6453):eaax1566, 2019.

[10] Steven V Ley, Daniel E Fitzpatrick, Richard J Ingham, and Rebecca M Myers. *Organic synthesis: march of the machines.* Angewandte Chemie International Edition, 54(11):3449–3464, 2015.

[11] Florian Häse, Loïc M Roch, and Alán Aspuru-Guzik. *Next-generation experimentation with self-driving laboratories.* Trends in Chemistry, 2019.

[12] Jordan A Daponte, Yuejun Guo, Rebecca T Ruck, and Jason E Hein. *Using an automated monitoring platform for investigations of biphasic reactions.* ACS Catalysis, 9(12):11484–11491,

2019.

[13] Fang Ren, Logan Ward, Travis Williams, Kevin J Laws, Christopher Wolverton, Jason Hattrick-Simpers, and Apurva Mehta. *Accelerated discovery of metallic glasses through iteration of machine learning and high-throughput experiments.* Science advances, 4(4):eaaq1566, 2018.

[14] Zhi Li, Mansoor Ani Najeeb, Liana Alves, Alyssa Sherman, Peter Cruz Parrilla, Ian M Pendleton, Matthias Zeller, Joshua Schrier, Alexander J Norquist, and Emory Chan. *Robot-accelerated perovskite investigation and discovery (rapid): 1. inverse temperature crystallization.* 2019.

[15] Yuri Boykov and Vladimir Kolmogorov. *An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision.* IEEE Transactions on Pattern Analysis & Machine Intelligence, (9):1124–1137, 2004.

[16] John Canny. *A computational approach to edge detection. IEEE Transactions on pattern*

analysis and machine intelligence, (6):679–698, 1986.

[17] Weiqi Yuan and Desheng Li. *Measurement of liquid interface based on vision.* In Fifth World Congress on Intelligent Control and Automation (IEEE Cat. No. 04EX788), volume 4, pages 3709–3713. IEEE, 2004.

[18] Kunal J Pithadiya, Chintan K Modi, and Jayesh D Chauhan. *Comparison of optimal edge*

detection algorithms for liquid level inspection in bottles. In 2009 Second International Conference on Emerging Trends in Engineering & Technology, pages 447–452. IEEE, 2009.

[19] Sagi Eppel and Tal Kachman. *Computer vision-based recognition of liquid surfaces and phase boundaries in transparent vessels*, with emphasis on chemistry applications. arXiv preprint arXiv:1404.7174, 2014.

[20] Matthew O'Brien, Peter Koos, Duncan L Browne, and Steven V Ley. *A prototype continuous flow liquid–liquid extraction system using open-source technology.* Organic & biomolecular chemistry, 10(35):7031–7036, 2012.

[21] Qing Liu, Bo Chu, Jinye Peng, and Sheng Tang. *A visual measurement of water content of crude oil based on image grayscale accumulated value difference.* Sensors, 19(13):2963, 2019.10

[22] Ti-Ho Wang, Ming-Chih Lu, Chen-Chien Hsu, Cheng-Chuan Chen, and Jia-Dong Tan. *Liquid-level measurement using a single digital camera.* Measurement, 42(4):604–610, 2009.

[23 Sagi Eppel. *Tracing liquid level and material boundaries in transparent vessels using the graph cut computer vision approach.* arXiv preprint arXiv:1602.00177 , 2016.

[24] Scott Mayer McKinney, Marcin Sieniek, Varun Godbole, Jonathan Godwin, Natasha Antropova, Hutan Ashrafian, Trevor Back, Mary Chesus, Greg C Corrado, Ara Darzi, et al. *International evaluation of an ai system for breast cancer screening.* Nature, 577(7788):89–94, 2020.

[25] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. *Imagenet classification with deep convolutional neural networks.* In Advances in neural information processing systems, pages 1097–1105, 2012.

[26] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. *Imagenet: A large-scale hierarchical image database.* In 2009 IEEE conference on computer vision and pattern recognition, pages 248–255. Ieee, 2009.

[27] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. *Microsoft coco: Common objects in context.* In European conference on computer vision, pages 740–755. Springer, 2014.

[28] Sagi Eppel. *Setting an attention region for convolutional neural networks using region selective features, for recognition of materials within glass vessels.* arXiv preprint arXiv:1708.08711, 2017.

[29] 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.

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

[31] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. *Microsoft COCO: Common objects in context*. In

ECCV, 2014.

[32] Jonathan Long, Evan Shelhamer, and Trevor Darrell. *Fully convolutional networks for semantic segmentation*. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 3431–3440, 2015.

[33] Holger Caesar, Jasper Uijlings, and Vittorio Ferrari. *COCO-Stuff: Thing and stuff classes in*

context. In CVPR, 2018.

[34] Abhishek Dutta and Andrew Zisserman. *The VIA annotation software for images, audio and video.* In Proceedings of the 27th ACM International Conference on Multimedia, MM '19, New York, NY, USA, 2019. ACM.

[35] 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.

[36] Kuo Men, Huaizhi Geng, Chingyun Cheng, Haoyu Zhong, Mi Huang, Yong Fan, John P Plastaras, Alexander Lin, and Ying Xiao. *More accurate and efficient segmentation of organs-at-risk in radiotherapy with convolutional neural networks cascades*. Medical physics, 46(1):286–292, 2019.

[37] Alexander Kirillov, Kaiming He, Ross Girshick, Carsten Rother, and Piotr Dollár. *Panoptic segmentation.* In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 9404–9413, 2019.

[38] Sagi Eppel. *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, 2019.11

[39] Konstantin Sofiiuk, Olga Barinova, and Anton Konushin. *Adaptis: Adaptive instance selection Network*. arXiv preprint arXiv:1909.07829, 2019.

[40] 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.

Supporting information

Training Mask-RCNN

For the Mask R-CNN network[29] backbone, we used a ResNet 50,[40] FPN model pretrained on the COCO dataset.[31] 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 super-classes based on their subclass label. The other model has a more fine-grain separation, where instances are classified into three super-classes, 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 (Figure 7c) is built on a ResNet encoder[40] that receives the image, a pointer point and an ROI mask.[38] Both the pointer point and ROI mask are represented as binary masks (Figure 7c) and 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 7c). The final convolutional layer of the Resnet is connected to three different heads: the segmentation head consists of a standard PSP head followed by three upsampling layers, while the classification head is similar to the ResNet standard classification head 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.

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 7a) was chosen as the region of the vessel containing the material. The ROI for the vessel GES net (Figure 7b) 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 pretrained on the COCO panoptic dataset.[30,38] Due to the small size of the dataset, significant augmentation was used, including deforming, noise adding, cropping and color modifications. Each of the nets was trained on a single Titan XP GPU for about 200 epochs.