Supplemental Material for paper ID 479

Localizing Grouped Instances for Efficient Detection in Low-Resource Scenarios

This complementary supplemental describes experiments for "Grouped Instances for Object Detection in Low-Resource Systems". First, we report additional qualitative results of ODGI in Figure 4, Figure 5, Figure 6, Figure 7 and Figure 8.

In Section 1, we report extended results for ODGI with various number of extracted crops in the first stage. In Section 2, we list values of ODGI hyperparameters we obtain on the validation set, as described in the main paper. Finally, in Section 3 , Section 5 and Section 5 we detail the results of our ablation experiments.

1. Effect of the number of crops, γ_1

In Figure 1 and 2, we report on experiments comparing baselines to ODGI for all number of crops $\gamma_1 \in [1, 10]$. For easier readability, We display results as a 2D plot of `MAP0.5` versus computation time (CPU), each value represented as a percentage of the baseline (*i.e. tiny-yolo* with input size 1024x1024 pixels for VEDAI, *yolo* and *MobileNet-100* with input size 1024x1024 pixels for SDD). As expected, increasing the number of crops always increase the final detection accuracy, although the increase is not worth the additional computational effort in some cases. Nonetheless, the curves corresponding to ODGI methods are always above the ones for the standard detectors baselines, even for large number of crops, showing the proposed method is computationally interesting for different budget allocations. For completeness, we also report the corresponding numeric values to these curves in Table 6, Table 7 and Table 8, as well as the corresponding timings on Raspberry Pi and GPU.

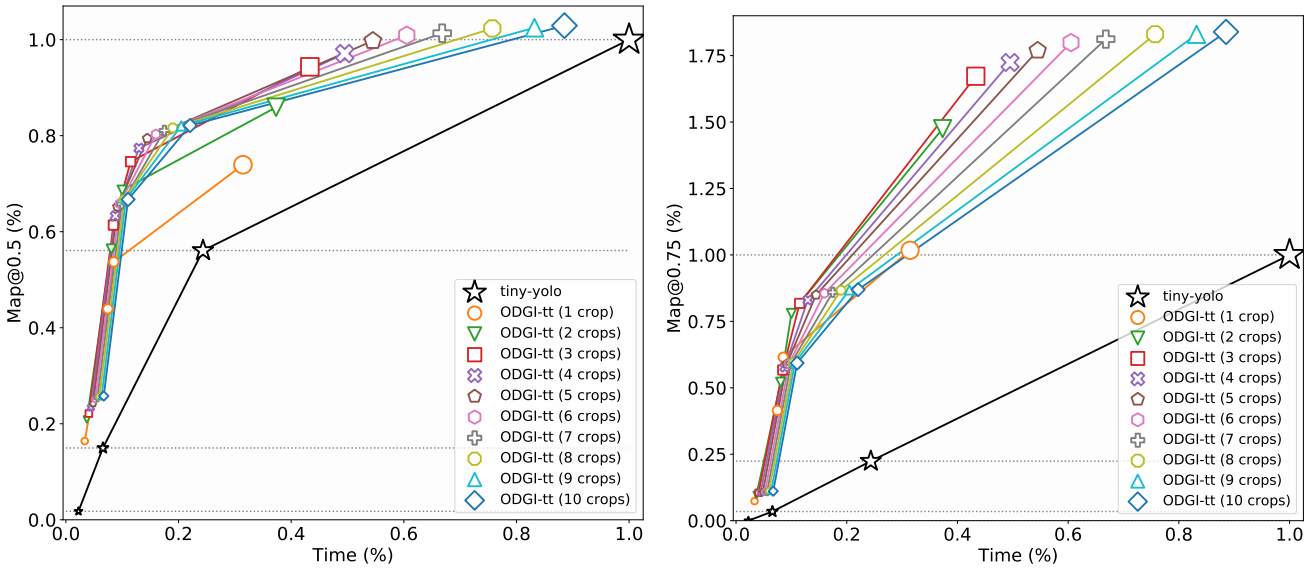


Figure 1: Plots of `MAP0.5` (left) and `MAP0.75` (right) versus runtime (CPU) for VEDAI. The metrics are reported in percentage, relatively to the baseline run at full resolution. The black dashed line represents the baseline model run at different resolutions. Each colored line corresponds to a specific number of extracted crops, γ_1, each point being a specific input resolution.

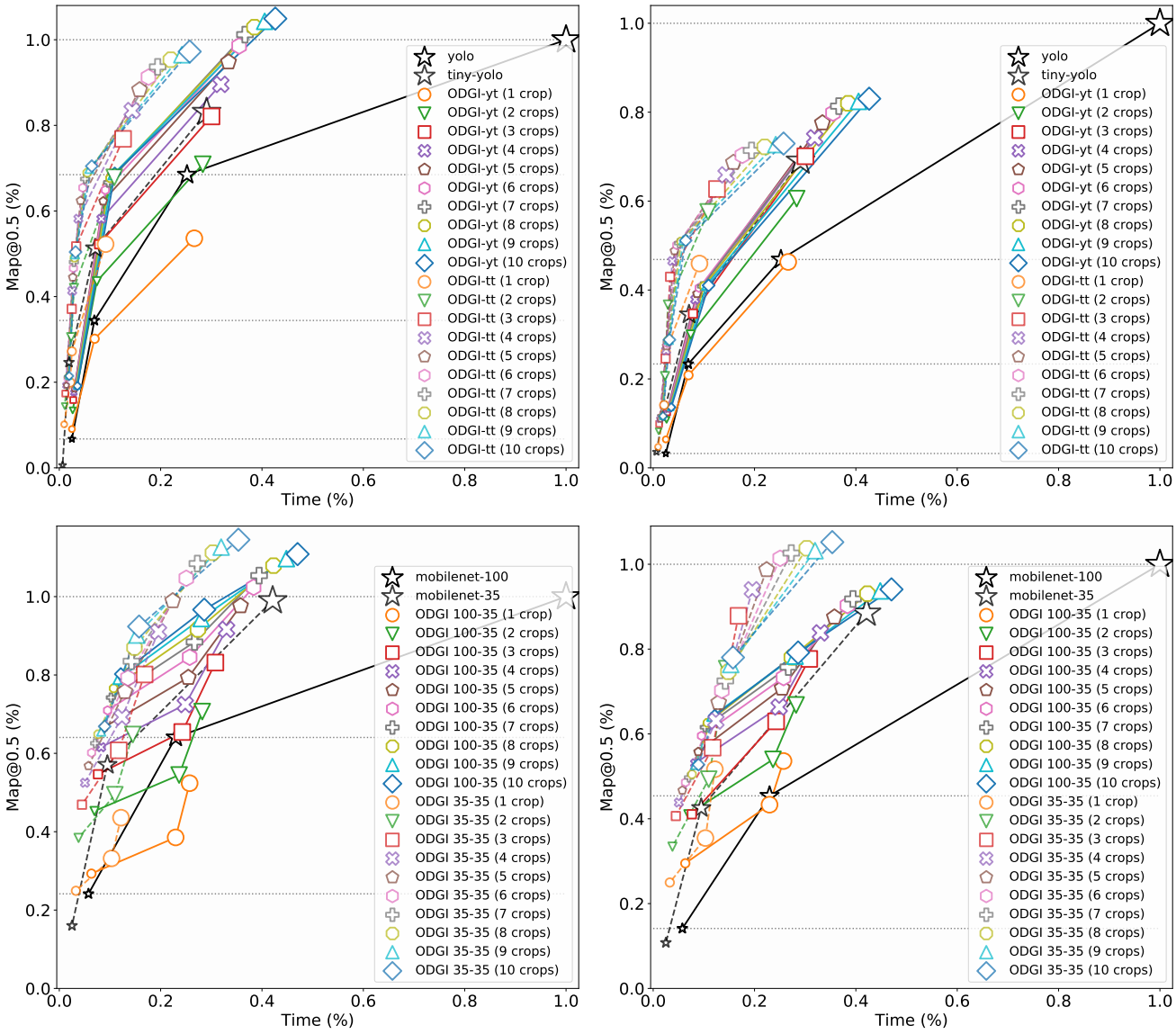


Figure 2: Plots of MAP versus runtime (CPU) for YOLO and tiny-YOLO based (top) and MobileNet based (bottom) models on SDD.

2. Hyperparameters

In Table 1, we report the values of hyperparameters obtained from the validation set for all the ODGI 512-256 methods: τ_{low} is almost always 0, showing that most low confidence patches are indeed true negatives and do not need to be filtered out. The value of τ_{nms} is generally around 0.25, which shows that it is beneficial to prevent patch overlap as to increase the surface of the image refined by the second stage. Finally, $\tau_{\text{high}} \in \{0.8, 0.9\}$ for VEDAI and $\tau_{\text{high}} \in \{0.6, 0.7\}$ for SDD. This reflects intrinsic properties of each dataset: in VEDAI, images mostly contain only few objects which can be covered by all the extracted crops. It is always beneficial to refine these predictions hence a high value of τ_{high}. On the more challenging SDD, ODGI tends to more often use the shortcut for confident individuals in stage 1, and instead focuses on groups and lower-confidence individuals which can benefit from refinement more.

Note that for the lower resolutions (256-128) we observe the same trends for hyperparameters: values are the same for τ_{low} and τ_{high}, while τ_{nms} is generally higher (0.5), matching the fact that patches overlap more often at lower resolutions.

ODGI-yt 512-256 on SDD	τ_{low}	τ_{high}	τ_{nms}
(1 crop)	0.0	0.6	0.25
(2 crops)	0.0	0.6	0.5
(3 crops)	0.1	0.6	0.5
(4 crops)	0.0	0.6	0.25
(5 crops)	0.0	0.6	0.25
(6 crops)	0.0	0.6	0.25
(7 crops)	0.1	0.6	0.25
(8 crops)	0.0	0.6	0.5
(9 crops)	0.0	0.6	0.5
(10 crops)	0.0	0.6	0.5

ODGI-tt 512-256 on SDD	τ_{low}	τ_{high}	τ_{nms}
(1 crop)	0.0	0.6	0.25
(2 crops)	0.0	0.6	0.25
(3 crops)	0.0	0.6	0.25
(4 crops)	0.1	0.6	0.25
(5 crops)	0.1	0.6	0.25
(6 crops)	0.0	0.6	0.25
(7 crops)	0.0	0.6	0.25
(8 crops)	0.0	0.7	0.25
(9 crops)	0.1	0.7	0.25
(10 crops)	0.0	0.7	0.25

ODGI-tt 512-256 on VEDAI	τ_{low}	τ_{high}	τ_{nms}
(1 crop)	0.0	0.7	0.25
(2 crops)	0.0	0.7	0.25
(3 crops)	0.0	0.8	0.5
(4 crops)	0.0	0.8	0.5
(5 crops)	0.0	0.9	0.5
(6 crops)	0.0	0.8	0.25
(7 crops)	0.0	0.8	0.25
(8 crops)	0.1	0.8	0.25
(9 crops)	0.0	0.8	0.25
(10 crops)	0.0	0.8	0.25

ODGI-100-35 512-256 on SDD	τ_{low}	τ_{high}	τ_{nms}
(1 crop)	0.0	0.6	0.25
(2 crops)	0.0	0.6	0.25
(3 crops)	0.0	0.6	0.25
(4 crops)	0.3	0.6	0.25
(5 crops)	0.3	0.6	0.25
(6 crops)	0.1	0.6	0.25
(7 crops)	0.1	0.6	0.25
(8 crops)	0.3	0.6	0.25
(9 crops)	0.1	0.6	0.25
(10 crops)	0.1	0.6	0.25

ODGI-35-35 512-256 on VEDAI	τ_{low}	τ_{high}	τ_{nms}
(1 crop)	0.0	0.6	0.25
(2 crops)	0.0	0.6	0.25
(3 crops)	0.0	0.6	0.25
(4 crops)	0.3	0.6	0.25
(5 crops)	0.3	0.6	0.25
(6 crops)	0.1	0.6	0.25
(7 crops)	0.1	0.6	0.25
(8 crops)	0.3	0.6	0.25
(9 crops)	0.1	0.7	0.25
(10 crops)	0.1	0.6	0.25

Table 1: Summary of hyperparameters for the ODGI 512-256 models, optimized on the validation set.

3. Ablation experiments: No Groups

We compare ODGI with ODGI$^{\text{singles}}$, a variant without group information: the loss term $\mathcal{L}_{\text{group}}$ in our training objective disappears and we ignore group flags in the transition between stages. In Table 2 and Table 3, we observe that ODGI consistantly improves over ODGI $^{\text{singles}}$ in terms of detection accuracies for equivalent number of crops, γ_1. However, we also observe that both methods behave qualitatively differently. Intuitively, detecting groups in early stages is more efficient as larger image regions, accounting for multiple individuals, can be extracted then propagated down the pipeline. Without that ability, ODGI $^{\text{singles}}$ usually require more crops, and thus computation, to achieve similar detection accuracies as ODGI. On the other hand, ODGI $^{\text{singles}}$ focuses on detecting individuals, hence the individual boxes early-exiting the pipeline after the first stage are usually better individual predictions. Patches extracted by ODGI provide better coverage of relevant regions than ODGI$^{\text{singles}}$, but ODGI$^{\text{singles}}$ typically provides more confident individual objects at the early exit of stage 1.

`MAP0.5`	$\gamma_1 = 1$	$\gamma_1 = 2$	$\gamma_1 = 3$	$\gamma_1 = 4$	$\gamma_1 = 5$	$\gamma_1 = 6$	$\gamma_1 = 7$	$\gamma_1 = 8$	$\gamma_1 = 9$	$\gamma_1 = 10$
ODGI-tt 512-256	0.2455	0.3188	0.3610	0.3921	0.4148	0.4288	0.4400	0.4479	0.4533	0.4568
no groups	0.2252	0.2788	0.3211	0.3549	0.3806	0.3991	0.4132	0.4240	0.4321	0.4386
ODGI-tt 256 128	0.1278	0.1974	0.2430	0.2733	0.2932	0.3074	0.3170	0.3238	0.3280	0.3311
no groups	0.1220	0.1823	0.2297	0.2597	0.2821	0.2973	0.3088	0.3167	0.3223	0.3265

Table 2: `MAP0.5` results. Comparing two ODGI models on the SDD dataset versus their counterpart not exploiting group structures. ODGI consistantly outperforms the no groups variant for equivalent number of crops.

MAP0.75	$\gamma_1 = 1$	$\gamma_1 = 2$	$\gamma_1 = 3$	$\gamma_1 = 4$	$\gamma_1 = 5$	$\gamma_1 = 6$	$\gamma_1 = 7$	$\gamma_1 = 8$	$\gamma_1 = 9$	$\gamma_1 = 10$
ODGI-tt 512-256	0.0399	0.0500	0.0545	0.0573	0.0597	0.0611	0.0621	0.0628	0.0632	0.0634
no groups	0.0346	0.0421	0.0473	0.0513	0.0547	0.0563	0.0576	0.0585	0.0592	0.0599
ODGI-tt 256 128	0.0227	0.0319	0.0374	0.0404	0.0424	0.0435	0.0439	0.0443	0.0445	0.0445
no groups	0.0234	0.0300	0.0348	0.0381	0.0409	0.0426	0.0440	0.0446	0.0449	0.0453

Table 3: MAP0.75 results comparing ODGI to its "no groups" counter part.

4. Ablation Experiments: No Offsets

The rescaling step introduced in the main paper is very fast to compute and yet strongly impacts the detection accuracy. Intuitively, if the learned scale offsets are too low, the extracted patches might cut objects rather than englobe them, hence leading to missed detections. If they are too high, the extracted regions become very large, making detection harder for subsequent stages as their input resolution is smaller. In practice, we set the offset margin δ to 0.0025 in the definition of the offsets loss (see manuscript), which corresponds to roughly half the average size object size in our datasets.

To analyze the influence of the rescaling, we perform two sets of ablation experiments.

First, we test the model with fixed offsets rather than learned ones (ODGI$^{\text{cst-off}}$), with offset values fixed to $\frac{2}{3}$, *i.e.* 50% expansion of the bounding boxes(row *fixed offsets*), and corresponds to the value of the offsets margin δ we chose for standard ODGI. As shown in Table 4, this variant is always outperformed by ODGI, which shows that the model benefits from learning offsets tailored to its predictions. For instance a predicted box which is off-center, with low confidence, might need higher offsets to compensate. While a box which is already large to begin with should not need to be extended further.

Second, we entirely ignore the learned offsets during the patch extraction step (ODGI$^{\text{no-off}}$). This again negatively affects the MAP: extracted crops are well localized around relevant objects, but do not actually fully enclose them. Consequently, the second stage retrieves partial objects, but with very high confidence, leading to strong false positives predictions. Most correct detections come instead from early-exit predictions at stage 1, hence the MAP does not increase when adding more crops (see Table 4).

MAP0.5	$\gamma_1 = 1$	$\gamma_1 = 2$	$\gamma_1 = 3$	$\gamma_1 = 4$	$\gamma_1 = 5$	$\gamma_1 = 6$	$\gamma_1 = 7$	$\gamma_1 = 8$	$\gamma_1 = 9$	$\gamma_1 = 10$
ODGI-tt 512-256	0.2455	0.3188	0.3610	0.3921	0.4148	0.4288	0.4400	0.4479	0.4533	0.4568
no offsets	0.1269	0.1278	0.1270	0.1264	0.1254	0.1247	0.1238	0.1232	0.1225	0.1218
constant offsets	0.1993	0.2286	0.2385	0.2440	0.2460	0.2467	0.1238	0.2460	0.2449	0.2438
ODGI-tt 256 128	0.1278	0.1974	0.2430	0.2733	0.2932	0.3074	0.3170	0.3238	0.3280	0.3311
no offsets	0.0339	0.0377	0.0391	0.0401	0.0401	0.0403	0.0401	0.0402	0.0399	0.0399
constant offsets	0.0882	0.1186	0.1358	0.1444	0.1499	0.1531	0.0401	0.1549	0.1548	0.1544

MAP0.75	$\gamma_1 = 1$	$\gamma_1 = 2$	$\gamma_1 = 3$	$\gamma_1 = 4$	$\gamma_1 = 5$	$\gamma_1 = 6$	$\gamma_1 = 7$	$\gamma_1 = 8$	$\gamma_1 = 9$	$\gamma_1 = 10$
ODGI-tt 512-256	0.0399	0.0500	0.0545	0.0573	0.0597	0.0611	0.0621	0.0628	0.0632	0.0634
no offsets	0.0212	0.0210	0.0209	0.0208	0.0207	0.0207	0.0206	0.0206	0.0206	0.0205
constant offsets	0.0303	0.0319	0.0327	0.0330	0.0333	0.0333	0.0206	0.0332	0.0332	0.0331
ODGI-tt 256 128	0.0227	0.0319	0.0374	0.0404	0.0424	0.0435	0.0439	0.0443	0.0445	0.0445
no offsets	0.0033	0.0033	0.0033	0.0033	0.0032	0.0032	0.0032	0.0032	0.0032	0.0032
constant offsets	0.0115	0.0144	0.0161	0.0168	0.0172	0.0173	0.0032	0.0173	0.0171	0.0170

Table 4: Comparing two ODGI models on the SDD dataset versus their counterpart with fixed offsets or no offsets at all. ODGI consistantly outperforms the no groups variant for equivalent number of crops.

5. Ablation Experiments: Shared Weights

A disadvantage of ODGI as we formulate it in the manuscript is that requires S backbone networks, one for each stage. While this is acceptable for lightweight architectures, *e.g*. MobileNet, this can be prohibitive for wider and deeper backbones. To palliate this problem, we also experimented with sharing weights across different stages: In this setting, we have only backbone networks, common to the first and second stages, while only the last fully connected layer is specific to each stage. We make two observations: (i) In that setting, it is often beneficial to have a slightly delayed training schedule rather than train jointly from the start, i.e to first train the detections for stage 1 only, and after a few epoch to incorporate the stage 2 contributions. (ii) While this decreases the number of model parameters, weights sharing significantly decreases the detection performance. See Table 5 for quantitative results.

MAP0.5 ($\gamma = 6$)	no sharing	sharing
ODGI-tt 512-256	0.429	0.385
ODGI-tt 256-128	0.307	0.197
ODGI-tt 128-64	0.098	0.051

Table 5: Results of weights sharing experiments on SDD for the ODGI teeny-tiny models. Sharing weights halves the number of model parameters but significantly hurts the model performance

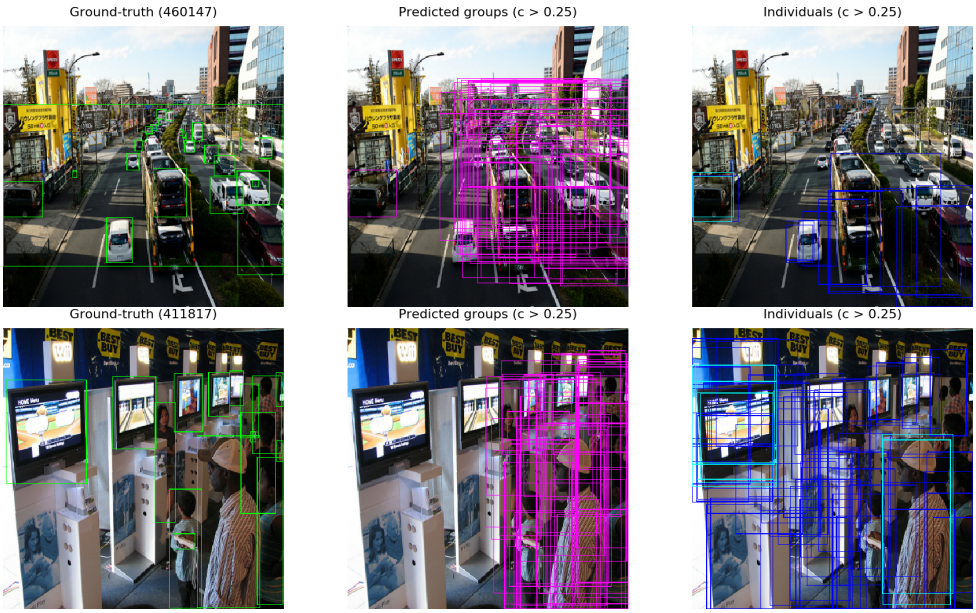
In fact, the visual appearance of input images to stage 1 (small relevant regions in large sparse images) and stage 2 (often densely covered patch pre-selected by stage 1) are drastically different. This can be seen from the qualitative examples in Figure 4 and following. This introduces a *visual domain shift* between the two stages which explains why sharing representations for these two domains might not be adequate. More complex settings, e.g. sharing only the early weights would be possible but are out of the scope of this paper.

6. Qualitative evaluation of groups

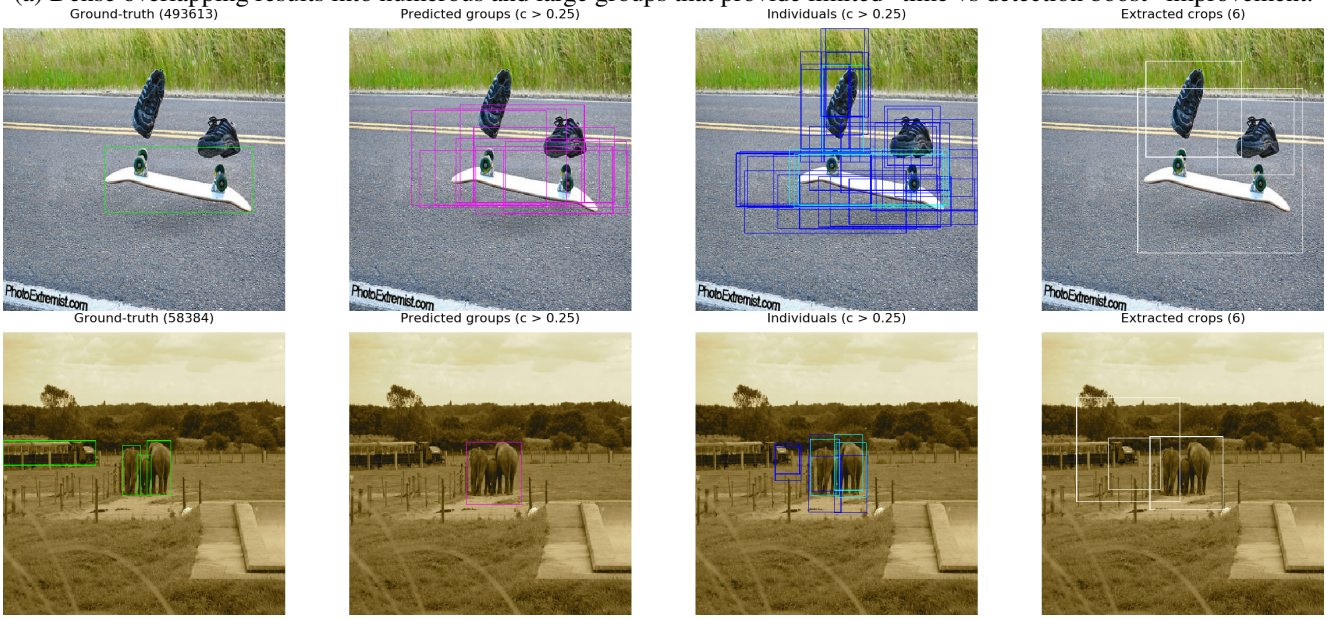
In this section, we briefly discuss the relevance of the ground-truth and predicted groups with respect to the dataset properties.

Small groups in large sparse areas. In Figure 4 and following, we report examples of groups detected by the first stage of ODGI on test samples from the SDD dataset (Subfigure (c)). We observe that the detected groups exhibit three nice properties: *First,* they are of relatively small size, as a consequence the crops extracted and fed as inputs to stage 2 are well localized and effectively provide a generous "zoom-in" effect on relevant regions. *Second,* they contain in general only a few objects, which makes the detection task for stage 2 easier. *Finally*, the sparse distribution of objects in the input images leads to only a few, non-overlapping, generated relevant regions. Combined, these conditions contribute to improving the speed-vs-accuracy trade-off by providing the second stage with few relevant regions that each contain a detection problem easier than the one provided to stage 1.

Densely distributed medium-to-large sized objects. In contrast, benchmarks with large objects and dense object distributions, such as MS-COCO, provide drastically different conditions. In fact when objects overlap too densely, the notion of groups becomes fuzzy and the extracted relevant regions are often quite large and numerous. We show an example of this in Figure 3 (a). Consequently, the crops received by the second stage often contain a detection problem almost as hard as the first stage. Furthermore they come in large numbers, which require numerous feed-forward passes of the second stage. When the object distribution is sparser (Figure 3 (b-c), the detected groups define relevant localized image regions, as was the case in the aerial view dataset settings. However the objects being quite large to begin with, they are often well detected as individuals (see third image in each row): As a result, feeding the extracted crop to the second stage might help refine the bounding box coordinates, however the potential improvement is limited, relatively to the cost of an additional feed-forward pass, considering that objects in the group are often already well detected at the individual level.



(a) Dense overlapping results into numerous and large groups that provide limited "time vs detection boost" improvement.



(b) In sparser scenes, fewer groups are detected but they yield large image regions as the objects themselve are quite large to begin with.

(c) Example with no object overlap: no groups detected.

Figure 3: Qualitative examples of applying ODGI to a dataset with dense distribution of rather large objects such as MS-COCO. (a) In many cases, objects densely overlap and the ground-truth and detected groups lead to rather large and *numerous* image regions that might slightly help detection but come at a high computational cost. (b) For sparser distributions, the learned groups lead to informative and well localized relevant image regions. Finally (c), when objects do not overlap, with respect to the grid size (13x13 here), no groups are detected.

	map@0.5	map@0.75	CPU (s)	Pi (s)	GPU (ms)
tiny 1024	0.684	0.252	1.926	10.473	14.279
tiny 512	0.384	0.056	0.469	2.619	8.158
tiny 256	0.102	0.009	0.127	0.695	6.971

(a) tiny-YOLO baselines results

	map@0.5	map@0.75	CPU (s)	Pi (s)	GPU (ms)
1 crop					
ODGI-tt 512-256	0.506	0.256	0.606	3.490	12.836
ODGI-tt 256-128	0.368	0.155	0.164	1.009	11.713
ODGI-tt 256-64	0.301	0.105	0.144	0.844	11.595
ODGI-tt 128-64	0.112	0.019	0.064	0.423	11.620
2 crops					
ODGI-tt 512-256	0.588	0.372	0.719	4.193	13.176
ODGI-tt 256-128	0.470	0.197	0.193	1.185	11.692
ODGI-tt 256-64	0.386	0.131	0.155	0.873	11.748
ODGI-tt 128-64	0.143	0.025	0.072	0.444	11.731
3 crops					
ODGI-tt 512-256	0.646	0.422	0.834	4.888	13.947
ODGI-tt 256-128	0.510	0.206	0.221	1.327	11.922
ODGI-tt 256-64	0.420	0.143	0.163	0.910	11.702
ODGI-tt 128-64	0.151	0.026	0.079	0.475	11.761
4 crops					
ODGI-tt 512-256	0.665	0.435	0.953	5.596	14.374
ODGI-tt 256-128	0.530	0.209	0.250	1.458	11.907
ODGI-tt 256-64	0.433	0.146	0.169	0.981	11.630
ODGI-tt 128-64	0.160	0.026	0.086	0.543	11.482
5 crops					
ODGI-tt 512-256	0.683	0.446	1.050	6.323	14.463
ODGI-tt 256-128	0.544	0.214	0.278	1.615	12.251
ODGI-tt 256-64	0.446	0.149	0.177	0.978	12.049
ODGI-tt 128-64	0.166	0.027	0.093	0.527	11.907

	map@0.5	map@0.75	CPU (s)	Pi (s)	GPU (ms)
6 crops					
ODGI-tt 512-256	0.691	0.454	1.166	7.026	14.685
ODGI-tt 256-128	0.549	0.216	0.308	1.778	11.970
ODGI-tt 256-64	0.450	0.149	0.184	1.016	12.081
ODGI-tt 128-64	0.172	0.027	0.101	0.559	12.042
7 crops					
ODGI-tt 512-256	0.693	0.457	1.287	7.719	14.919
ODGI-tt 256-128	0.555	0.217	0.337	1.933	12.434
ODGI-tt 256-64	0.450	0.149	0.191	1.046	12.111
ODGI-tt 128-64	0.174	0.027	0.108	0.587	11.965
8 crops					
ODGI-tt 512-256	0.700	0.461	1.458	8.418	15.841
ODGI-tt 256-128	0.559	0.219	0.365	2.095	12.115
ODGI-tt 256-64	0.456	0.150	0.199	1.160	11.838
ODGI-tt 128-64	0.175	0.028	0.115	0.608	11.647
9 crops					
ODGI-tt 512-256	0.701	0.461	1.602	9.121	16.609
ODGI-tt 256-128	0.561	0.219	0.394	2.260	12.606
ODGI-tt 256-64	0.457	0.150	0.205	1.121	12.486
ODGI-tt 128-64	0.177	0.028	0.122	0.652	12.348
10 crops					
ODGI-tt 512-256	0.704	0.464	1.705	9.806	17.015
ODGI-tt 256-128	0.562	0.219	0.425	2.420	12.268
ODGI-tt 256-64	0.457	0.149	0.213	1.169	12.354
ODGI-tt 128-64	0.176	0.028	0.129	0.687	12.234

(b) Results for ODGI methods

Table 6: Detailed results for each number of crops $\gamma_1 \in [1, 10]$ for experiments on the VEDAI dataset.

	map@0.5	map@0.75	CPU (s)	Pi (s)	GPU (ms)
yolo 1024	0.470	0.087	6.625	46.935	34.663
yolo 512	0.322	0.041	1.670	12.056	16.872
yolo 256	0.162	0.020	0.459	3.418	13.513
tiny 1024	0.390	0.060	1.926	10.473	14.279
tiny 512	0.241	0.030	0.469	2.619	8.158
tiny 256	0.116	0.010	0.127	0.695	6.971

(a) YOLO and tiny-YOLO baselines results

	map@0.5	map@0.75	CPU (s)	Pi (s)	GPU (ms)
1 crop					
ODGI-yt 512-256	0.252	0.040	1.765	12.901	22.625
ODGI-tt 512-256	0.245	0.040	0.606	3.490	12.836
ODGI-yt 256-128	0.142	0.018	0.462	3.707	18.751
ODGI-tt 256-128	0.128	0.023	0.164	1.009	11.713
ODGI-tt 256-64	0.094	0.012	0.144	0.844	11.595
2 crops					
ODGI-yt 512-256	0.333	0.053	1.876	13.626	23.219
ODGI-tt 512-256	0.319	0.050	0.719	4.193	13.176
ODGI-yt 256-128	0.205	0.026	0.491	3.896	18.600
ODGI-tt 256-128	0.197	0.032	0.193	1.185	11.692
ODGI-tt 256-64	0.143	0.018	0.155	0.873	11.748
3 crops					
ODGI-yt 512-256	0.386	0.061	1.990	14.319	22.807
ODGI-tt 512-256	0.361	0.055	0.834	4.888	13.947
ODGI-yt 256-128	0.245	0.030	0.521	4.053	18.931
ODGI-tt 256-128	0.243	0.037	0.221	1.327	11.922
ODGI-tt 256-64	0.174	0.021	0.163	0.910	11.702
4 crops					
ODGI-yt 512-256	0.421	0.065	2.112	15.002	23.713
ODGI-tt 512-256	0.392	0.057	0.953	5.596	14.374
ODGI-yt 256-128	0.273	0.033	0.550	4.202	18.679
ODGI-tt 256-128	0.273	0.040	0.250	1.458	11.907
ODGI-tt 256-64	0.194	0.023	0.169	0.981	11.630
5 crops					
ODGI-yt 512-256	0.445	0.067	2.213	15.727	24.654
ODGI-tt 512-256	0.415	0.060	1.050	6.323	14.463
ODGI-yt 256-128	0.293	0.034	0.574	4.374	18.902
ODGI-tt 256-128	0.293	0.042	0.278	1.615	12.251
ODGI-tt 256-64	0.209	0.024	0.177	0.978	12.049

	map@0.5	map@0.75	CPU (s)	Pi (s)	GPU (ms)
6 crops					
ODGI-yt 512-256	0.463	0.069	2.351	16.404	24.486
ODGI-tt 512-256	0.429	0.061	1.166	7.026	14.685
ODGI-yt 256-128	0.305	0.035	0.602	4.550	18.745
ODGI-tt 256-128	0.307	0.044	0.308	1.778	11.970
ODGI-tt 256-64	0.219	0.024	0.184	1.016	12.081
7 crops					
ODGI-yt 512-256	0.475	0.071	2.427	17.064	25.220
ODGI-tt 512-256	0.440	0.062	1.287	7.719	14.919
ODGI-yt 256-128	0.313	0.035	0.639	4.725	19.286
ODGI-tt 256-128	0.317	0.044	0.337	1.933	12.434
ODGI-tt 256-64	0.226	0.025	0.191	1.046	12.111
8 crops					
ODGI-yt 512-256	0.484	0.071	2.551	17.756	25.089
ODGI-tt 512-256	0.448	0.063	1.458	8.418	15.841
ODGI-yt 256-128	0.318	0.036	0.665	4.899	19.212
ODGI-tt 256-128	0.324	0.044	0.365	2.095	12.115
ODGI-tt 256-64	0.231	0.025	0.199	1.160	11.838
9 crops					
ODGI-yt 512-256	0.490	0.072	2.685	18.474	25.825
ODGI-tt 512-256	0.453	0.063	1.602	9.121	16.609
ODGI-yt 256-128	0.321	0.036	0.695	5.064	19.038
ODGI-tt 256-128	0.328	0.044	0.394	2.260	12.606
ODGI-tt 256-64	0.234	0.025	0.205	1.121	12.486
10 crops					
ODGI-yt 512-256	0.493	0.072	2.824	19.150	26.388
ODGI-tt 512-256	0.457	0.063	1.705	9.806	17.015
ODGI-yt 256-128	0.323	0.036	0.725	5.233	19.120
ODGI-tt 256-128	0.331	0.044	0.425	2.420	12.268
ODGI-tt 256-64	0.237	0.025	0.213	1.169	12.354

(b) Results for ODGI-yt and ODGI-tt methods

Table 7: Detailed results for each number of crops $\gamma_1 \in [1, 10]$ for experiments on the SDD dataset with YOLO and tiny-YOLO based models.

	map@0.5	map@0.75	CPU (s)	Pi (s)	GPU (ms)
MobileNet-100 1024	0.415	0.061	1.991	17.346	23.102
MobileNet-100 512	0.266	0.028	0.457	4.014	10.980
MobileNet-100 256	0.100	0.009	0.115	0.923	9.494
MobileNet-35 1024	0.411	0.054	0.838	6.792	13.911
MobileNet-35 512	0.237	0.026	0.189	1.506	9.810
MobileNet-35 256	0.067	0.007	0.050	0.419	9.320

(a) YMobileNet-35 and MobileNet-100 baselines results

	map@0.5	map@0.75	CPU (s)	Pi (s)	GPU (ms)
1 crop					
ODGI-100-35 512-256	0.217	0.033	0.512	4.424	19.676
ODGI-35-35 512-256	0.181	0.031	0.242	1.992	17.518
ODGI-100-35 512-128	0.160	0.026	0.457	4.170	19.587
ODGI-35-35 512-128	0.138	0.021	0.205	1.685	17.202
ODGI-100-35 256-128	0.122	0.018	0.126	1.099	17.215
ODGI-35-35 256-128	0.103	0.015	0.065	0.616	17.679
2 crops					
ODGI-100-35 512-256	0.293	0.041	0.562	4.864	20.201
ODGI-35-35 512-256	0.269	0.046	0.289	2.370	17.573
ODGI-100-35 512-128	0.226	0.033	0.471	4.262	19.944
ODGI-35-35 512-128	0.205	0.030	0.218	1.771	17.345
ODGI-100-35 256-128	0.188	0.025	0.138	1.183	17.501
ODGI-35-35 256-128	0.159	0.020	0.076	0.695	17.429
3 crops					
ODGI-100-35 512-256	0.345	0.047	0.613	5.281	19.760
ODGI-35-35 512-256	0.332	0.053	0.335	2.766	17.939
ODGI-100-35 512-128	0.272	0.038	0.483	4.367	19.656
ODGI-35-35 512-128	0.252	0.034	0.234	1.870	17.351
ODGI-100-35 256-128	0.227	0.025	0.152	1.276	17.514
ODGI-35-35 256-128	0.195	0.025	0.089	0.766	17.331
4 crops					
ODGI-100-35 512-256	0.380	0.051	0.657	5.728	20.100
ODGI-35-35 512-256	0.378	0.057	0.390	3.184	17.430
ODGI-100-35 512-128	0.301	0.040	0.495	4.489	19.958
ODGI-35-35 512-128	0.287	0.038	0.246	1.983	17.096
ODGI-100-35 256-128	0.255	0.032	0.165	1.378	17.946
ODGI-35-35 256-128	0.218	0.027	0.101	0.853	17.520
5 crops					
ODGI-100-35 512-256	0.405	0.053	0.712	6.178	20.189
ODGI-35-35 512-256	0.410	0.060	0.446	3.642	17.883
ODGI-100-35 512-128	0.329	0.043	0.507	4.594	20.164
ODGI-35-35 512-128	0.314	0.041	0.257	2.078	17.225
ODGI-100-35 256-128	0.277	0.034	0.177	1.477	17.343
ODGI-35-35 256-128	0.236	0.028	0.115	0.933	17.360

	map@0.5	map@0.75	CPU (s)	Pi (s)	GPU (ms)
6 crops					
ODGI-100-35 512-256	0.425	0.055	0.763	6.631	19.893
ODGI-35-35 512-256	0.434	0.061	0.499	4.086	17.832
ODGI-100-35 512-128	0.351	0.044	0.512	4.686	20.192
ODGI-35-35 512-128	0.329	0.043	0.270	2.163	17.453
ODGI-100-35 256-128	0.294	0.036	0.190	1.553	17.623
ODGI-35-35 256-128	0.250	0.029	0.127	1.007	17.396
7 crops					
ODGI-100-35 512-256	0.437	0.056	0.785	7.096	20.020
ODGI-35-35 512-256	0.450	0.062	0.543	4.519	17.639
ODGI-100-35 512-128	0.367	0.046	0.531	4.785	19.998
ODGI-35-35 512-128	0.346	0.044	0.283	2.250	17.111
ODGI-100-35 256-128	0.308	0.037	0.203	1.653	18.121
ODGI-35-35 256-128	0.260	0.030	0.141	1.096	17.340
8 crops					
ODGI-yt 512-256	0.484	0.071	3.16	18.26	27.1
ODGI-100-35 512-256	0.448	0.056	0.842	7.553	19.844
ODGI-35-35 512-256	0.461	0.063	0.603	4.956	17.764
ODGI-100-35 512-128	0.380	0.047	0.545	4.880	19.996
ODGI-35-35 512-128	0.361	0.045	0.295	2.359	17.640
ODGI-100-35 256-128	0.318	0.038	0.213	1.751	17.258
ODGI-35-35 256-128	0.269	0.031	0.153	1.193	16.919
9 crops					
ODGI-100-35 512-256	0.455	0.057	0.892	7.980	20.206
ODGI-35-35 512-256	0.467	0.063	0.636	5.394	18.002
ODGI-100-35 512-128	0.392	0.047	0.558	4.981	20.516
ODGI-35-35 512-128	0.374	0.046	0.305	2.460	17.501
ODGI-100-35 256-128	0.327	0.039	0.229	1.861	17.746
ODGI-35-35 256-128	0.272	0.032	0.165	1.288	17.814
10 crops					
ODGI-100-35 512-256	0.460	0.057	0.936	8.423	20.717
ODGI-35-35 512-256	0.475	0.064	0.704	5.863	18.721
ODGI-100-35 512-128	0.401	0.048	0.569	5.080	20.067
ODGI-35-35 512-128	0.383	0.047	0.314	2.559	17.629
ODGI-100-35 256-128	0.333	0.039	0.242	1.960	17.833
ODGI-35-35 256-128	0.277	0.032	0.178	1.382	17.810

(b) Results for ODGI-100-35 and ODGI-35-35 methods

Table 8: Detailed results for each number of crops $\gamma_1 \in [1, 10]$ for experiments on the SDD dataset with MobileNet based models.

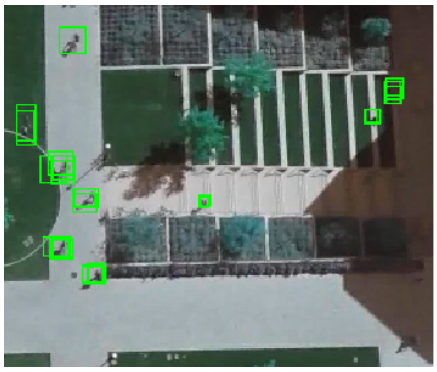
(a) Ground-truth

(b) ODGI stage 1: detected object boxes (cyan: confidence above τ_{high})

(c) ODGI stage 1: detected group boxes

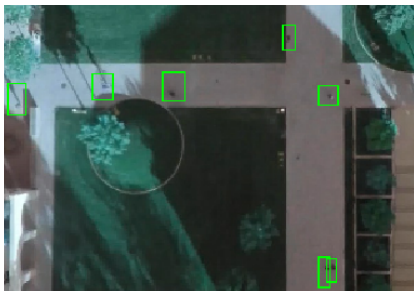
(d) ODGI stage 1: regions passed to stage 2

(e) ODGI stage 2: detected object boxes

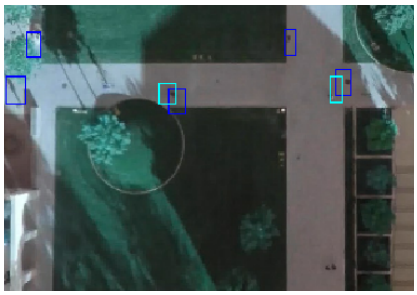


(f) ODGI: overall detected object boxes

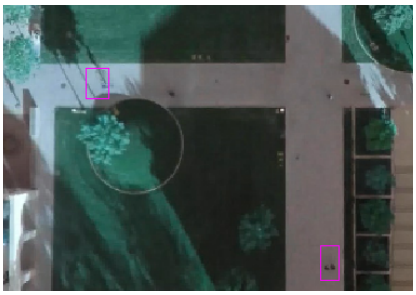
Figure 4: Qualitative results for ODGI. No filtering step was applied here, but for readability we only display boxes predicted with confidence at least 0.5. Best seen on PDF with zoom. Additional figures are provided in the supplemental material.



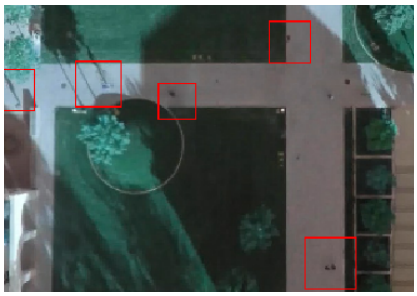
(a) Ground-truth



(b) ODGI stage 1: detected object boxes (cyan: confidence above τ_{high})

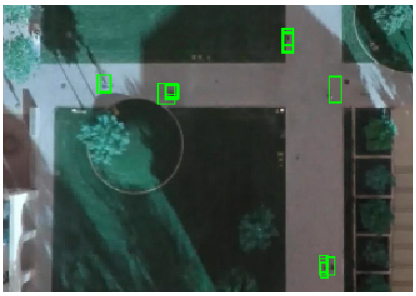


(c) ODGI stage 1: detected group boxes



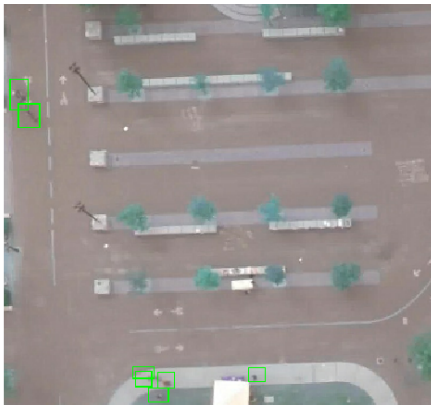
(d) ODGI stage 1: regions passed to stage 2

(e) ODGI stage 2: detected object boxes



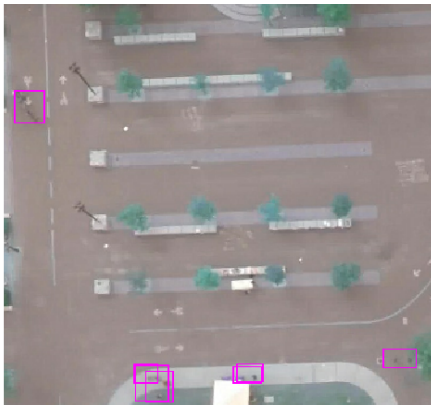
(f) ODGI: overall detected object boxes

Figure 5: Qualitative results for ODGI (continued).



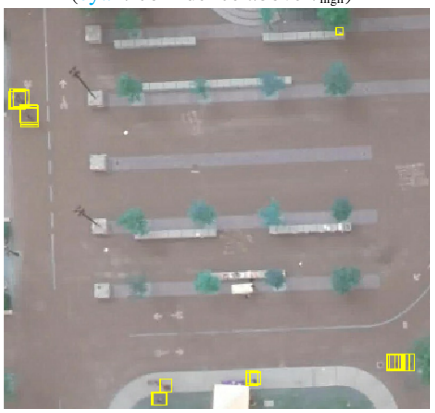
(a) Ground-truth

(b) ODGI stage 1: detected object boxes (cyan: confidence above τ_{high})

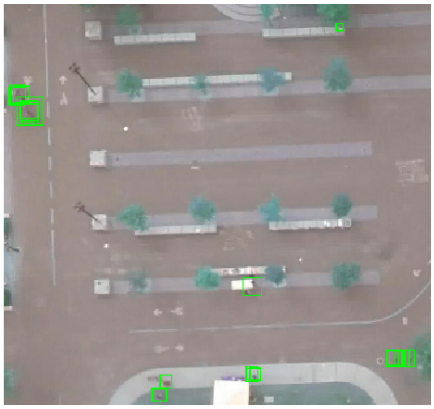


(c) ODGI stage 1: detected group boxes

(d) ODGI stage 1: regions passed to stage 2



(e) ODGI stage 2: detected object boxes



(f) ODGI: overall detected object boxes

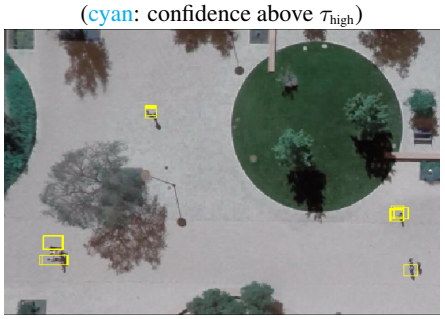
Figure 6: Qualitative results for ODGI (continued).

(a) Ground-truth

(b) ODGI stage 1: detected object boxes (cyan: confidence above τ_{high})

(c) ODGI stage 1: detected group boxes

(d) ODGI stage 1: regions passed to stage 2



(e) ODGI stage 2: detected object boxes

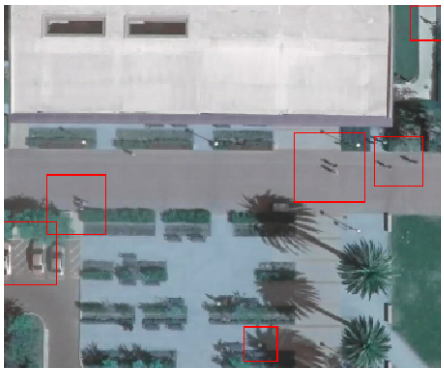
(f) ODGI: overall detected object boxes

Figure 7: Qualitative results for ODGI (continued).

(a) Ground-truth

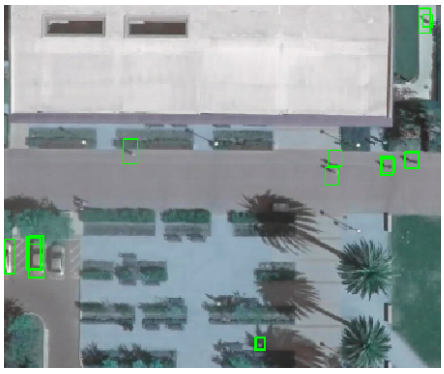
(b) ODGI stage 1: detected object boxes (cyan: confidence above τ_{high})

(c) ODGI stage 1: detected group boxes



(d) ODGI stage 1: regions passed to stage 2

(e) ODGI stage 2: detected object boxes



(f) ODGI: overall detected object boxes

Figure 8: Qualitative results for ODGI (end).