Low Altitude Image Analysis using Panoptic Segmentation

Emmanouil Christakis, Zisis Batzos, Konstantinos Konstantoudakis,
Kiriaki Christaki, Prodromos Boutis, Dimitrios Sainidis, Dimitrios Tsiakmakis,
Georgios Almpanis, Tasos Dimou, Petros Daras
Visual Computing Lab, Information Technologies Institute
6^{th} Km Charilaou-Thermi,, Thessaloniki, Greece
{manchr, zisisbatzos, k.konstantoudakis, kchristaki,
prod, dsainidis, tsiakmakis, galbanis, dimou, daras}@iti.gr

Abstract

In this report we present the submission of the VCL-CERTH team in the Trecvid 2021 Disaster Scene Description and Indexing (DSDI) task. The dataset provided for this task, LADI, contains only labels as ground truth data indicating the presence or absence of each of the 32 features of interest in the images of the dataset. However, aerial images are often captured from high altitude and as such the features that the systems participating in the task are asked to detect, often appear tiny in an image. That being the case, we believe that just a label indicating the presence of the feature as ground truth is not sufficient to guide the system to detect this feature. For this reason we opted to approach the task as a panoptic segmentation one for the vast majority of the 32 features. Since a panoptic segmentation network can not be trained on image labels we had to manually create segmentation annotations for a small part of the LADI images ourselves and train a panoptic segmentation networks using these annotations.

1. Introduction

The purpose of the Trecvid [1] DSDI task was the detection of various features of interest in aerial videos containing natural disasters scenes. These features were 32 in total and could be split into 5 larger categories, namely infrastructure, vehicles, water, environment and damage. More specifically, the participating teams were given a number of short video clips and for each of the 32 features, they had to detect which clips contained this feature and furthermore the clips should be ranked according to the team's confidence that the feature is present at each clip. In order to train their systems, the teams were given a dataset consisting of images taken from the LADI [5] dataset and their corresponding ground truth labels. Given these labels, a straight forward approach to deal with the task at hand, would be to train a multi-label convolutional network based image classifier that would be able, given an image, to output whether this image contains each of the 32 features. However, taking into account the nature of the LADI dataset, containing high altitude aerial images, it is often the case that features like cars, buildings, boats and other objects, appear tiny in the image. We believe that just a label pointing to the presence of such an object in the image is not enough to guide the network to focus on such a small area of the image where the object would be located. Furthermore, label based training could lead to various unwanted biases and associations of the network. As an example when the network detects a boat it could output that a water related feature is also detected, even if this is not necessarily the case, since in the training dataset boats and water are most likely both present in an image. These kind of generalizations could come up for various other feature pairs like cars and roads. To avoid such issues and to give the trained system the ability to detect small size objects, we opted to take a different route than a multi-label classifier. Instead, we chose to view the task as an object detection and semantic segmentation problem. Many of the 32 features can be thought of as ”things”, meaning that they can be considered as separate countable objects that be clearly located in the image. Such features include cars, trucks, boats, water towers and others. The detection of ”things” can be accomplished by object detection networks. For other features, while clearly located in the image, an instance count is not meaningful. For example one can point where ”grass” is located in the image but can not count how many instances of ”grass” exist in the image. That is the case for various other features like roads, trees, ocean etc, and they can be can be thought of as ”stuff”. To locate these features in an image a semantic segmentation network can be utilized. In semantic segmentation each pixel in an image is given a label based on the class it belongs. However, the task of segmenting ”things” and

”stuff” in an image using a single network is called panoptic segmentation and there are various architectures in the literature that perform well on this task. Using a single network to segment both kinds of objects, offers faster network training and inference time. For these reasons we chose to handle ”things” and ”stuff” using a panoptic segmentation network. To train such a network, detailed instance and semantic segmentation annotations have to provided as ground truth during the training process. These annotations are not available in the LADI dataset and we had to create them from scratch. Producing such annotations is very time consuming and we were only able to annotate only a small number of the images in the LADI dataset. Finally, there is a third category of features that are both uncountable and can not be precisely located in the image. These belong in the damages category including flooding, landslide, rubble, damage (misc), smoke / fire. For these while one can determine their presence in the image, it can not be determined which pixels exactly depict these features. These features can not be detected by the panoptic network and we had to use a multi-label classifier to deal with them.

2. Preparing the Dataset

The dataset preparation process involved two separate steps. The first was about the creation of instance and semantic segmentation annotations to train the panoptic segmentation network. The second dealt with the label based annotations for training the classifier that would handle the damage related features.

2.1. Instance and Semantic Annotations

The training of a panoptic segmentation network to detect ”things” and ”stuff” features, required detailed instance and semantic annotations on LADI images. A sample of the annotations we created can be seen in figure 1. As mentioned above, these kind of annotations are very time consuming and we were only able to annotate only 300 images in total. Most of the images were part of LADI. However, since Trecvid 2020 DSDI test video clips were also available for this year’s participating teams, we annotated some video frames extracted from these clips as well.

2.2. Label processing for the Damage Classifier

Regarding the damages super-category, and more specifically the damage flooding, rubble/debris, damage/misc, road washout, smoke/fire, landslide, a multi-label classifier was utilized. Labels for these classes needed to be on image level, as provided by the LADI dataset. In order to optimize the classifier’s performance, we decided to manipulate the dataset. At first, LADI dataset contains labels from different labelers and there are many cases of images that are not label consistent. Some of these cases were reasonable, considering that the abstract nature of these classes creates ambiguity, while other cases were clearly image mislabeling. To counter this label inconsistency issue, we decided to keep only the images of the dataset in which all labelers agreed completely. While using this method cleaned the data quite well, it also led to the side effect of some classes (road washout, smoke/fire, landslide) being underrepresented in the dataset. To resolve this class imbalance issue, for these three classes we decided to loosen the constrain of full agreement between the labelers. In fact, because of the small number of images labeled with these classes, we visually checked every image labeled as road washout, smoke/fire or landslide by at least one labeler. Although the visual inspection provided some more images, it was not enough to counter the class imbalance problem. Considering the large number of images representing the dominant classes, as a final step we dropped from the dataset a number of images labeled only with these dominant classes.

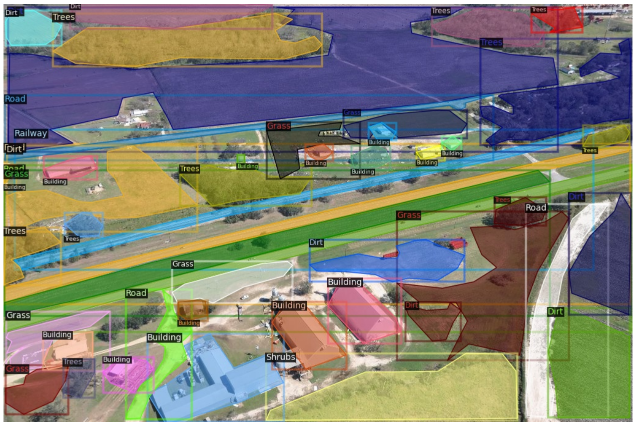

Figure 1. Panoptic network annotation sample.

3. Panoptic network and Image classifier training

Having created the panoptic annotations, we were then able to train the panoptic network. For this task we utilized the Detectron2 [6] library that provides state-of-the-art detection and segmentation algorithms. More specifically, we used the panoptic-FPN [4] architecture. The initial network was trained on the COCO [4] panoptic dataset and we fine tuned on our annotated images. An inference sample on an frame extracted from the 2021 DSDI test clips is shown in figure 2. For the damage classifier we employed a a Resnet-50 [3] classifier, pretrained on the Imagenet [2] dataset and fine-tuned on the damage labels.

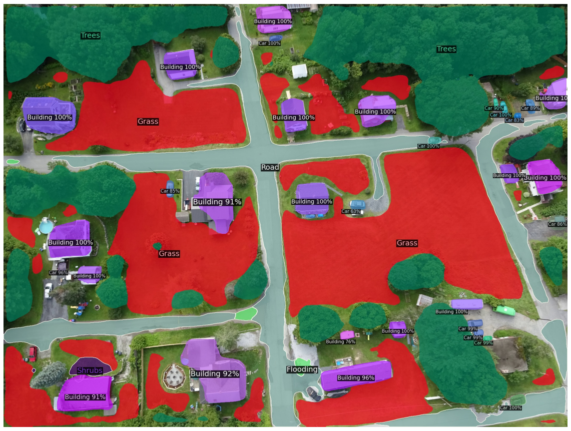

Figure 2. Panoptic network inference sample.

4. Our Submissions

All of our submissions utilized the panoptic network to detect "things" and "stuff" and the damage classifier to detect damages in the video clips. The distinction between them was the use of different criteria to rank the video clips for each feature. Both the panoptic network and the damage classifier perform inference on individual images. Performing inference on every video frame of a clip would be highly redundant and would dramatically slow down the processing time of our system. For this reason we chose to only process one out every 10 video frames. Then we combined the outputs for all the processed frames to draw conclusions for the clip as a whole.

The process for the damage classification of a clip was the same for all our submissions. More specifically the score S^i for a clip in regards to the damage feature i is calculated as

$$S^{i'} = max(\left[S^i_1, .., S^i_k, .., S^i_N\right])$$

$$S^i = \begin{cases} 0 & S^{i'} \leq thr \\ S^{i'} & S^{i'} \geq thr \end{cases}$$

where N is the number of processed frames in the clip, S^i_k is the confidence level of the classifier that the damage feature i is present in the $k \in (0, N)$ processed frame and thr is a predefined constant threshold. The ranking of the clips for this damage feature would be based around this score. If the score was zero we would conclude that this damage feature is not present in the clip.

For the "things" features, in the first submission the score S^j for a clip for the "thing" feature j was calculated as

$$S^{j'} = max(\left[S^j_1, .., S^j_l, .., S^j_M\right])$$

$$S^j = \begin{cases} 0 & S^{j'} \leq thr \\ S^{j'} & S^{j'} \geq thr \end{cases}$$

where M is the total number of instances of the "thing" feature j that were detected in the all the processed frames of the clip, S^j_l is the confidence level of the panoptic network for the detection of the l-th instance of the feature where $l \in (0, M)$. Notice that M can be larger than the number of frames N since many instances of this feature can be detected in each processed frame. For the second submission

$$S^j = sum(\left[S^{j'}_1, .., S^{j'}_l, .., S^{j'}_M\right])$$

$$S^{j'}_l = \begin{cases} 0 & S^j_l \leq thr \\ S^j_l & S^j_l \geq thr \end{cases}$$

where as before S^j_l is the confidence level of the panoptic network for the detection of the l-th instance. So in this submission, we take the confidence levels for all the detected instances of the "thing" feature j that exceed a threshold in the whole clip and sum them up to get a score for the clip in regards to the feature j. The thinking behind this, is that if for example the network detects 10 cars with enough confidence in a clip while in another it only detects 2 cars, we would consider more likely the first clip to contain cars than the second and as such the first clip would have higher score. For the third submission, we would as also consider the pixel area of the detected instances. Instances with larger pixel area would be given a bigger weight in the previous sum given for the second submission. The thinking was that detections with larger pixel area would be more reliable. However, this proved to not be the case as indicated by the results presented in the results section.

Finally the scoring for the "stuff" features was the same for all 3 submissions and given as

$$S^h = sum(\left[area^h_1, .., area^h_k, .., area^h_N\right])$$

where S^h is the score of a clip for the "stuff" feature h and $area^h_k$ is the pixel area of this feature in the k-th processed frame where $k \in (0, N)$. Again here the justification for this score, is that in a clip where a "stuff" feature covers a large pixel area for many frames in the clip should score higher (for this feature) than one where where this feature covers only a small pixel area. A difference between the first and the second submission was that in the second only pixels segmented with a confidence level above a certain threshold would contribute to the pixel area of the of the stuff feature.

5. Results and analysis

In table 1 the submission type, submission name and the mean average precision achieved for all participating submissions is presented. "O" type submissions utilized extra data not provided by the DSDI track while the "L" ones only

Submission Type	Submission Name	mAP
O	FIU_UM 3	0.339
O	FIU_UM 2	0.331
O	FIU_UM 4	0.298
L	VCL_CERTH 2	0.282
L	VCL_CERTH 1	0.268
L	FIU_UM 1	0.268
L	FIU_UM 2	0.254
L	FIU_UM 3	0.250
L	VCL_CERTH 3	0.211
L	BUPT_MCPRL 2	0.159
L	BUPT_MCPRL 1	0.129

Table 1. Submission type, submission name and mAP achieved for all participating submissions.

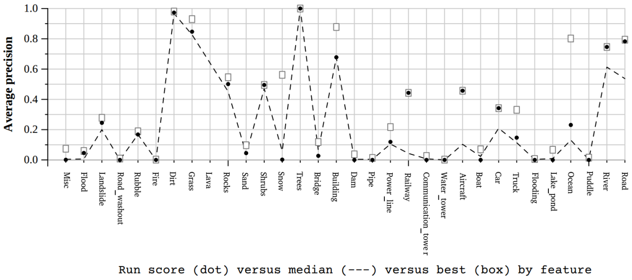

Figure 3. VCL-CERTH Submission 1 average precision against median and best for all 32 features.

used these data. Among ”L” submissions VCL_CERTH 3 achieves the highest mAP. In figures 3, 4, 5 you can see the average precision achieved by our submissions for each of the 32 features against the best and the median average precision among all submissions. In general, we noticed that the panoptic network performed better for low altitude images where objects and stuff were clearly visible and easily separated. For high altitude images it was challenging even for the annotators to properly segment the images in the dataset. Another issue with our approach was the small number of images we were able to annotate, which were only 300 in total. This resulted in many features having too few samples, for example communications and water towers. For these we generally under performed, however this was an issue for the other teams as well ,indicated by the low average precision of all submissions for these rare classes. For a number of features like ”Car”,”Aircraft”,”Road” we achieved the best average precision. Our third submission performed poorly against the other two. The hypothesis that detections with larger pixel areas would be reliable is not validated by the results. Among the first 2 submissions there are negligible changes in the overall performance and the idea of ranking a clip with many instances higher than one were only a few appear had mixed results, boosting for example the performance for the ”Building” feature while degrading it for the ”Car” feature.

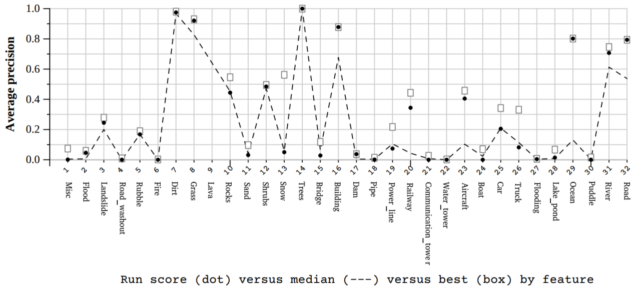

Figure 4. VCL-CERTH Submission 2 average precision against median and best for all 32 features.

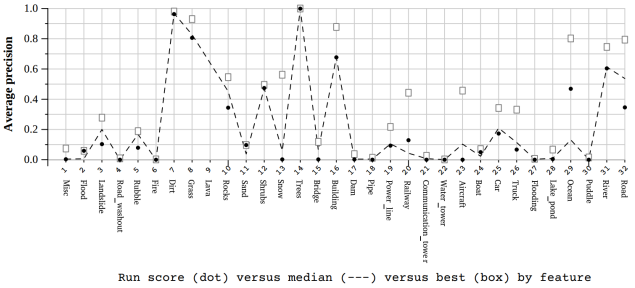

Figure 5. VCL-CERTH Submission 3 average precision against median and best for all 32 features.

6. Conclusion

While for the DSDI track, the dataset provided was in the form of labels and a multi label classifier would be the obvious approach, we opted, for most the 32 features of interest, to view the problem as a panoptic segmentation one. The features could be split into ”things”, ”stuff” and damages. ”Things” and ”stuff” were detected using a panoptic segmentation network, while damage features where detected using a ResNet multi label classifier. To train the panoptic network we had to create our own instance and semantic segmentation annotations for a small number (300 in total) of the LADI images. To train the classifier we used the labels provided by Trecvid, after processing them to filter out inconsistent labels. Despite the small number of panoptic annotations we trained our system on, our submissions performed competitively. Generally, the panoptic network performed better for low altitude images. We are of the opinion that panoptic segmentation is a promising approach to the DSDI task and if we were able to annotate more images to train our system, we believe we could significantly boost the performance, especially if more samples of rare classes were included in the annotated images.

References

[1] G. Awad, A. A. Butt, K. Curtis, J. Fiscus, A. Godil, Y. Lee, A. Delgado, J. Zhang, E. Godard, B. Chocot, L. Diduch,

J. Liu, Y. Graham, G. J. F. Jones, , and G. Quénot. Evaluating multiple video understanding and retrieval tasks at trecvid 2021. In *Proceedings of TRECVID 2021*. NIST, USA, 2021.

[2] J. Deng, W. Dong, R. Socher, L. 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, 2009.

[3] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[4] A. Kirillov, R. B. Girshick, K. He, and P. Dollár. Panoptic feature pyramid networks. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6392–6401, 2019.

[5] J. Liu, D. Strohschein, S. Samsi, and A. Weinert. Large scale organization and inference of an imagery dataset for public safety. In *2019 IEEE High Performance Extreme Computing Conference (HPEC)*, pages 1–6, Sep. 2019.

[6] Y. Wu, A. Kirillov, F. Massa, W.-Y. Lo, and R. Girshick. Detectron2. `https://github.com/facebookresearch/detectron2`, 2019.