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
6th Km Charilaou-Thermi,, Thessaloniki, Greece
{manchr, zisisbatzos, k.konstantoudakis, kchristaki,
prod, dsainidis, tsiakmakis, galbanis, dimou, daras}@iti.gr

Disaster Scene Description and Indexing task

- **Objective**: Detection of features of interest in aerial videos containing natural disasters scenes.

- 32 features, split into 5 categories infrastructure, vehicles, water, environment and damage.

- The teams were given short video clips and for each of the 32 features, the clips should be ranked according to the team's confidence that the feature is present at each clip.

- To train their systems, the teams were given a dataset consisting of images taken from the LADI dataset and their corresponding multilabel ground truth labels.

Panoptic Segmentation Motivation

- Given the ground truth labels, a straightforward approach would be to train a multi label convolutional classifier.

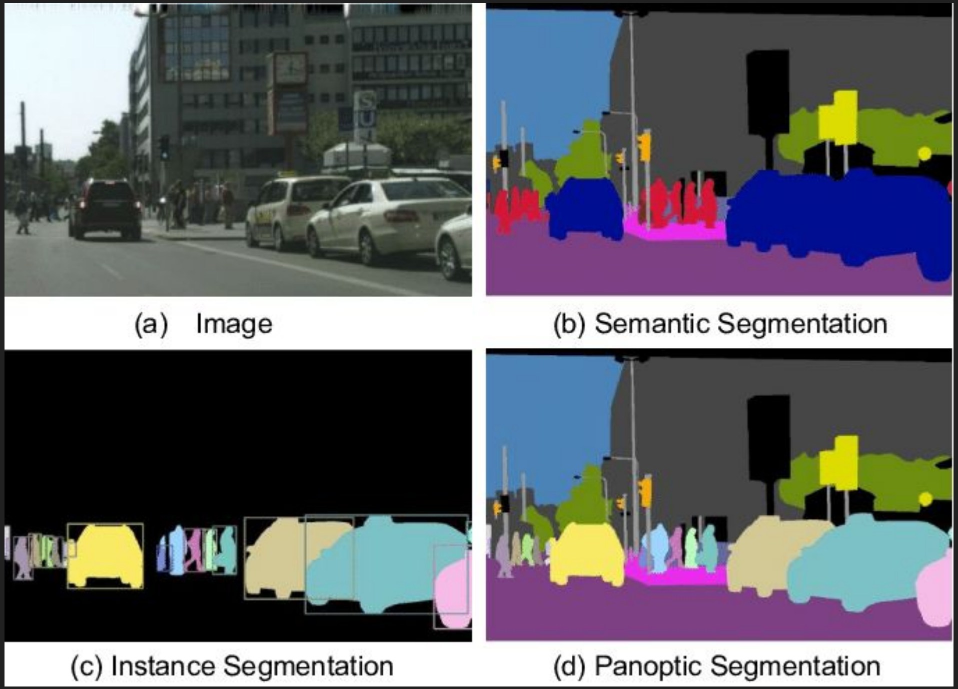
- However:
  - High altitude images: cars, buildings, boats and other objects, often appear tiny.
  - Just a label is not enough to guide the network to focus on such a small area.
  - Various unwanted associations: if the network detects a boat it could output that a water related feature is also detected since in the training dataset boats and water are most likely both present in an image.

Panoptic Segmentation

- We chose to view the task as an object detection and semantic segmentation problem.
- The 32 features can be split on whether they are countable and clearly located in the image.
- “Things” are countable and clearly located, “stuff” are uncountable but clearly located and damage related features are both uncountable and non clearly located.
- “Things” can be dealt with object detection and “stuff” with semantic segmentation.
- For damage related features we utilize convolutional classifiers

Instance vs Semantic vs Panoptic Segmentation

- Panoptic segmentation combines instance and semantic segmentation



(a) Image

(b) Semantic Segmentation

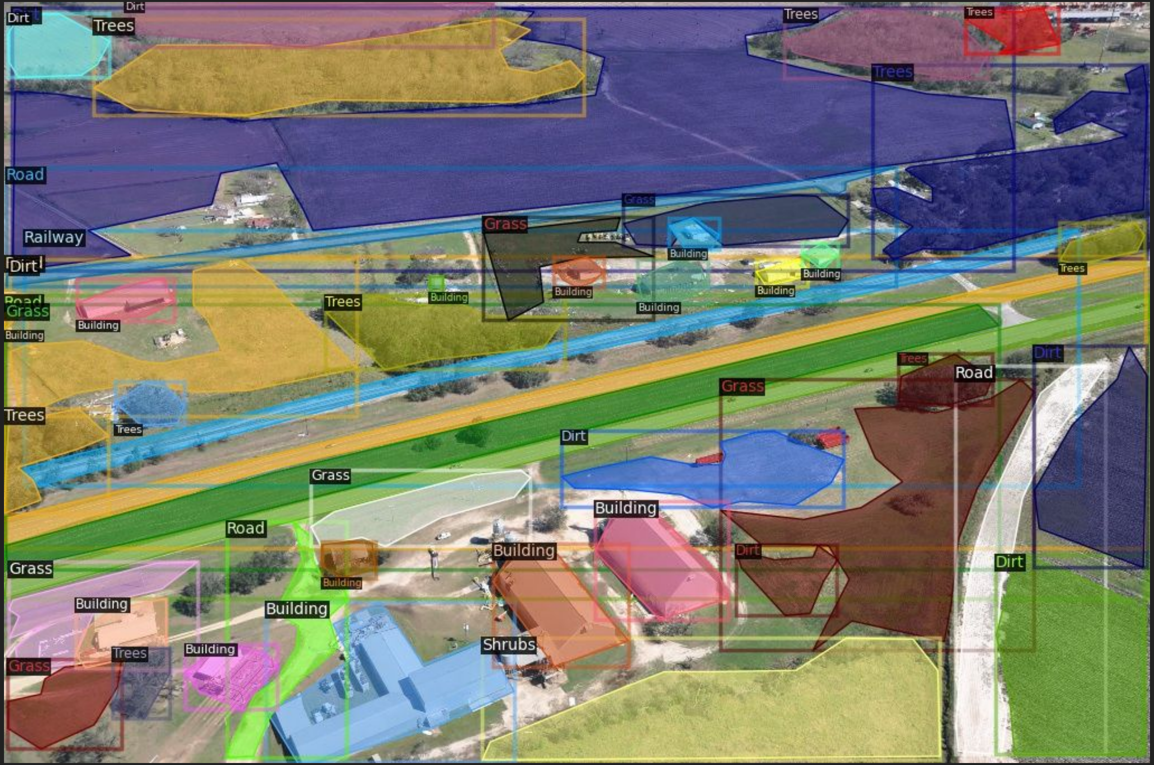
(c) Instance Segmentation

(d) Panoptic Segmentation

Preparing the Dataset

- Instance and Semantic Annotations
  - Panoptic segmentation requires instance and semantic annotations.
  - LADI only includes labels for ground truth.
  - We had to create our own annotations to train the panoptic network
  - Very time consuming, only a few LADI images annotated and some frames from DSDI 2020 test video clips, 300 in total.

Panoptic annotation sample



Preparing the Dataset

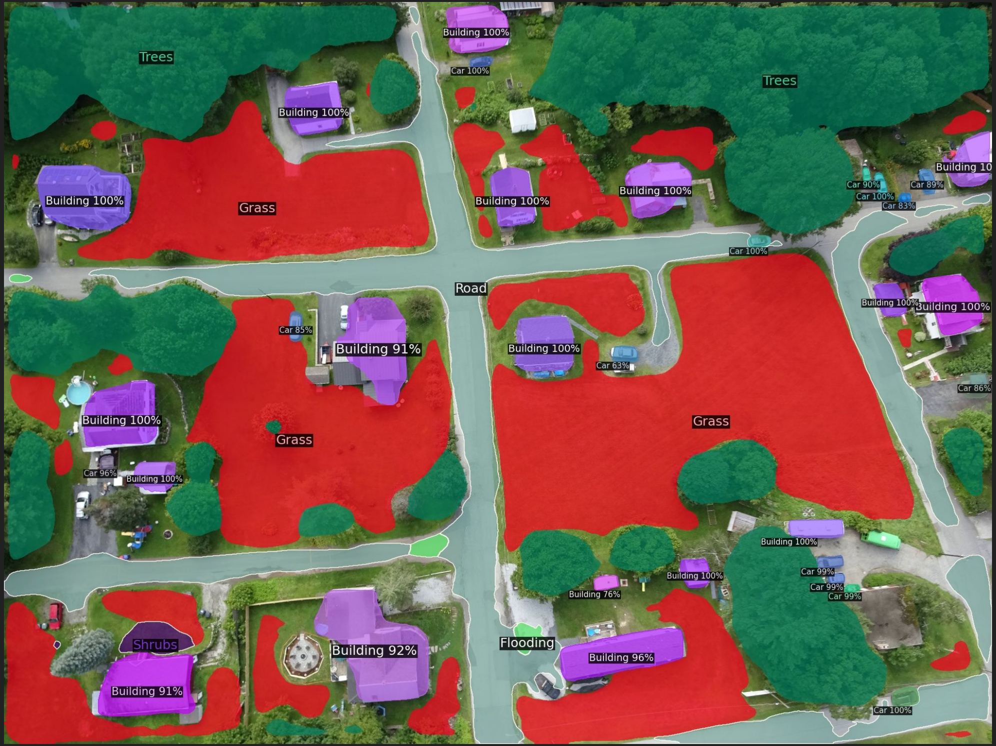
- Label processing for the Damage Classifier
  - LADI contains labels from different labelers and there are many cases of images that are not label consistent across the labelers.
  - First, keep only images in which all labelers agreed unanimously for the damage features.
  - This led to some classes (road washout, smoke/fire, landslide) being underrepresented.
  - Included images with non unanimous agreement after visually inspecting them.
  - Dropped from the dataset a number of images labeled only with the dominant classes.

Panoptic Network and Damage Classifier Training

- Panoptic Network
  - Used our own annotations on 300 images
  - Utilized Facebook’s Detectron2 library which provides state-of-the-art segmentation algorithms
  - Fine tuned the panoptic-FPN architecture, pretrained on the COCO panoptic dataset.

- Damage Classifier
  - we employed a a Resnet-50 classifier,
  - Pretrained on the Imagenet dataset and fine-tuned on the damage labels

Panoptic Network Inference Sample



Submissions

- All submissions used the panoptic network for ”things” and ”stuff” and the damage classifier for damages.

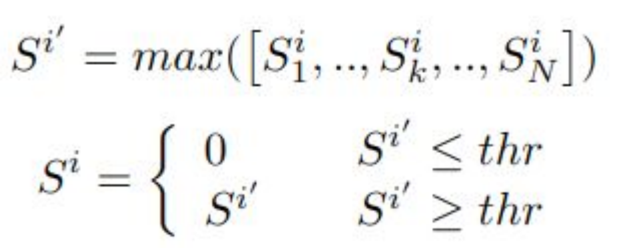
- The distinction was the use of different criteria to rank the video clips for each feature.

- For performance reasons we chose to only process one out every 10 video frames in each test video clip.

- We combined the outputs for all the processed frames to draw conclusions for the clip as a whole.

Damage Features Scoring

- The process for the damage classification of a clip was the same for all our submissions.
- The score S of a video clip in regards to a feature i would be given as



$$S^{i'} = max([S_1^i, .., S_k^i, .., S_N^i])$$

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

where N is the number of the processed frames.

“Things” Scoring

- For the first submission the score of a clip for the “thing” feature j was given as shown where M is the number of instances of this feature detected across all processed frames. We take the max score and If it exceeds a threshold we give this score to this clip.

$$S^{j'} = max(\left[S_1^j, ..., S_l^j, ..., S_M^j\right])$$

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

- For the second we again take the same scores as the first but now we add them up to produce the score. For example if the network detects 10 cars with enough confidence in a clip while in another it only detects 2 cars, the first would have a higher score for the car feature.

$$S^j = sum(\left[S_1^{j'}, ..., S_l^{j'}, ..., S_M^{j'}\right])$$

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

- For the third we also took into account the pixel area of the detected instances thinking that the detection of larger instances would be more reliable.

“Stuff” Scoring

- The scoring for the ”stuff” features was almost the same for all 3 submissions.
- The score of a clip for a stuff feature h was given as

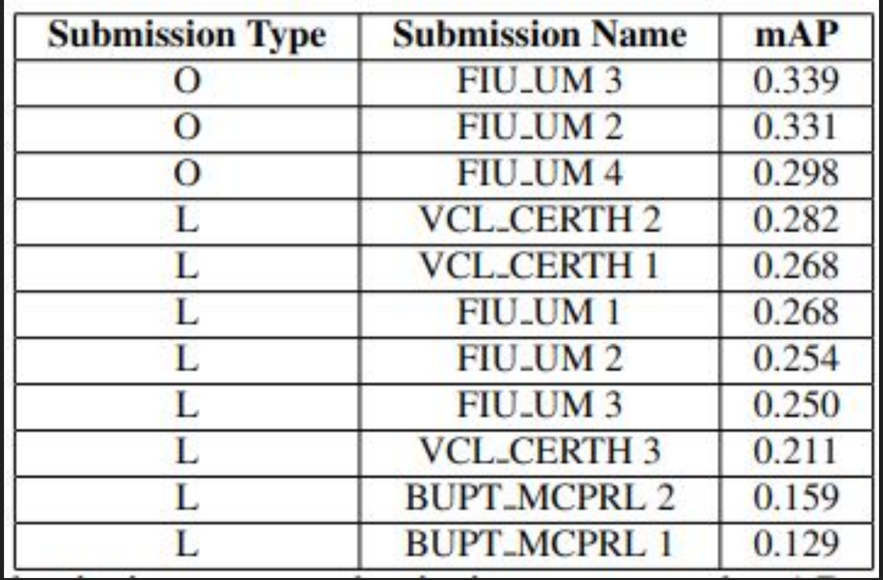
$$S^h = sum([area_1^h, .., area_k^h, .., area_N^h])$$

where $area^h_i$ is the pixel area of this feature in the i-th processed frame.

- This way if the total pixel area of a stuff feature across all processed frames was larger in clip than in another the first clip would get a higher score for this feature.

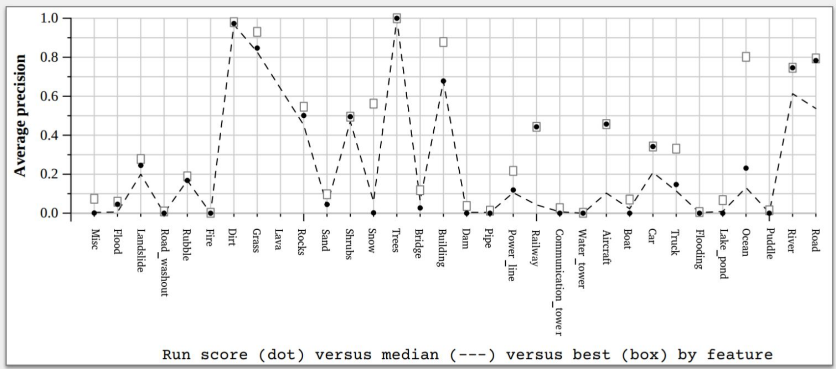
- A minor distinction between the first and the second submission was that in the second only pixels with a classification score over a threshold would contribute to the above sum.

Mean Average Precision Results

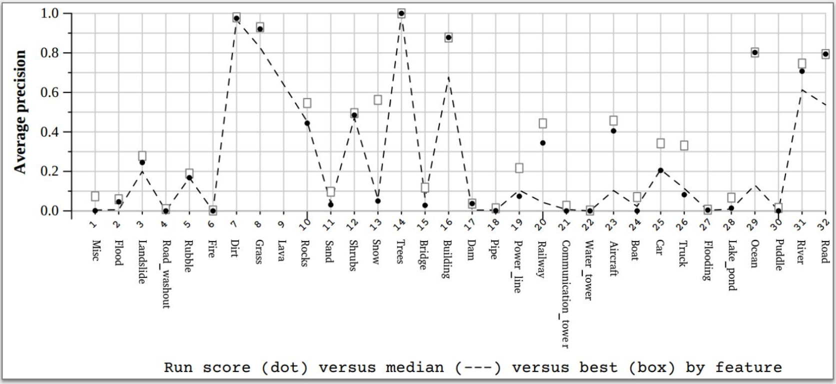


Submission Type	Submission Name	mAP
O	FIU_UM 3	0.339
O	FIU_UM 2	0.331
O	FIU_UM 4	0.298
L	VCL_CERTH 2	0.282
L	VCL_CERTH 1	0.268
L	FIU_UM 1	0.268
L	FIU_UM 2	0.254
L	FIU_UM 3	0.250
L	VCL_CERTH 3	0.211
L	BUPT_MCPRL 2	0.159
L	BUPT_MCPRL 1	0.129

Average Precision per Feature



Run score (dot) versus median (---) versus best (box) by feature



Run score (dot) versus median (---) versus best (box) by feature

Conclusions

- The panoptic network performed better for low altitude images where objects and stuff were clearly visible and easily separated.
- Time consuming annotation, only 300 images used to train the panoptic network
- This had the result of too few samples for various features like communications and water towers.
- Despite the small number of annotated images, we achieve competitive results.
- Given a more sufficient number of images annotated with instance and semantic annotations, we believe that panoptic segmentation is a promising approach for the DSDI task.