



# Visual Computing Lab @ TRECVID 2020

## Disaster Scene Description and Indexing (DSDI)

Emmanouil Christakis, Stefanos Demertzis, Konstantinos Stavridis, Athanasios Psaltis, **Anastasios (Tassos) Dimou**, Petros Daras

*Visual Computing Lab - Information Technologies Institute*

# Preparing the Dataset

- ➔ The given dataset contained images of very high resolution.
- ➔ Downsized  images to (224x224) due to computational constraints.
- ➔ Assumed the feature was present if at least 1 worker annotated it
- ➔ Create a frame export strategy (2 - 4 frames per given video clip)
  - ◆ 1 out of 3 frames if clip had less than 10 fps
  - ◆ 1 out of 100 frames if the clip had less than 20 fps
  - ◆ 1 out of 200 frames if the clip had more or equal to 20 fps

# Preparing the Dataset: Object-based Annotations

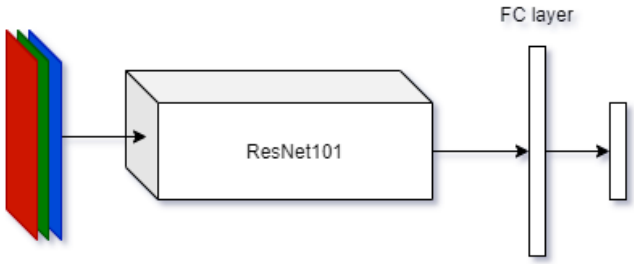
- ➔ The dataset was annotated with 32 labels.
- ➔ We argue that the localization of core image elements, would lead to improved performance in overall scene understanding.
- ➔ Opted to train object detection networks.
- ➔ Required annotation effort from our part with bounding boxes for a small portion of the dataset

# Our Submissions

- ➔ Five Independent ResNet101 Classifiers (L_VCL_1)
- ➔ A multi task classifier (L_VCL_5)
- ➔ Classifiers for concept / Object Detection for objects (O_VCL_3)
- ➔ Classifiers over Faster R-CNN Features Map (O_VCL_6)
- ➔ Classifiers over Faster R-CNN Features Map with attention (O_VCL_2)
- ➔ Five Independent ResNet101 Classifiers with attention (O_VCL_4)

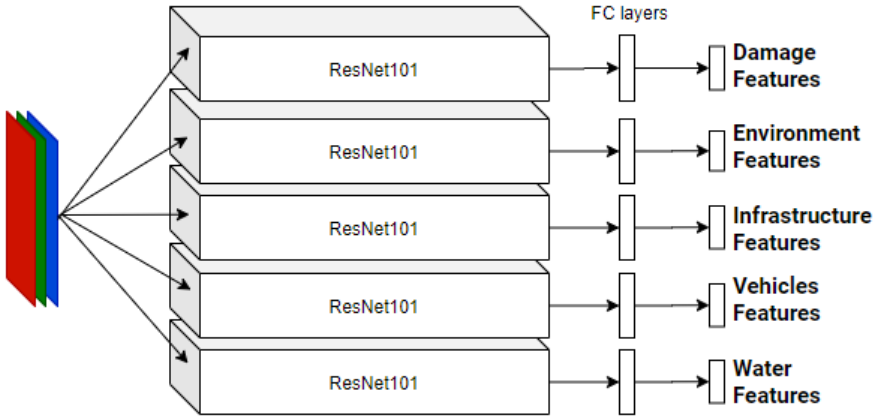
# Five Independent ResNet101 Classifiers (L_VCL_1)

- ➔ Five different Resnet101 classifiers, one for each category
- ➔ The original dataset was *split into 5 smaller parts*, each one:
  - ◆ contained only images along with their annotations for one of the 5 categories
  - ◆ used to train a different classifier



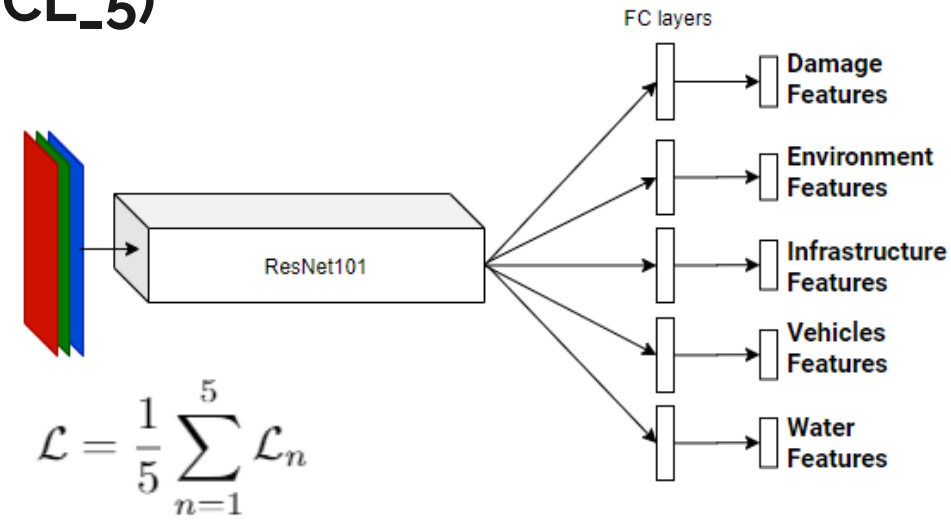
# Five Independent ResNet101 Classifiers (L_VCL_1)

- ➔ The classifiers were fine-tuned on the 5 smaller datasets
- ➔ We regard this approach as our baseline with which to compare the other submissions



# A multi task classifier (L_VCL_5)

- ➔ Common ResNet101 backbone
- ➔ 5 separate classifier heads on top
- ➔ Calculate the loss for each task
- ➔ Aggregate Losses on a global loss
- ➔ Perform back propagation once

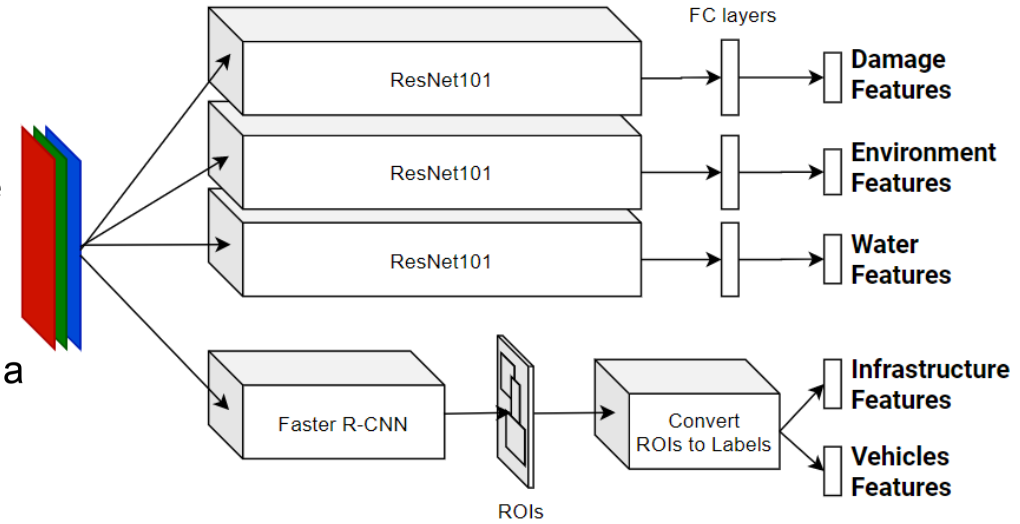



$$\mathcal{L} = \frac{1}{5}\sum_{n=1}^{5}\mathcal{L}_n$$

The average precision drops substantially using this approach.

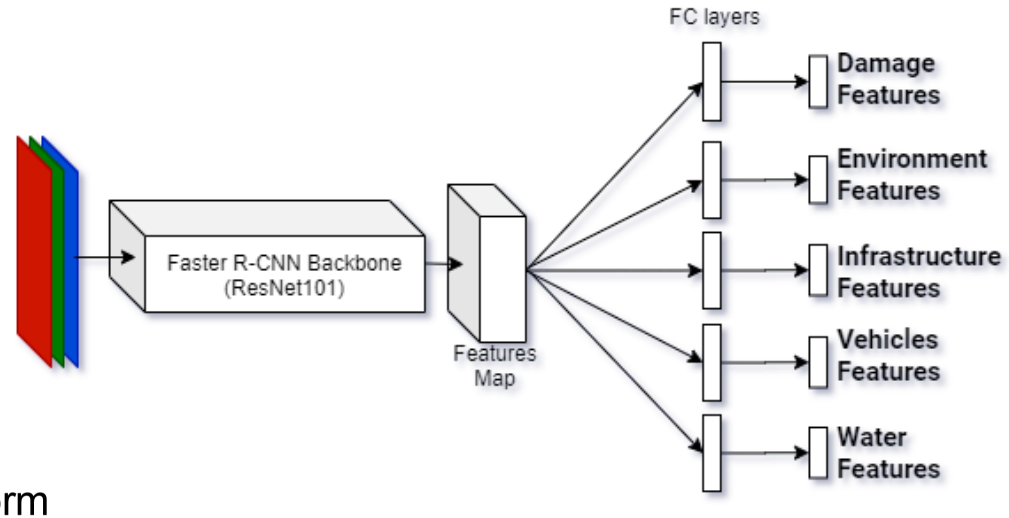
# Classifiers for concept / Object Detection for objects (O_VCL_3)

- ➔ Utilize the formed object localization annotations
- ➔ Faster R-CNN object detector, using the object-specific annotations
- ➔ Convert the output of Faster R-CNN to predict only the presence or absence of a feature
- ➔ For the features in the three concept categories we use the classifiers implemented in L_VCL_1.



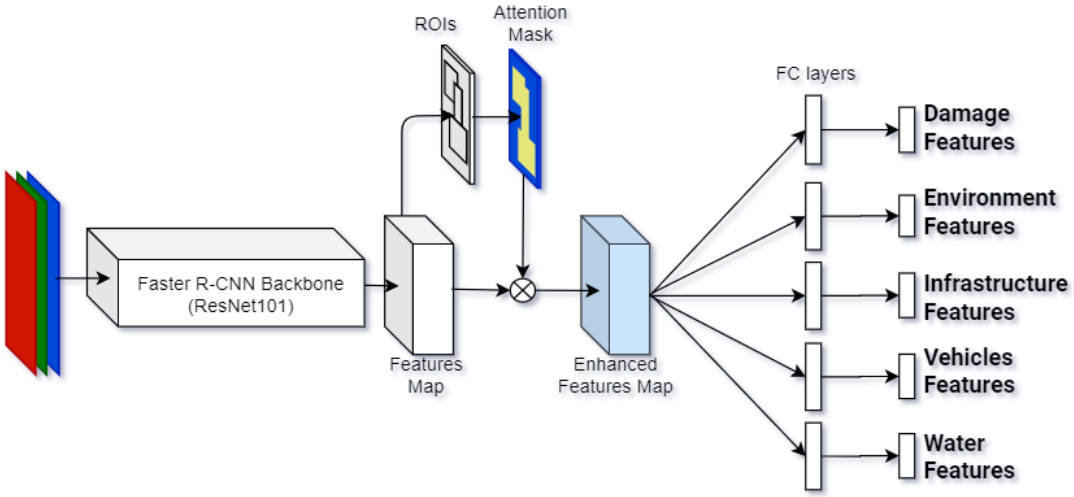
# Classifiers over Faster R-CNN Features Map
## (O_VCL_6)

- ➔ Common ResNet101 backbone
- ➔ 5 separate classifier heads on top
- ➔ Using the Resnet101 backbone from trained Faster R-CNN from O_VCL_3
- ➔ 5 dataloaders to iterate over datasets separately
- ➔ Calculate the loss for each task / Perform back propagation 5 Times



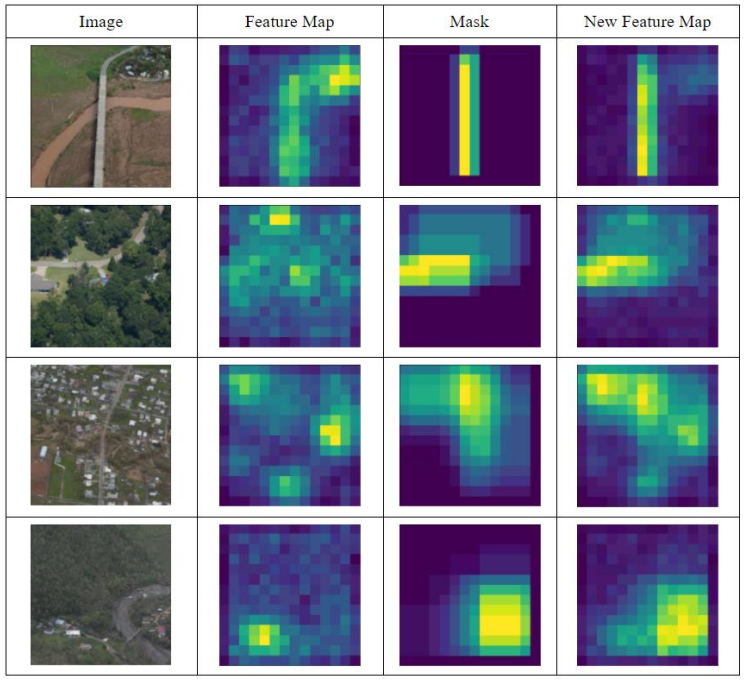
# Classifiers over Faster R-CNN Features Map with attention (O_VCL_2)

- ➔ Create attention masks from the detected bounding boxes
- ➔ Apply these on masks on the base features map
- ➔ 5 dataloaders to iterate over datasets separately
- ➔ Calculate the loss for each task / Perform back propagation 5 Times
- ➔ Backbone is kept frozen during the training



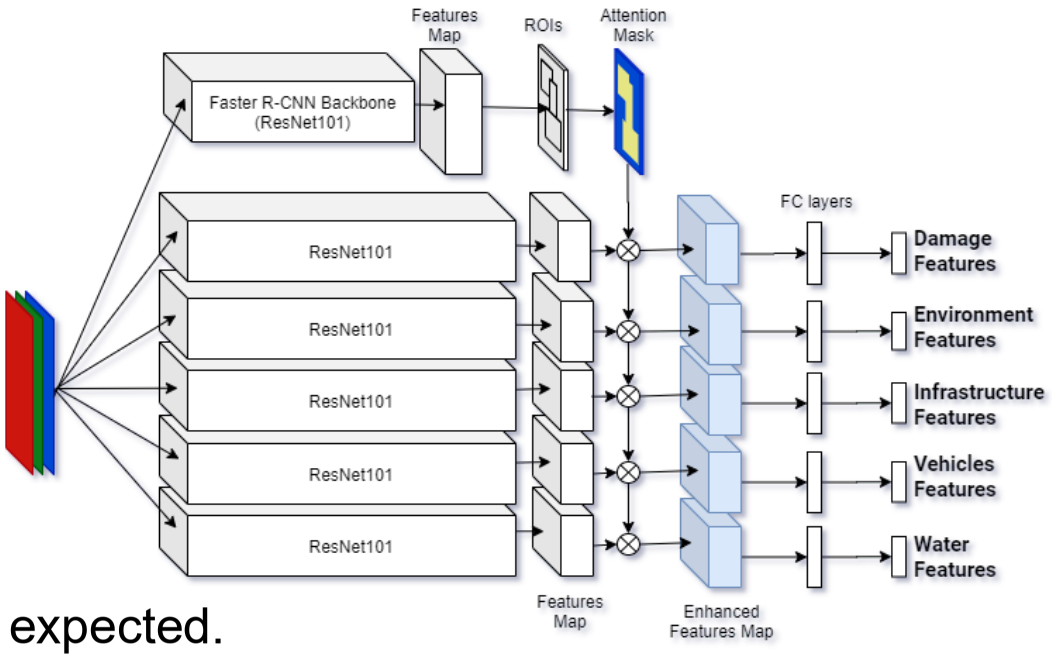
# Classifiers over Faster R-CNN Features Map with attention (O_VCL_2)

- ➔ Create attention masks from the detected bounding boxes
- ➔ Apply these on masks on the base features map
- ➔ 5 dataloaders to iterate over datasets separately
- ➔ Calculate the loss for each task / Perform back propagation 5 Times
- ➔ Backbone is kept frozen during the training



# Five Independent ResNet101 Classifiers with attention (O_VCL_4)

- ➔ 5 different classifiers, one for each category
- ➔ Apply the normalized attention mask over base features of C3 layer of the ResNet101 backbone




This method did not perform as expected.

# Results and analysis

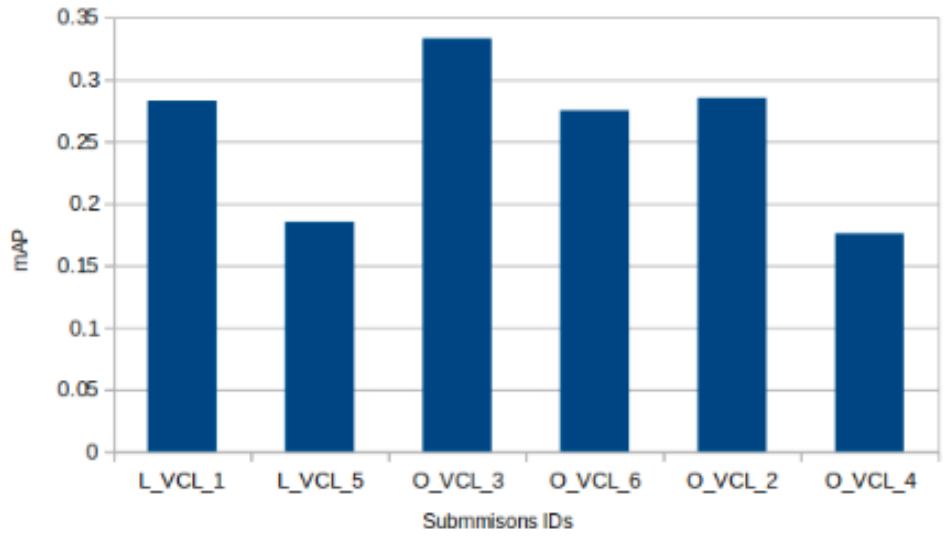



Figure 2. mAP achieved on the test videos by our submissions.

# Results and analysis

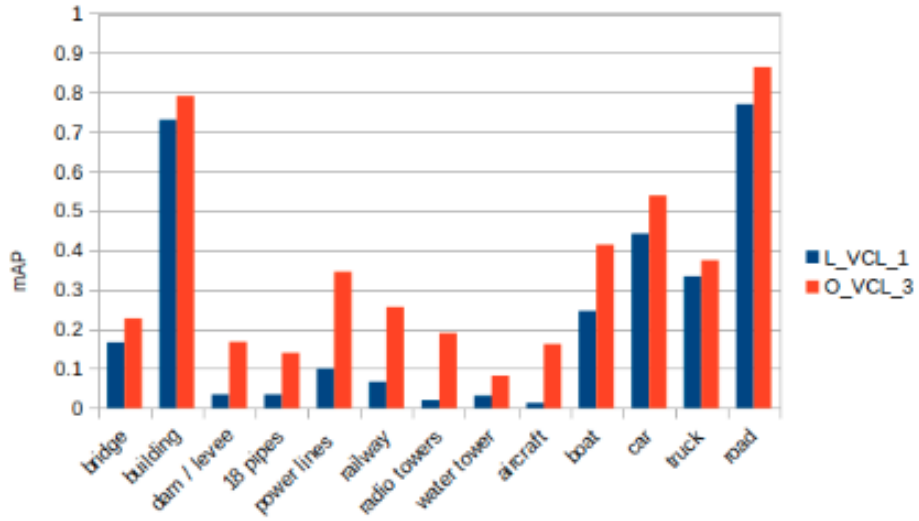



Figure 3. mAP comparison for submissions 1 and 3 for features in infrastructure and vehicle categories.

# Results and analysis

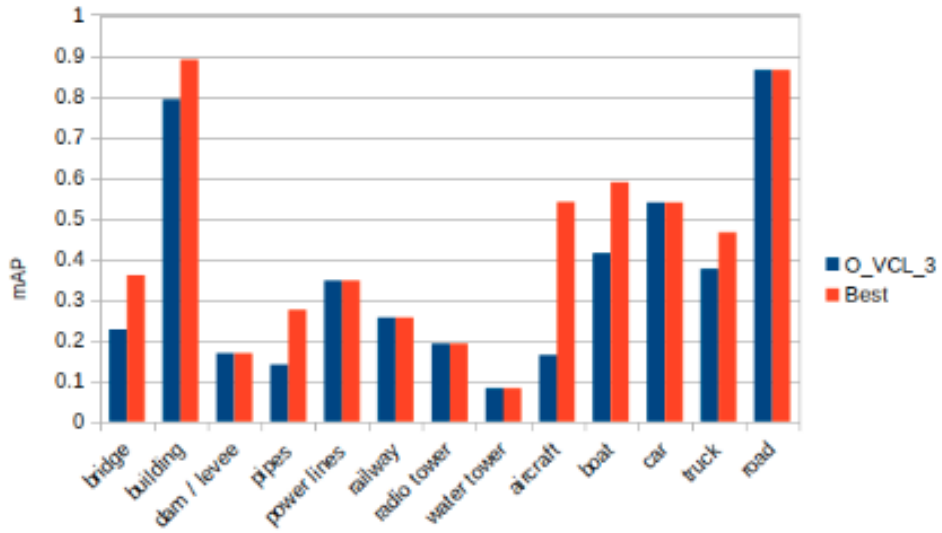



Figure 4. mAP of submission 3 vs best mAP for each feature in the infrastructure and vehicle categories.

# Conclusion

- ➔ Initially utilized ResNet101 classifiers
  - a) 5 different classifiers one for each feature category
  - b) one classifier with a common ResNet101 backbone and 5 classifiers heads on top
- ➔ Afterwards, relied on Faster R-CNN object detection network for vehicle and infrastructure
  - combination of Faster R-CNN and 3 of the 5 classifiers for the other 3 categories from the initial approach performed the best
  - utilized the Faster R-CNN backbone as a feature extractor and 5 classifier heads on top
  - features extracted from the Faster R-CNN backbone were refined with masks formed by the Faster R-CNN detected bounding boxes,

# Thank you

www.faster-project.eu
https://www.youtube.com/channel/UCshNttcAHEzbEmJN0z0PKTA

dimou@iti.gr

Co-funded by the Horizon 2020 programme
of the European Union