Flood Region Segmentation on Drone Images

E. Vlachos, D. Papaioannou, Prof. I. Pitas
Aristotle University of Thessaloniki
pitas@csd.auth.gr
www.aiia.csd.auth.gr

Version 2.0

Artificial Intelligence & Information Analysis Lab

Flood Region Segmentation on Drone Images

Natural Disaster Management

Recent catastrophic flood in Thessaly (2023).

- Due to climate change, **flash floods** are more usual than ever, affecting the lives of millions of people.
- There is an imminent need for cutting-edge Natural Disaster Management systems (NDM)

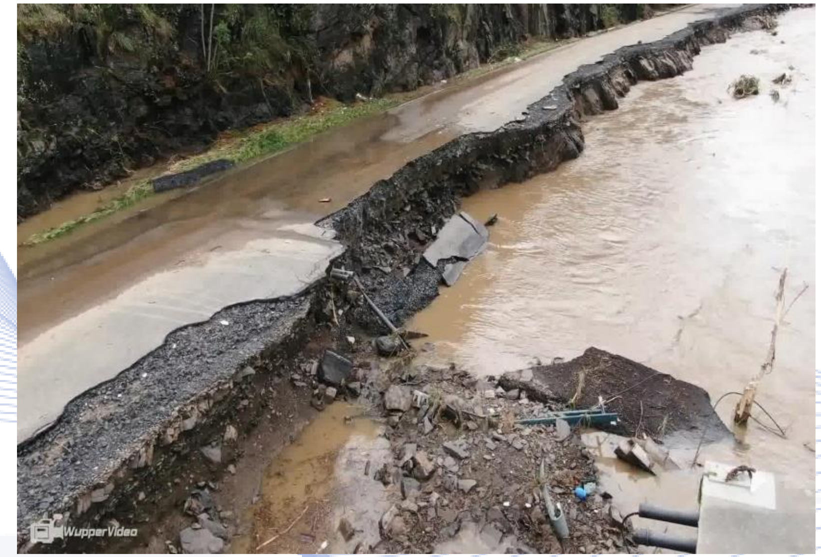
Artificial Intelligence & Information Analysis Lab

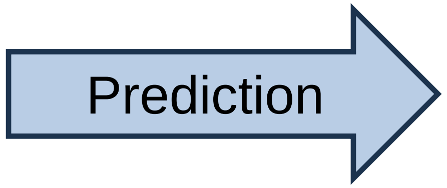
Natural Disaster Management

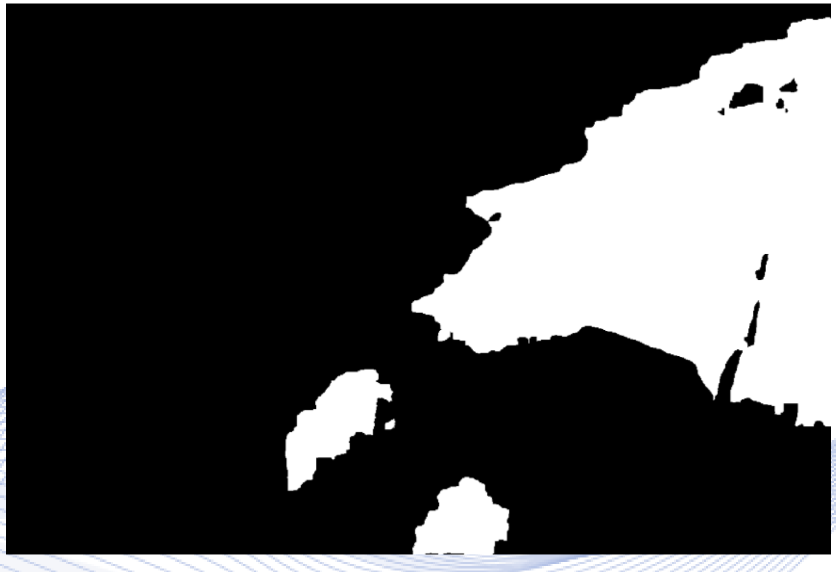
- ***Unmanned Aerial Vehicles (UAVs)*** can fly over areas which humans cannot access and capture useful footage.
- State-of-the-art computer vision models can perform tasks like ***flood segmentation*** or ***person detection*** and produce valuable insights from multimedia.
- Evolution on ***edge computing*** enables us to deploy computer vision models on UAVs.

UAV monitors flood damage in Peru.

Flood Region Segmentation on Drone Images







Flood Region Segmentation of a flood event on the use case of Arthal, Western Germany.

Artificial Intelligence & Information Analysis Lab

Flood Region Segmentation on Drone Images

- An image domain $\mathcal{X}$ must be segment in N different regions $\mathcal{R}_1 \dots, \mathcal{R}_N$.
- The segmentation rule is a logical predicate of the form $P(\mathcal{R})$.
- Image segmentation partitions the set $\mathcal{X}$ into the subsets $\mathcal{R}_i, i = 1, \dots, N$, having the following properties:

$$\mathcal{X} = \bigcup_{i=1}^{N} \mathcal{R}_i \,,$$

$$\mathcal{R}_i \cap \mathcal{R}_j = \emptyset, \quad i \neq j,$$

$$P(\mathcal{R}_i) = TRUE, \qquad i = 1, \dots, N,$$

$$P(\mathcal{R}_i \cup \mathcal{R}_j) = FALSE, \qquad i \neq j$$

Artificial Intelligence & Information Analysis Lab

Flood Region Segmentation on Drone Images

- Flood region segmentation is a task where each pixel in an image domain $\mathcal{X}$ is classified as either belonging to a flooded region (e.g., foreground) or to a non-flooded region (e.g., background).
- Given a function f, a segmentation mask is produced as $S = f(\mathcal{X})$.
- $f(\mathcal{X})$ generates a probability map $P(x, y)$ where $P(x, y) = P(S(x, y) = 1|\mathcal{X})$ is the probability that the pixel (x, y) belongs to the flooded region.
- Given a threshold T the segmentation mask is given as:

$$S(x,y) = \begin{cases} 1, & if\ P(x,y) \geq T \\ 0, & otherwise \end{cases}.$$

Artificial Intelligence & Information Analysis Lab

Flood Region Segmentation on Drone Images

Artificial Intelligence & Information Analysis Lab

Deep Semantic Segmentation

- Goal of deep flood semantic image segmentation is to classify each pixel of the input image to a flood or non-flood class using DNNs.

- **Feature Extraction:** Extract relevant information for the image, identifying visual patterns and characteristics that can assist in distinguishing the different regions.
- **Dense prediction:** DNN predictions are made at pixel level.

Deep Semantic Segmentation

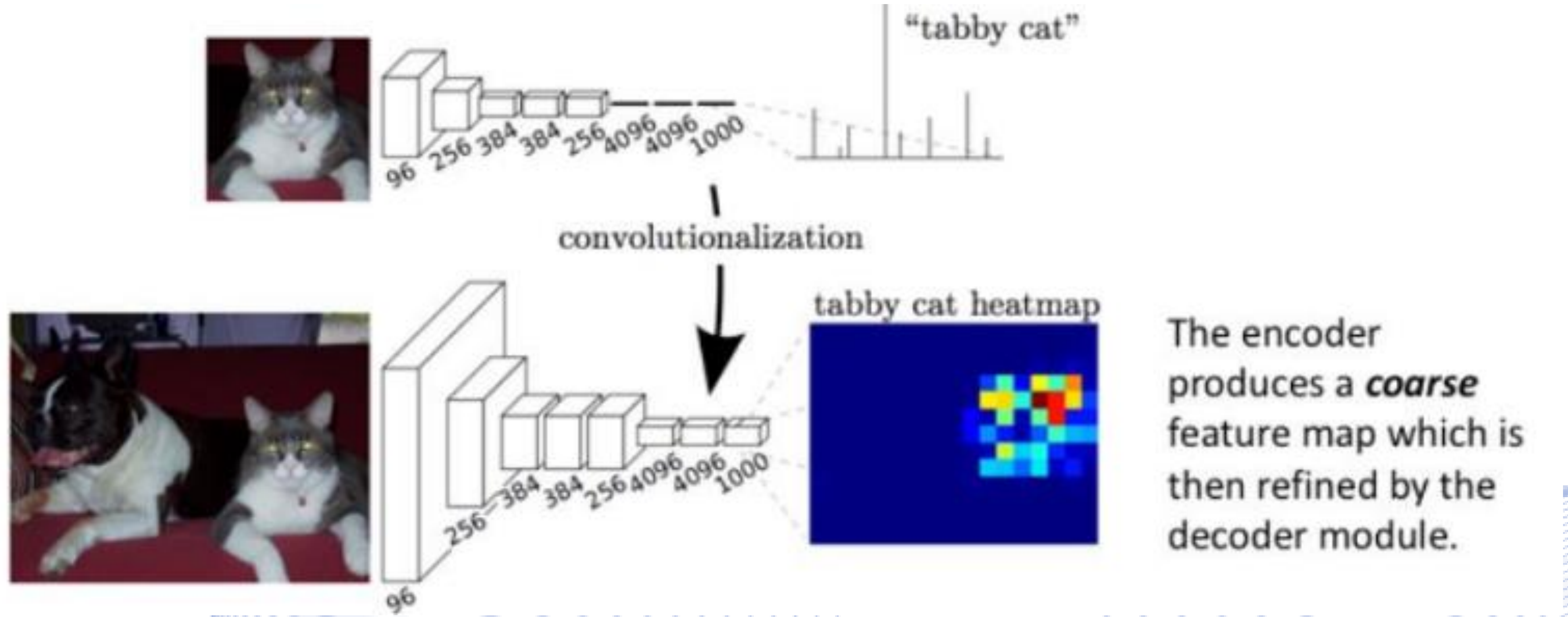
- Transforming the fully connected layers of image classification networks into convolution layers enables the transformed network to output **heatmaps**.

- End-to-end dense prediction learning is possible by adding extra layers and using an appropriate loss function.

- Encoder-Decoder network architecture.

Deep Semantic Segmentation

- Replacing fully connected layers with convolutional ones.



Coarse feature map for semantic image segmentation [LON2015].

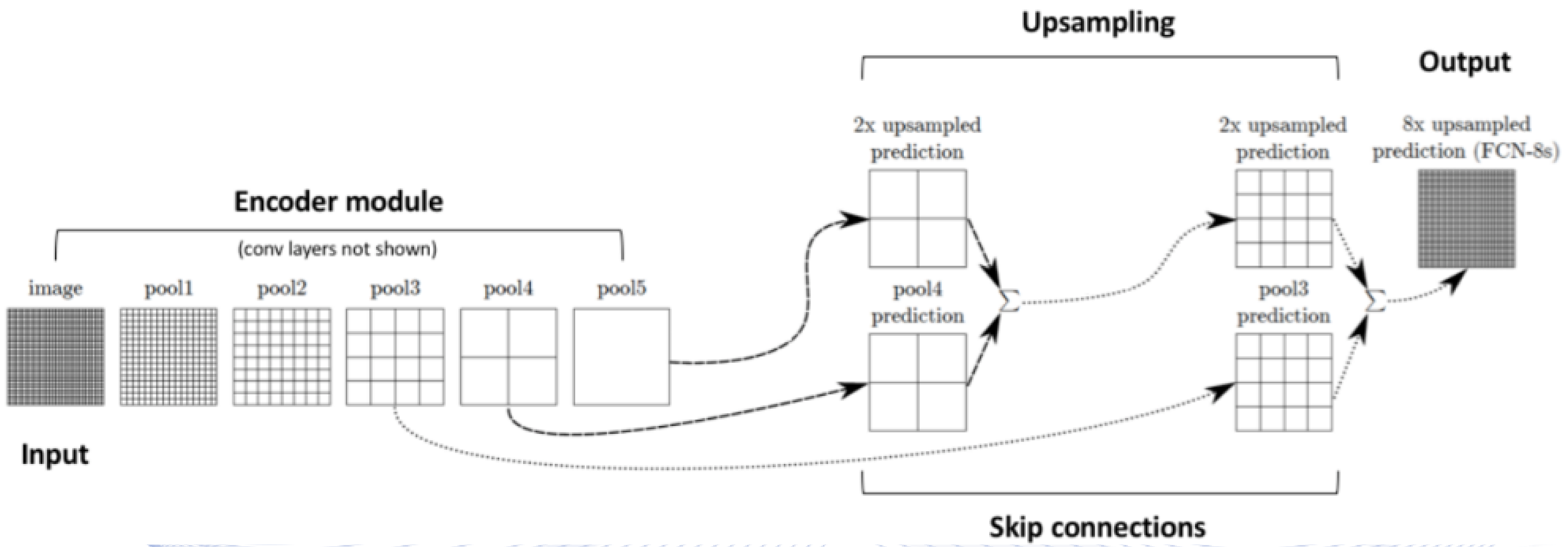
Deep Semantic Segmentation

- However, as the encoder radically reduces the resolution of the input image the decoder fails to produce fine-grained segmentations.

- ***Skip network connections*** are added in fully convolutional network that combine the final prediction layer with previous fine-grained layers.

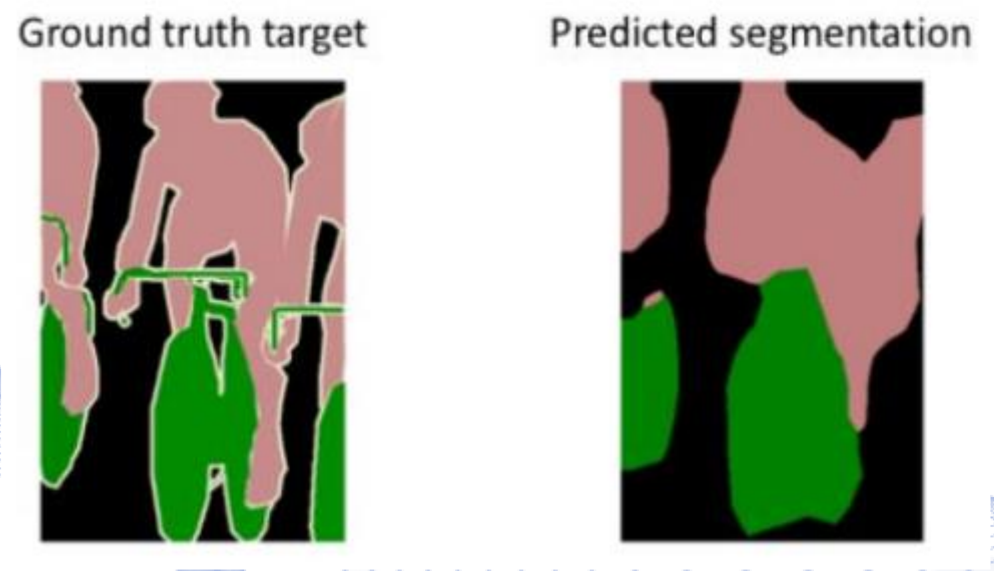
- Combine fine layers and coarse layers allow the model to make local predictions that respect global structure.

Deep Semantic Segmentation

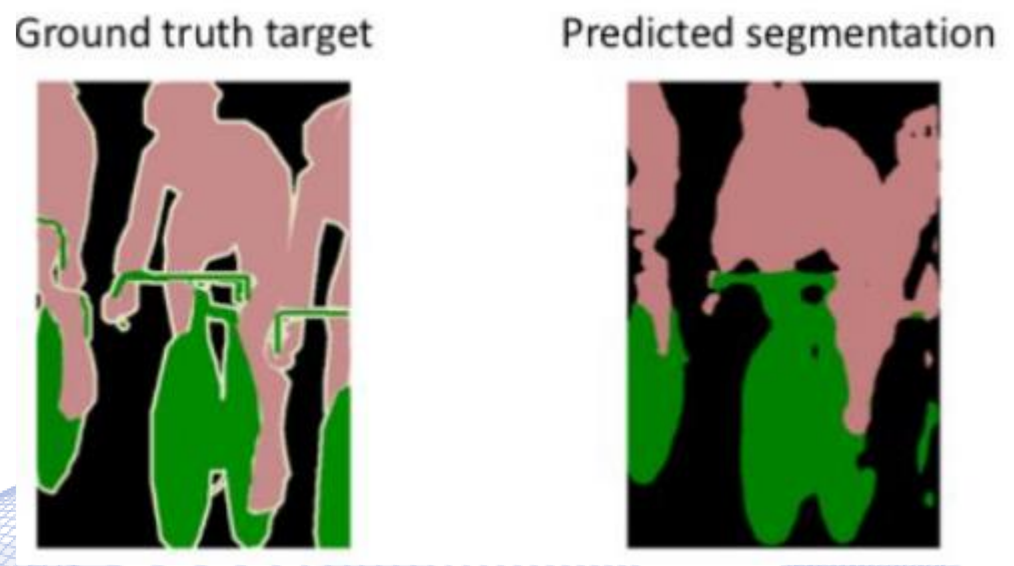


Skip CNN layer connections [LON2015].

Deep Semantic Segmentation



Coarse image segmentation[LON2015].



Improved segmentation results with skip connections [LON2015].

Deep Semantic Segmentation

Types of semantic segmentation learning procedures:

- ***Supervised Segmentation:*** A fully labeled dataset is used, where each image has a ground truth segmentation mask. The objective is to minimize the difference between the ground truth and the prediction mask using a loss function (e.g., cross entropy).
- ***Unsupervised Segmentation:*** Segmentation is performed without labeled data, relying on clustering or representation learning techniques to produce the segmentation masks.
- ***Semi-Supervised Segmentation:*** Both labeled and unlabeled images is utilized with a ratio of $1{:}5$. It combines supervised learning on labeled data with consistency regularization on unlabeled data.

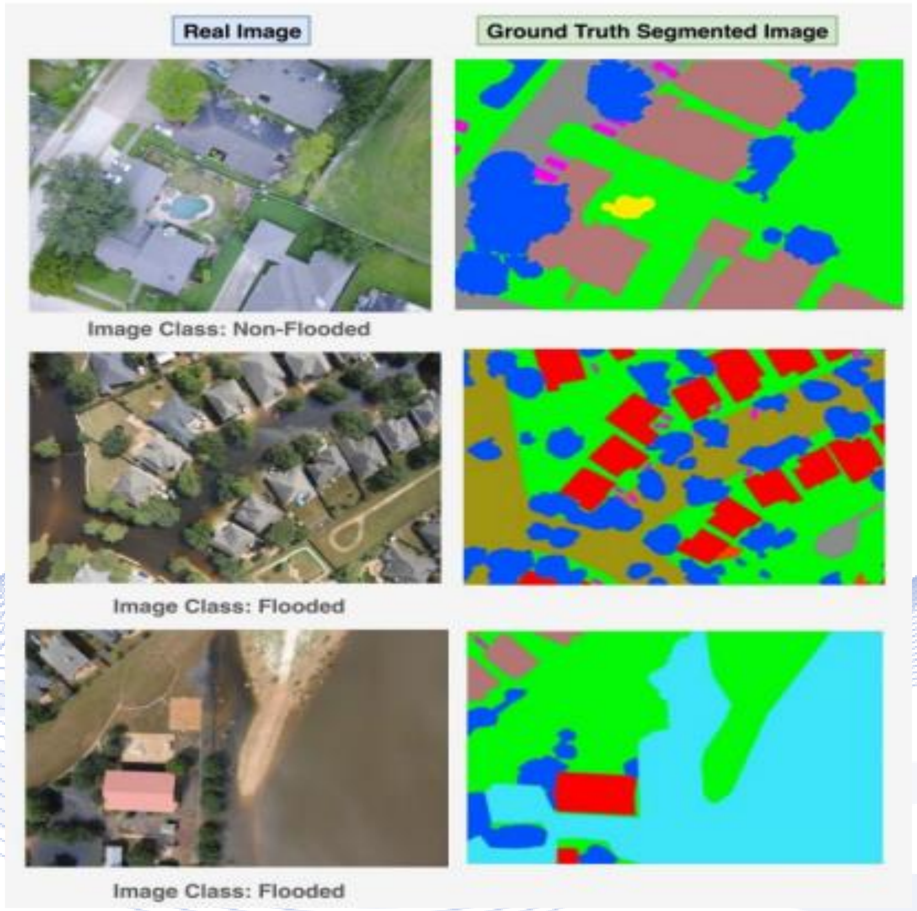
Flood Region Segmentation on Drone Images

Flood Region Segmentation

- FloodNet Dataset, consisting of UAV imagery for post-flood semantic segmentation.

- This dataset however is more suitable for post destruction assessment applications.

- V-floodnet includes thousands of images depicting regular waterbodies in their dataset, thus making the flood a “minority”.



Sample images and ground truth masks from FloodNet Dataset.

Flood Region Segmentation

FloodSeg: A novel dataset for efficient Flood Segmentation is introduced.

- ***High-Resolution Aerial View:*** The dataset captures detailed, high resolution aerial images of flood-affected areas from drone footage providing a detailed view of the landscape and water coverage.

- ***Annotated Flood Regions:*** Each image is carefully labeled to identify flooded regions, allowing for precise segmentation of water boundaries from surrounding terrain.

Artificial Intelligence & Information Analysis Lab

Flood Region Segmentation

The details of the proposed dataset are as follows:

FloodSeg: Train, Val, Test

- 548 annotated flood images from 3 different sources
- Train-Validation split (ratio 3:1)
- 2 manually annotated videos for testing purposes
- Greek video: 567 frames Italian video: 1204 frames

Extended FloodSeg Dataset

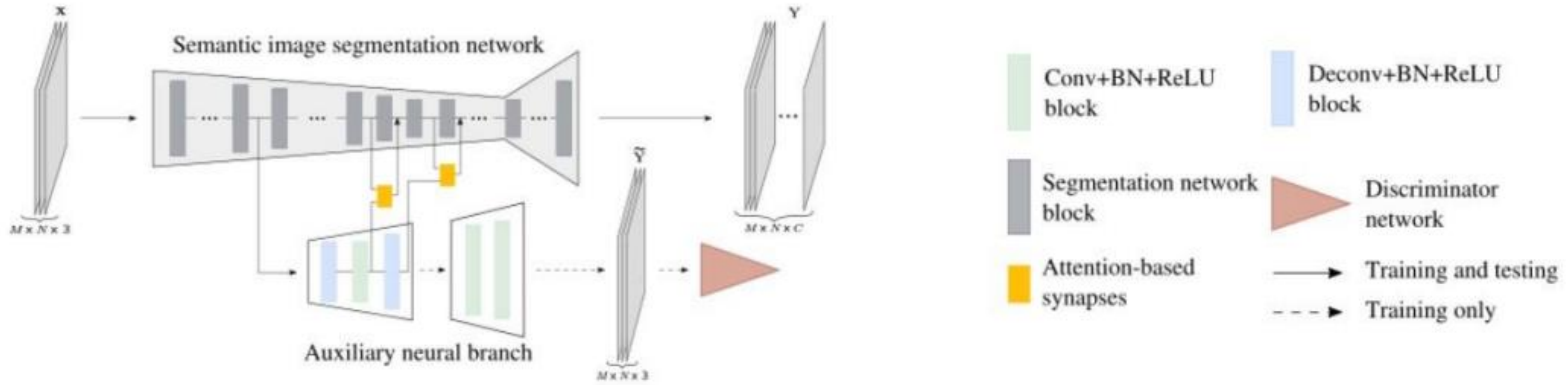
- Included unlabeled subsets to use via semi-supervised training.
- Included unlabeled subsets as well as new validation and test sets for distinct geographic regions (Greek and central European floods).

Flood Region Segmentation

Sample images from the FloodSeg Training Dataset, each one from a different source.

Flood Segmentation – CNN i2i

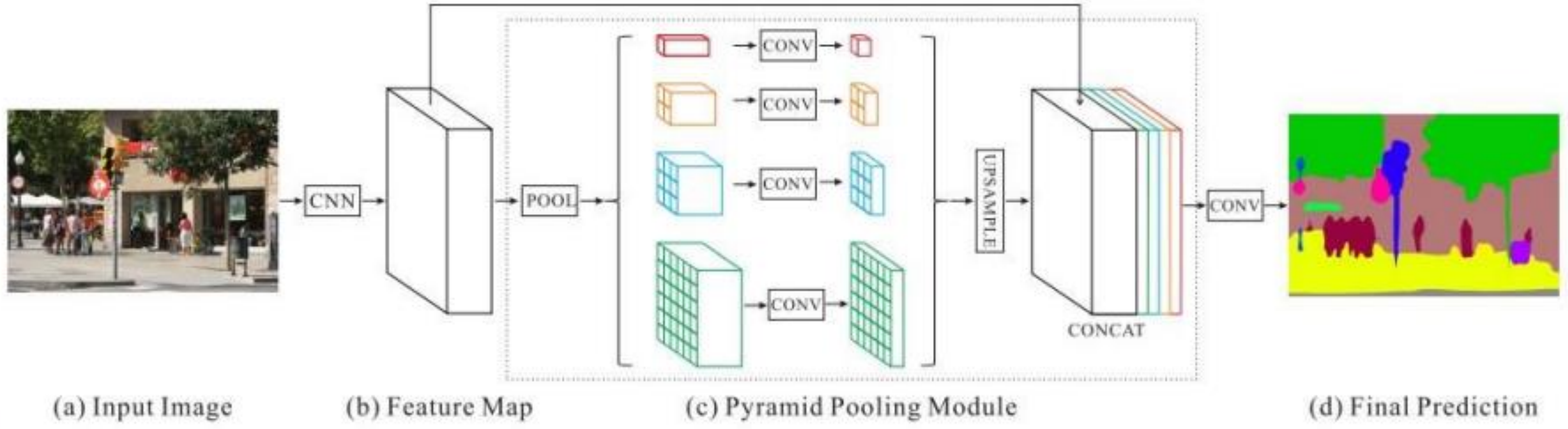
- **CNN-i2i** is a state-of-the-art semantic segmentation model, capable of making real-time predictions.
- A parallel neural generative branch is tasked to reconstruct the semantic masks in RGB format, and its useful features are propagated to the main branch via attention-based synapses.
- In our case, the main branch is a **BiseNet Network** with **Resnet18** as backbone.



Architecture of CNN i2i.

Flood Segmentation – PSPnet

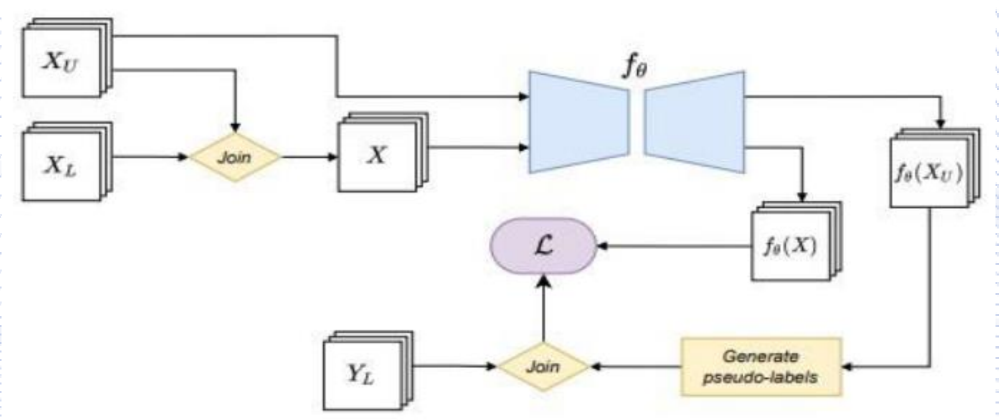
- **PSPnet** is considered a benchmark model for semantic segmentation tasks.
- It employs a Pyramid Pooling module, concatenates feature maps of different scales and captures “global” features.
- **Resnet50** is used as backbone for a lighter implementation.



Architecture of PSPnet.

Flood Segmentation – Semi Supervised Learning

- **ST++**, a state-of-the-art self-training method is utilized for semi-supervised training.
- This method pseudo-labels the unlabeled images in two steps (reliable, unreliable).
- Strong Image Augmentations are injected in unlabeled images to prevent overfitting to wrong predictions.



Graphical depiction of the ST++ method.

Artificial Intelligence & Information Analysis Lab

Flood Segmentation Results

Model	FloodSeg val (mIoU)	Greek Test (mIoU)	Italian test (mIoU)	Speed (ms)	FPS
CNN-i2i	**87.65%**	81.48%	**83.07%**	**9.82**	**101.85**
PSPnet	87.49%	**82.94%**	81.84%	12	79.5

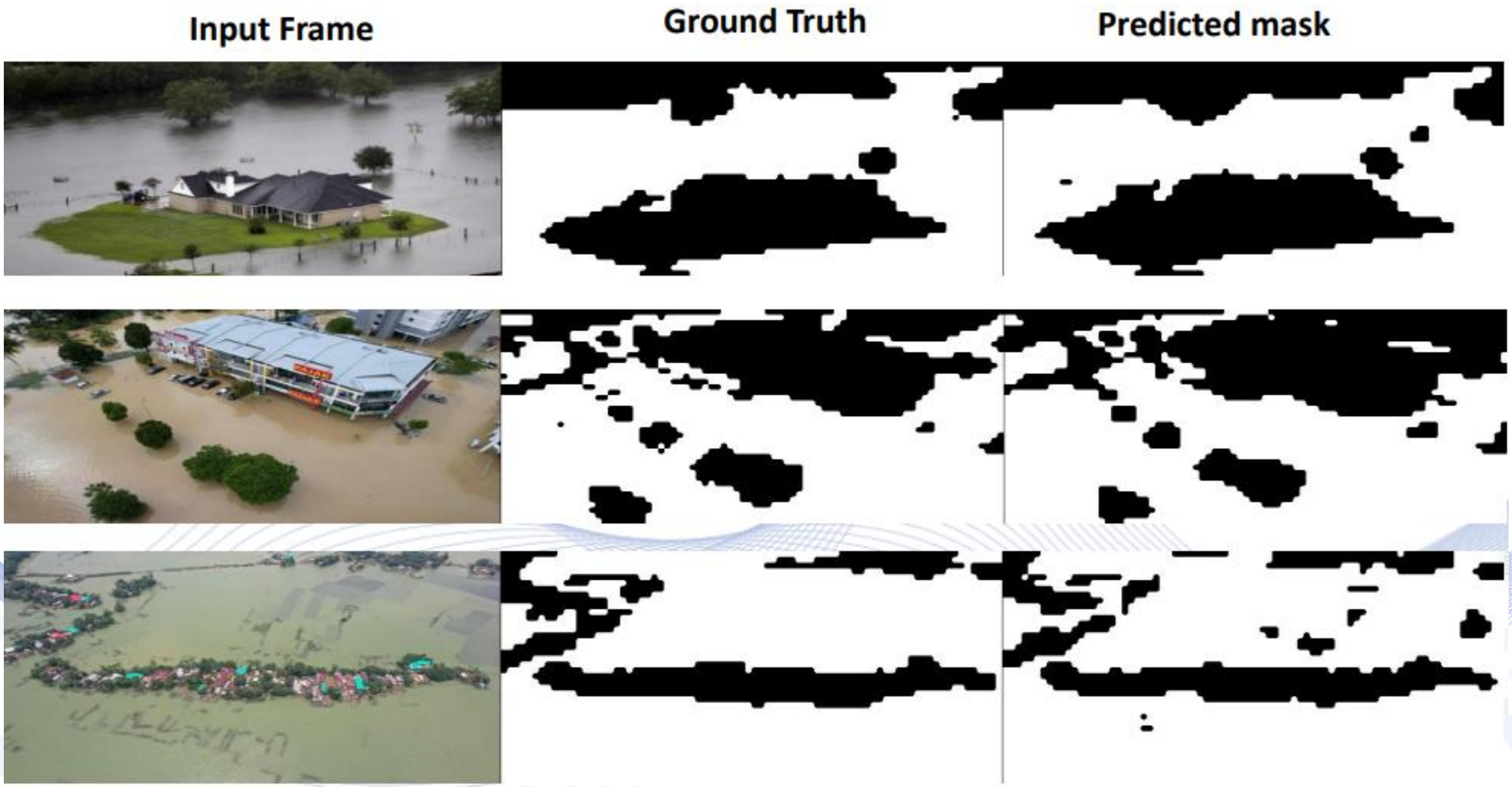
Table 1. Experimental evaluation of both supervised segmentation architectures trained on FloodSeg Dataset.

Unlabeled data amount	FloodSeg val (mIoU)	Greek Test (mIoU)	Italian test (mIoU)
827	88.43%	86.06%	**84.36%**
2430	88.74%	**86.55%**	83.88%
4647	**88.94%**	86.51%	83.89%

Table 2. Results of semi-supervised training with varying amount of Flood Images as unlabeled set (model PSPnet).

Artificial Intelligence & Information Analysis Lab

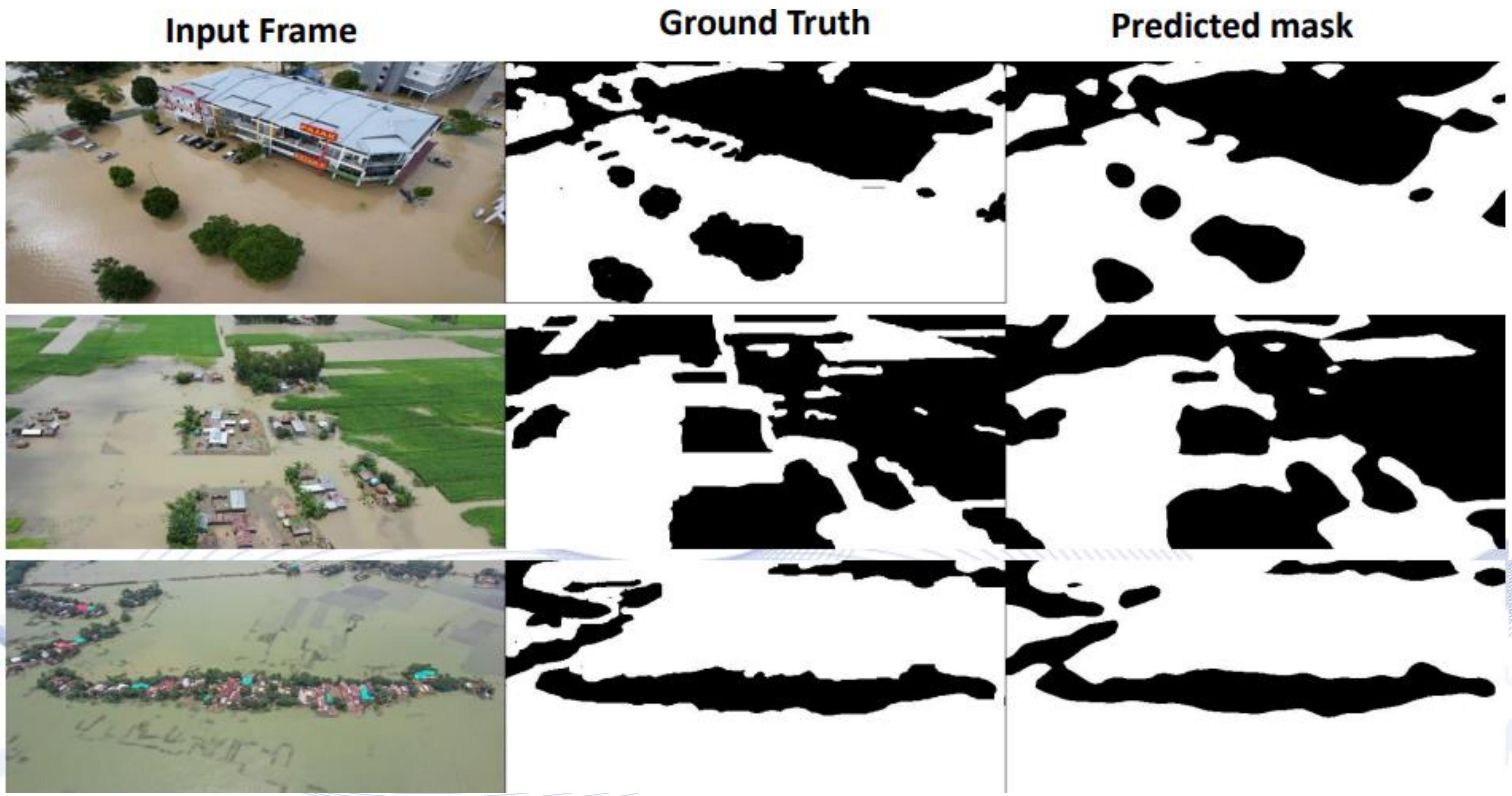
Flood Segmentation



Segmented outputs with **CNN-i2i**.

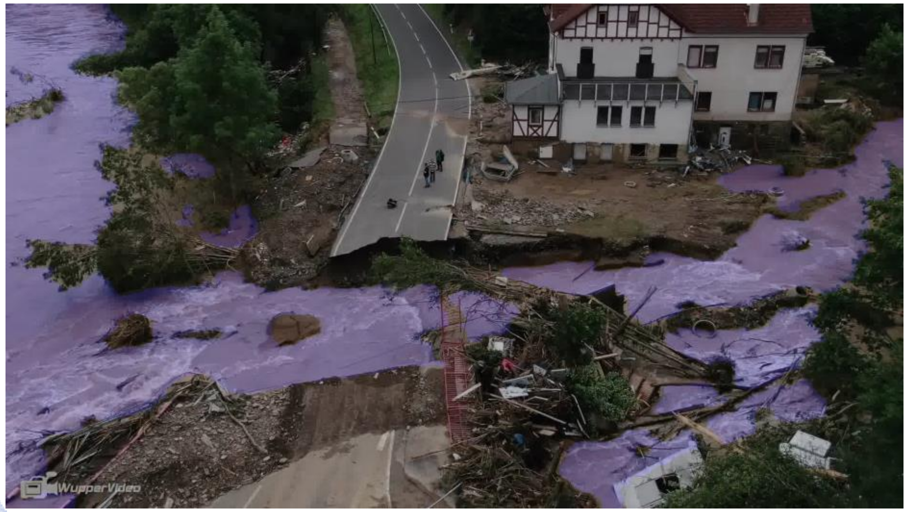
Artificial Intelligence & Information Analysis Lab

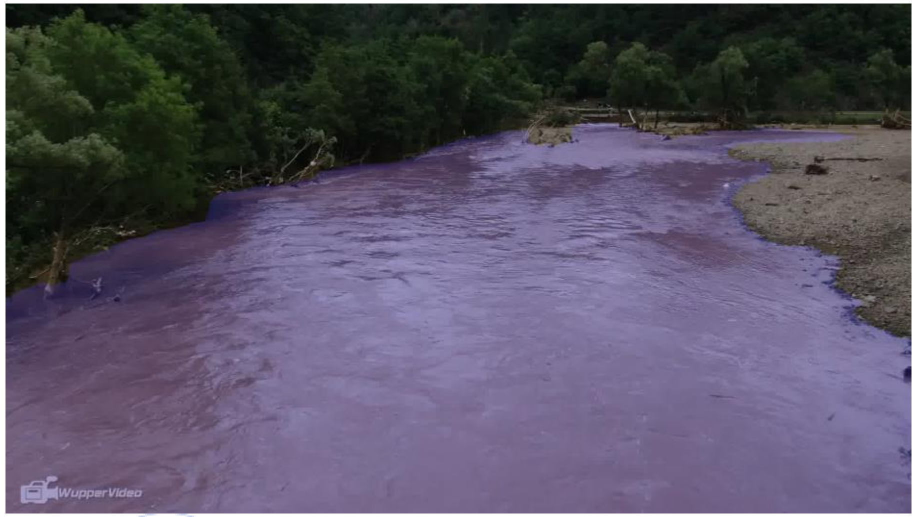
Flood Segmentation

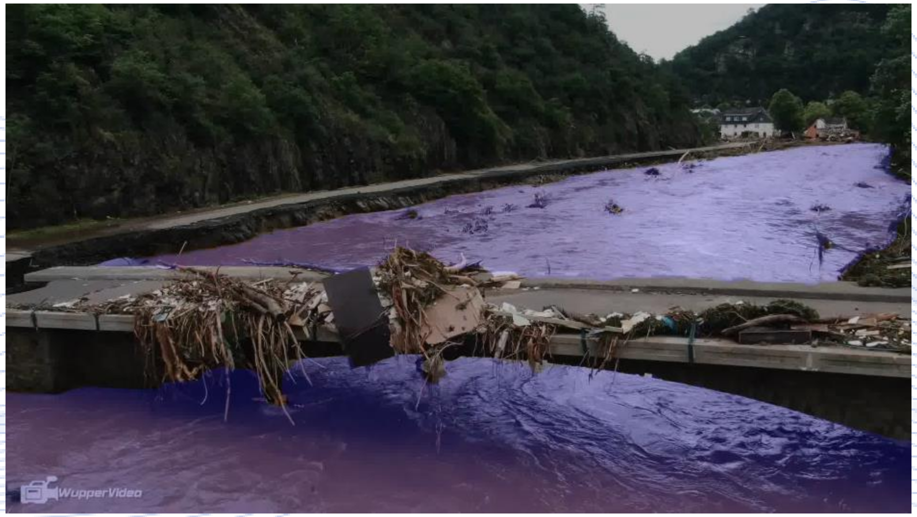


Segmented outputs with **PSPnet**.

Flood Segmentation



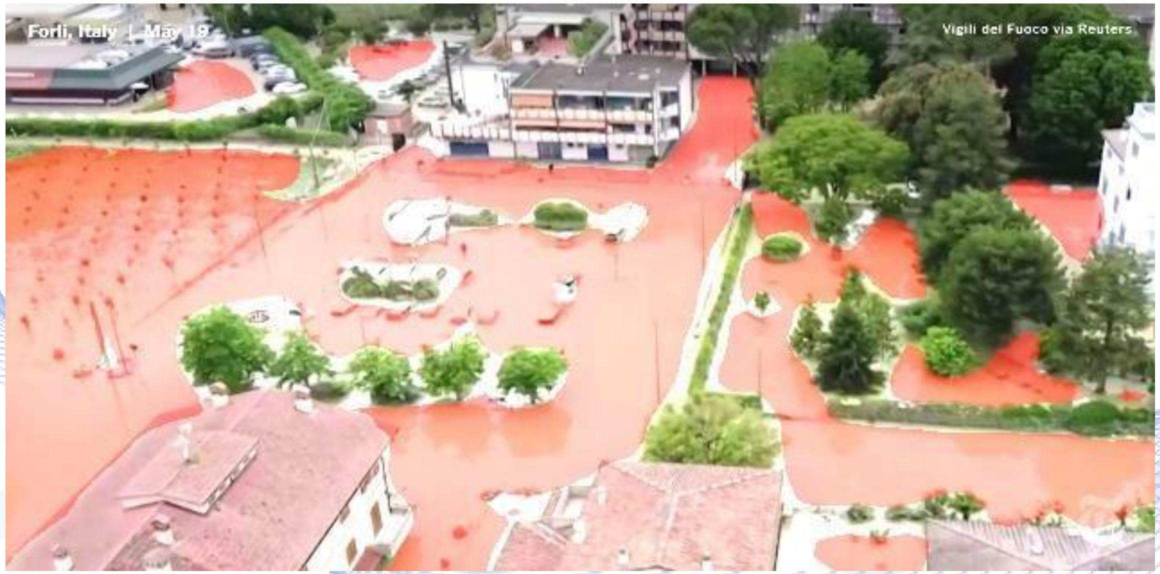


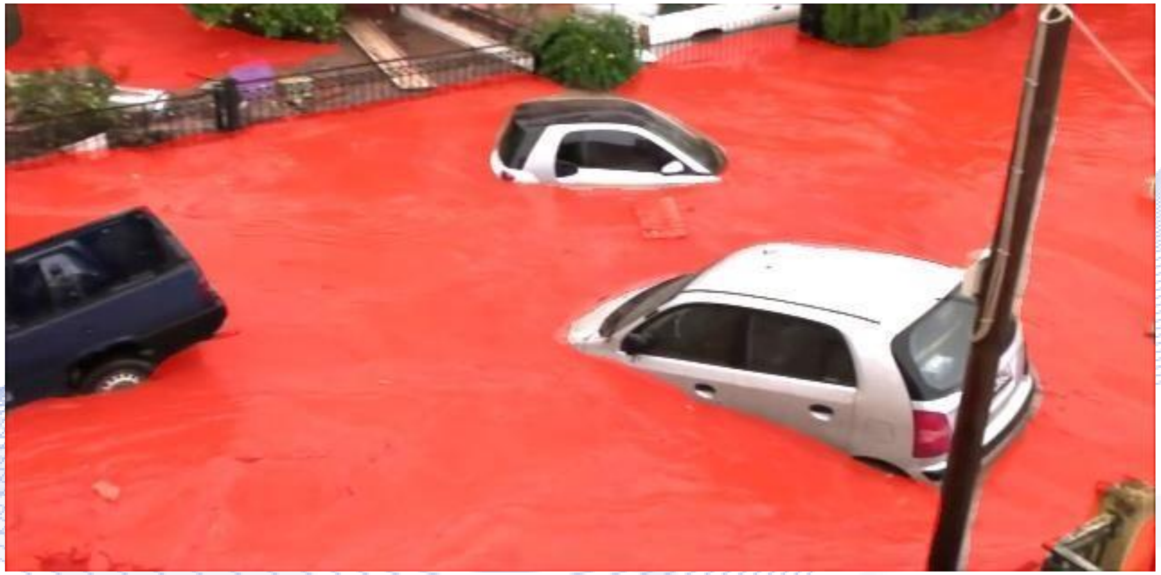


Segmented Results of Arthal West Germany flash flood case using CNN-i2i Architecture.

Artificial Intelligence & Information Analysis Lab

Visualized Segmentation Results





Visualized flood segmentation masks for two sample input frames, from the real-world testing videos that we annotated.

Flood Region Segmentation on Drone Images

- Introduction
- Deep Semantic Segmentation
- Flood Region Segmentation
- **Object Detection**
- Person/Vehicle Detection in Flooded Regions
- House-Roof Detection in Flooded Regions
- Flood Monitoring System

Object Detection

- Object detection = Classification + Localization:
- Find ***what*** is in a picture as well as ***where*** it is.

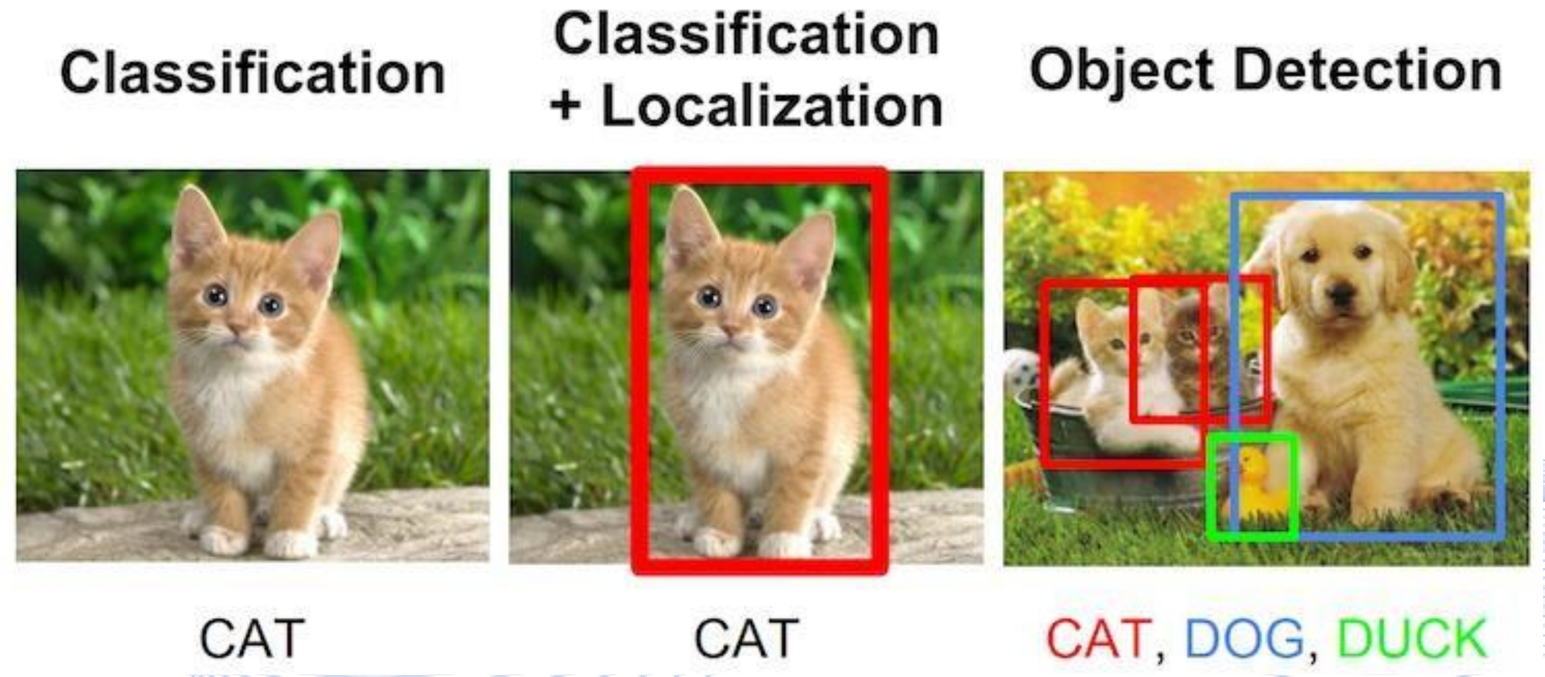


Figure: http://cs231n.stanford.edu/slides/2016/winter1516_lecture8.pdf

Object Detection

- ***Input***: an image.
- ***Output***: ***bounding boxes*** containing depicted objects.
  - Each image may contain a different number of detected objects.
- Old approach: train a specialized classifier and deploy in ***sliding-window style*** to detect all object of that class.
  - Very inefficient, quite ineffective.
- ***Goal***: combine classification and localization into a ***single architecture for multiple, multiclass object detection***.

Classification/Recognition/ Identification

- Given a set of classes $\mathcal{C} = \{\mathcal{C}_i, i = 1, \dots, m\}$ and a sample $\mathbf{x} \in \mathbb{R}^n$, the ML model $\hat{\mathbf{y}} = \boldsymbol{f}(\mathbf{x};\ \boldsymbol{\theta})$ predicts a class label vector $\hat{\mathbf{y}} \in [0, 1]^m$ for input sample $\mathbf{x}$, where $\boldsymbol{\theta}$ are the learnable model parameters.

- Essentially, a probabilistic distribution $P(\hat{\mathbf{y}}|\mathbf{x})$is computed.
- Interpretation: likelihood of the given sample $\mathbf{x}$ belonging to each class $\mathcal{C}_i$.

- Single-target classification:
  - classes $\mathcal{C}_i, i = 1, \dots, m$ are not mutually exclusive : $||\hat{\mathbf{y}}||_1 = 1.$
- Multi-target classification:
  - classes $\mathcal{C}_i, i = 1, \dots, m$ are not mutually exclusive : $||\hat{\mathbf{y}}||_1 \geq 1.$

Regression

- **Regression:**
  - Example: In object detection, localize the object:
  - regress object ROI parameters: ROI center $(\boldsymbol{x}_c, \boldsymbol{y}_c)$, width w, height h).
  - **Function approximation:** it is essentially regression, when the function $\mathbf{y} = \boldsymbol{f}(\mathbf{x})$ is known.

Object Detection

Object detection is a ***multitask machine learning*** problem:

- combination of classification and regression.
- Given a set of classes $\mathcal{C} = \{\mathcal{C}_i, i = 1, \dots, m\}$ and an image sample $\mathbf{x} \in \mathbb{R}^n$, the model predicts (for one object instance only) an output vector $\hat{\boldsymbol{y}} = [\hat{\boldsymbol{y}}_1^T | \hat{\boldsymbol{y}}_2^T]^T$ consisting of:
  - A class vector $\hat{\boldsymbol{y}}_1 \in [0,1]^m$ and
  - A bounding box parameter vector $\hat{\boldsymbol{y}}_2 = [x, y, w, h]^T$ corresponding to object ROI.
- Optimization of a joint cost function:

$$\min_{\boldsymbol{\theta}} J(\boldsymbol{y}, \hat{\boldsymbol{y}}) = \alpha_1 J_1(\boldsymbol{y}_1, \hat{\boldsymbol{y}}_1) + \alpha_2 J_2(\boldsymbol{y}_2, \hat{\boldsymbol{y}}_2)$$

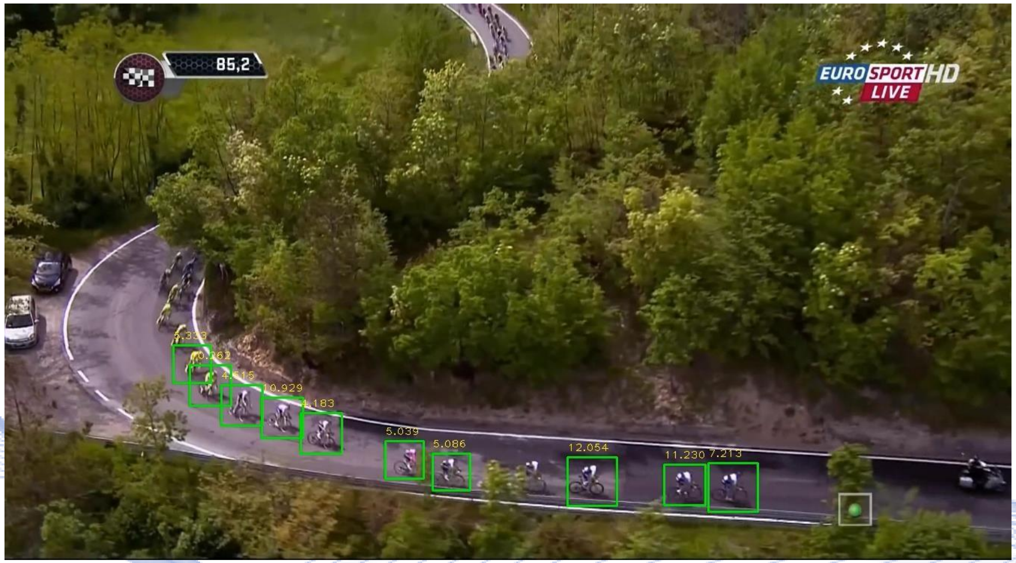
- The above vector pair will be computed for every possible target detected in the image sample $\mathbf{x}$.

Non-Maximum Suppression (NMS)

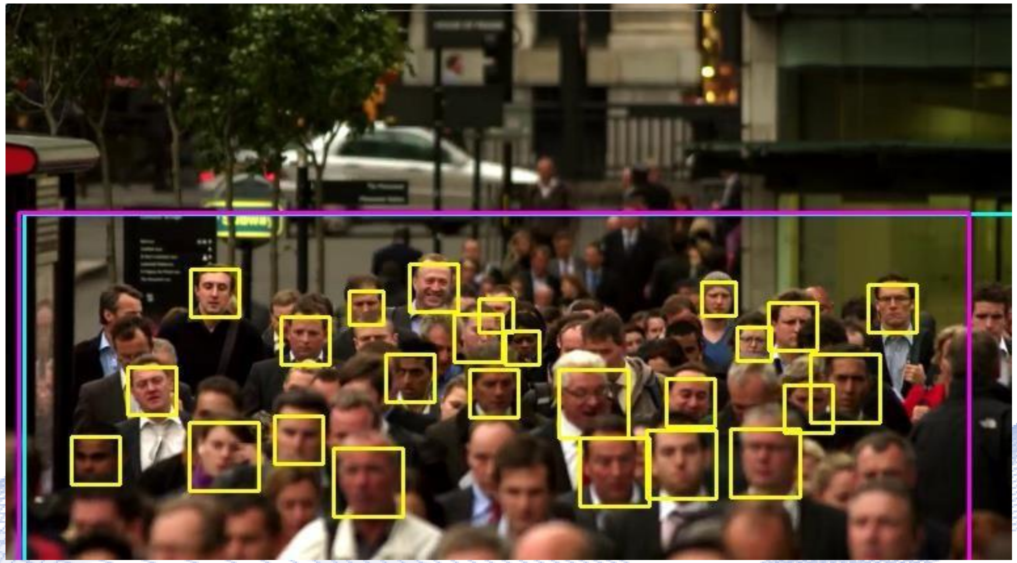
- **Challenge:** Object detectors often output many overlapping detections.

- **Solution:** Post process raw detections using NMS
  - For each category:
    - Select next highest-scoring box
    - Eliminate lower-scoring boxes with IoU>threshold(e.g. 0.5)
    - If remaining boxes, go to first step.

Object Detection Inference Examples



Bicycle Detection.



Face Detection.

CNN-Object Detection

Region proposal-based detectors

- R-CNN, Fast R-CNN, Faster R-CNN
- R-FCN

Single Stage Detectors

- YOLO
- SSD
- YOLO v2, v3, v4
- RetinaNet, RBFnet
- CornerNet, CenterNet

Transformer Detectors

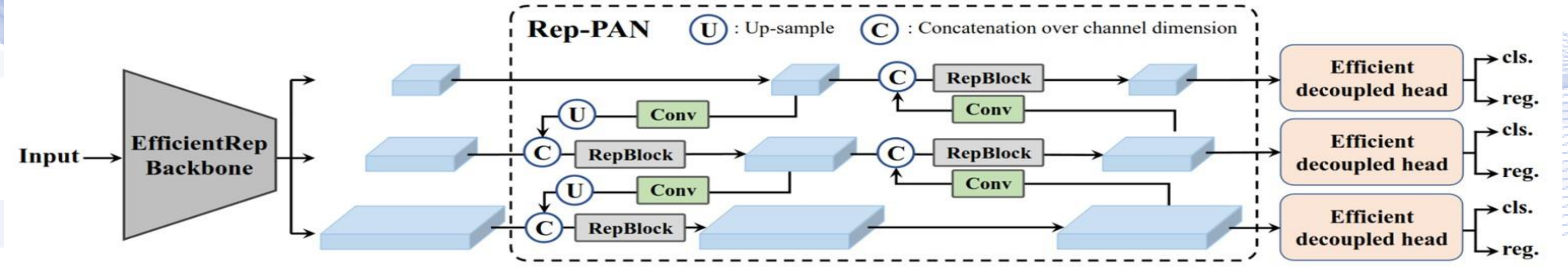
- DETR.

CNN-Object Detection Architectures

- **Backbone** refers to a CNN used for image feature extraction:
  - ResNet, MobileNet, VGG etc.
- **Neck** is an extra object detector layer that goes on top of the backbone. I t extracts different feature maps from different stages of the backbone.
  - FPN, PANet, Bi-FPN etc.
- ***Head*** network performs actual object detection: classification (probability of $m+1$ classes) and regression of RoI parameters (x, y, h, w).

YOLO v6

- ***EfficientRep Backbone***: RepVGG and CSPStackRep blocks are combined to optimize speed and accuracy across model sizes.
- ***Rep-PAN Neck***: RepVGG and CSPStackRep blocks are used to enhance feature integration at multiple scales.
- ***Efficient Decoupled Head***: a hybrid-channel strategy is used to reduce the computation costs.



YOLOv6 Architecture [LI2022].

YOLO Models

YOLO versions have evolved significantly, each uniquely enhancements to the single-stage detector architecture:

- ***YOLOX***
- ***YOLOv7***
- ***YOLOv8***
- ***YOLOv9***
- ***YOLOv10***
- ***YOLOv11.***

Each model improves detection accuracy, speed, and robustness for various applications.

DETR

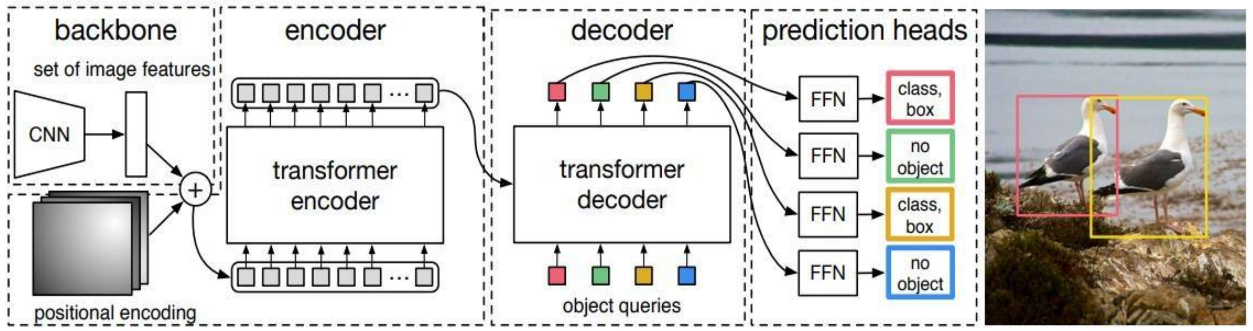
- ***Detection Transformer*** (***DETR***) views object detection as a direct set prediction problem, while removing many hand - designed components like Non-Maximum Suppression (NMS) or anchor generation.

- DETR utilizes an encoder-decoder sequence processing model called ***Transformer*** [VAS2017] and a bipartite matching loss.

DETR

DETR architecture has three main components:

- A RestNet50/101 CNN backbone for feature extraction.
- An encoder-decoder transformer model.
- A feed-forward head network makes the final detection predictions.



DETR architecture [CAR2020].

DETR

Given a set of ground truth $\mathcal{D} = \{(\mathbf{x}_i, \mathbf{y}_i), i = 1, \dots, N\}$ of N images and a set of predictions $\hat{\boldsymbol{y}}_i, i = 1, \dots, N$ we search for a permutation of N elements $\sigma \in \mathcal{G}_N$ with the lowest cost:

$$\hat{\sigma} = \arg\min_{\sigma \in \mathcal{G}_N} \sum^{\mathrm{N}} J(\hat{\boldsymbol{y}}_i, \hat{\boldsymbol{y}}_{\sigma(i)}).$$

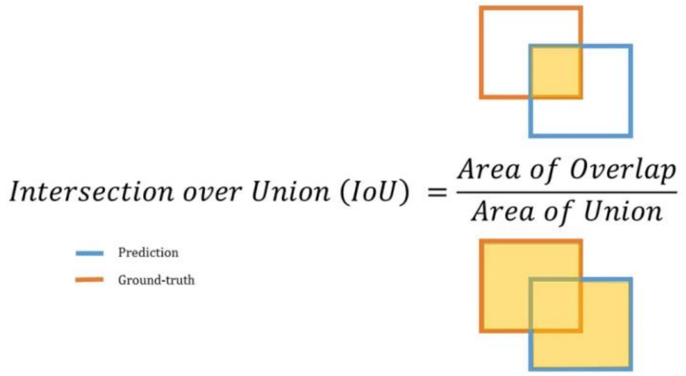
Bipartite matching loss J^i is a pair-wise matching cost between ground truth $\mathbf{y}_i$ and a prediction $\hat{\boldsymbol{y}}_{\sigma(i)}$.

- $\mathbf{y}_i = \{\mathbf{y}_{1i}, \boldsymbol{y}_{2i}\}$ where $\mathbf{y}_{1i}$ is the target class label and $\mathbf{y}_{2i}$ defines ground truth box coordinates.
- J significantly reduces low-quality predictions and eliminates the need for output reductions, e.g., by using NMS.

Object Detection Performance Metrics

- ***Intersection over Union (IoU)***:

$$J(\mathcal{A},\mathcal{B}) = \frac{|\mathcal{A}\cap\mathcal{B}|}{|\mathcal{A}\cup\mathcal{B}|}.$$


$Intersection\ over\ Union\ (IoU) = \frac{Area\ of\ Overlap}{Area\ of\ Union}$
Prediction
Ground-truth

- $\mathcal{A},\mathcal{B}$: estimated, ground truth ROIs (sets, bounding boxes).
- $|\mathcal{A}|$: set cardinality (area counted in pixels).
- Also called ***Jaccard Similarity Coefficient*** or ***Overlap Score.***

Object Detection Performance Metrics

- Object detection on images $i = 1, \dots, N_t$:

bounding boxes $\mathcal{A}_{ij}$ and confidence scores s_{ij}.

- If $\mathcal{A}_{ij}$ is matched to a ground truth box $\mathcal{B}_{ik}$:

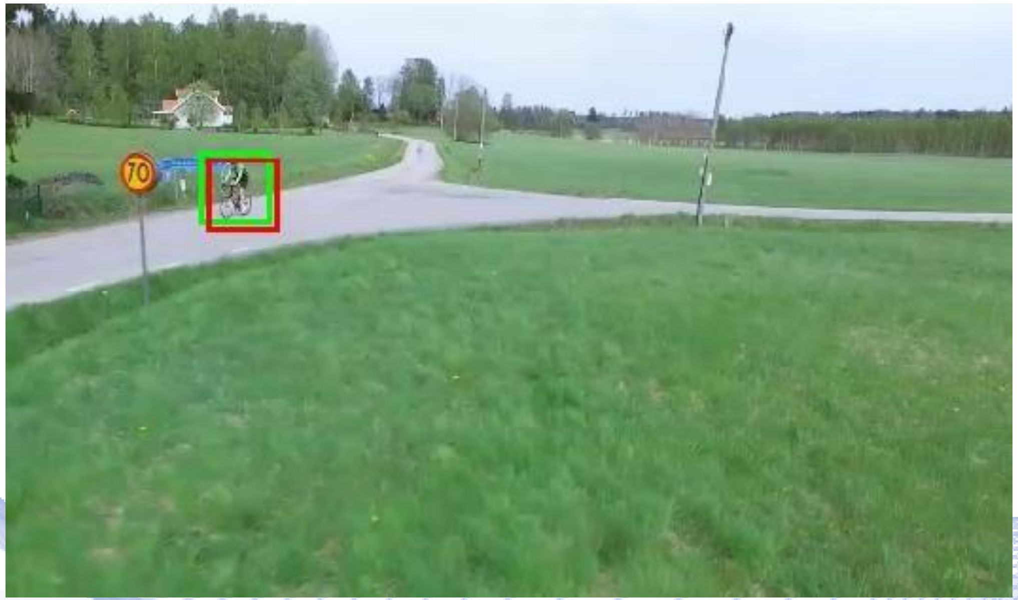
$$J(\mathcal{A}_{ij}, \mathcal{B}_{ik}) > T(\mathcal{B}_{ik}), \qquad \text{then } z_{ij} = 1.$$

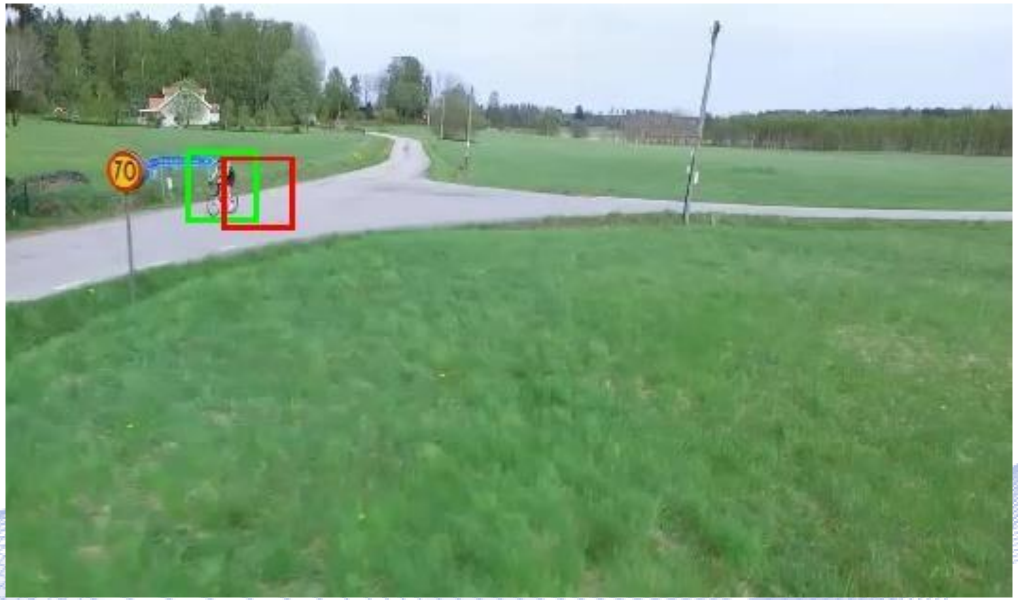
- The threshold $T(\mathcal{B}_{ik})$ depends on the box size:

$$T(\mathcal{B}_{ik}) = \min(0.5, hw/(h+1)(w+1)).$$

Artificial Intelligence & Information Analysis Lab

Object Localization Performance Metrics Example





Object localization performance: a) $J(\mathcal{A},\mathcal{B}) = 0.67$; b) $J(\mathcal{A},\mathcal{B}) = 0.27$.

Object Detection Performance Metrics

For M ground truth object ROIs on all N_t images:

- Let $n_{ij} = 1$ for a successful classification at ***confidence threshold*** t $(s_{ij} \geq t)$:
- ***Recall, Precision*** definitions (modified):

$$r(t) = \frac{\sigma_{ij}\, n_{ij} z_{ij}}{M},$$

$$p(t) = \frac{\sigma_{ij}\, n_{ij} z_{ij}}{\sigma_{ij}\, n_{ij}}.$$

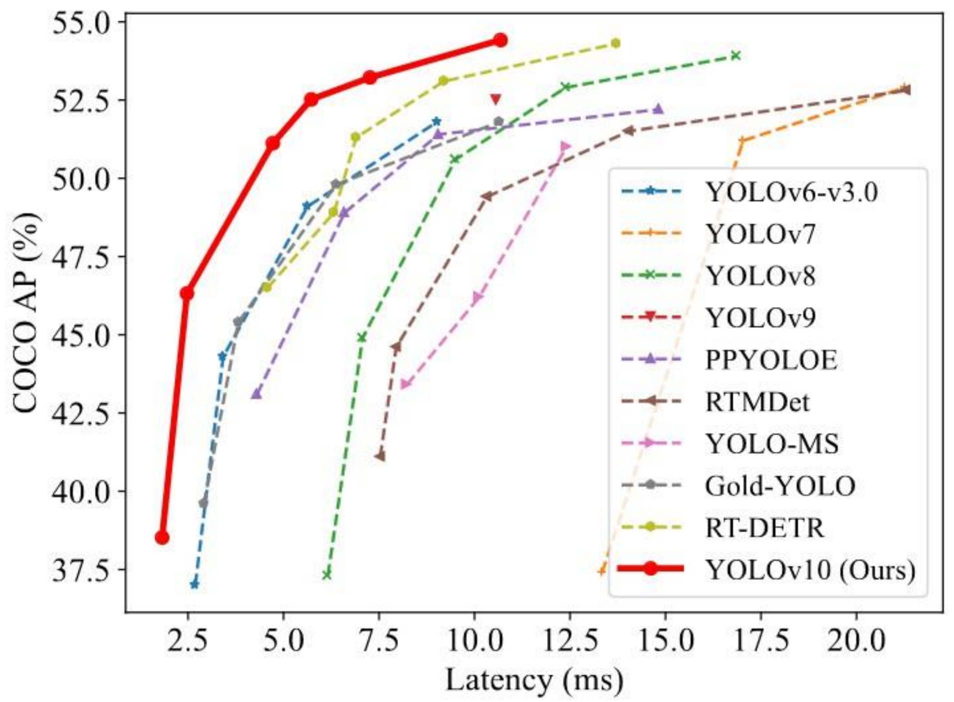
Object Detection Performance Metrics

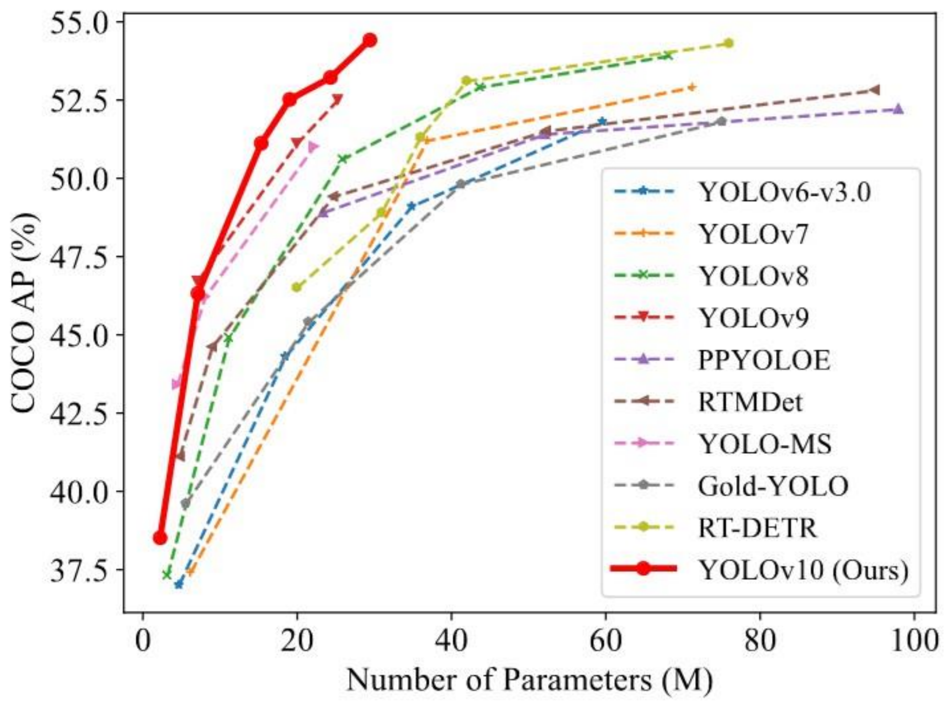
Mean Average Precision (***mAP***):

- It is calculated over N levels of confidence threshold $t_n, n = 1, \dots, N$:

$$mAP = \frac{1}{N}\Sigma_n\, p(t_n).$$

Real-Time Object Detectors



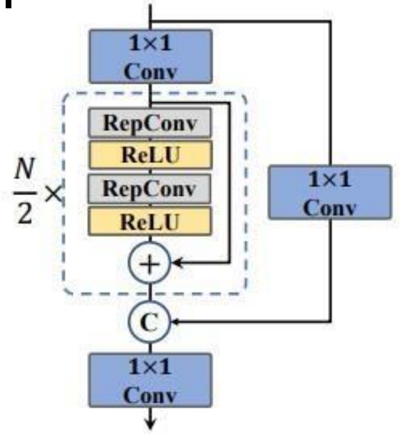


Real time object detectors comparison on COCO dataset [WAN2024].

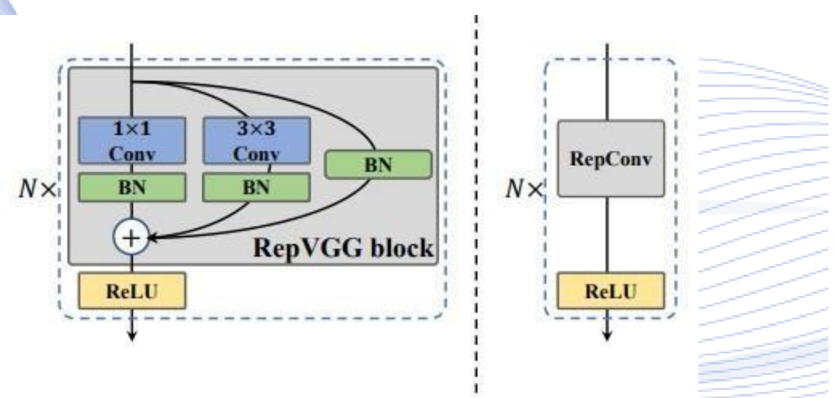
Flood Region Segmentation on Drone Images

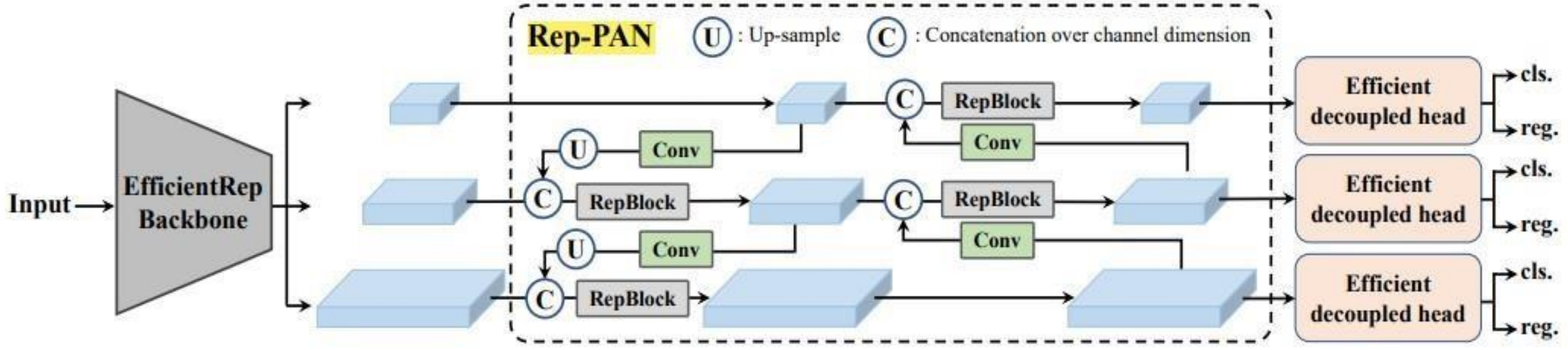
Person/Vehicle Detection

- Opted for the vanilla approach of using a model pretrained on
- COCO dataset.
- Yolov6 [LI2023] is a state-of-the-art model in real-time object
- detection.
- Small and Large models were selected and compared.



CSPStackRep building block.





RepVGG [DING2021] building block.

Yolov6 small model architecture.

Person/Vehicle Detection Results

The detectors were evaluated using a **detection test set** that we created and manually annotated.
It consists of many images, with objects labeled as person, car, truck. We focused on **submerged objects**, and **people that were in danger**.

	$mAP_{50:95}$	mAP_{50}	Latency (ms)	FPS
YOLOv6-S	0.53	0.73	**13.8**	**72.6**
YOLOv6-L	**0.61**	**0.80**	30.3	33

Table 3. Object detection results measured with mAP metric.

	Person $AP_{50:95}$	Person mAP_{50}
YOLOv6-S	0.92	0.63
YOLOv6-L	**0.94**	**0.69**

Table 4. Object detection results for **person** class.

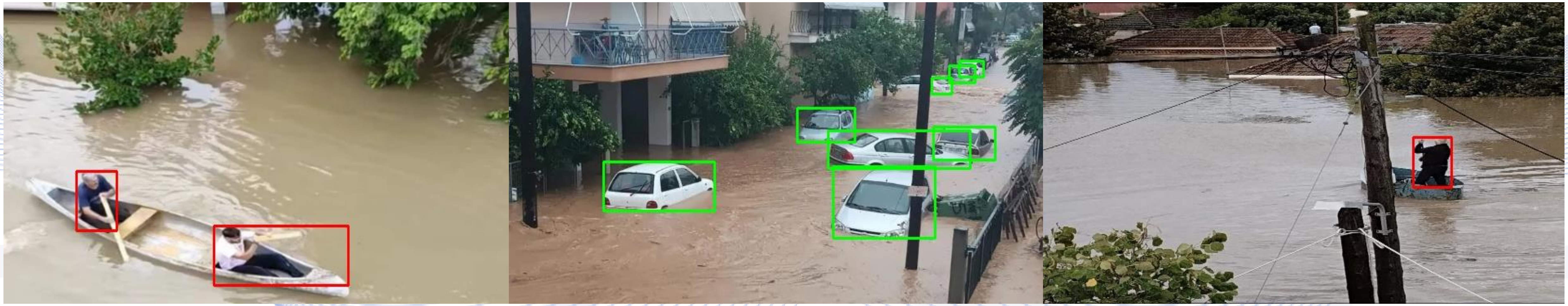
	Car $AP_{50:95}$	Car mAP_{50}
YOLOv6-S	0.84	0.59
YOLOv6-L	**0.88**	**0.64**

Table 5. Object detection results for **car** class.

Artificial Intelligence & Information Analysis Lab

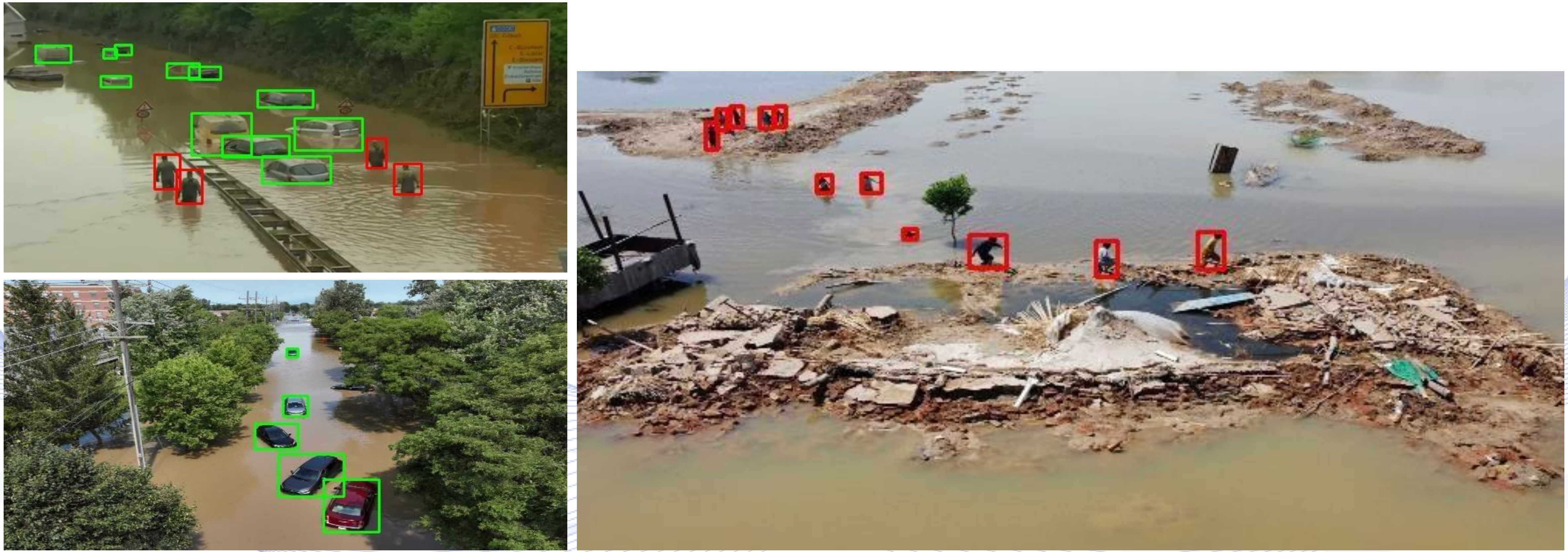
Object Detection and Tracking in Floods

DNN models, pretrained on COCO dataset were used to detect classes of interest (***cars, persons***) that may be in danger).



YOLOv6 4.0 small version in person, car detection in Thessaly floods, Greece (September 2023).

Detection Results Visualized



Sample images from our detection test set, with visualized bounding boxes (red boxes for **persons**, green boxes for **cars**).

Flood Region Segmentation on Drone Images

- Introduction
- Deep Semantic Segmentation
- Flood Region Segmentation
- Object Detection
- Person/Vehicle Detection in Flooded Regions
- **House-Roof Detection in Flooded Regions**
- Flood Monitoring System

Flooded House Detection on Drone Images

- **House-roof detection** is a task where the goal is to locate and classify bounding boxes around house roofs in an image domain X.
- Given a detection function f, the output is a set of bounding boxes $\mathrm{B}=\{b_1, b_2, \ldots, b_n\}$ where each $b_i = (\mathbf{x}_i, \mathbf{y}_i, \mathbf{W}_i, \mathbf{h}_i)$ defines the coordinates, width, and height of a bounding box that is predicted to contain a house roof.
- Each bounding box b_i is associated with a confidence score $\mathrm{P}(\boldsymbol{b}_i|\mathbf{X})$ representing the probability that b_i contains a house roof.
- Given a confidence threshold T, a bounding box b_i is considered to contain a house roof if: $\boldsymbol{b}_i = \begin{cases} 1, & if\ P(\boldsymbol{bi}|\boldsymbol{X}) \geq T \\ 0, & otherwise \end{cases}$

Artificial Intelligence & Information Analysis Lab

Datasets

Three main datasets have been used.

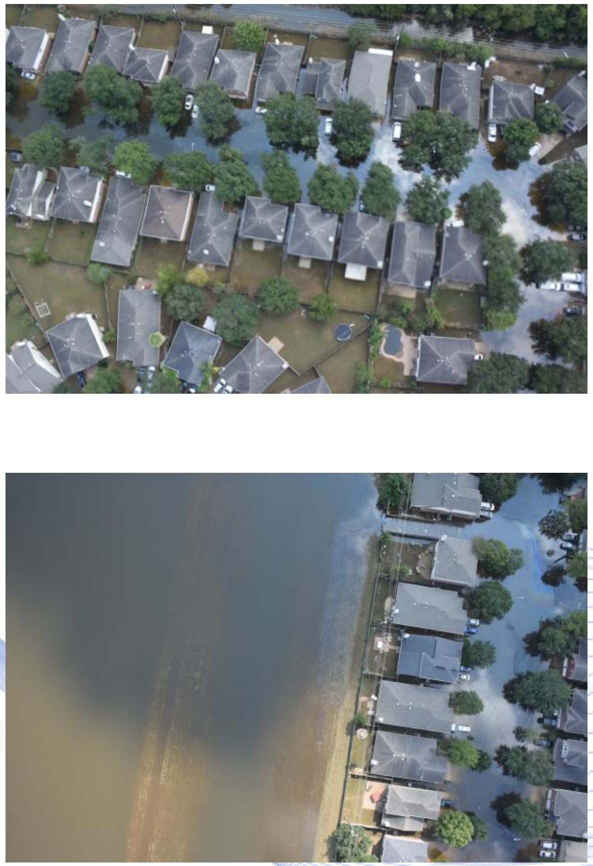
- FloodNet
- Giannitsa (images of the city of Giannitsa, Macedonia, Greece)
- RedRoofs (Google Maps Images)

The following datasets were created:

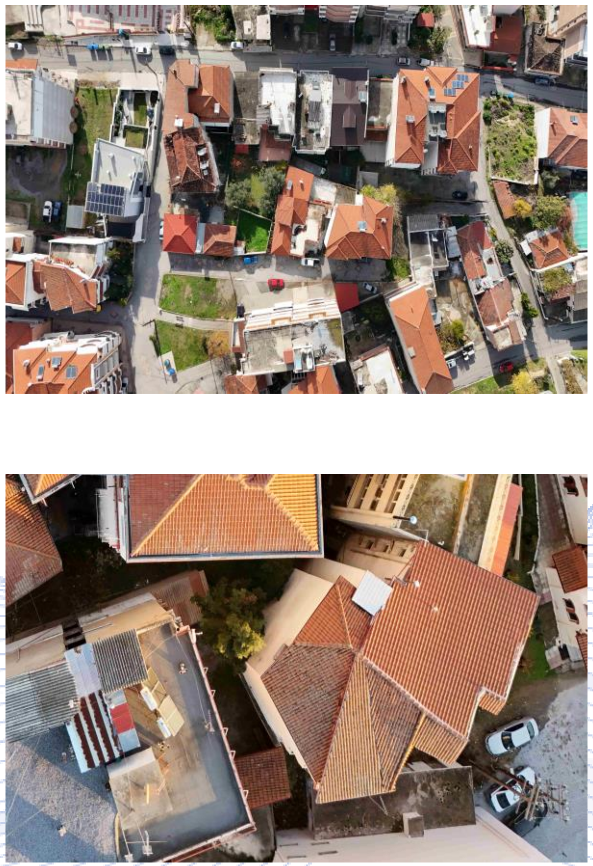
- FRG (Floodnet + RedRoofs + Giannitsa)
- MixedAreas (images from the FRG dataset and internet-sourced images depicting flooded houses)
- NoWater (
  - images without flooded regions for training
  - Images with flooded regions for testing)

Datasets

FloodNet



Giannitsa



RedRoofs

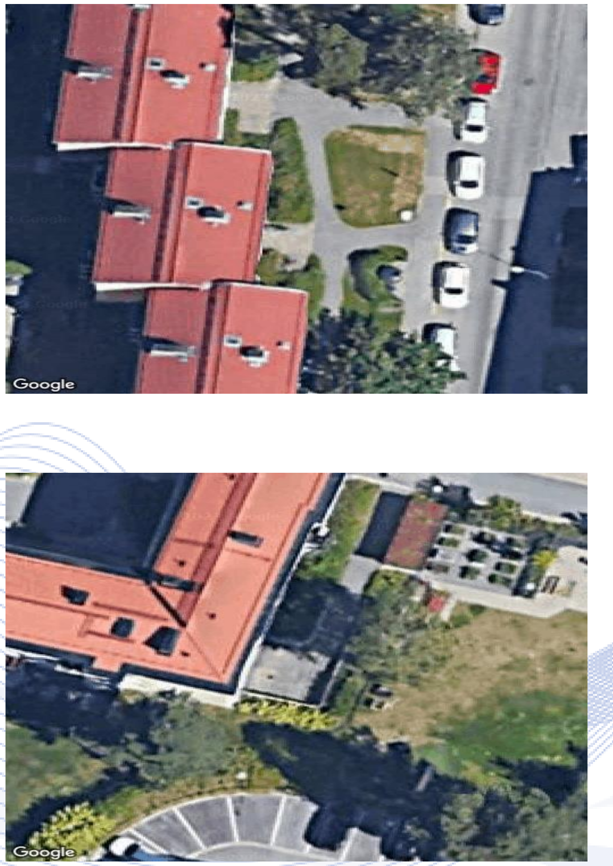


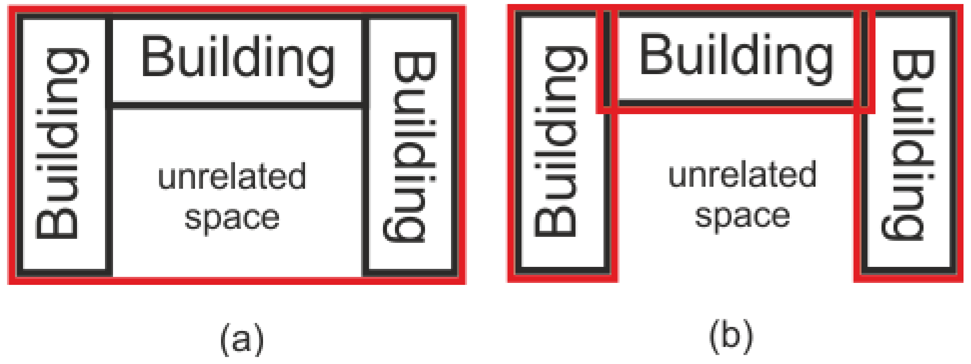
Image samples from the three main datasets.

Datasets

Annotation:

Two ways for image annotation:

a) One bounding box
b) Multiple bounding boxes



Example of annotating a C-shaped building.

Advantages of (b):

- Improved Object Description
- Fewer unrelated pixels
- Reduced false detections

Disadvantages of (b):

- Higher number of bounding boxes
- Longer annotation time

Inference Time

Models	Roof detection inference time	
	MixedAreas test dataset	NoWater test dataset
YOLOv6n Finetuned	7.48 ms	8.42 ms
YOLOv6s Finetuned	8.17 ms	9.20 ms
YOLOv6m Finetuned	15.27 ms	16.73 ms
YOLOv6m6 Finetuned	19.38 ms	21.18 ms
YOLOv6l Finetuned	21.11 ms	22.48 ms
DETR	105.1 ms	108.5 ms
DETR	103.6 ms	180.9 ms
DETR	104.5 ms	111.0 ms

Table 9. Evaluating Inference Speeds for Different Detection Models.

Results

Models	Detector trained on FRG dataset			
	MixedAreas test dataset		NoWater test dataset	
	mAP 0.5	mAP 0.5:0.95	mAP 0.5	mAP 0.5:0.95
YOLOv6m Finetuned	83.0%	64.4%	82.6%	62.7%
YOLOv6l Finetuned	**84.2%**	**66.7%**	**83.5%**	**64.7%**

Table 10. Performance of YOLOv6 Finetuned Models.

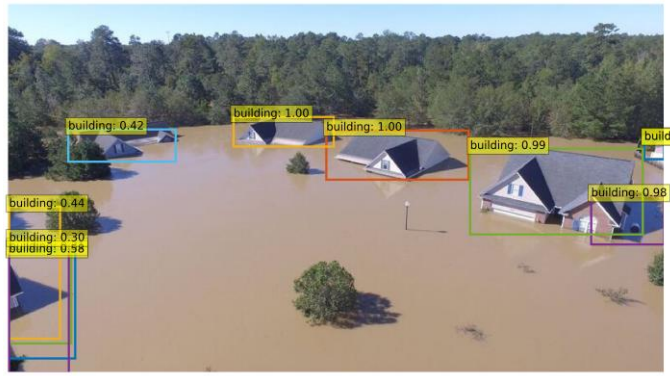
Best detector: YOLOv6l Finetuned

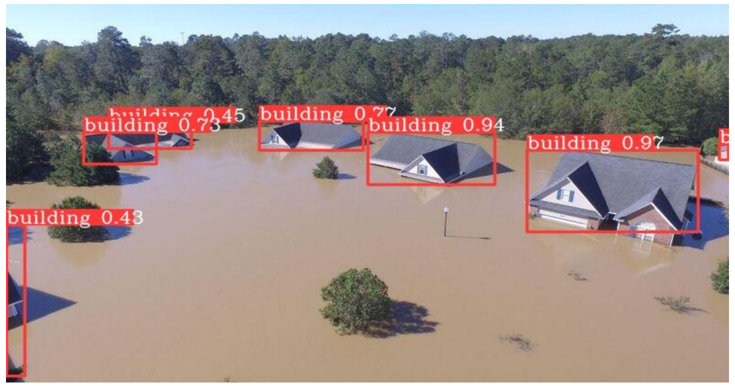
Suitable detector for our drone: YOLOv6m Finetuned

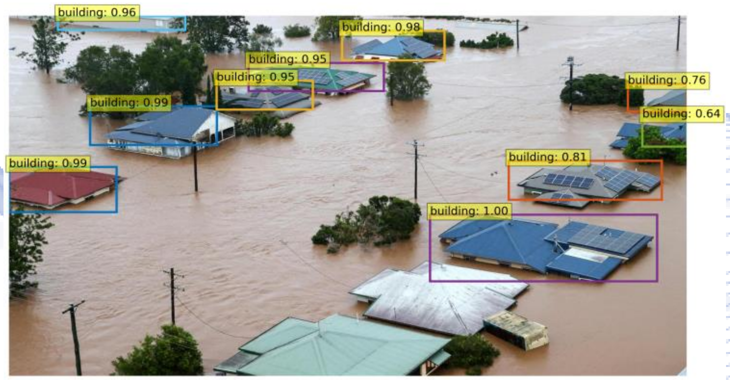
Inference Example

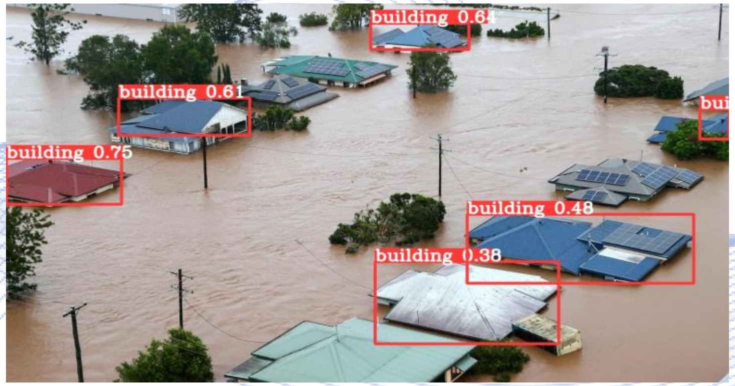
DETR

YOLOv6m









Comparison of DETR and YOLOv6m Inference Results.

Artificial Intelligence & Information Analysis Lab

Flood Region Segmentation on Drone Images

Artificial Intelligence & Information Analysis Lab

Flood Monitoring System

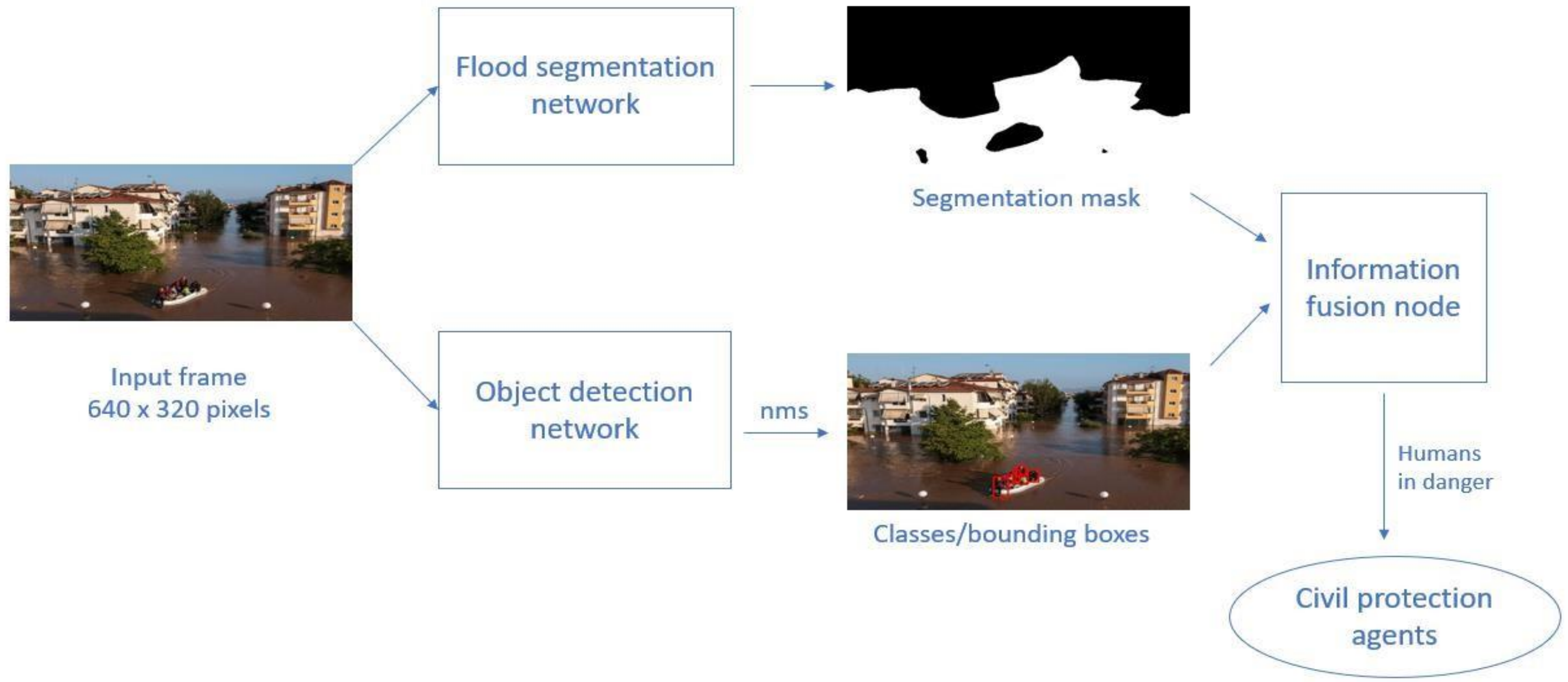
We developed a Deep Learning based Flood Monitoring System, capable of being deployed on the edge.

This system can:

- Segment the flood water, providing its end users with a precise flood mapping tool.
- Detect humans or vehicles in the flood context, even when they are partially submerged in the water.
- Fuse the two outputs and generate rescue alert if the detected object is in danger of the flood.

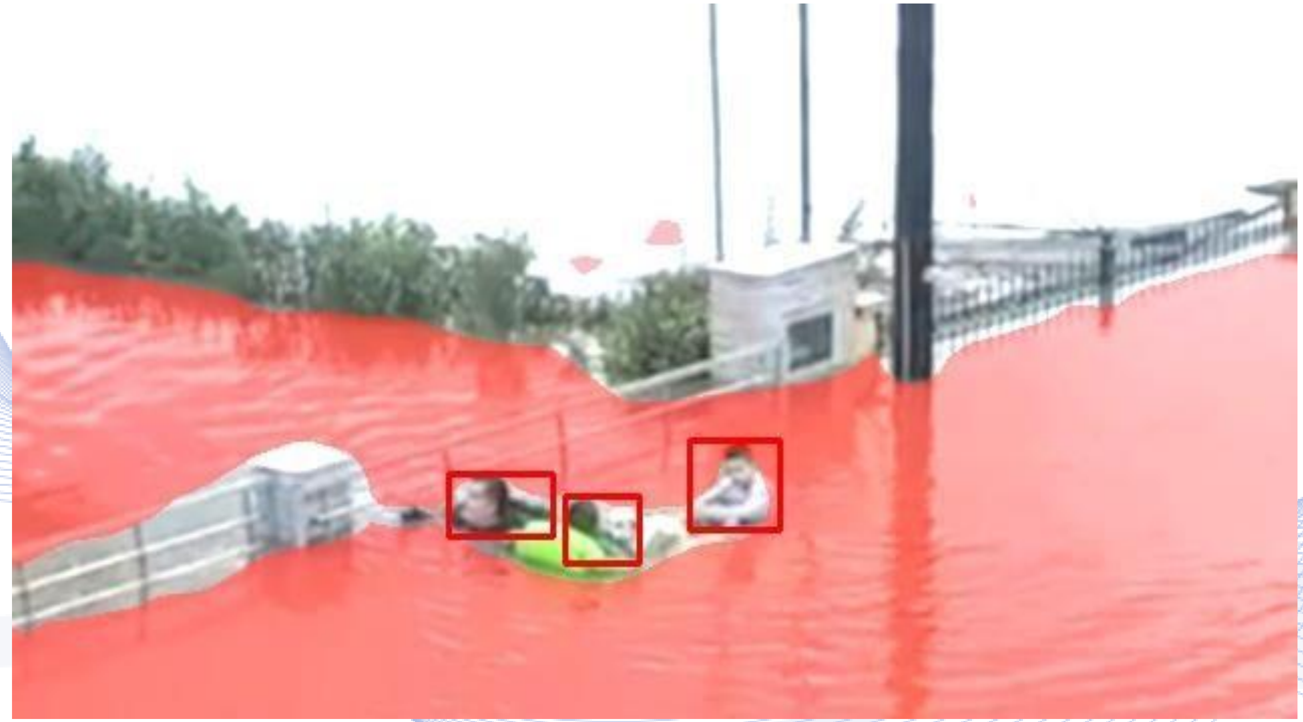
Our system can process big visual data in near real time and send meaningful information to the end users (e.g., the Civil Protection's agents).

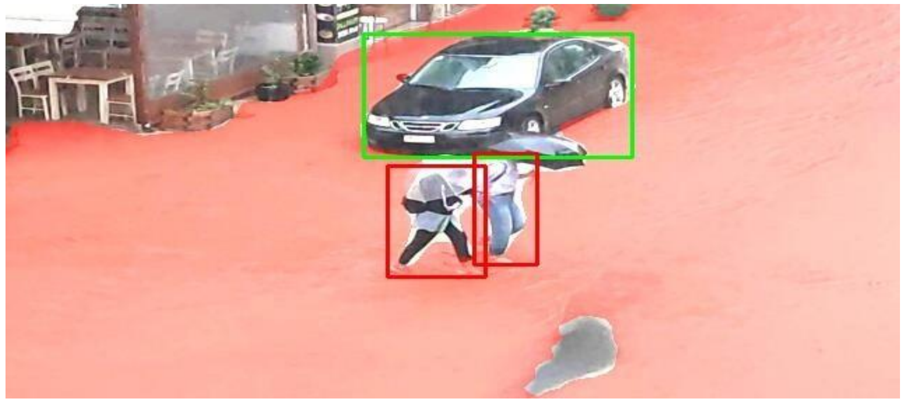
Unified system

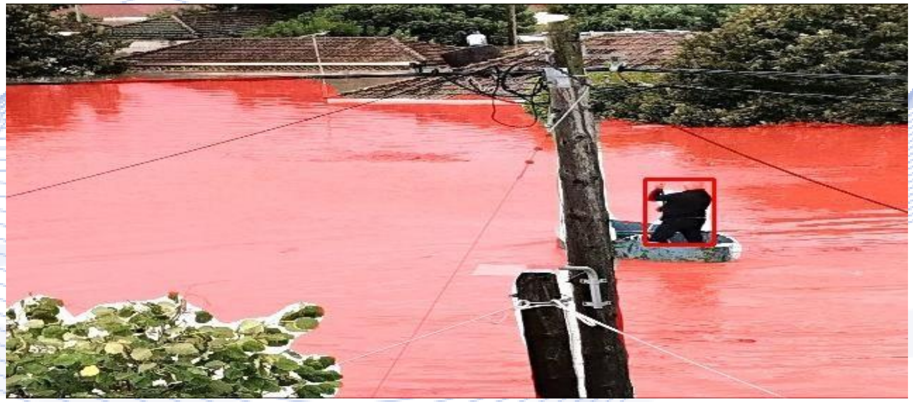


Graphical depiction of our system's abstract architecture.

Rescue Alert examples: Case 1

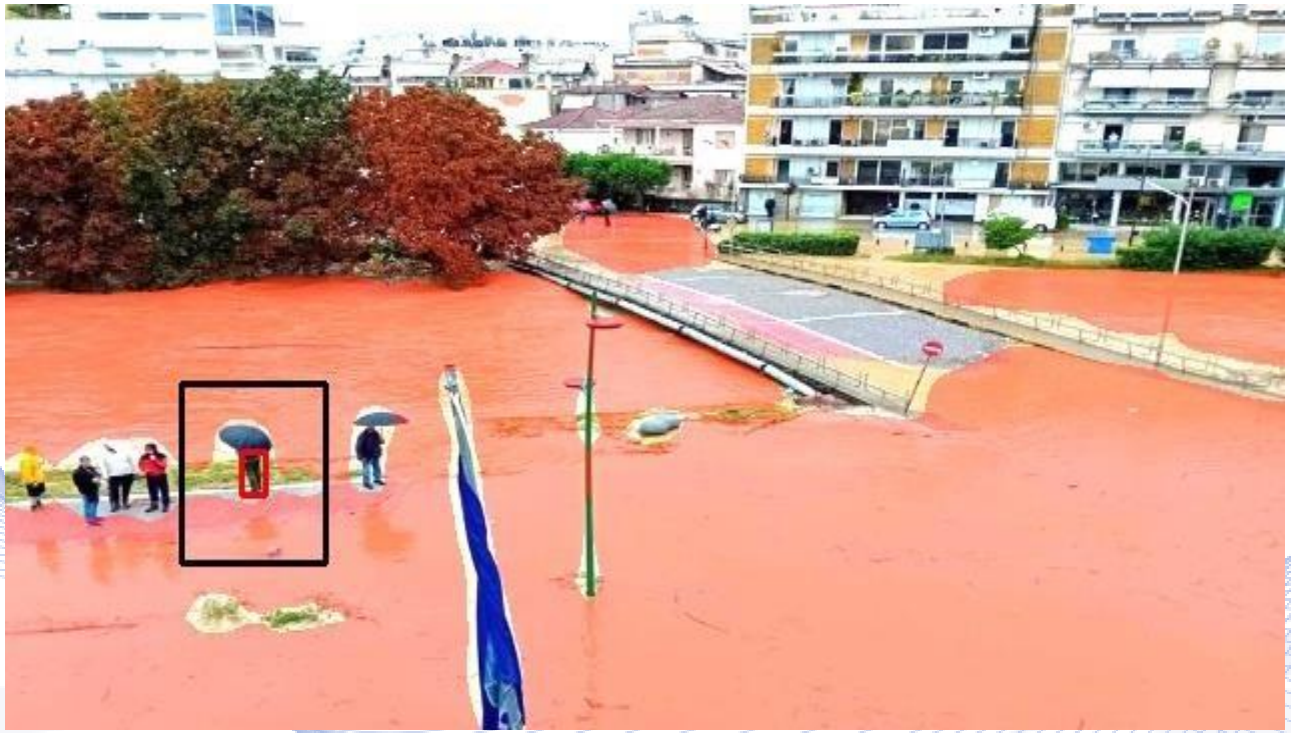


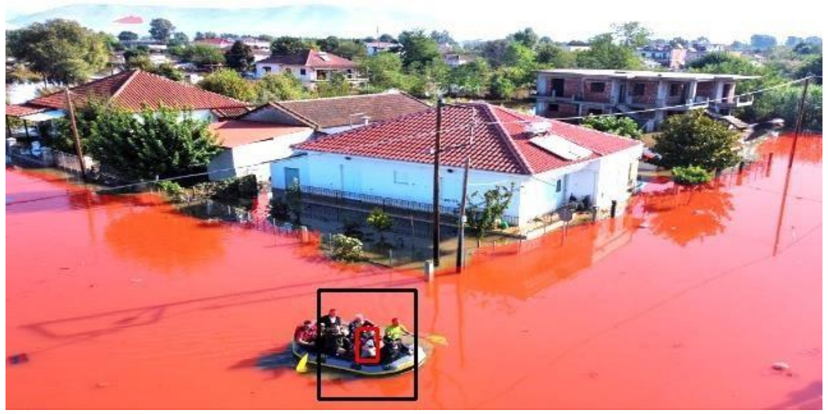


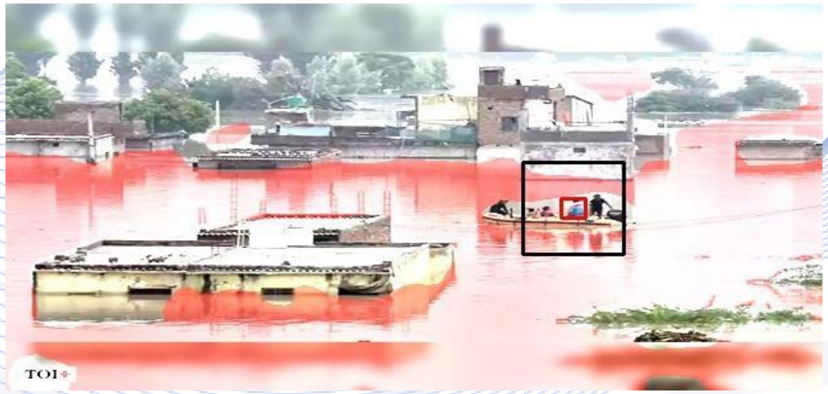


Images from the recent floods in Thessaly.
Our proposed system would generate alerts due to the presence of water pixels inside the bounding boxes.

Rescue Alert examples: Case 2







Sample images displaying the extended bounding boxes.
Our proposed system would generate alerts due to the percentage of water inside the surrounding area.

References

[LI2022] C.Li, L.Li, H.Jiang, et al. “YOLOv6: A Single-Stage Object Detection Framework for Industrial Applications”. arXiv, 2022.

[VAS2017] A. Vaswani, Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I. “Attention is all you need”. NeurIPS (2017).

[CAR2020] N. Carion, F. Massa, G. Synnaeve, N. Usanier, A. Kirillov, S. Zagoruyko. “End-to-end object detection with transformers”. arXiv, 2020.

[WAN2024] A.Wang, H.Chen, et al. “YOLOv10: Real-Time End-to-End Object Detection”. arXiv, 2024.

[LI2023] C. Li, L. Li, Y. Geng, H. Jiang, M. Cheng, B. Zhang, Z. Ke, X. Xu, and X. Chu, “Yolov6 v3. 0: A fullscale reloading,” arXiv preprint arXiv:2301.05586, 2023.

[DING2021] X. Ding, X. Zhang, N. Ma, J. Han, G. Ding, and J. Sun, “Repvgg: Making vgg-style convnets great again,” in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 2021, pp. 13 733–13 742.

Q & A

Thank you very much for your attention!

Contact: Prof. I. Pitas
pitas@csd.auth.gr