A Multi-Agent System for flooding events

Andrea Rafanelli[a,b],
Stefania Costantini[b],
Giovanni De Gasperis[b]

[a]Dipartimento di Informatica – Università di Pisa

[b]Dipartimento di Informatica – Scienza e Ingegneria
e Matematica (DISIM) – Università dell'Aquila

23rd **Workshop "From Objects to Agents",**
September 01–03, 2022, Genova, Italy

Table of Contents

Introduction

Context

- The amount of damage caused by the weather has increased dramatically in recent years;
- Important: the *efficiency* and *efficacy* with which these disasters are identified and addressed;
- Deep Learning (DL) algorithms enable the extraction and classification of visual characteristics with incredible precision.

Problems (motivations for this work)

- Data available are limited;
- The complexity of flood phenomena $\Rightarrow$ the development of reliable meteorological monitoring applications;
- Not only the detection of the problem but also its reporting to the appropriate authorities is needed.

Contributions

1. Development of an agent-based system for real-time flood recognition;
2. Integration of image analysis with the decision-making capabilities of intelligent agents $\Rightarrow$ reactivity and proactivity in emergency contexts;

Overall Framework

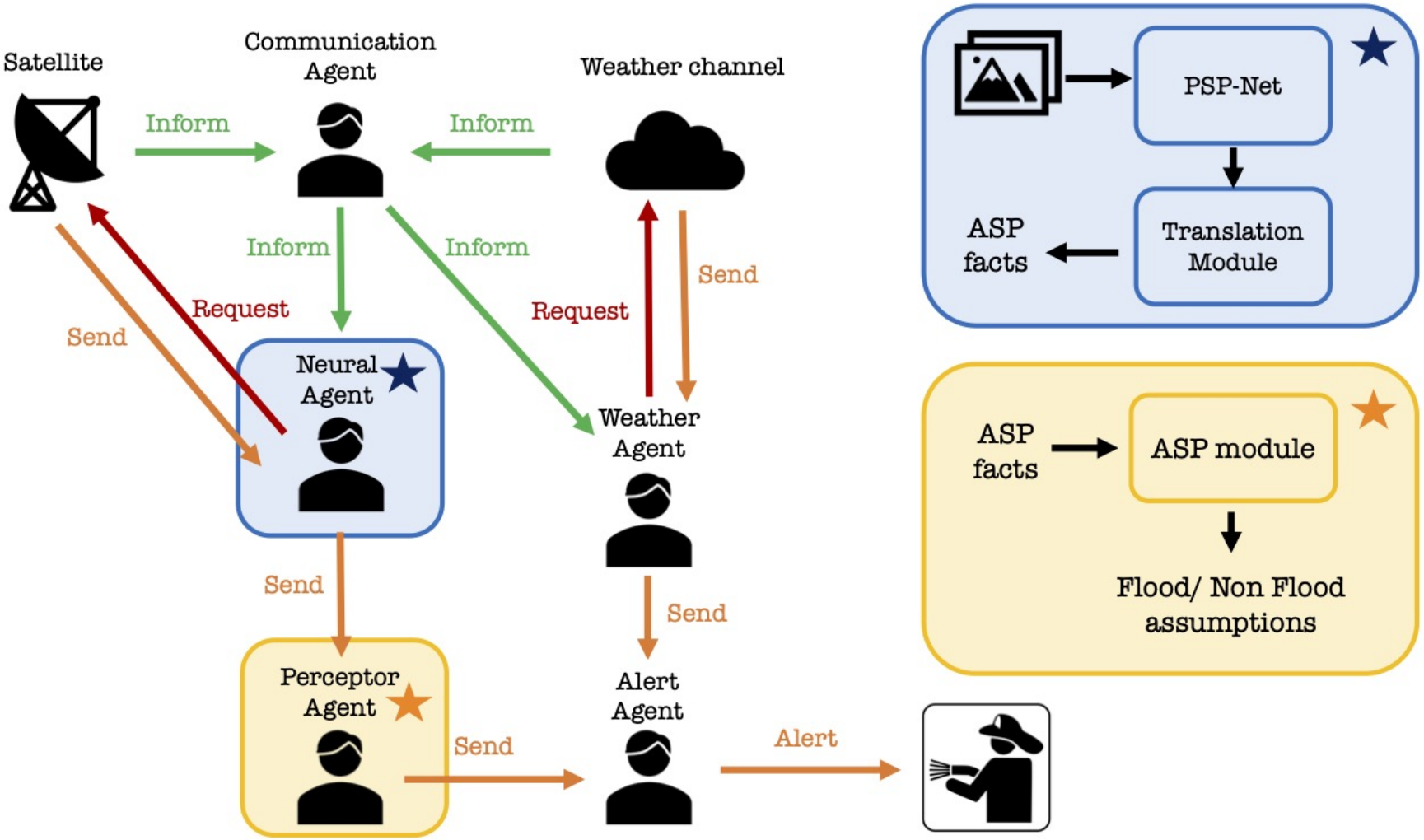


Figure: General framework

Methods I
DALI

- DALI [Costantini and Tocchio, 2002] [Costantini and Tocchio, 2004] is an Agent-Oriented Logic Programming language;
- The DALI communication architecture implements the DALI/ FIPA protocol, which consists of the main FIPA primitives;
- There are some extension to DALI: ASP-DALI[1]and RedisClient[2]a Prolog library that allows DALI MAS agents to send messages to Redis.

Formulation

An agent is defined as: $Ag = <P, E, I, A>$,
where Ag is the agent name, PAg is the "agent program", $E = E_1, ..., E_n$ is the set of the external events, $I = I_1, ..., I_m$ is the set of internal events and $A = A_1, ..., A_k$ is the set of possible actions.

[1] `https://github.com/AAAI-DISIM-UnivAQ/ASP_DALI`
[2] `https://github.com/AAAI-DISIM-UnivAQ/RedisClient`

Methods II
PSP-Net

- Semantic segmentation refers to the process of identifying which elements are present in an image and where they are located;
- Most semantic segmentation models contains two parts, i.e an Encoder and a Decoder;
- The Encoder is responsible for the extracting out features from the image;
- The Decoder is the one which predicts the class of the pixel at the end.

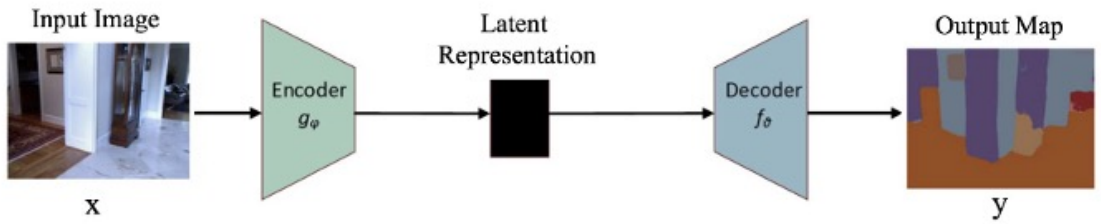


Figure: General semantic segmentation framework, source [Minaee et al., 2022]

Methods II
PSP-Net

- PSP-Net, or Pyramid Scene Parsing Network, is a semantic segmentation model;
- Its encoder contains the CNN backbone and the pyramid pooling module;
- The pretrained CNN serves to extract the feature map;
- The pyramid pooling module exploits the global information by aggregating the overall context;
- The decoder is just another network which takes in features and produce in our case a segmented image.

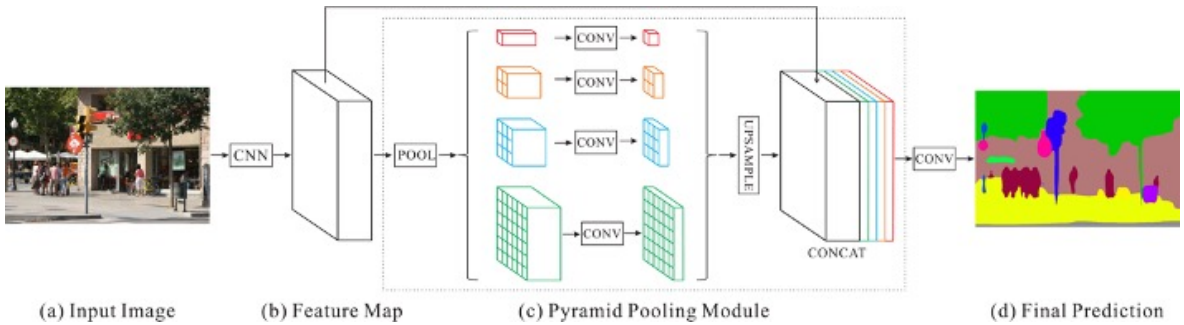


Figure: source [Zhao et al., 2017]

Methods III
Translation module

- The translation module is responsible for translating the neural network's output into an ASP program consisting of ground facts;

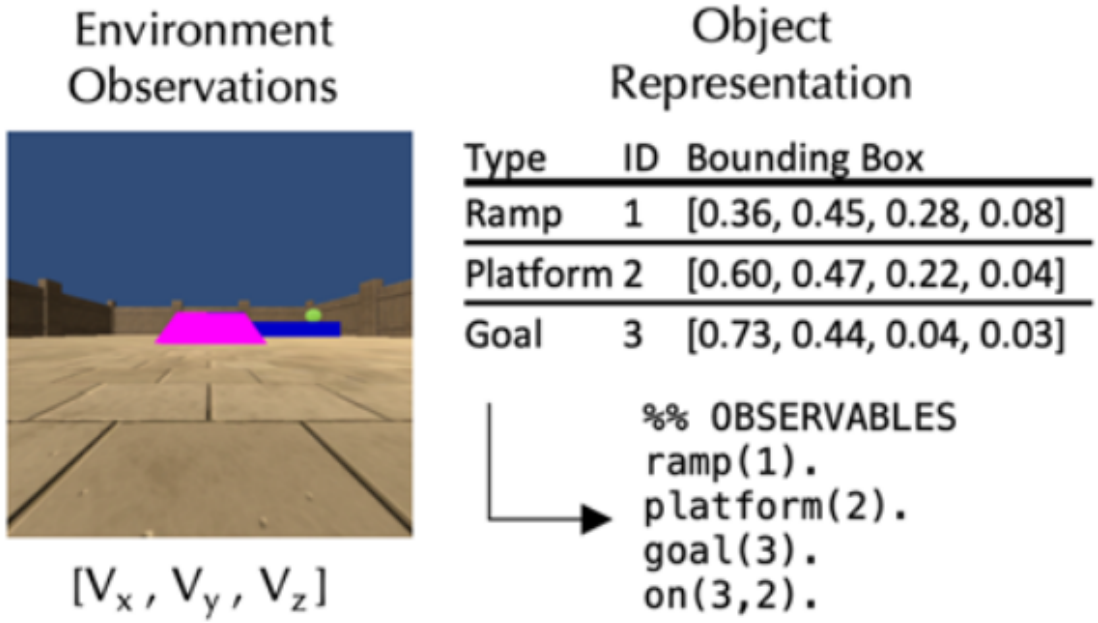


Figure: source [Mitchener et al., 2022]

Methods III
Translation module

- An example of the facts that we want to generate is the following:
  road/3.
  water/3.
  building/3.

Idea:

road(54, 23, 160).
We will find the radius, R, and the positions of objects, (X,Y), with 2D coordinates corresponding to the centroids of objects in aerial/satellite images.

Spatial relations

inside(X,Y).
surrounded(X,Y).
beside(X,Y).
outside(X,Y).
adjacent(X,Y).
etc.

Agents description I
Neural Agent

- It is the agent that receives, processes, and extracts logical predicates from satellite images;
- This agent consists of two modules: the neural network, and the translation module.

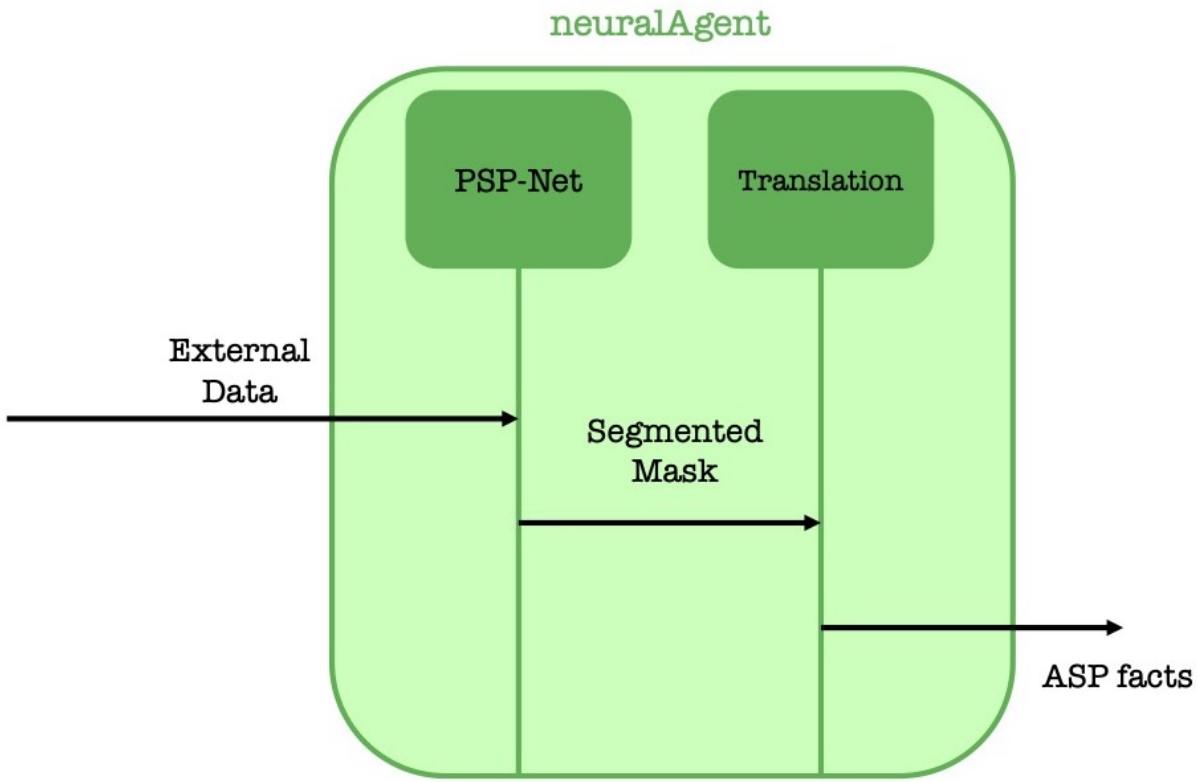


Figure: Neural agent

Agents description I
Neural Agent: PSP-Net experiment

- FloodNet dataset[3]:

Object Class	Images
Building-flooded	245
Building-non-flooded	880
Road-flooded	264
Road-non-flooded	1175
Vehicle	813
Pool	531
Tree	1885
Water	984

Figure: Label distribution,
source [Rahnemoonfar et al., 2021]

- Change number of classes;
- Augmentation: rotation of the original images by 90, 180, 270 degrees.

[3] https://drive.google.com/drive/folders/17cUNXoOAQPaAzYXByoNK9d29jpg2OOY7

Agents description I
Neural Agent: PSP-Net experiment

- **Epochs:** 300
- **Batch size:** 8
- **Encoder:** ResNet101
- **Weights:** Imagenet
- **Optimizer:** SGD
- **Momentum:** 0.9
- **Learning rate:** 0.1

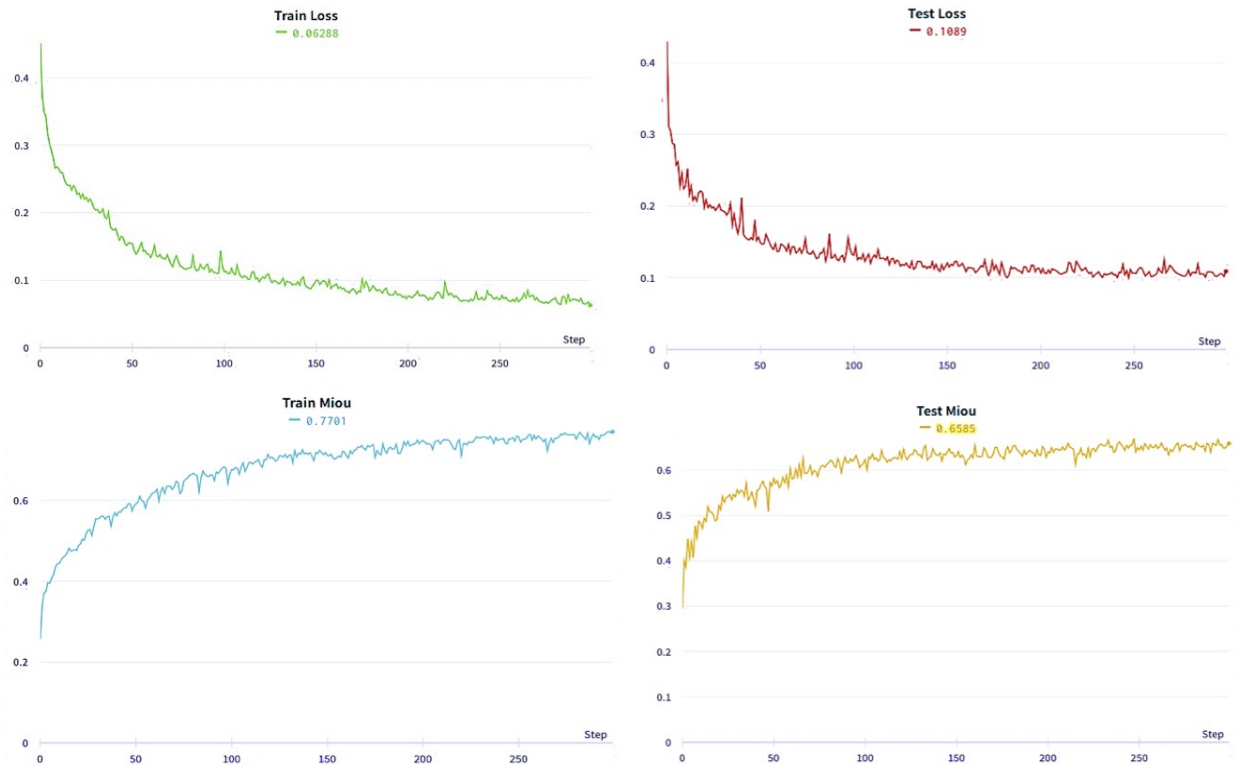
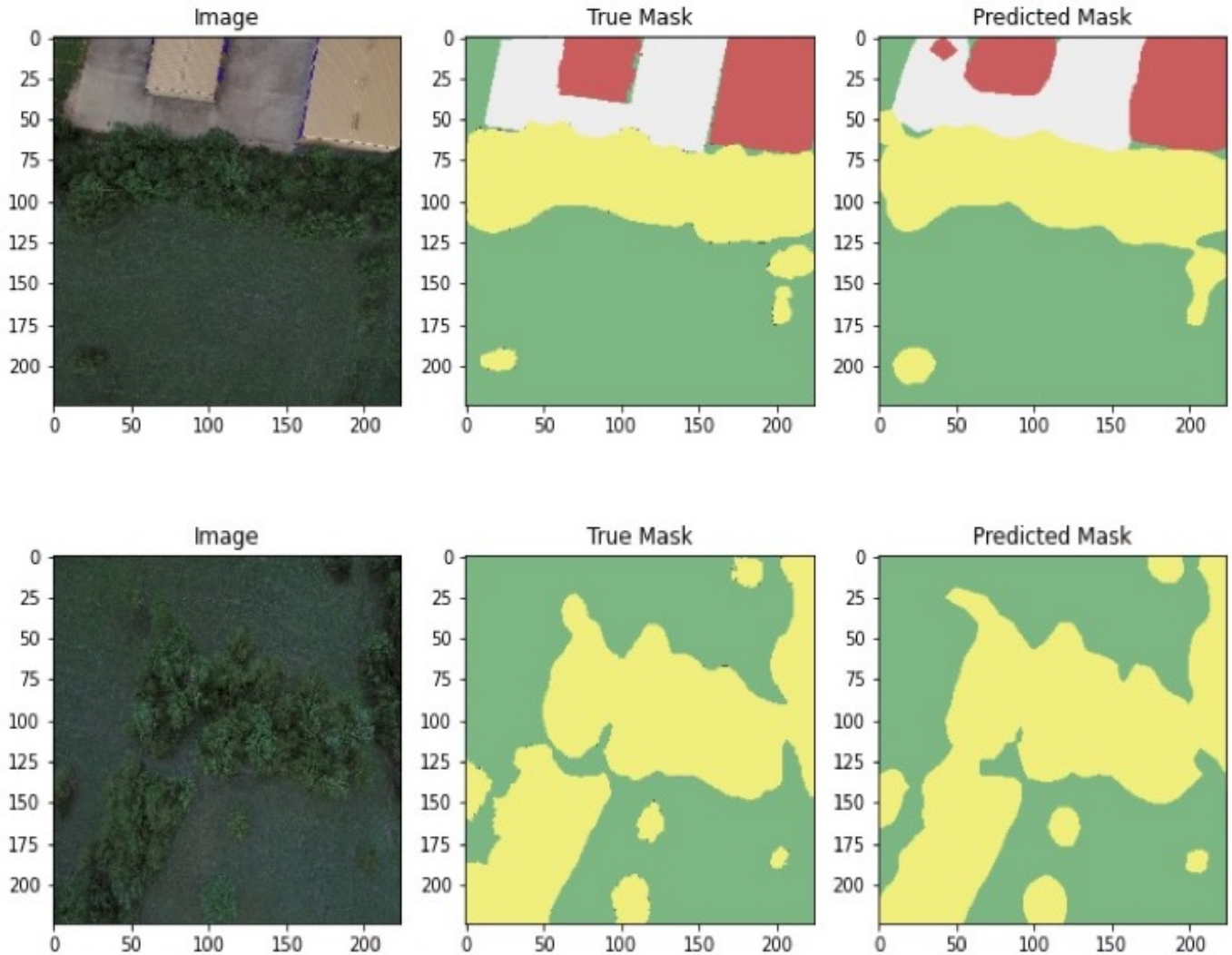


Figure: PSP-Net experiment

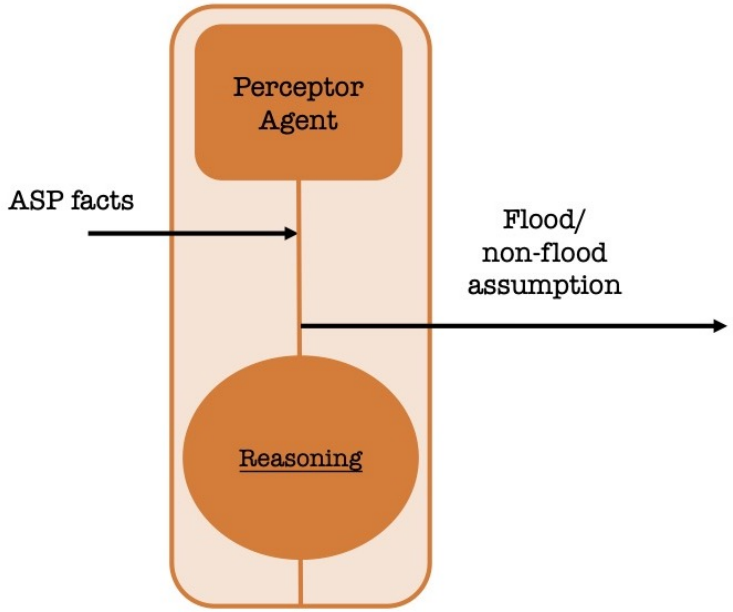
[4] https://github.com/andrearafanelli/Aerial-image-segmentation

Agents description I
Neural Agent: PSP-Net experiment



Agent description II
Perceptor Agent

- It reasons on the retrieved facts (or logical predicates) from the neural network;
- It is connected to an ASP extension module that attempts to determine the image's content;
- It receives the ground facts and augments the observation space with common-sense rules.




```
flooded(building(X, Y, R)) :- building(X, Y, R), surrounded_water(X, Y, R).
surrounded_water(X, Y, R) :- water(V, W, R1), X+R <= V+R1, Y+R <= W+R1.
```

Agent description III
Weather Agent

- It is the agent responsable of managing weather information;
- If there are weather alerts, or alarming situations, its role is to communicate with Alert agent.

GREEN	Intense and dangerous phenomena are not expected
YELLOW	Intense phenomena expected, locally dangerous
ORANGE	More intense than normal phenomena expected : dangerous
RED	Extreme phenomena expected : very dangerous

```
red(X,Y) :- heavy_rain(X,Y).
heavy_rain(X,Y) :- assert(danger(X,Y,1)), assert(meteo_danger).
meteo_dangerI :> send_message(alertAgent, danger(X,Y)).
```

Agent description IV
Alert Agent

- It processes the communications received by Perceptor agent and Weather agent;

Actions:

It takes two possible actions:

1. If both agents inform that there are flooding problems in the area of interest: notifies the authorities in real time (ALERT);
2. If one agent provides notification of a possible flooding problem: inform the authorities in real time (PRE-ALERT).

Conclusion

Conclusions

- An integrated system to provide assistance and support during catastrophic weather events;
- Effectiveness of combination of Multi-Agent Systems and Machine Learning (ML).

Future works

- Translation module full implementation;
- ASP program full implementation;
- Integration of external sources (through API) to get data;
- Detectron2 library to get better results;

References I

Costantini, S. and Tocchio, A. (2002).
A logic programming language for multi-agent systems.
In Flesca, S., Greco, S., Ianni, G., and Leone, N., editors, *Logics in Artificial Intelligence*, pages 1–13, Berlin, Heidelberg. Springer Berlin Heidelberg.

Costantini, S. and Tocchio, A. (2004).
The dali logic programming agent-oriented language.
In Alferes, J. J. and Leite, J., editors, *Logics in Artificial Intelligence*, pages 685–688, Berlin, Heidelberg. Springer Berlin Heidelberg.

Minaee, S., Boykov, Y., Porikli, F., Plaza, A., Kehtarnavaz, N., and Terzopoulos, D. (2022).
Image segmentation using deep learning: A survey.
IEEE Transactions on Pattern Analysis and Machine Intelligence, 44(7):3523–3542.

References II


Mitchener, L., Tuckey, D., Crosby, M., and Russo, A. (2022).
Detect, understand, act: A neuro-symbolic hierarchical reinforcement learning framework.
Mach. Learn., 111(4):1523–1549.


Rahnemoonfar, M., Chowdhury, T., Sarkar, A., Varshney, D., Yari, M., and Murphy, R. (2021).
Floodnet: A high resolution aerial imagery dataset for post flood scene understanding.
IEEE Access, 9:89644–89654.


Zhao, H., Shi, J., Qi, X., Wang, X., and Jia, J. (2017).
Pyramid scene parsing network.
In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.