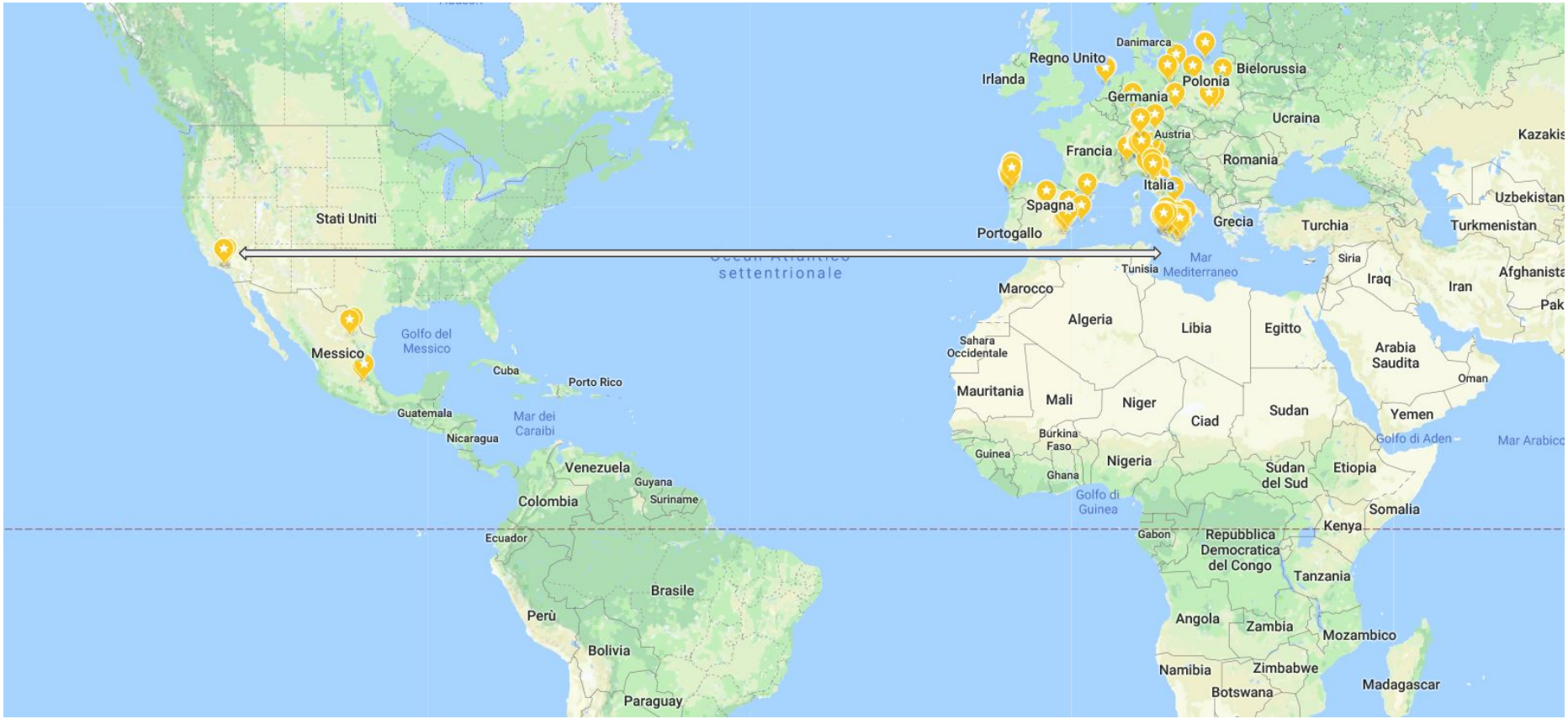
# CoArse-to-Fine Q-Learning

## for Object Localisation on VHR Images

:: Dario Zanca ::

SAILab SEMINARS

17 Maggio 2018


Danimarca
Regno Unito
Irlanda
Bielorussia
Polonia
Germania
Ucraina
Austria
Francia
Romania
Italia
Spagna
Portogallo
Grecia
Turchia
Uzbekistan
Turkmenistan
Kazakis
Stati Uniti
Oceano Atlantico settentrionale
Mar Mediterraneo
Tunisia
Siria
Iraq
Iran
Afghanista
Pak
Marocco
Algeria
Libia
Egitto
Sahara Occidentale
Arabia Saudita
Oman
Mauritania
Mali
Niger
Ciad
Sudan
Yemen
Golfo di Aden
Mar Arabico
Golfo del Messico
Messico
Cuba
Porto Rico
Guatemala
Nicaragua
Mar dei Caraibi
Burkina Faso
Guinea
Nigeria
Ghana
Golfo di Guinea
Sudan del Sud
Etiopia
Somalia
Kenya
Venezuela
Guyana
Suriname
Colombia
Ecuador
Gabon
Repubblica Democratica del Congo
Tanzania
Brasile
Perù
Bolivia
Angola
Zambia
Mozambico
Namibia
Zimbabwe
Botswana
Madagascar
Paraguay


405
5
210
101
134
Pasadena
210
Beverly Hills
Los Angeles
Santa Monica
605
405
110
105
710
91
Torrance
Anaheim
5
Long Beach
Santa Ana
55
405
Irvine
Huntington
Beach
405
73
Google



Caltech



Sara Beery
PhD Student



Natalie Bernat
PhD Student



Eli Cole
PhD Student



Mason McGill
PhD Student



Serim Ryou
PhD Student



Cristina Segalin
Postdoc



Joe Marino
PhD Student



Caroline Murphy
Group Administrator



Oisin Mac Aodha
Postdoctoral Fellow



Matteo Ronchi
PhD Student



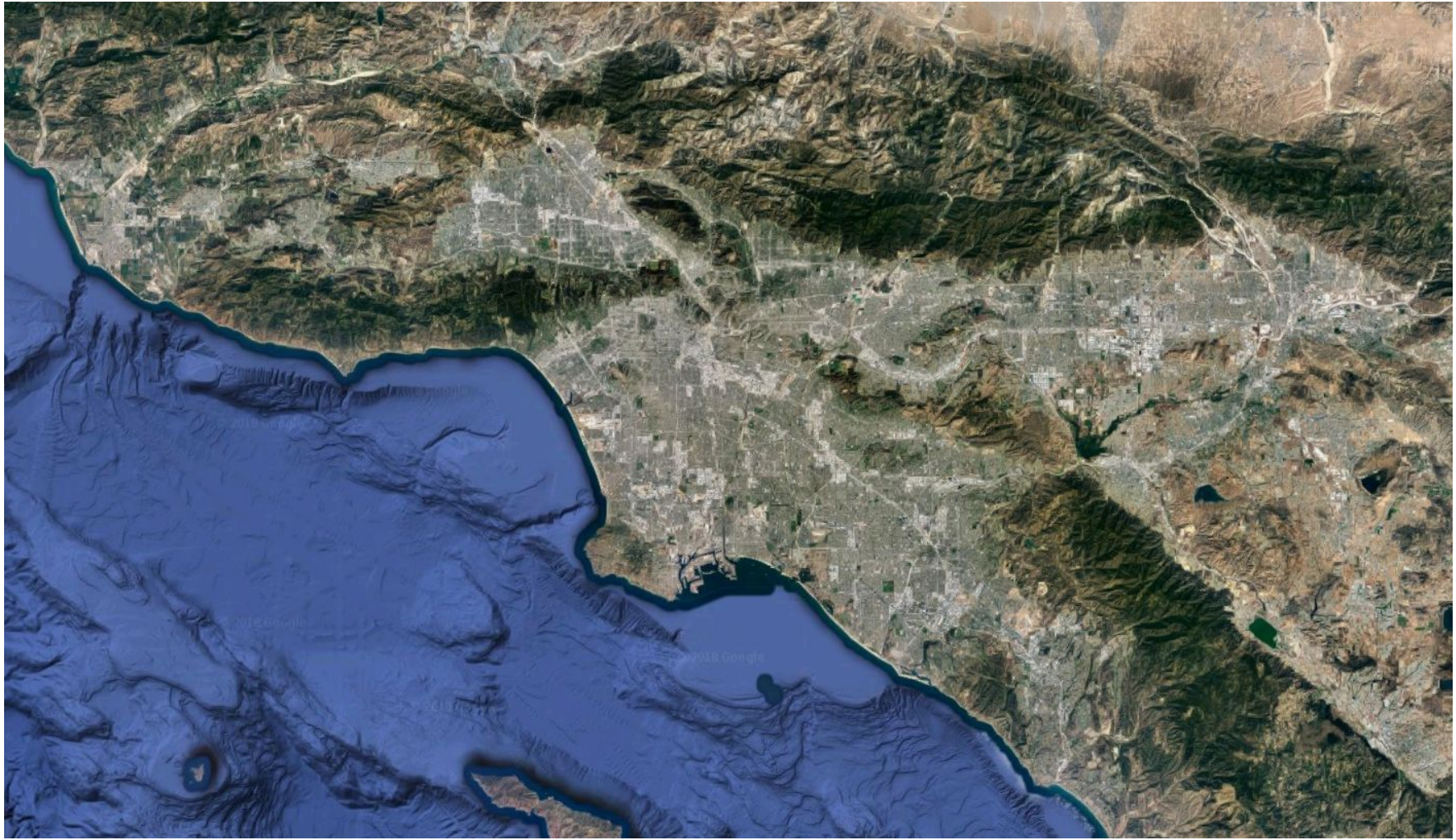
Grant Van Horn
PhD Student



Tony Zhang
PhD Student



Pietro Perona
Principal Investigator



# Environment

Given a map

$$\mathcal{M} : \mathcal{R}^2 \rightarrow \mathcal{R}^n$$

of pixels, we can make queries

$$q(x, y, z) = [\text{image}]$$

which output a crop of the map of fixed resolution $n \times n$, centered in a certain position $(x, y)$ and zoom $z$ .

# Environment

Assumptions about the environment:

- The set of possible zooms is discrete, i.e. $\mathcal{Z} := \{z_1, ..., z_k\}$, where $z_1$ provides the coarser scale and $z_k$ the finest.
- Assuming the fixed resolution $n \times n$, a query at zoom $z_i$ samples from an area four times bigger then $z_{i+1}$.

# Task

We want to localise all the instances of an object of interest (OOI) in $\mathcal{M}$.

Assumptions about the problem:

- OOI are visible only at the finest scale that correspond to $z_k$; a model $\mathcal{O}$ for OOI is given (pre-trained).
- $\mathcal{M}$ is a very high resolution (VHR) image.
- Sensing costs a lot. We want to minimize the number of queries performed. On the other hand, we assume cheap processing and infinite memory.
- OOI are sparsely distributed on $\mathcal{M}$; a linear search is not convenient. This implies that there exists at least one zoom level before the finest, in which feature observable at that level are correlated with the presence of OOI.

# Model

We want to determine the *value* function

$$f_\pi(s, a),$$

that defines the expected return starting from a state $s$ = [image], taking the action $a$, and thereafter following the policy $\pi$. The action $a$ indicates the position $(u, v)$ and the zoom $z$ at which make the next query,

$$a = (u, v, z).$$

# The value function

More precisely, and omitting the dependence on $\pi$,

$$f(s,a) = E\left[r + \gamma \max_{a'} f(s',a') | s,a\right],$$

Losing generality for the sake of simplicity, we assume that decisions at each level of zoom $z_i$ are independent. Given the set of function

$$\{f_1, ..., f_{k-1}\} \cup \{f_k \equiv 0\}$$

we define $f$ by cases

$$f(s,a) = \Big\{ f_i(s,a_i), \text{ if } z = z_i$$

# Activation path

At each step, we select a location for each zoom level, from the coarsest to the finest. We compute a *fixed saccade activation path*

$$a = (a_1, a_2, ..., a_{k-1}),$$

where,

$$a_i = \arg\max f_i(s_i, \cdot)$$

and, if $(u_o, v_0)$ are the origin's coordinates,

$$s_i = q(u_{i-1}, v_{i-1}, z_i).$$

This path uniquely determines a *final* patch

$$q(u_{k-1}, v_{k-1}, z_k) \subset \mathcal{M}$$

# Training

The reward signal is defined as follow,

$$r(a_i) = \begin{cases} 0, \ i < k-1 \\ \mathcal{O}(s_k), \ i = k-1 \end{cases}$$

$$f_i(s_i, a_i) = E\left[r(a_i) + \gamma \max_{a_{i+1}} f_{i+1}(s_{i+1}, a_{i+1}) | s_i, a_i\right]$$

$$L = \sum_{i=1}^{k-1} \left( \underbrace{r(a_i) + \gamma \max_{a_{i+1}} f_{i+1}(s_{i+1}, a_{i+1})} - f_i(s_i, a_i) \right)$$

# Training

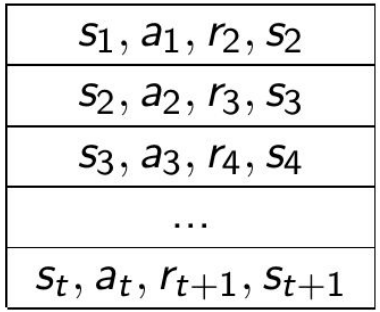
With NNs, this process diverges due to

- Correlation between examples
- Non-stationary targets

To solve this problem we:

- build data-set from agent's own experience, then train sampling from it

- define <experiencing> and <training> phases



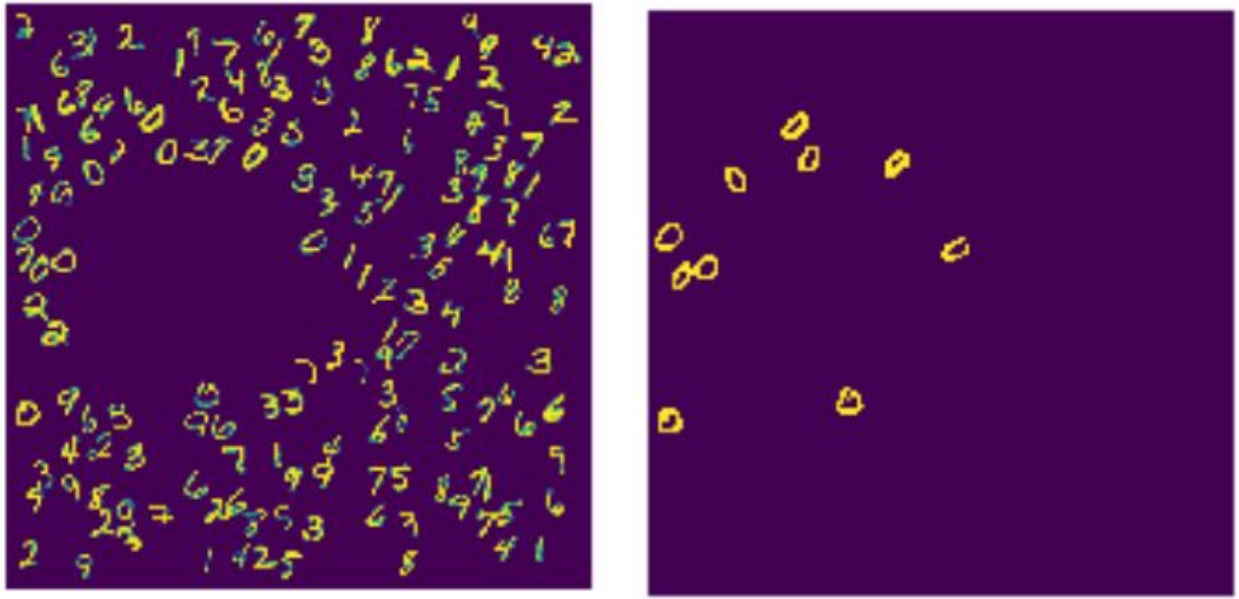
Experience buffer

# Memory

Since we are interested in the localisation of non-already-explored OOIs, we add the extra channel $\mathcal{V}$ in input that keeps memory of the locations already visited, so that,

$$f_i : \mathcal{R}^{n^2 \times m^2 \times 2} \rightarrow \mathcal{R}.$$

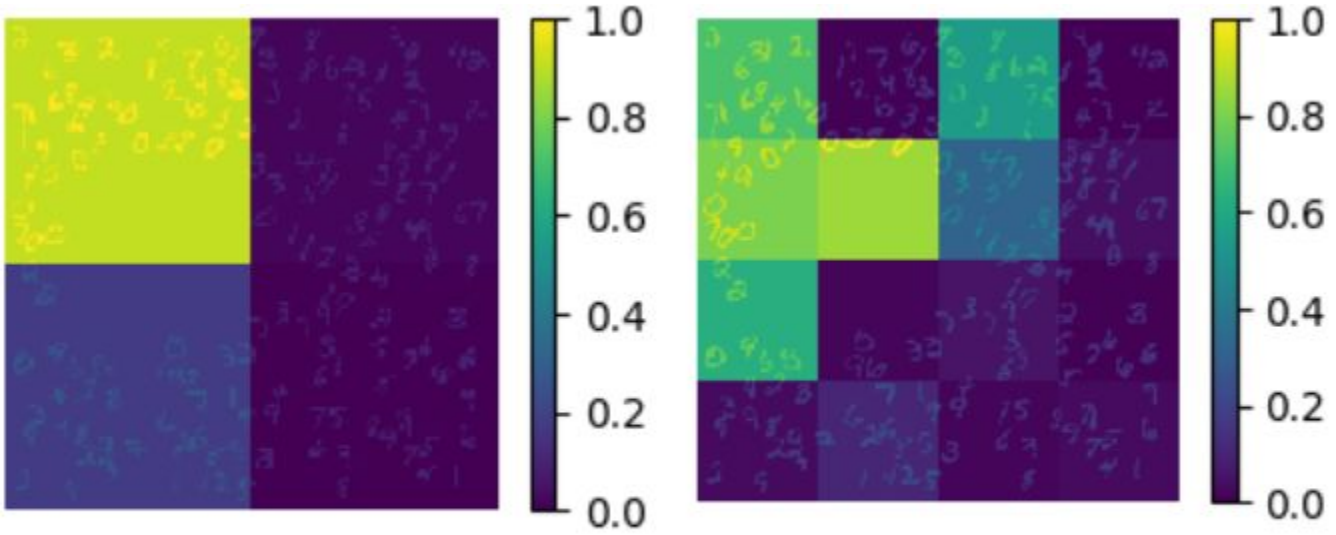
# Experiment (Environment)



(a) Map $\mathcal{M}$ (b) Labels

Figure 1: Example of environment with ten OOIs distributed close to empty square area.

# Experiment (value functions)




(a) $f_1$ (b) $f_2$

Figure 2: Example of value functions $f_1$ and $f_2$ after 5000 episodes of training.

# Experiment (reward during training)

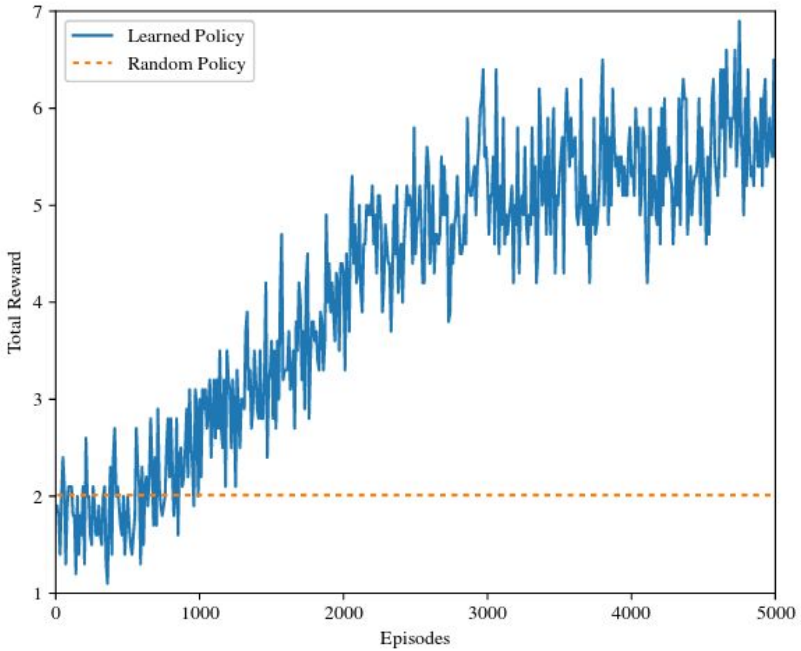



Figure 3: **Total reward during training.** For each episode, a new environment is generated.

# Learned behavior



After 1000 episodes



After 5000 episodes

Thank you!

Questions?