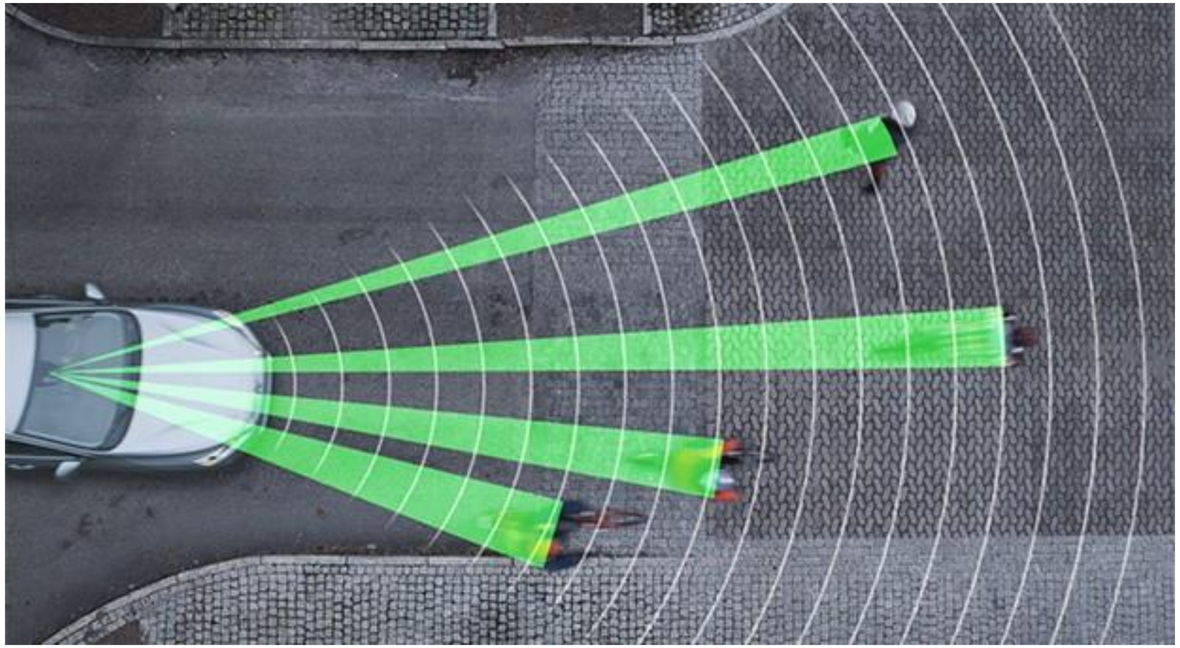
PSU Student Research Symposium 2017

Bayesian Optimization for Refining Object Proposals, with an Application to Pedestrian Detection

Anthony D. Rhodes
5/10/17

Portland State UNIVERSITY

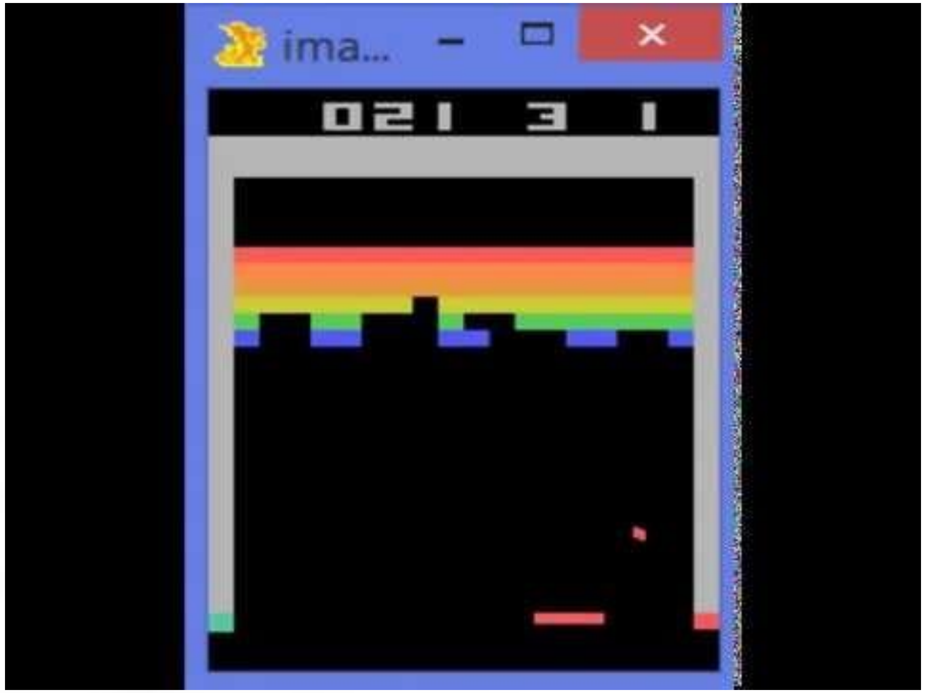


What is Machine Learning?

- Machine learning is a scientific paradigm that “gives computers the ability to learn without being explicitly programmed.”
- In the main, machine learning focuses on the task of pattern recognition in complex – sometimes “big” – data settings. Typically, such algorithms “learn” to perform *prediction* or *classification* through inference from many training examples.
- *Please feel free to ask questions and interrupt me at any time, if any material requires further clarification.

What is Machine Learning? (cont'd)

- There are many contemporary, cutting-edge applications of machine learning, including: cancer detection (a classification problem), natural language processing, data security and anomaly detection (unsupervised learning), automated vehicles (reinforcement learning) and recent state-of-the-art AI, such as *Watson*, the *alphaGO* project and automated Atari game-playing – to name but a few examples.



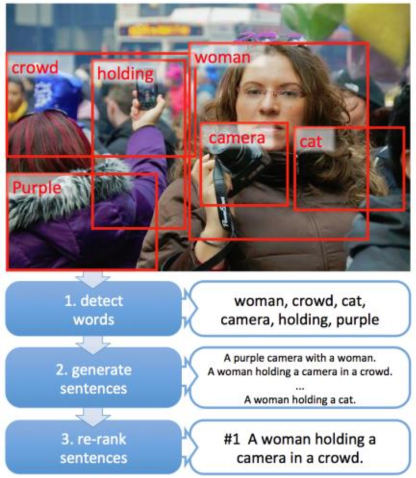
Google DeepMind Atari

Google DeepDream

Computer Vision

- In our research we focus on a sub-field of A.I. known as *computer vision*.

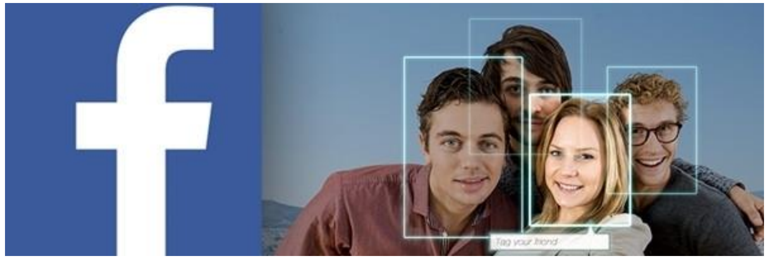
- Current work in computer vision commonly focuses on tasks/problems such as: *object detection* (e.g. is there a pedestrian in this image?), *object localization* (where is the pedestrian?), automated *image captioning*, situation and *activity recognition*, and *video tracking*.

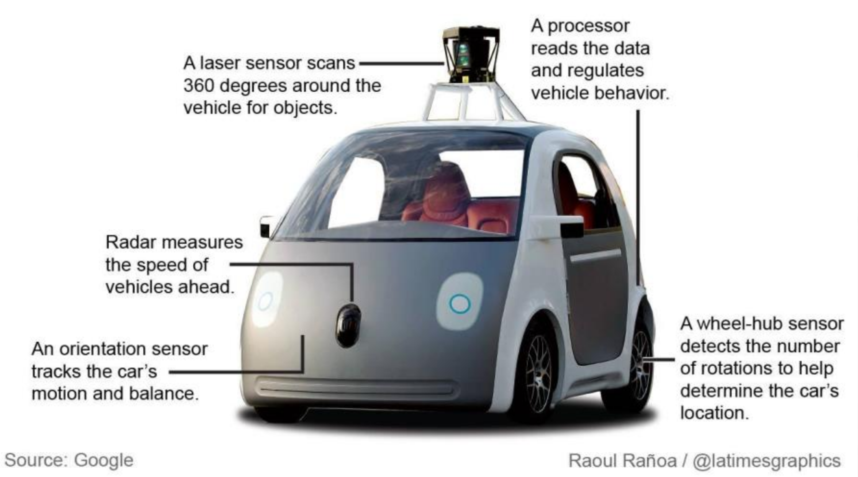


1. detect words — woman, crowd, cat, camera, holding, purple

2. generate sentences — A purple camera with a woman.
A woman holding a camera in a crowd.
...
A woman holding a cat.

3. re-rank sentences — #1 A woman holding a camera in a crowd.





A laser sensor scans 360 degrees around the vehicle for objects.

A processor reads the data and regulates vehicle behavior.

Radar measures the speed of vehicles ahead.

An orientation sensor tracks the car's motion and balance.

A wheel-hub sensor detects the number of rotations to help determine the car's location.

Source: Google

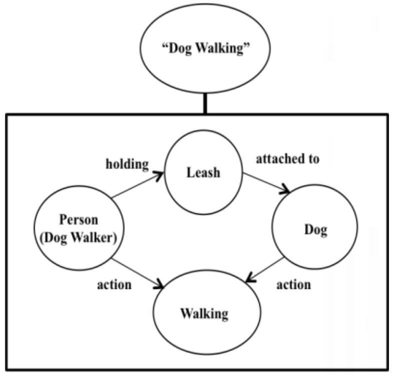
Raoul Rañoa / @latimesgraphics

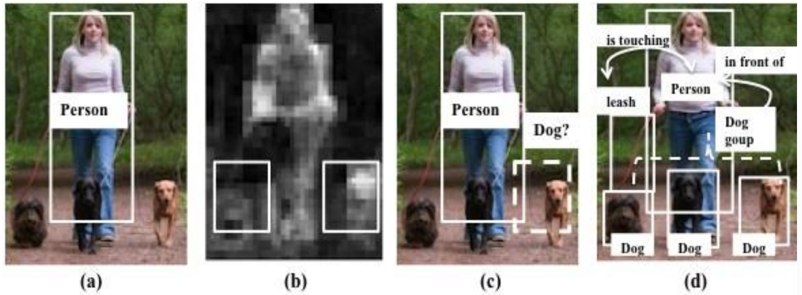
- The preeminent goal of computer vision/A.I. is to e an algorithm with true cognitive-visual intelligence.

Active Object Localization in Visual Situations

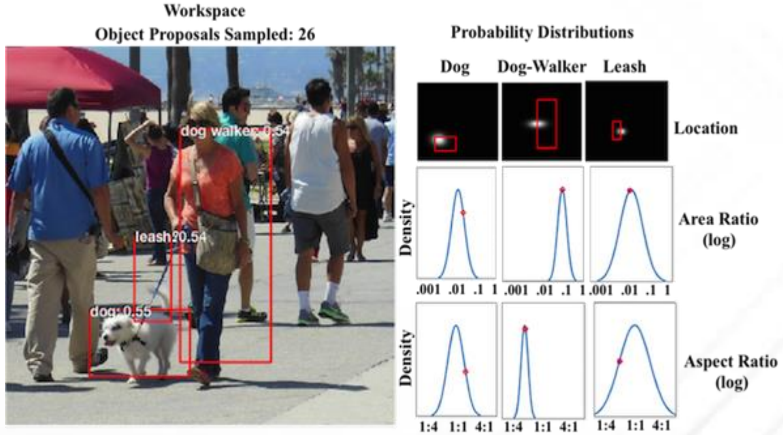
- *Situate* is a computer vision framework for active object localization is visual situations.

- We define a 'visual situation', *e.g*. 'dog-walking', as an abstract concept whose image instantiations are linked more by their common spatial and semantic structure than by low-level visual similarity.





- Our system combines given and learned knowledge of the structure of a particular situation, and adapts that knowledge to a new situation instance as it actively searches for objects.



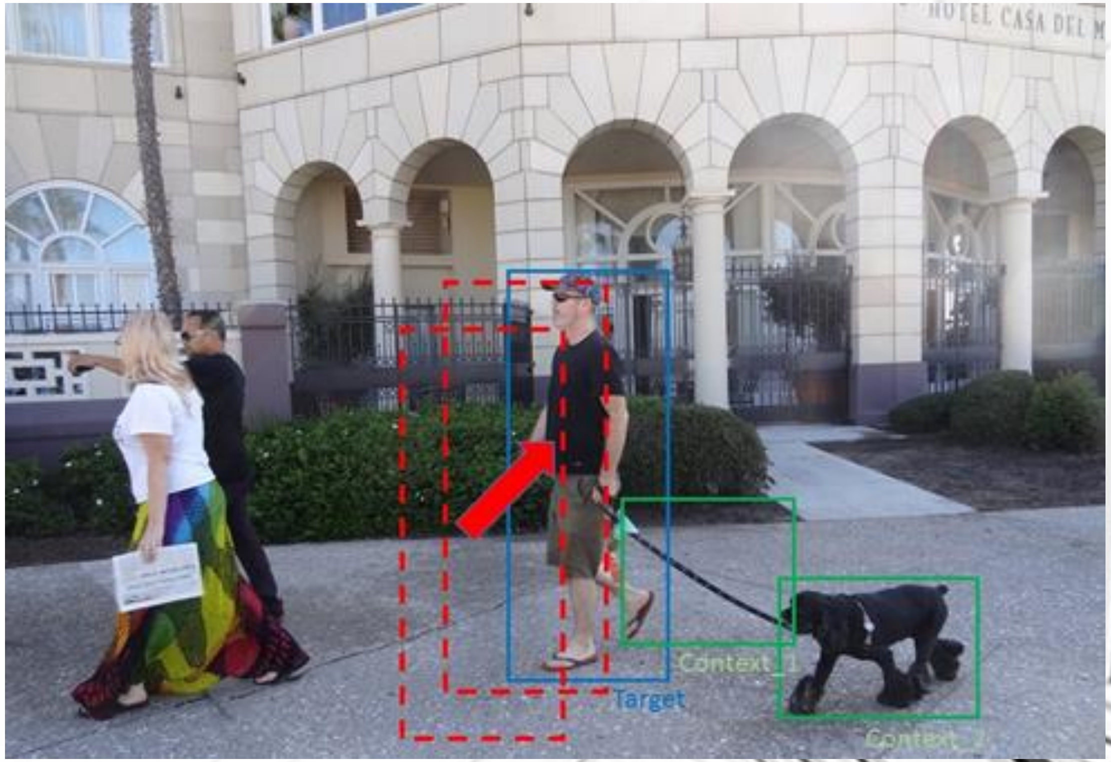
Project Objectives

- Develop efficient algorithm employing an *active* search for target objects, using, possibly known “situational context”; results shown for pedestrian detection.

- The quality of this search is generally determined by (2) criteria:

(1) How well does the proposal bounding-box match the ground truth (i.e. a tightly-cropped box) for the target object? This measure is commonly reported as the “overlap” or IOU (intersection over union).

(2) How efficient is the search? E.g. How “long” does it take, how many proposals are required?



Background

- Research presented here draws from (4) papers:

(1) Rhodes, A. D., Witte, J., Mitchell, M., and Jedynak, B. Using Gaussian Processes and Context for Active Object Localization. (2017)

(2) Rhodes, A. D., Witte, J., Mitchell, M., and Jedynak, B. Bayesian Optimization for Refining Object Proposals. (2017)

(3) Rhodes, A. D., Quinn, M. H., and Mitchell, M. Fast On-Line Kernel Density Estimation for Active Object Localization. (2016)

(4) Quinn, M. H., Rhodes, A. D., and Mitchell, M. Active Object Localization in Visual Situations. (2016)

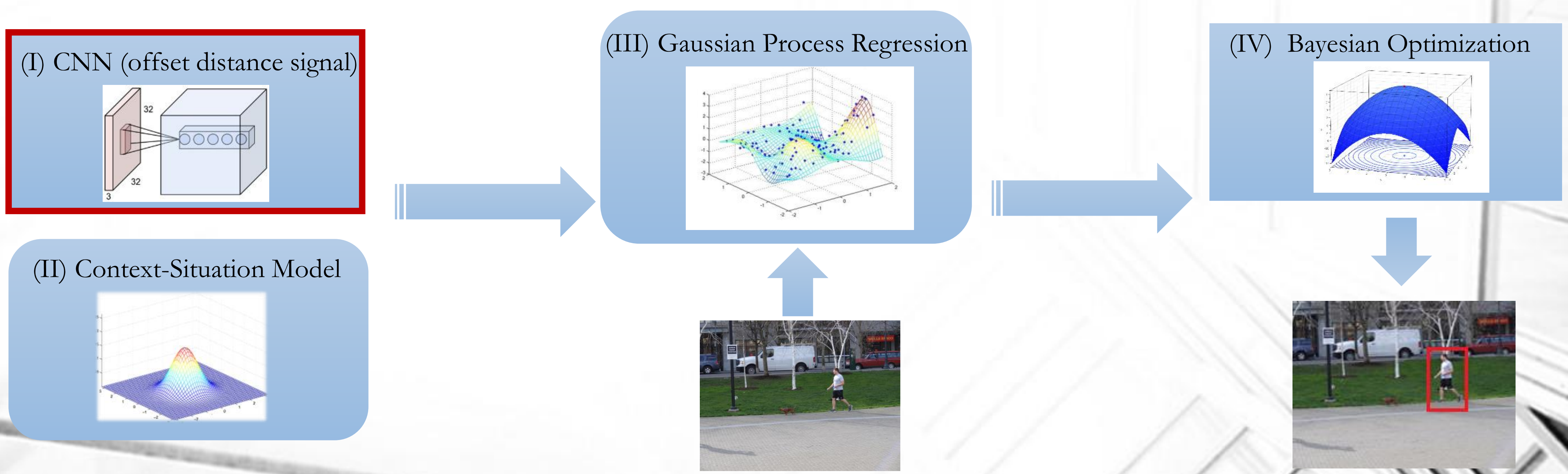
General Algorithm Pipeline

- (I) We train a convolutional neural network (CNN) to score bounding-box proposals to approximate an offset distance from the target object.

- (II) From training data, we develop context-situation model as a distribution of location and size parameters for a target object bounding-box, given various location and size parameters for a particular visual situation.

- (III) We apply a Gaussian Process (GP)to approximate this offset response signal over the (large) search space of the target.

- (IV) A Bayesian active search is then used to achieve fine-grained localization of the target.



(I) CNN (offset distance signal)

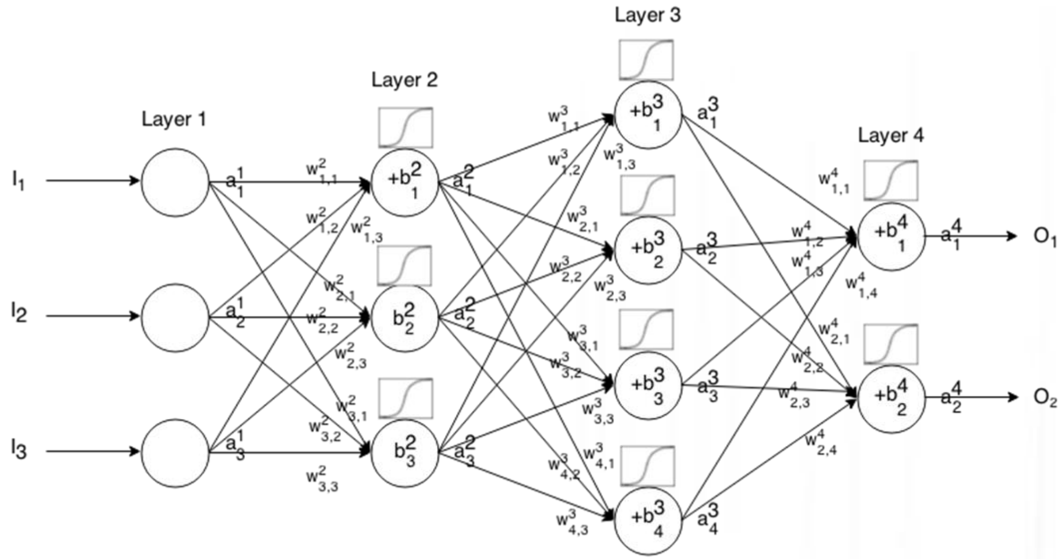
(II) Context-Situation Model

(III) Gaussian Process Regression

(IV) Bayesian Optimization

(I). A Short Digression (CNNs)

- We use a common architecture in machine learning know as a neural network (specifically: a CNN, a convolutional neural network) to extract features from an image patch that we then "score" in terms of the distance from an object of interest in an image.

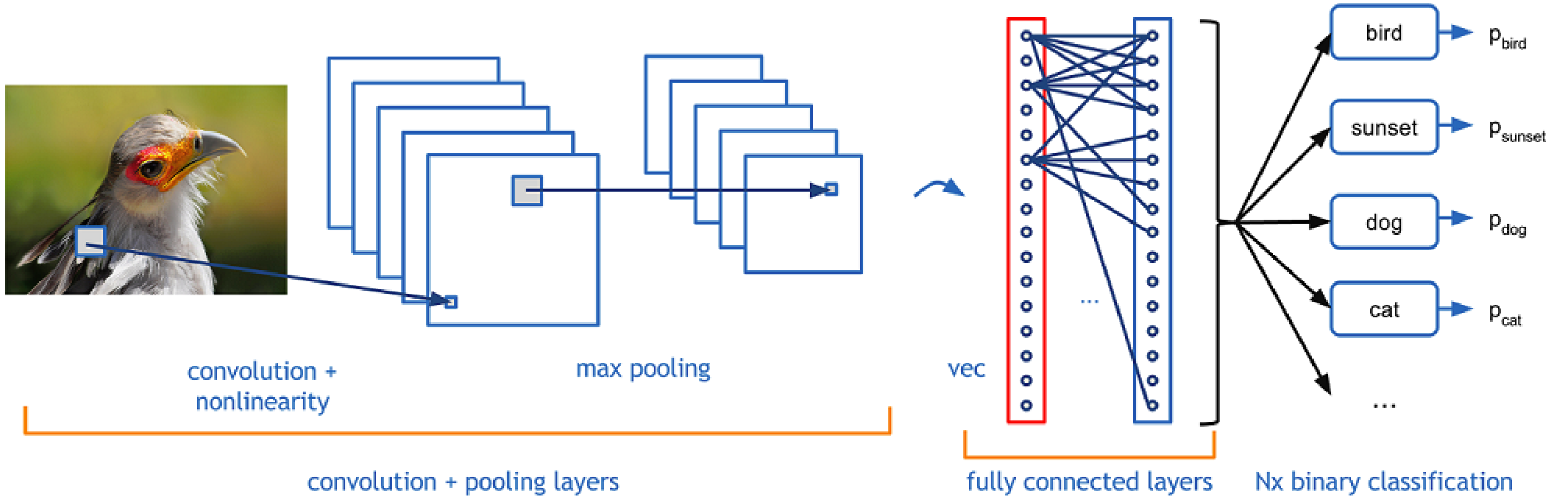


- Each neuron receives some inputs, performs a dot product and optionally follows it with a non-linearity (*e.g.* sigmoid/tanh).

- The whole point of this is that NNs/CCNs are often very effective at (automatically) learning complex patterns in data; NNs can serve as black box universal function approximators.

(I) A Short Digression (CNNs)

- Intuitively, the network will learn filters that activate when they see some type of visual feature such as an edge of some orientation or a blotch of some color. Now, we will have an entire set of filters in each CONV layer, and each of them will produce a separate 2-dimensional activations; these features are stacked along the depth dimension in the CNN and thus produce the output volume.



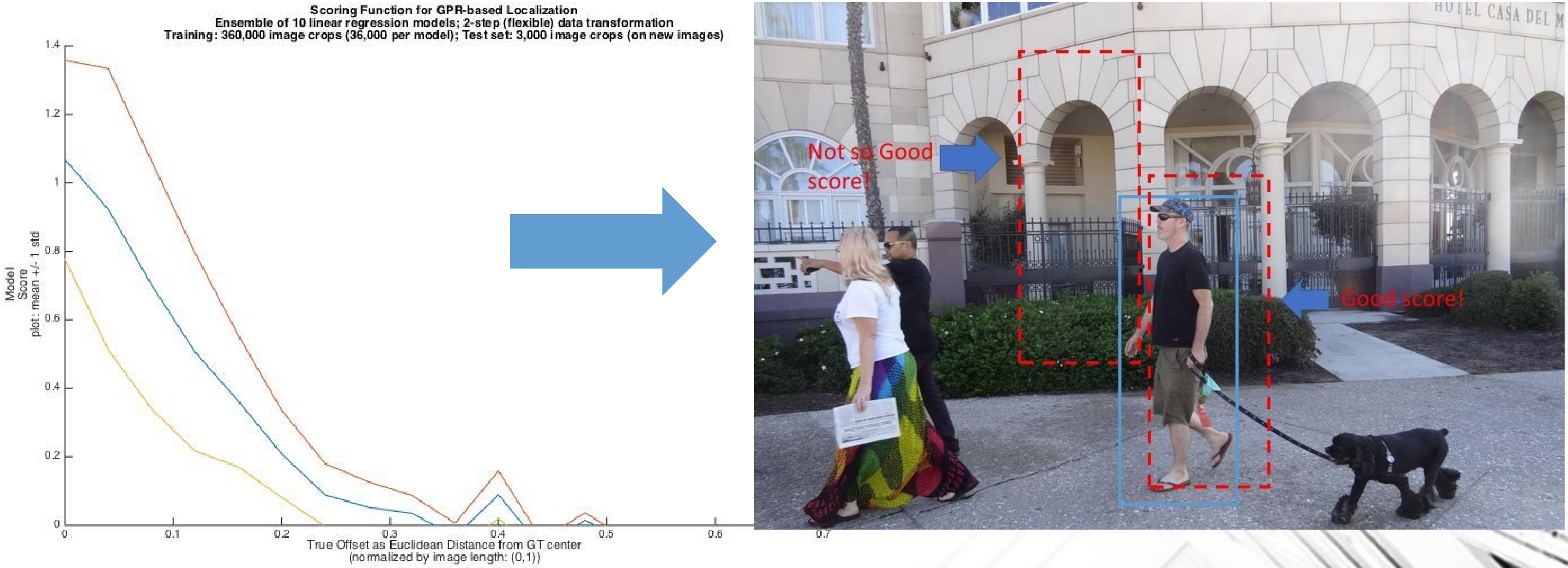
- A nice way to interpret this model via a brain analogy is to consider each entry in the 3D output volume as an <u>output of a neuron that looks at only a small region in the input</u> and <u>**shares parameters**</u> with all neurons to the left and right spatially (since the same filter is used).

- Each neuron is accordingly connected to only a local region of the input volume; the spatial extent of this connectivity is a hyperparameter called the receptive field.

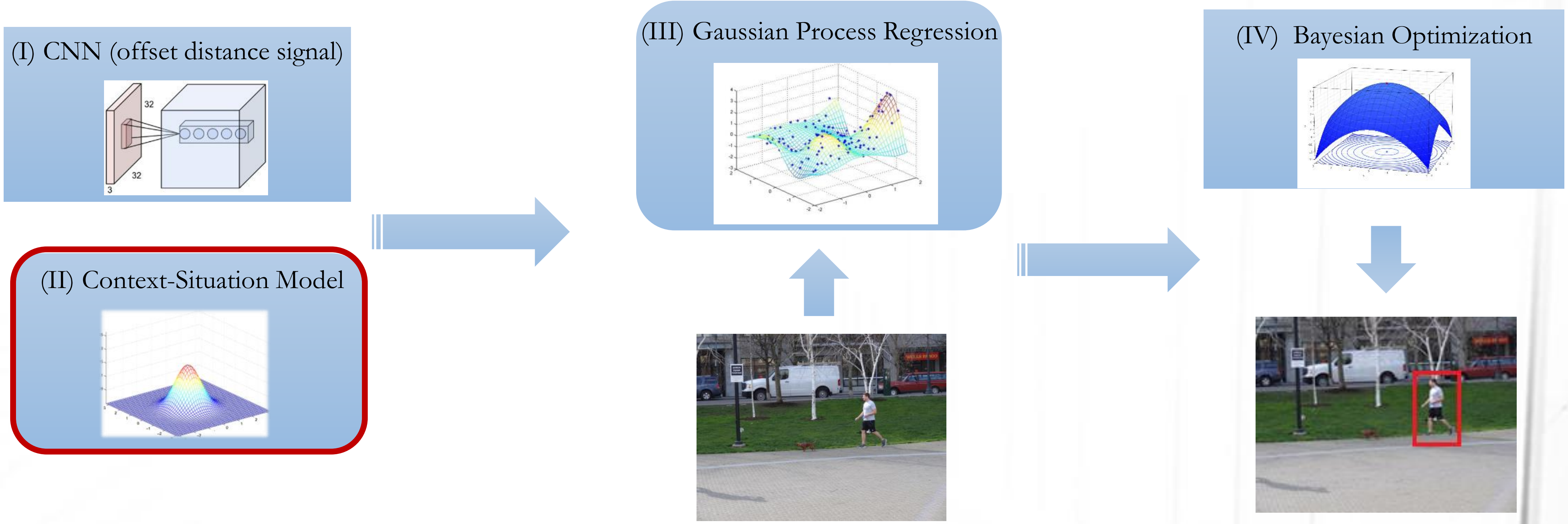
(I). Offset-score Function

- For training to learn the surrogate function, we extracted features from a pre-trained CNN on ~10^6 image patches per object category (e.g. human, dog), with 4096 features per patch, and performed ridge regression fitting and a multi-step data transformation to generate a workable function for object localization with GPR.

- An ensemble of these classifiers constitutes our scoring function.



Recall: Algorithm Pipeline



(I) CNN (offset distance signal)

(II) Context-Situation Model

(III) Gaussian Process Regression

(IV) Bayesian Optimization

(II). Context-Situation Model

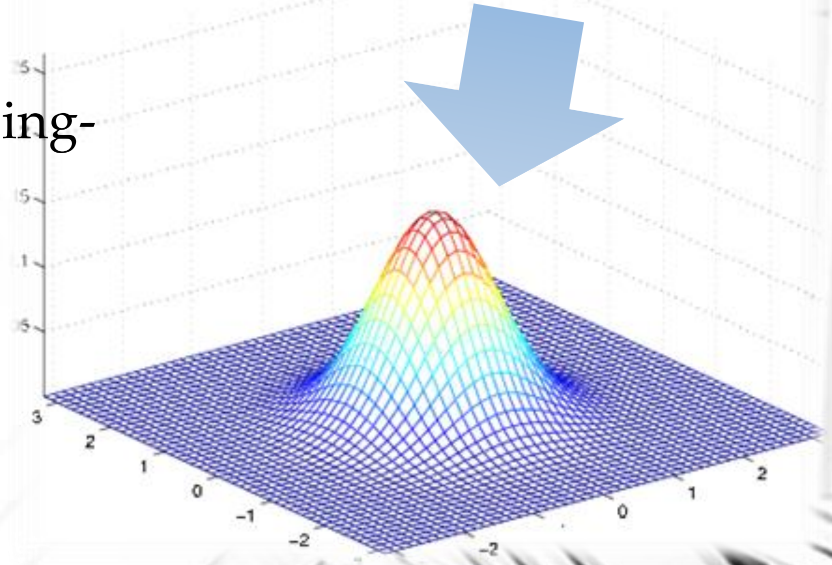
- We define a *context-situation model* as a distribution of location and size parameters for a target object bounding-box, given various location and size parameters for a particular visual situation: $p\left(x_{target}, s_{target} \middle| \{x_{context}, s_{context}\}_{1:C}\right)$.

- More specifically, this learned model consists of a set of probability distributions modeling the joint locations and sizes of three relevant objects (i.e., pedestrian, dog and leash).

- These distributions capture the expected relationships among the three objects with respect to location and size/shape of bounding-boxes.

- For simplicity and as a general proof of concept, we model context-situation MVN (multi-variate Normal) distributions.

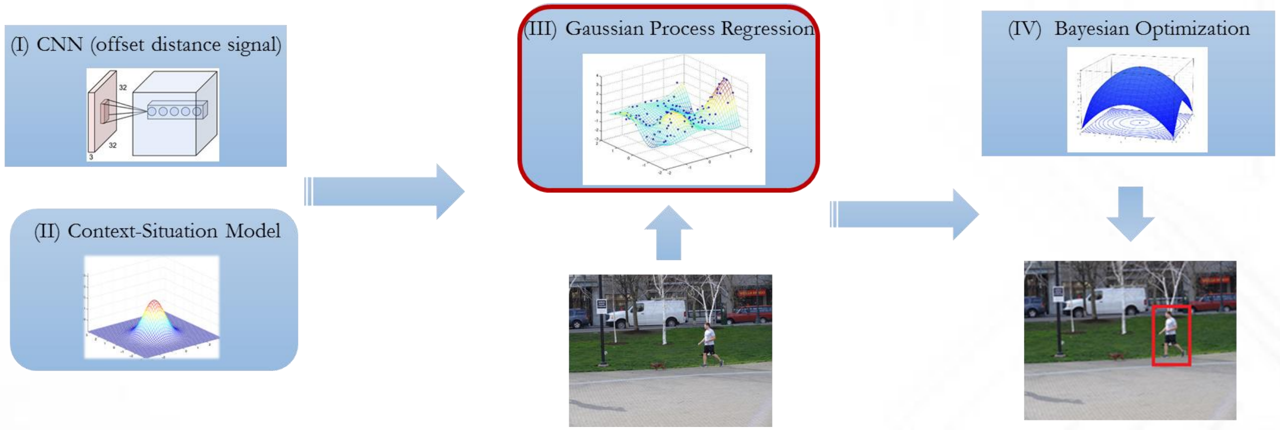
- We use the context-situation model to generate initial bounding-box proposals for a pedestrian in a test image.



*In prior work we additionally developed more flexible context-situation models using kernel methods.

(III). Gaussian Process Regression

- Gaussian Process Regression (GPR) is a flexible (Bayesian) regression scheme that defines a probability density over function output values, according to previously observed data.
- A Gaussian Process is uniquely defined by the choice of its *mean* and *covariance* functions, where the mean indicates the average function output and the covariance typically measures “similarity” between data values.
- GPR method is consequently *data-driven* (*i.e*. non-parametric) method offering several distinct advantages over traditional regression/interpolation approaches.


(I) CNN (offset distance signal)

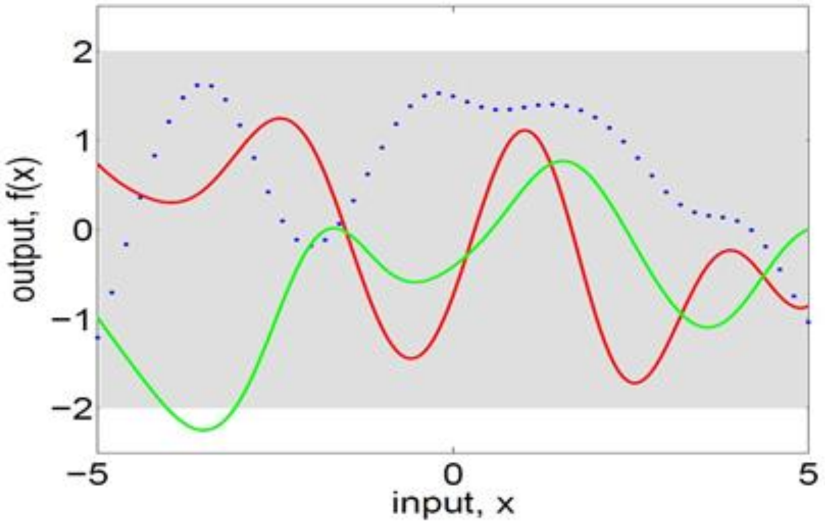
(II) Context-Situation Model

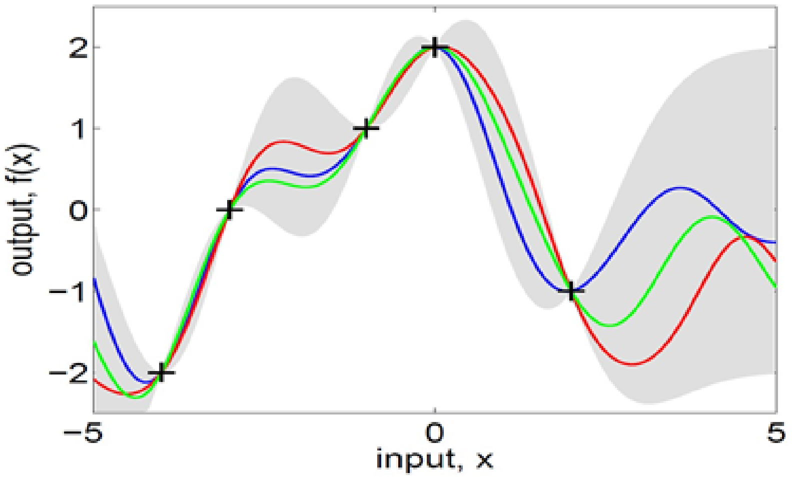
(III) Gaussian Process Regression

(IV) Bayesian Optimization

(III). GPR

- One can employ GPR in this framework as a probabilistic, *generative* function model.



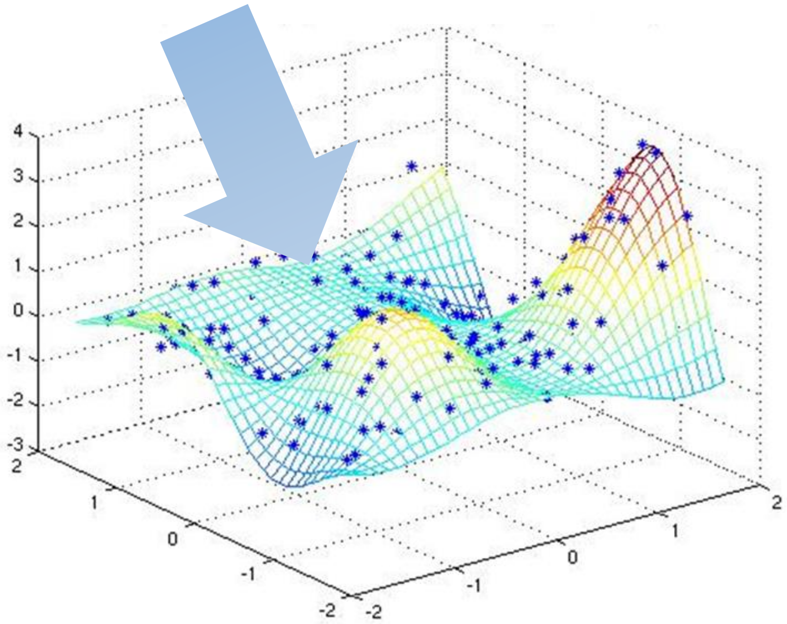


- The figure on the left shows samples drawn from a GPR prior (*i.e*. without data); the image on the right depicts samples taken from the posterior distribution; note the instance of perfect interpolation achieved due to noise-free modeling.

*Image credit: Rasmussen & Williams (2006)

(III). GPR

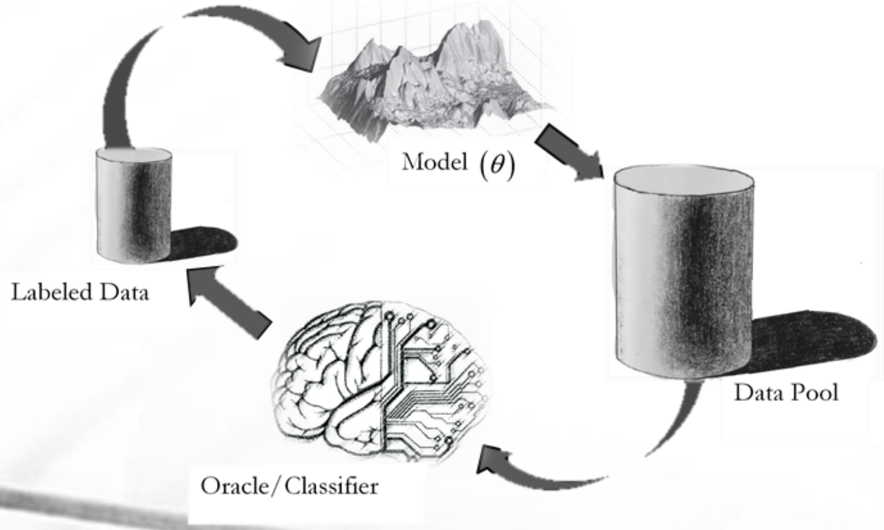
- Because it is computationally expensive to generate offset prediction values for a large number of bounding-box proposals (due to the CNN), we use GPR to approximate the offset prediction values over the *target search space* (*i.e.* a large grid of values).

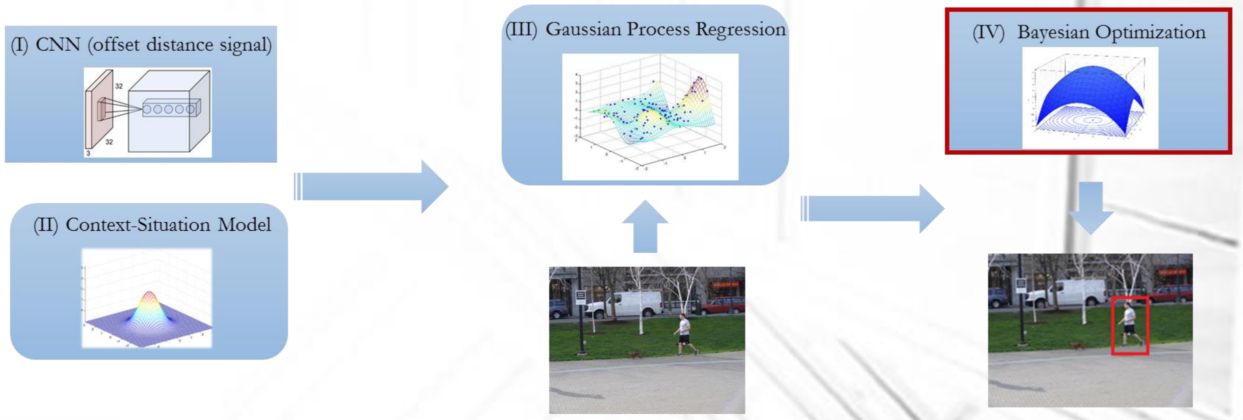


- Next, we actively search this space according to a Bayesian optimization scheme (IV) to find new proposals that are likely to capture the target object.

(IV). Active Learning Queries

- Next we consider the task of efficiently querying the search space for a target object generated by the GPR procedure.
- In choosing new data points, we naturally want to collect the "best" data available at a minimum computational cost.
- Why is this problem challenging? Because our search space is only an *approximation* to the true measure of the quality of a bounding-box proposals.
- The main undertaking in active learning is to make a decision as to which data points to query (formulating a query strategy); this decision is encapsulated formally through an *acquisition function*.





(I) CNN (offset distance signal)
(II) Context-Situation Model
(III) Gaussian Process Regression
(IV) Bayesian Optimization

(IV). Active Learning Queries

- Next we consider the task of efficiently querying the search space for a target object generated by the GPR procedure.

- In choosing new data points, we naturally want to collect the “best” data available at a minimum computational cost.

- Why is this problem challenging? Because our search space is only an *approximation* to the true measure of the quality of a bounding-box proposals.

- The main undertaking in active learning is to make a decision as to which data points to query (formulating a query strategy); this decision is encapsulated formally through an *acquisition function*.

- Ideally, in addition to *exploring* regions of high uncertainty, we should also *exploit*, to some degree, “regions of promise”, respecting our target object.

(IV). Active Learning Queries

- In the framework of Bayesian optimization, ***acquisition functions*** are used to guide the search for the optimum of the GPR approximation to the true objective function (whose maximum occurs, ideally, for a proposal that perfectly crops the pedestrian).

- Intuitively, acquisition functions are defined in such a way that *high acquisition* indicates greater likelihood of an objective function maximum.

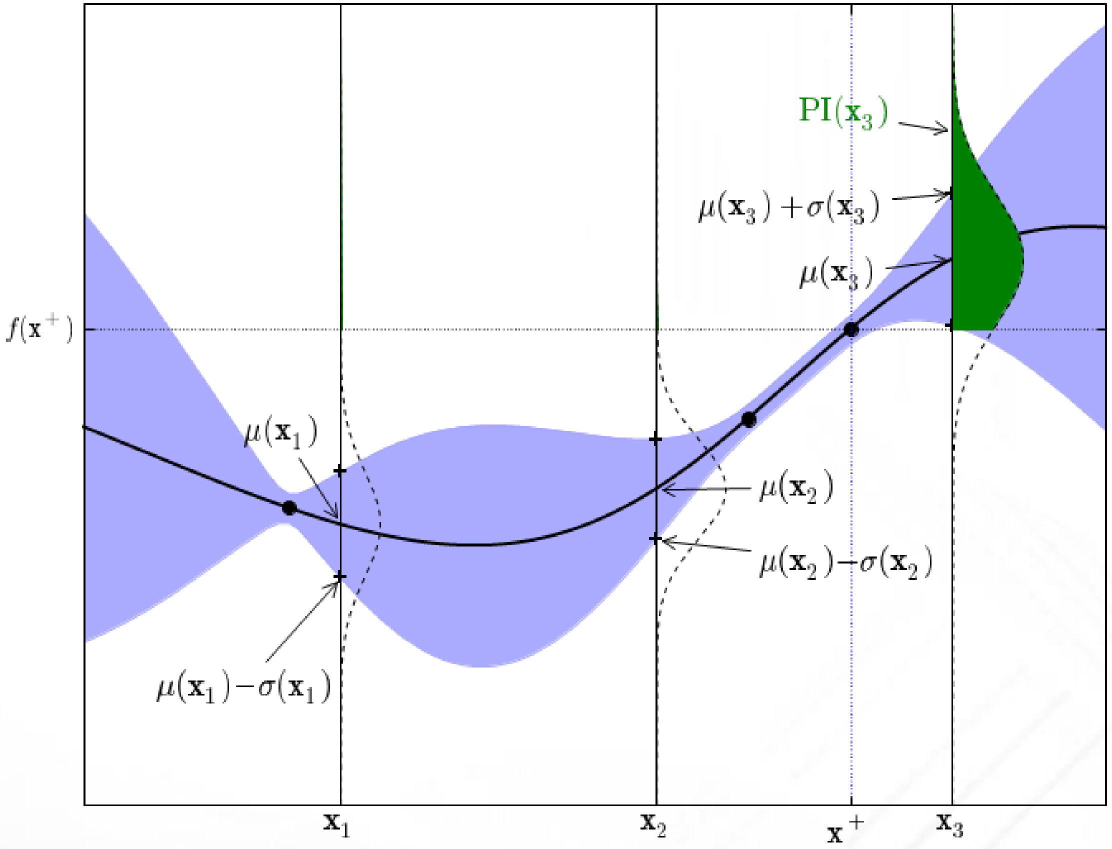
- Commonly used acquisition functions (we omit the details for brevity) in this setting include: *probable improvement* (PI) and *expected improvement* (EI).

(IV). Bayesian Optimization

- In the figure we display a Gaussian process showing the region of *probable improvement*. The maximum observation is at $\mathbf{x}^+$.

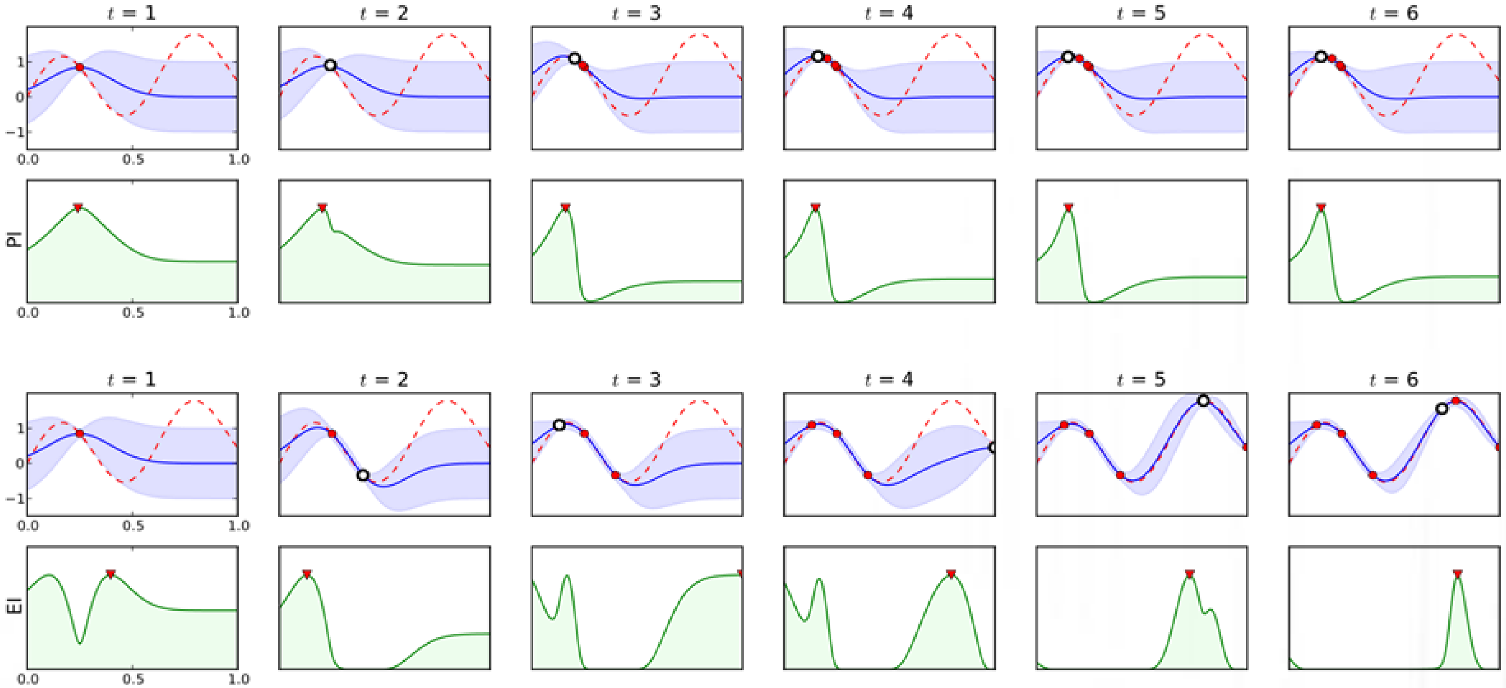
- The darkly-shaded area in the superimposed Gaussian above the dashed line can be used as a measure of improvement. The model predicts almost no possibility of improvement by observing at $\mathbf{x}_1$ or $\mathbf{x}_2$, while sampling at $\mathbf{x}_3$ is more likely to improve on $f(\mathbf{x}^+)$.

-



(IV). Bayesian Optimization

- Below are example iterations of both *PI* and *EI*-based active queries with GPR.



- In the current work, we use a variant of *EI* that is fine-tuned to our problem parameters.

GP-CL Algorithm

Algorithm: Gaussian Process Context Localization (GP-CL)

Input: Image *I*, a set of *C* context objects, trained model *y* giving response signals, learned context-situation model $p(x_{target}, s_{target} \mid \cdot)$, *n₀* initial bounding-box proposals for target object generated by the context-situation model, and corresponding response signal values: $D_{n_0} = \{(x_i, s_i), y(x_i, s_i)\}_{i=1}^{n_0}$, GP hyperparameters θ, size of GP realization space *M*, dynamic design parameter for Bayesian active search ξ, size of GP memory GP_{mem} (as number of generations used), batch size *n*, number of iterations *T*, current set of bounding-box proposals and response signals $D_{proposal}^{(t)}$.

```
Compute $n_0$ initial bounding box proposals:
$\{(x_i, s_i)\}_{i=1}^{n_0} \sim p(x_{target}, s_{target} \mid \cdot)$
$D_{proposal}^{(0)} \longleftarrow D_{n_0}$
for $t = 1$ to $T$ do
    Compute $\mu(x)^{(t)}$ and $\sigma(x)^{(t)}$ for the GP realization
        $f_M^{(t)}$ of $D_{proposal}^{(t-1)}$ over grid of $M$ points (Equation 4)
    for $i = 1$ to $n$ do
        $z_i = \underset{x}{\operatorname{argmax}}\, a_{CEI}\left(f_M^{(t)} \setminus \{z_j\}_{j=1}^{j=i-1}, \xi\right)$ (Equation 5)
        $sample: s_i \sim p(\cdot)_s$
        $p_i = (z_i, s_i)$
    end for
    $D^{(t)} \longleftarrow \{(x_i, s_i), y(x_i, s_i)\}_{i=1}^{n}$
    $D_{propos\ al}^{(t)} \longleftarrow \cup_{j=t-GP_{mem}}^{t} D^{(j)}$
end for
Return $\underset{x}{\operatorname{argmax}}\, \mu(x)^{(T)}$
```

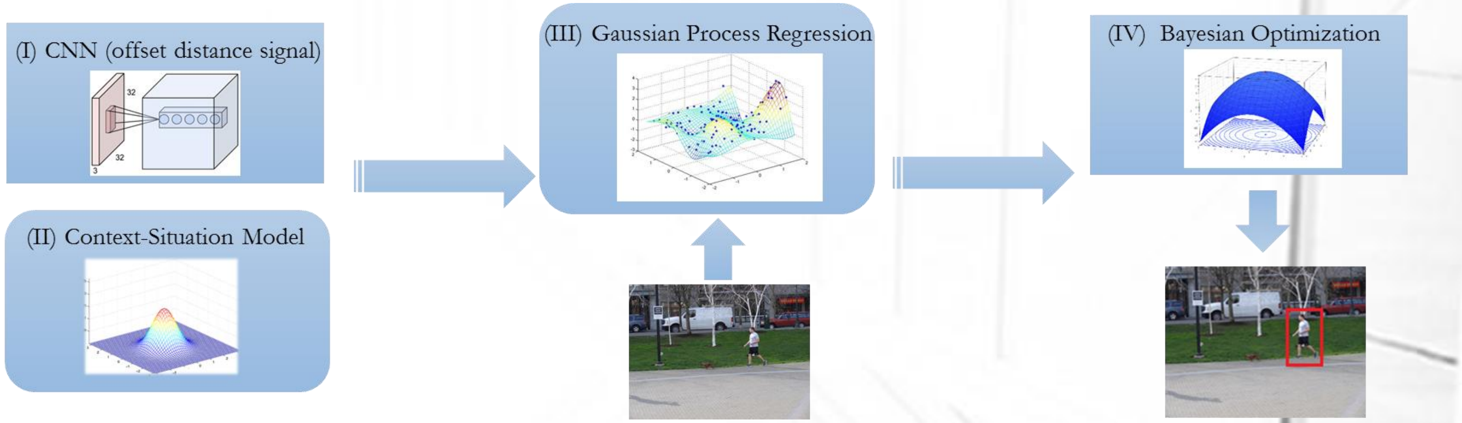
Step 1: Sample initial target proposals from context-situation model

Step 2: Score these proposals using the offset-prediction model (CNN)

Step 3: Compute GPR values over search space

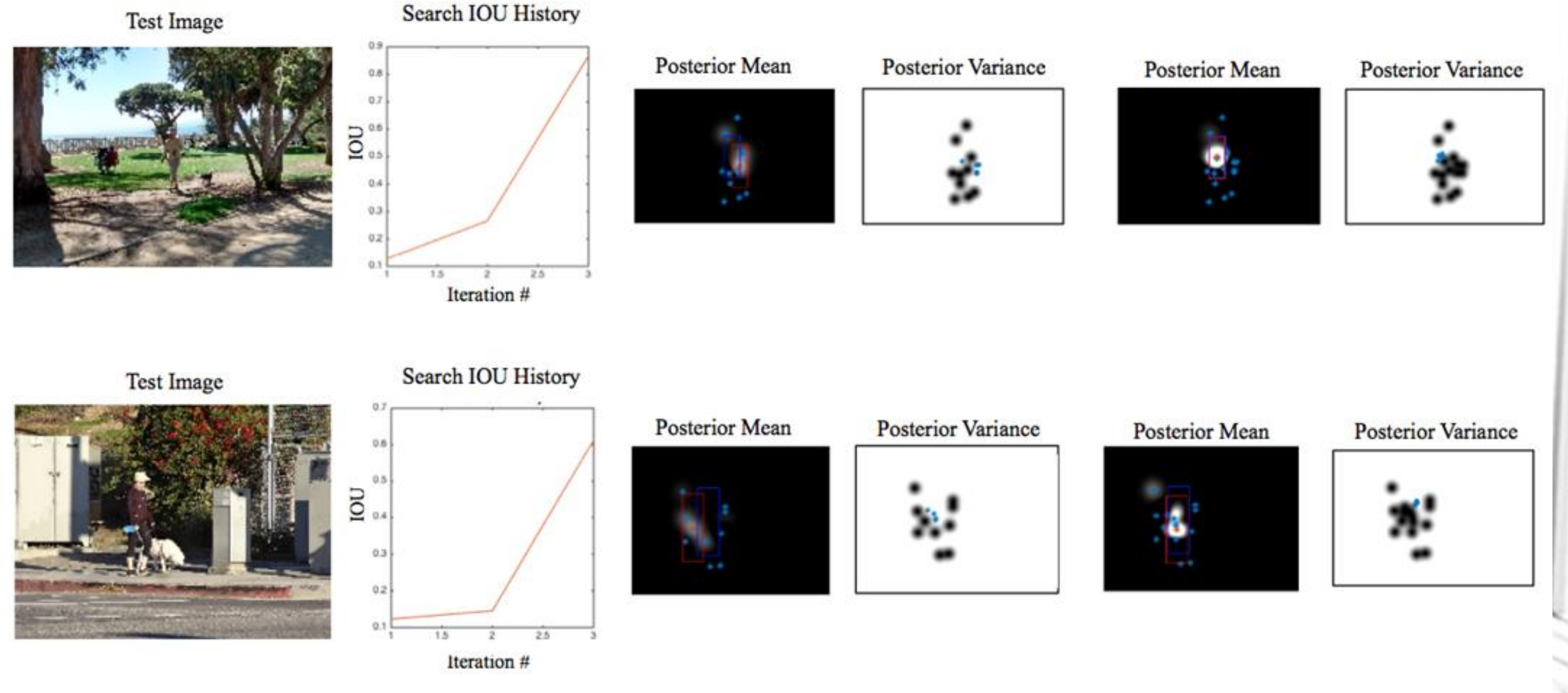
Step 4: Using Bayesian optimization procedure, return proposals in search space with maximum acquisition

Step 5: **Return** to Step 3 (loop)


(I) CNN (offset distance signal)
32
32
3
(II) Context-Situation Model
(III) Gaussian Process Regression
(IV) Bayesian Optimization

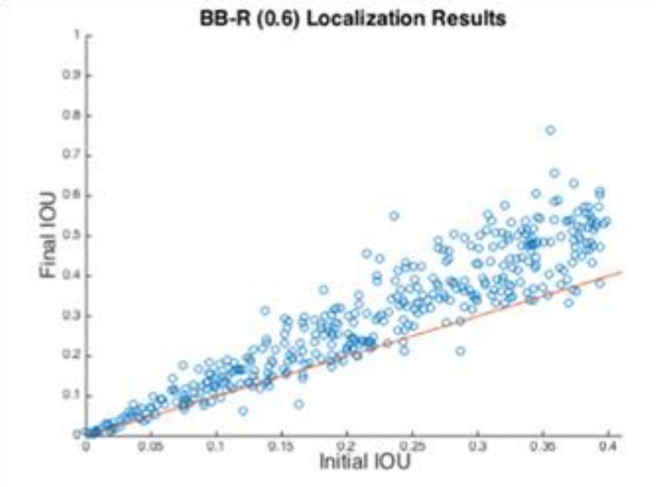
GP-CL Example Runs

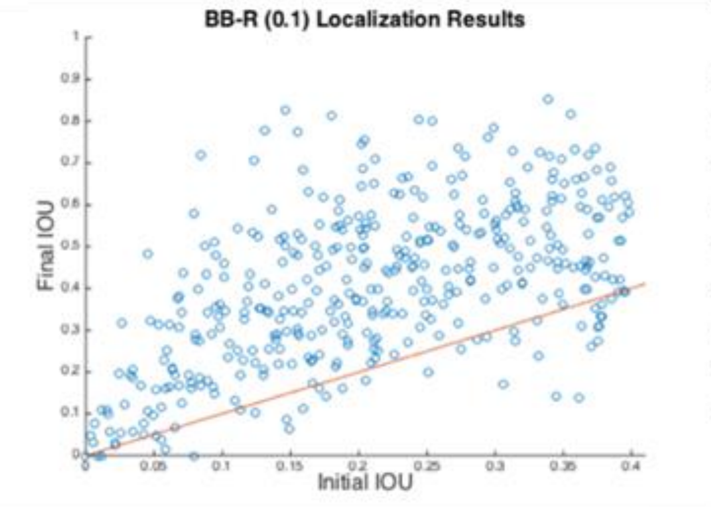
- Examples of runs on two test images with the GP-CL algorithm. In each row the test image is shown on the far-left; the “search IOU history” is displayed in the second column, with the algorithm iteration number on the horizontal axis and IOU with the ground-truth target bounding box on the vertical axis.

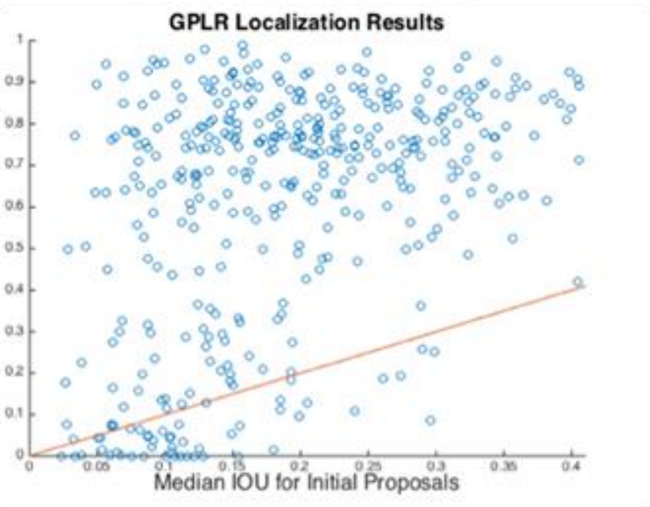


Experimental Results

- Graph of BB-R (0.6), BB-R (0.1) and GP-CL localization results for test images. The horizontal axis indicates the median IOU for the initial proposal bounding boxes, while the vertical axis designates the final IOU with the target object ground truth. The line depicted indicates "break-even" results.


BB-R (0.6) Localization Results


BB-R (0.1) Localization Results


GPLR Localization Results

Method	IOU Difference Median (SE)	Median Relative IOU Improvement	% of Test Set with IOU Improvement	% of Test Set Localized
BB-R (0.6)	.0614 (.0035)	34.62%	**90.1%**	12.3%
BB-R (0.1)	.1866 (.0077)	92.91%	90.0%	33.2%
GP-CL	**.4742** (.012)	**194.02%**	89.3%	**75.2%**

Thanks for listening, questions and comments are welcome.