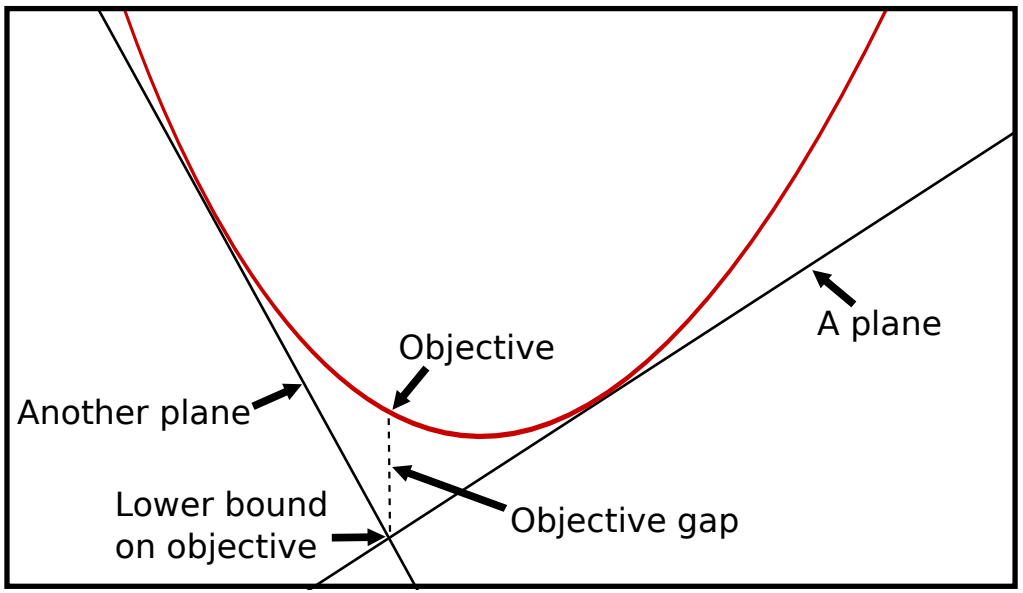
What does all this stuff mean?

The HOG trainer prints logs like this:

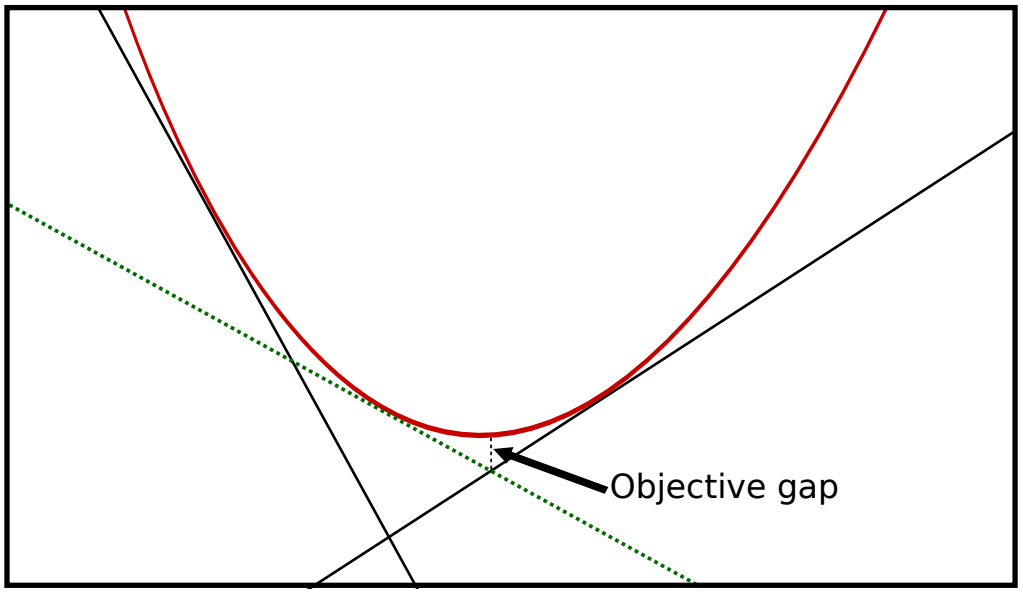
```
objective:                  8.5
objective gap:              0.4
risk:                       0.224
num planes:                 2
iter:                       12
```



What does all this stuff mean?

The HOG trainer prints logs like this:

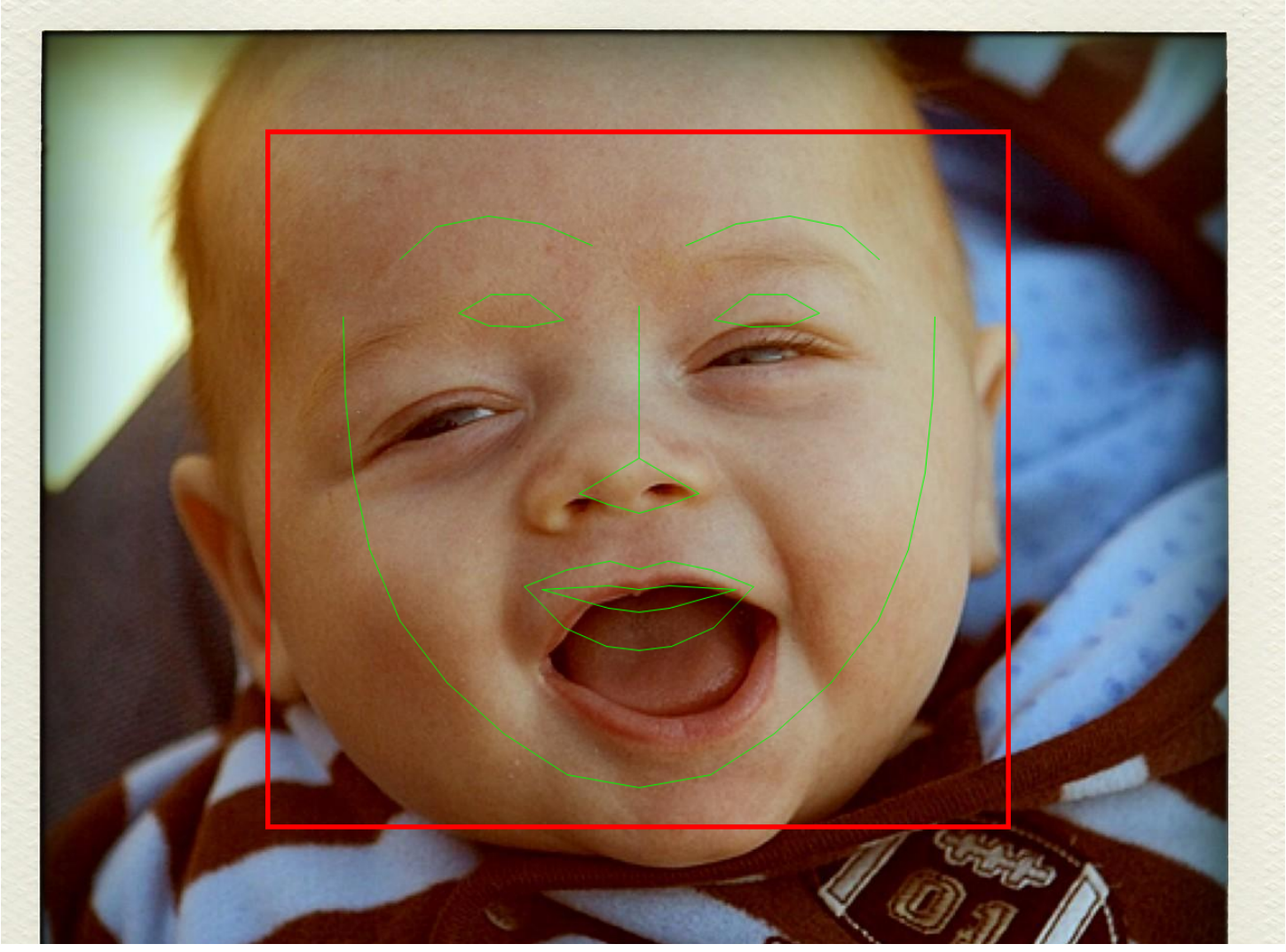
```
objective:                  8.3
objective gap:              0.1
risk:                       0.113
num planes:                 3
iter:                       13
```



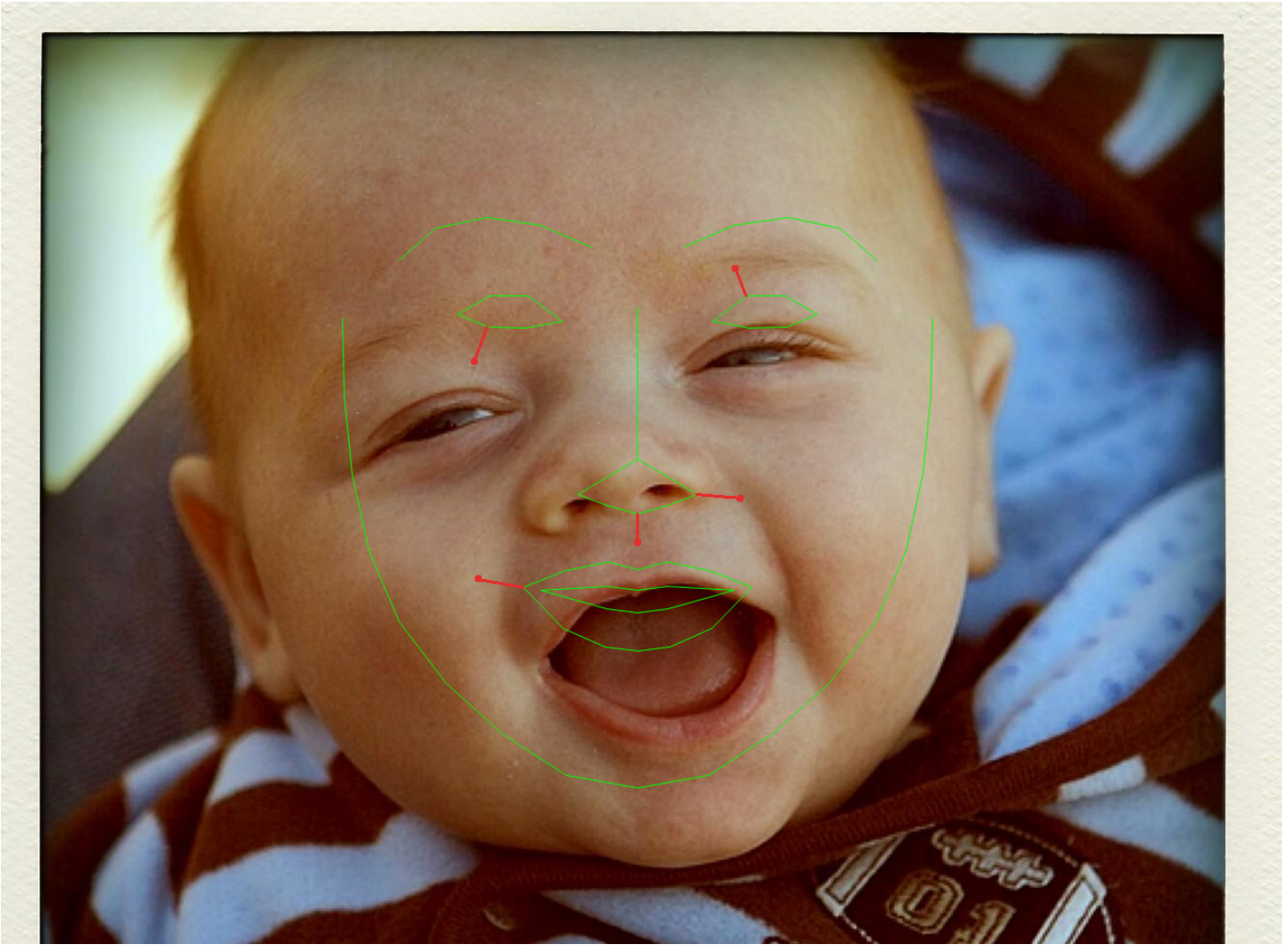
Object Pose Estimation

One Millisecond Face Alignment with an Ensemble of Regression Trees by Vahid Kazemi and Josephine Sullivan

Object Pose Estimation - Seeding



Relative Pixel Encoding



Object Pose Estimation - 1

Object Pose Estimation - 2

Object Pose Estimation - 3

Object Pose Estimation - 4

Object Pose Estimation - 5

Object Pose Estimation - 6

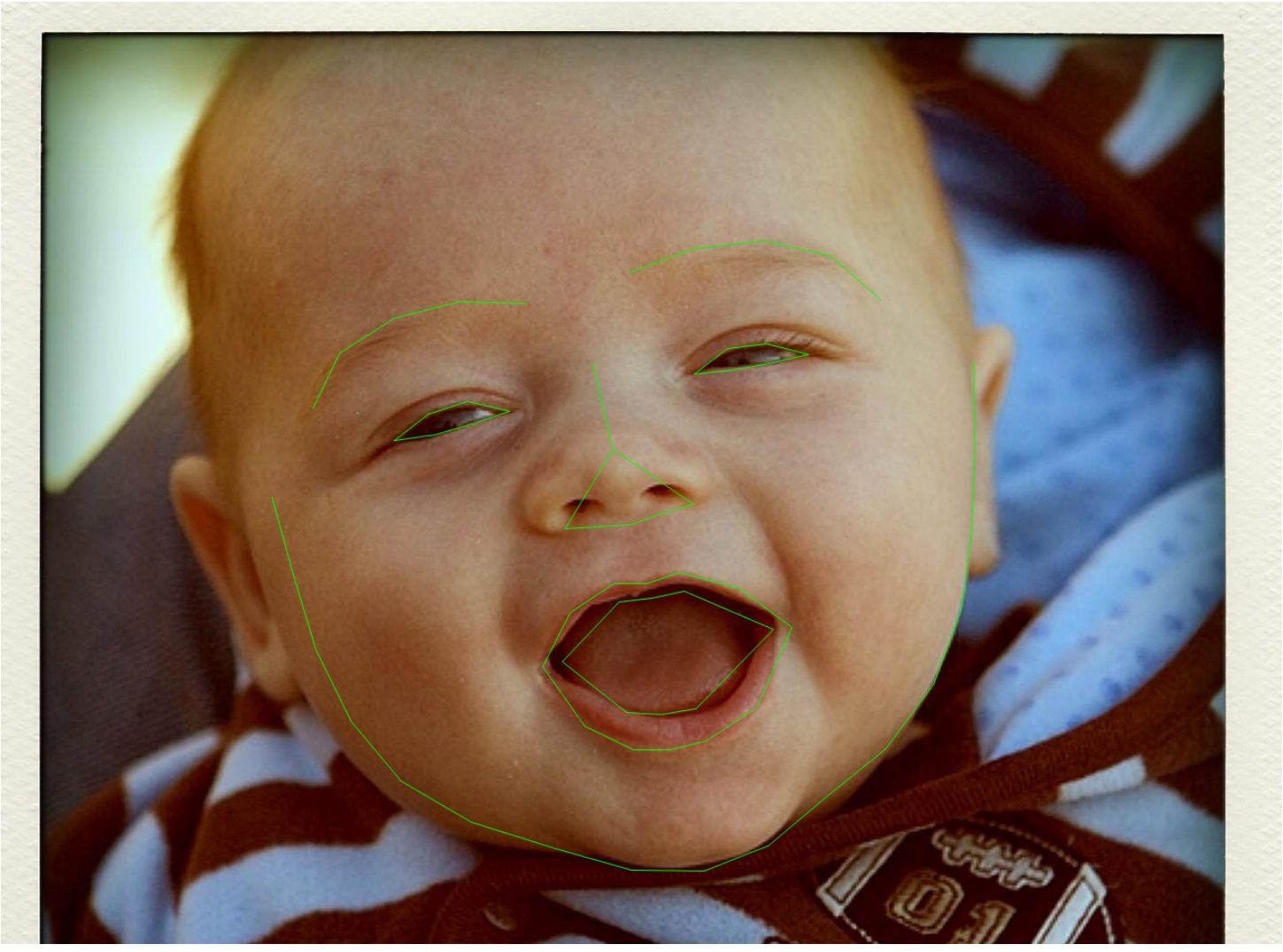
Object Pose Estimation - 7

Object Pose Estimation - 8

Object Pose Estimation - 9

Object Pose Estimation - 10

Object Pose Estimation - 11



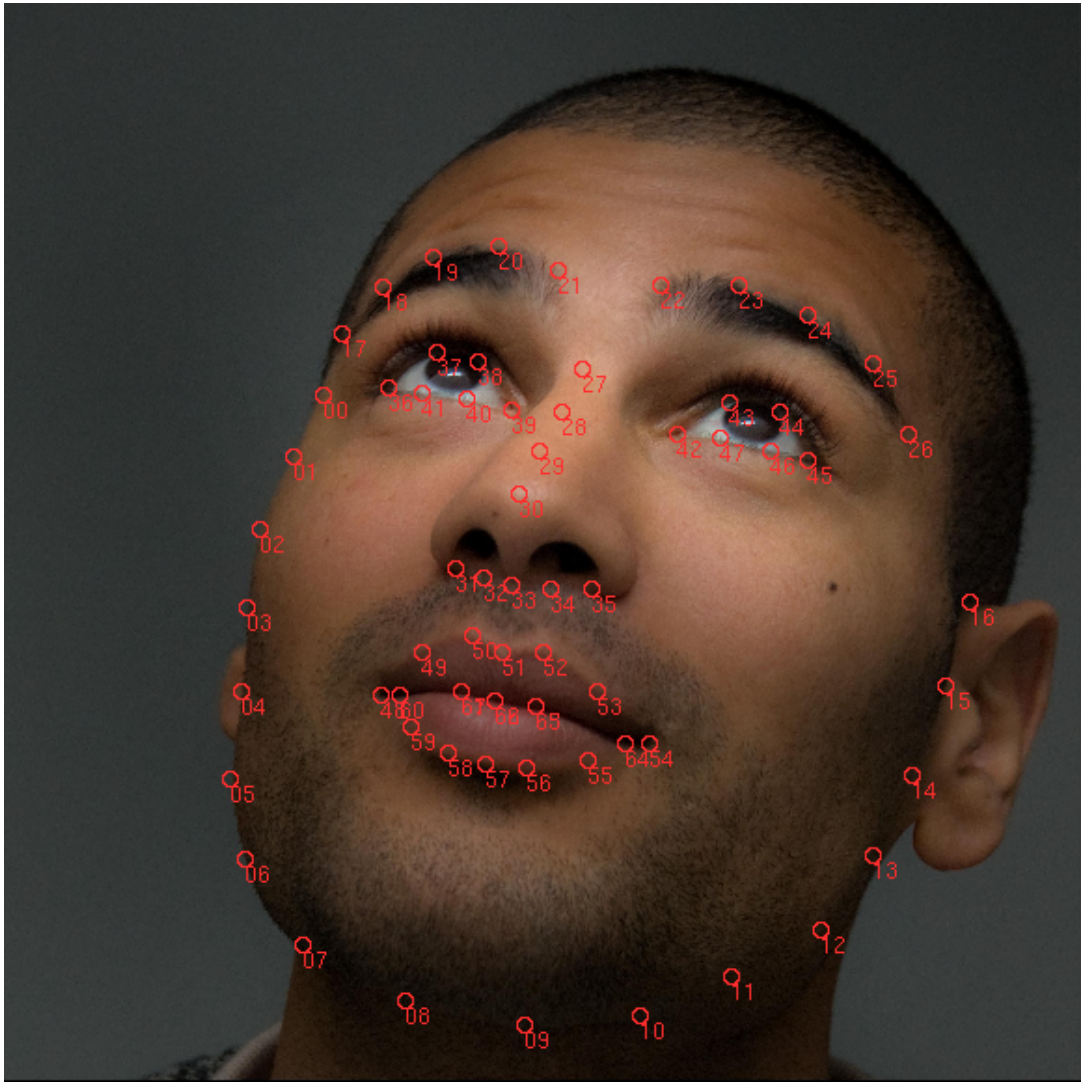
Object Pose Estimation - 12

Object Pose Estimation - 13

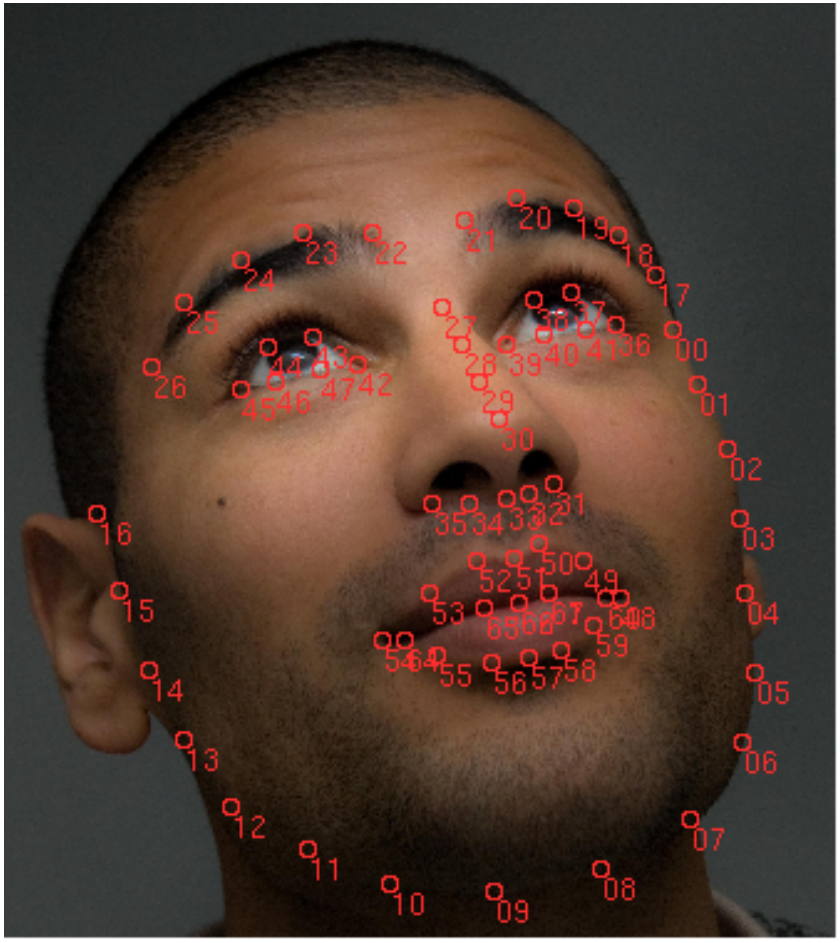
Object Pose Estimation - 14

Object Pose Estimation - 15

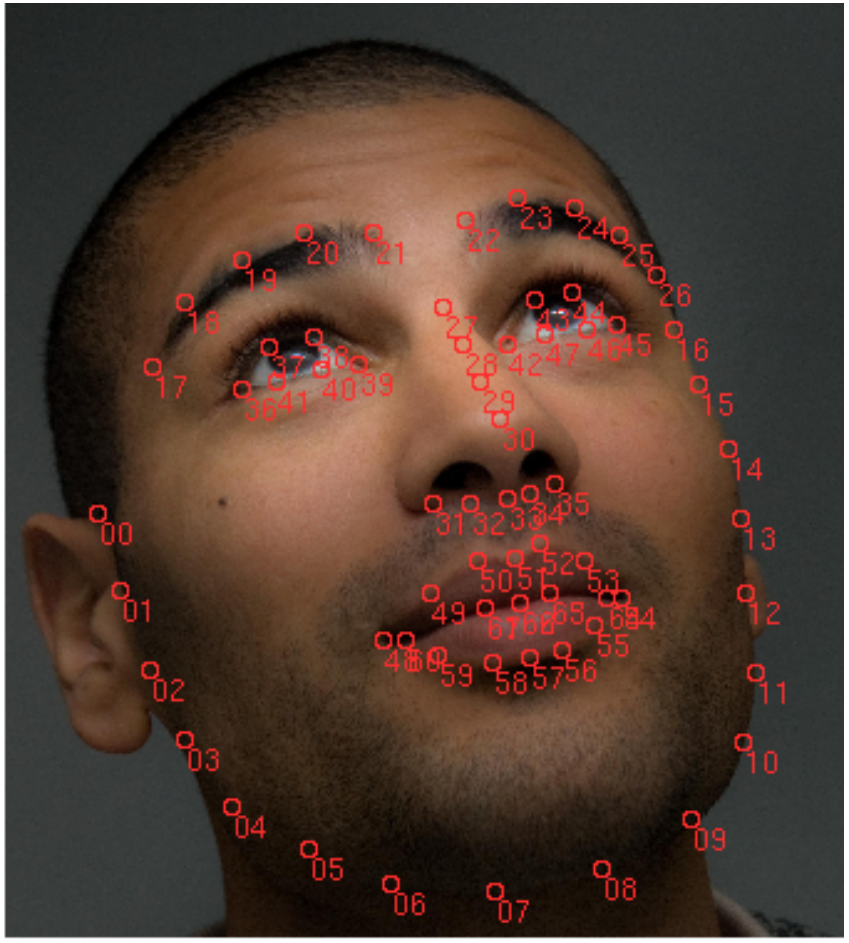
Correctly Mirroring Landmarks



Correctly Mirroring Landmarks



Simple Mirroring



Source Label Matching

Why isn't my detector working?

- You just have a bug in your code

Why isn't my detector working?

- You just have a bug in your code
- Your training data is bad

Why isn't my detector working?

- You just have a bug in your code
- Your training data is bad
- The algorithm just can't do what you are asking it to do.
  - Maybe the objects aren't included in the locations the detector looks. E.g. maybe you are using a sliding window+pyramid that doesn't look at objects as small as your targets.

Why isn't my detector working?

- You just have a bug in your code
- Your training data is bad
- The algorithm just can't do what you are asking it to do.
  - Maybe the objects aren't included in the locations the detector looks. E.g. maybe you are using a sliding window+pyramid that doesn't look at objects as small as your targets.
- What do you do?

Why isn't my detector working?

- You just have a bug in your code
- Your training data is bad
- The algorithm just can't do what you are asking it to do.
  - Maybe the objects aren't included in the locations the detector looks. E.g. maybe you are using a sliding window+pyramid that doesn't look at objects as small as your targets.
- What do you do?
  - Get it working on a small dataset. *Make sure the detector works on the images it was trained on.*

Why isn't my detector working?

- You just have a bug in your code
- Your training data is bad
- The algorithm just can't do what you are asking it to do.
  - Maybe the objects aren't included in the locations the detector looks. E.g. maybe you are using a sliding window+pyramid that doesn't look at objects as small as your targets.
- What do you do?
  - Get it working on a small dataset. *Make sure the detector works on the images it was trained on.*
  - If it works on training data but not on testing data **get more training data!**

Why isn't my detector working?

- You just have a bug in your code
- Your training data is bad
- The algorithm just can't do what you are asking it to do.
  - Maybe the objects aren't included in the locations the detector looks. E.g. maybe you are using a sliding window+pyramid that doesn't look at objects as small as your targets.
- What do you do?
  - Get it working on a small dataset. *Make sure the detector works on the images it was trained on.*
  - If it works on training data but not on testing data **get more training data!**
- It is working but your metric is bad. Always look at the output with your eyes and see what's really going on.

The danger of chasing metrics

- **Do not just look at a single evaluation metric**
- Always visually inspect the outputs to make sure things are working the way you expect.
  - Maybe you aren't calculating the metric correctly
  - The metric might just be inappropriate for your problem
  - There might be weird dataset bias



The danger of chasing metrics

From the wonderful "YOLOv3: An Incremental Improvement" by Joseph Redmon, Ali Farhadi:

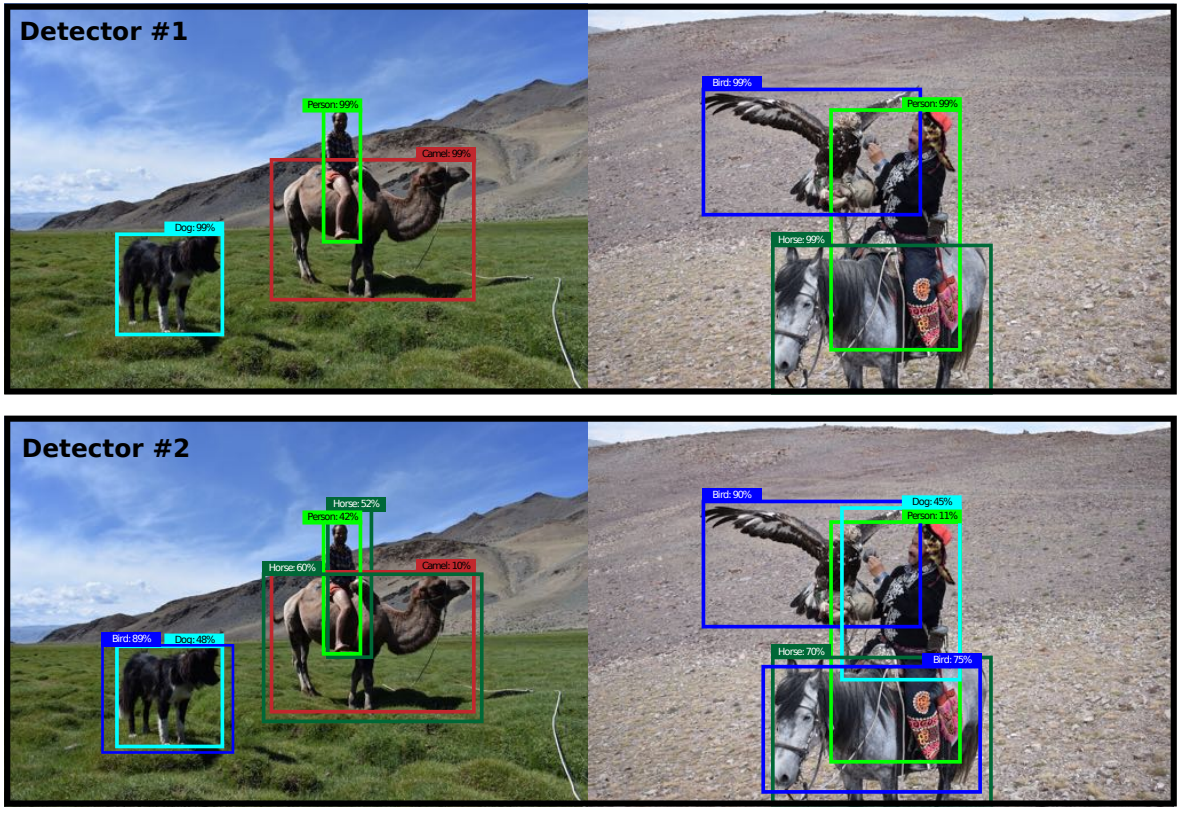


Figure 5. These two hypothetical detectors are perfect according to mAP over these two images. They are both perfect. Totally equal.

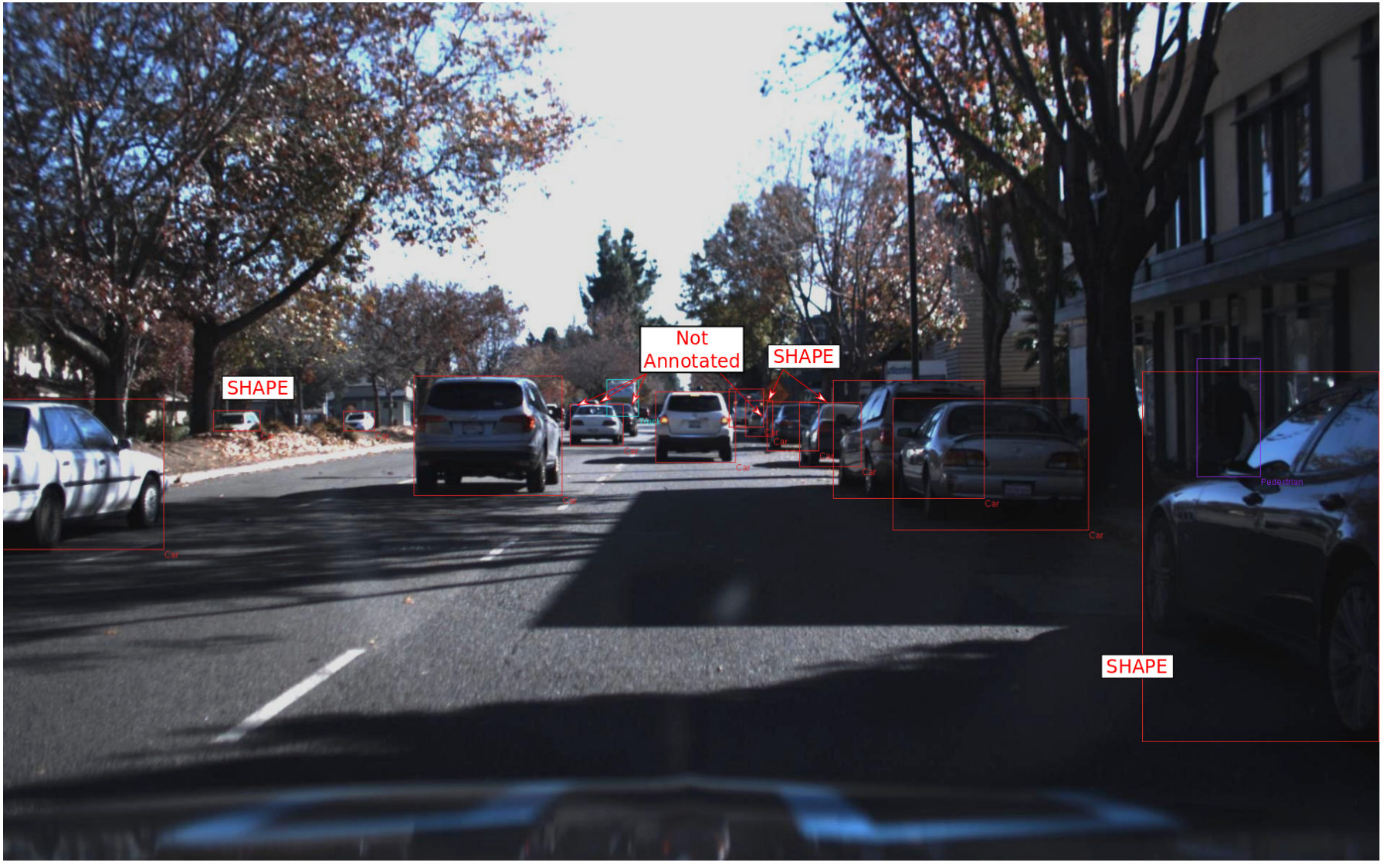
Always visually inspect the outputs, like seriously, always

- Datasets can have errors in them or use different annotation styles. e.g. FDDB many unannotated faces and boxes centered between the eyes.
- "False alarms" on FDDB:

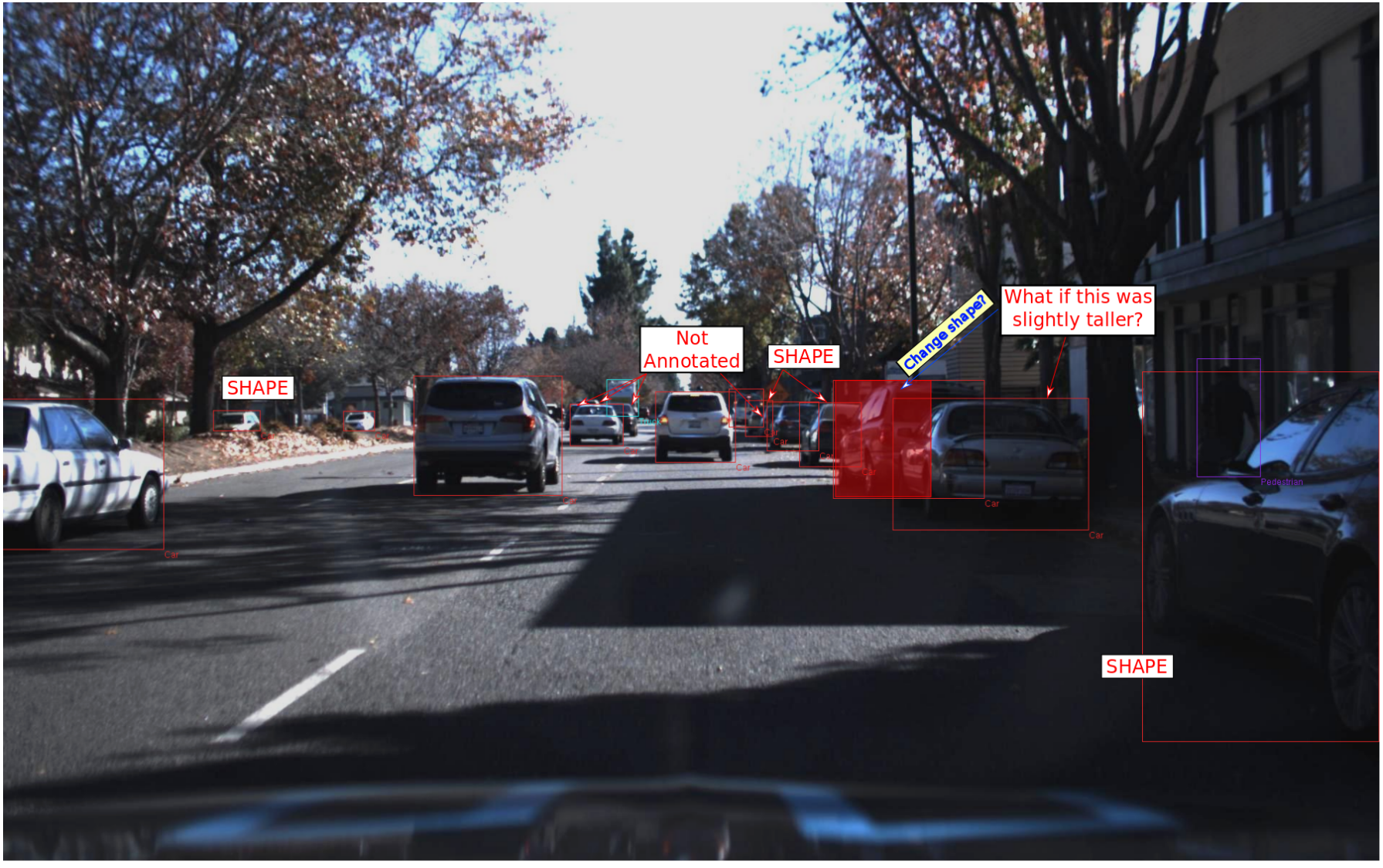
What makes a good dataset?

- The training data must look like the test data
- The training data should cover all the cases you care about
- You want at least several thousand training images
- The annotations (e.g. box positions) must be complete, accurate, and consistent
- Don't label things for detection if you know your algorithm can't possibly detect them
- Example, Udacity dataset. I trained the CNN+MMOD detector on it and it didn't want to find any cars. But then you look at the dataset and it's a mess. The lowest error model was the one that doesn't detect anything.

More Bad Training Data

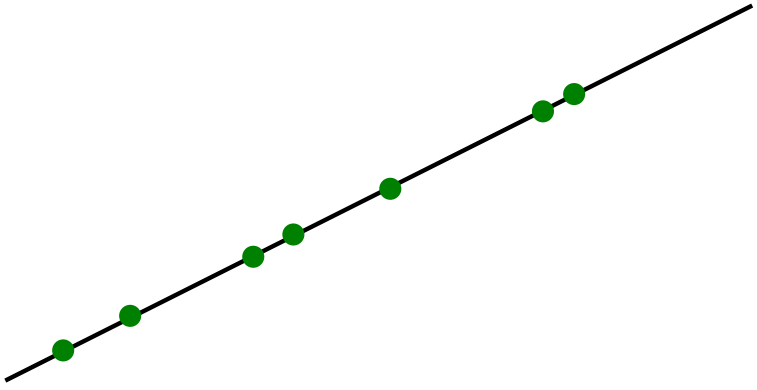


More Bad Training Data

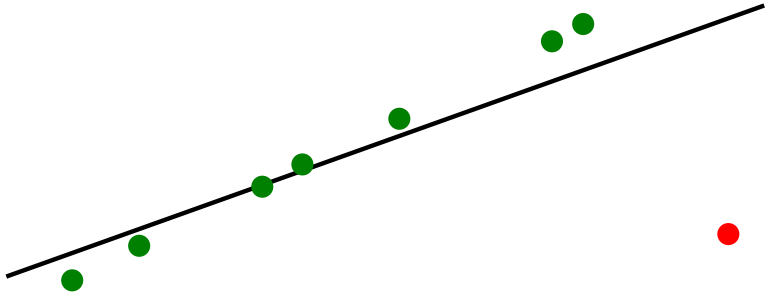


More data is not the same as better data

- Adding more training data isn't going to help if it's low quality.
- There are often really hard cases you don't care about. Exclude them from the dataset.



Great Fit



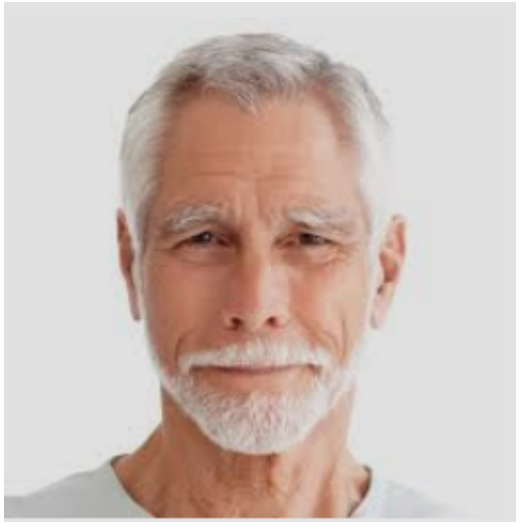
Bad Fit

More Bad Training Data

Suppose you want to train a face detector to find faces in images like this:

More Bad Training Data

Then this is horrible training data:

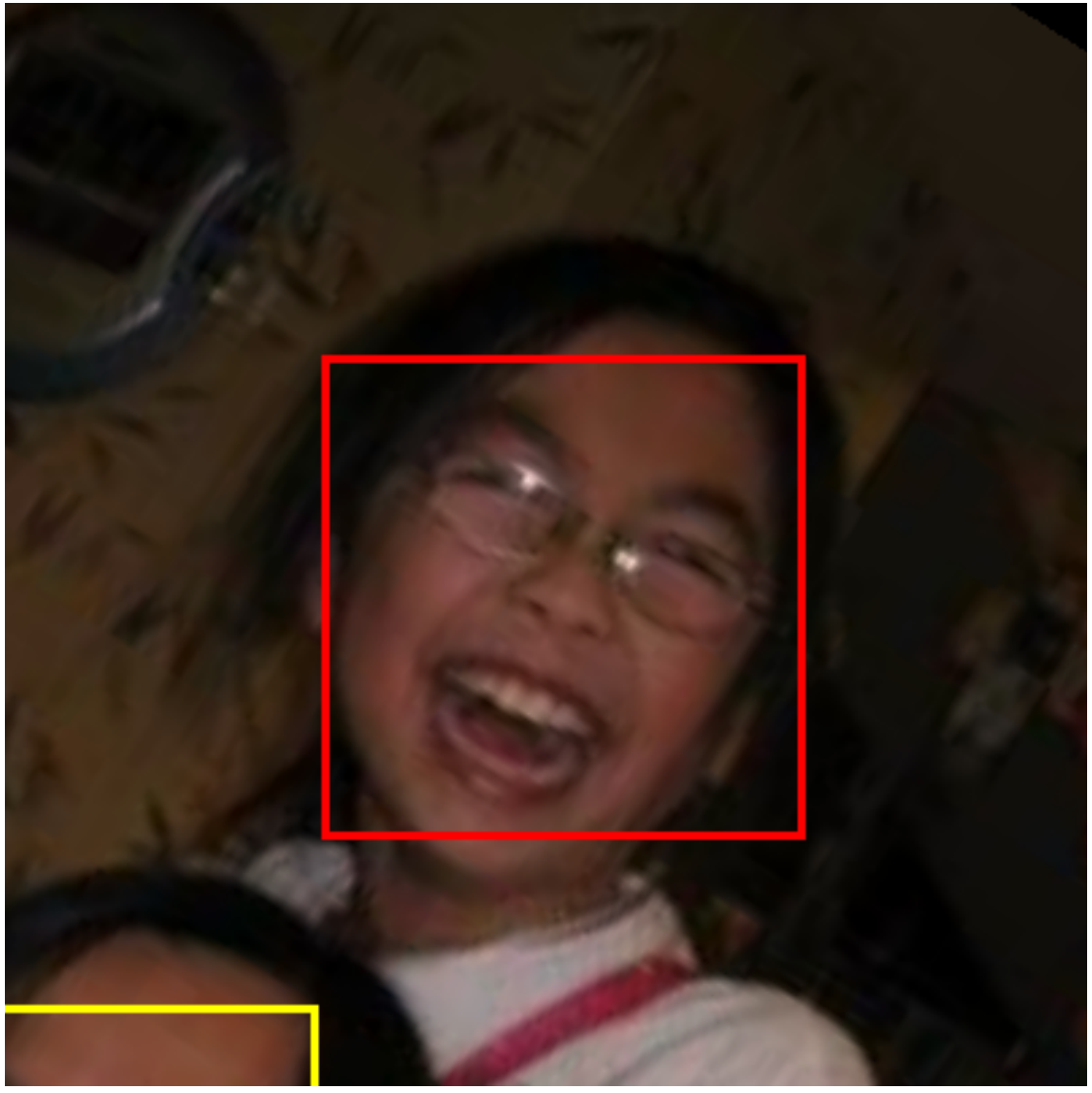





Data augmentation is really important

- You almost always want to do a lot of data augmentation
- E.g. randomly rotate, translate, mirror, change size, disturb colors/gamma
- Be careful about handling clipped objects

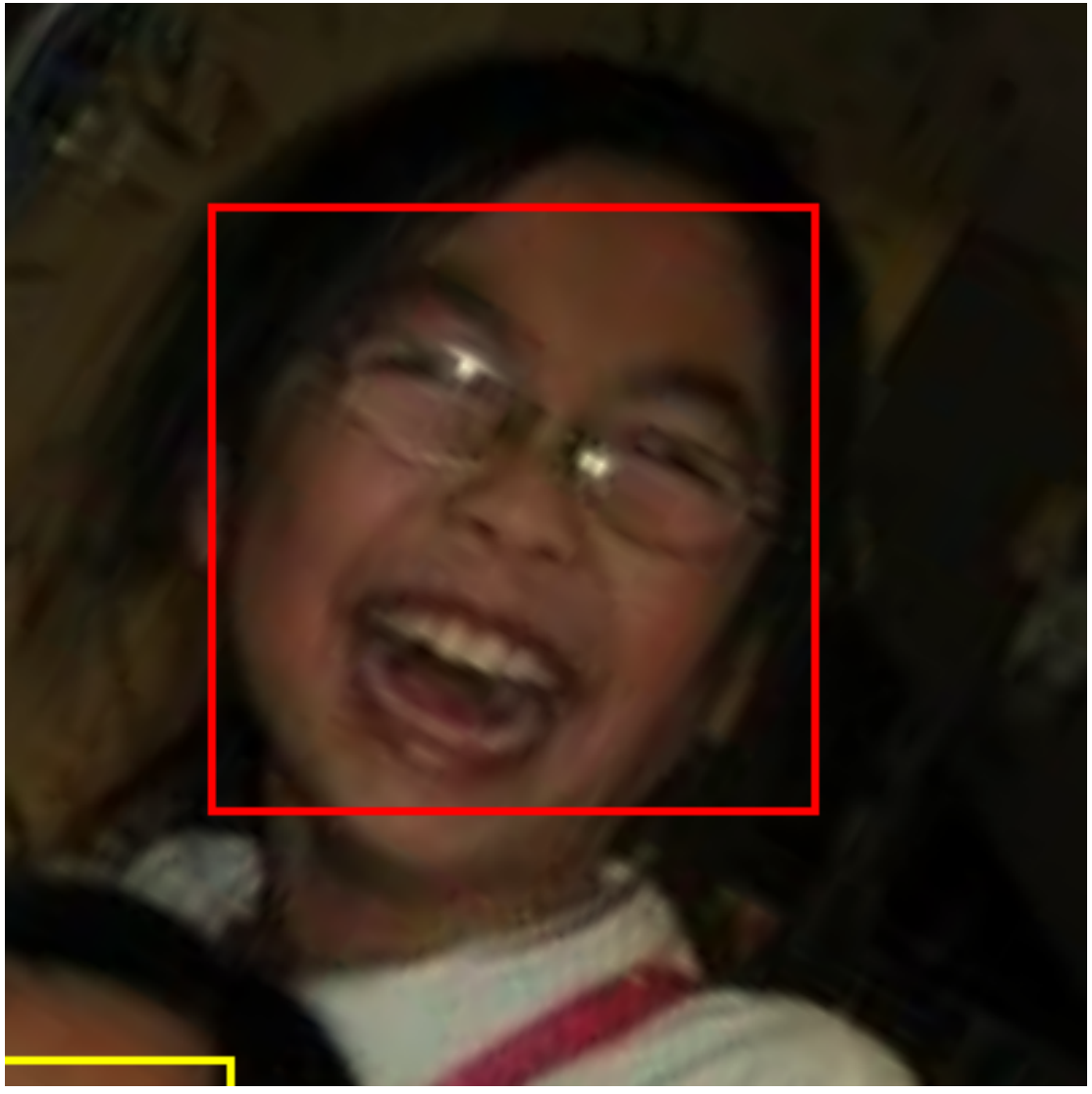
Data augmentation - Random scale,translation,gamma,mirroring,rotating



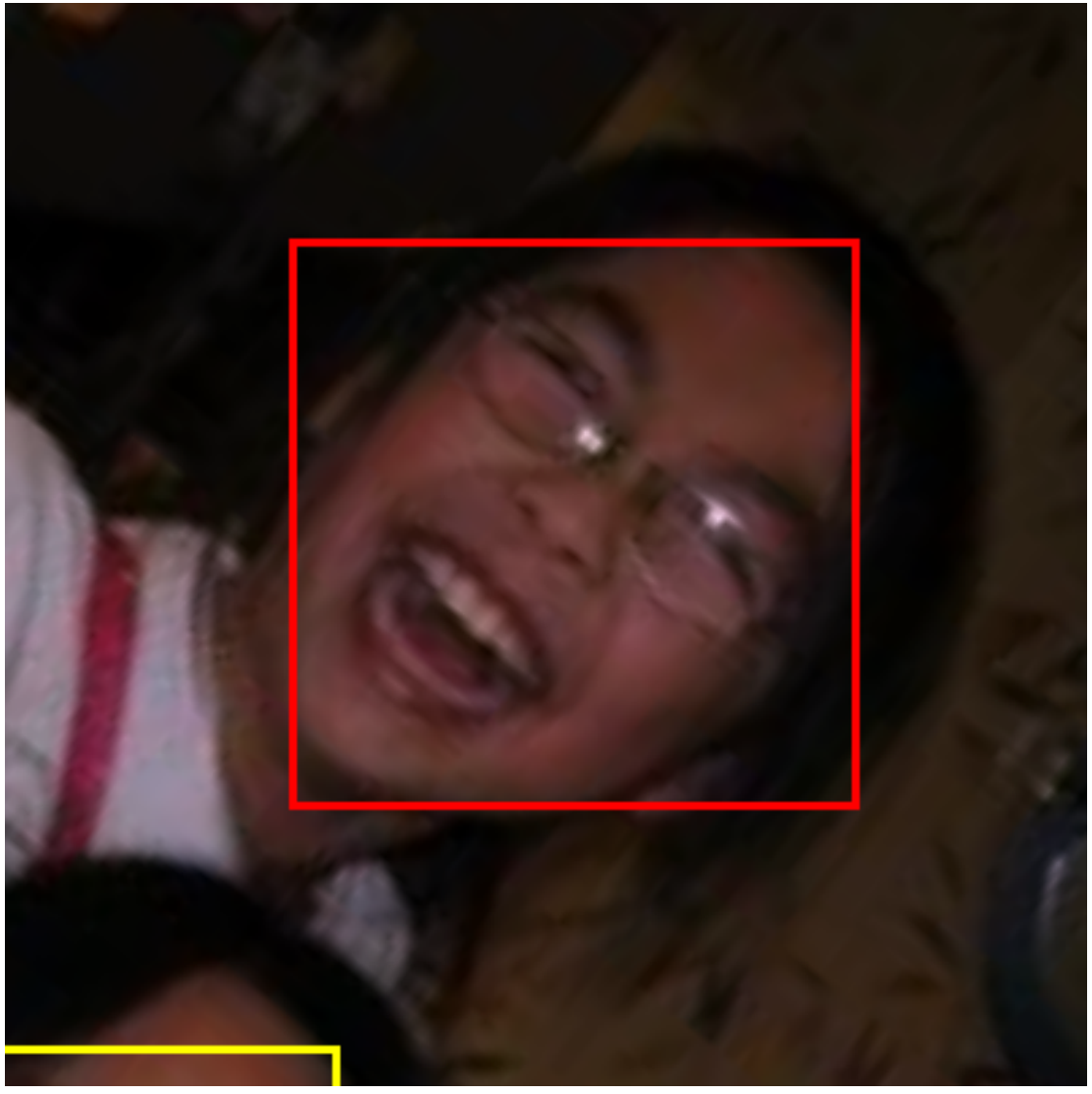
Data augmentation - Random scale,translation,gamma,mirroring,rotating

Data augmentation - Random scale,translation,gamma,mirroring,rotating

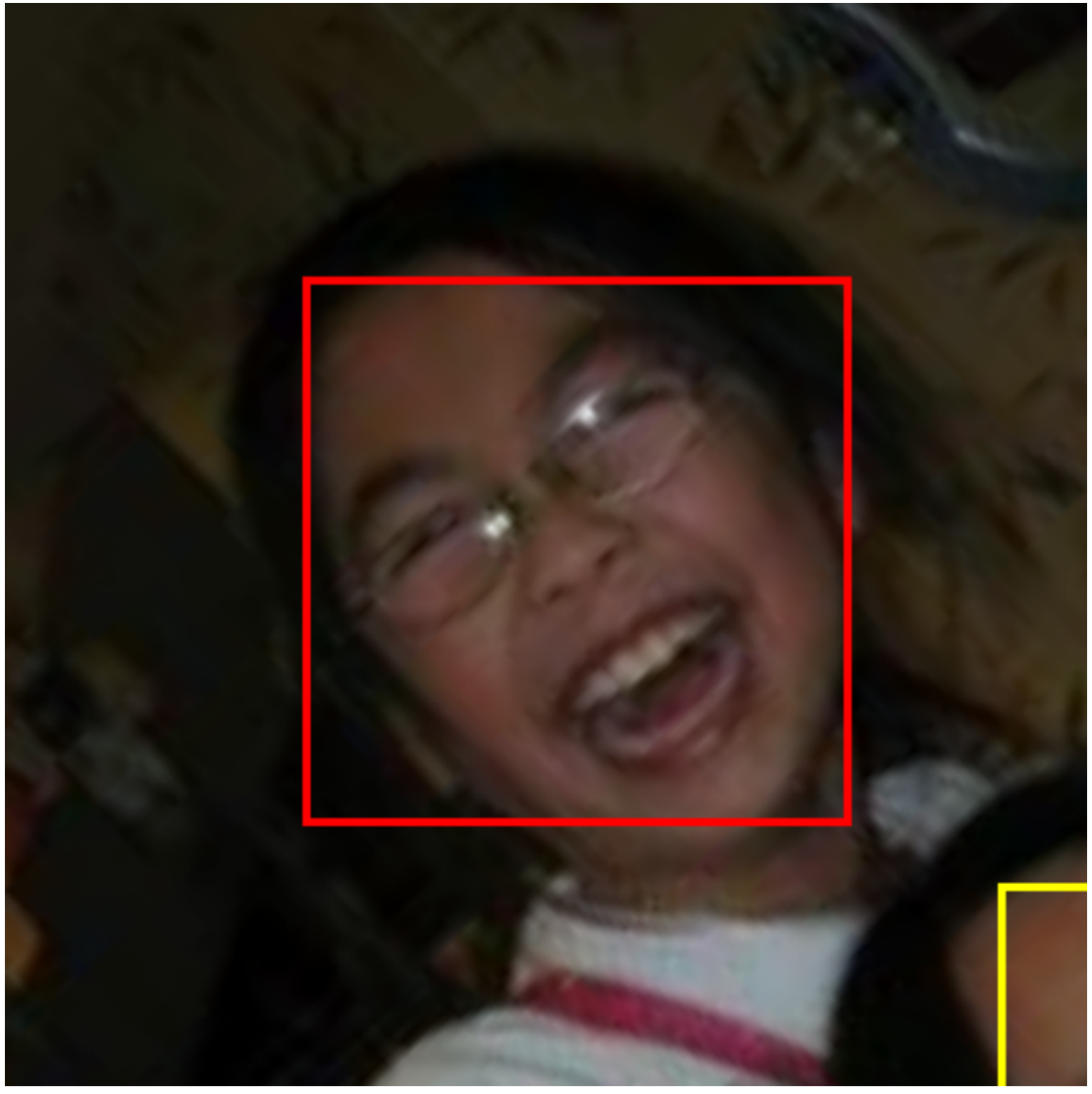
Data augmentation - Random scale,translation,gamma,mirroring,rotating



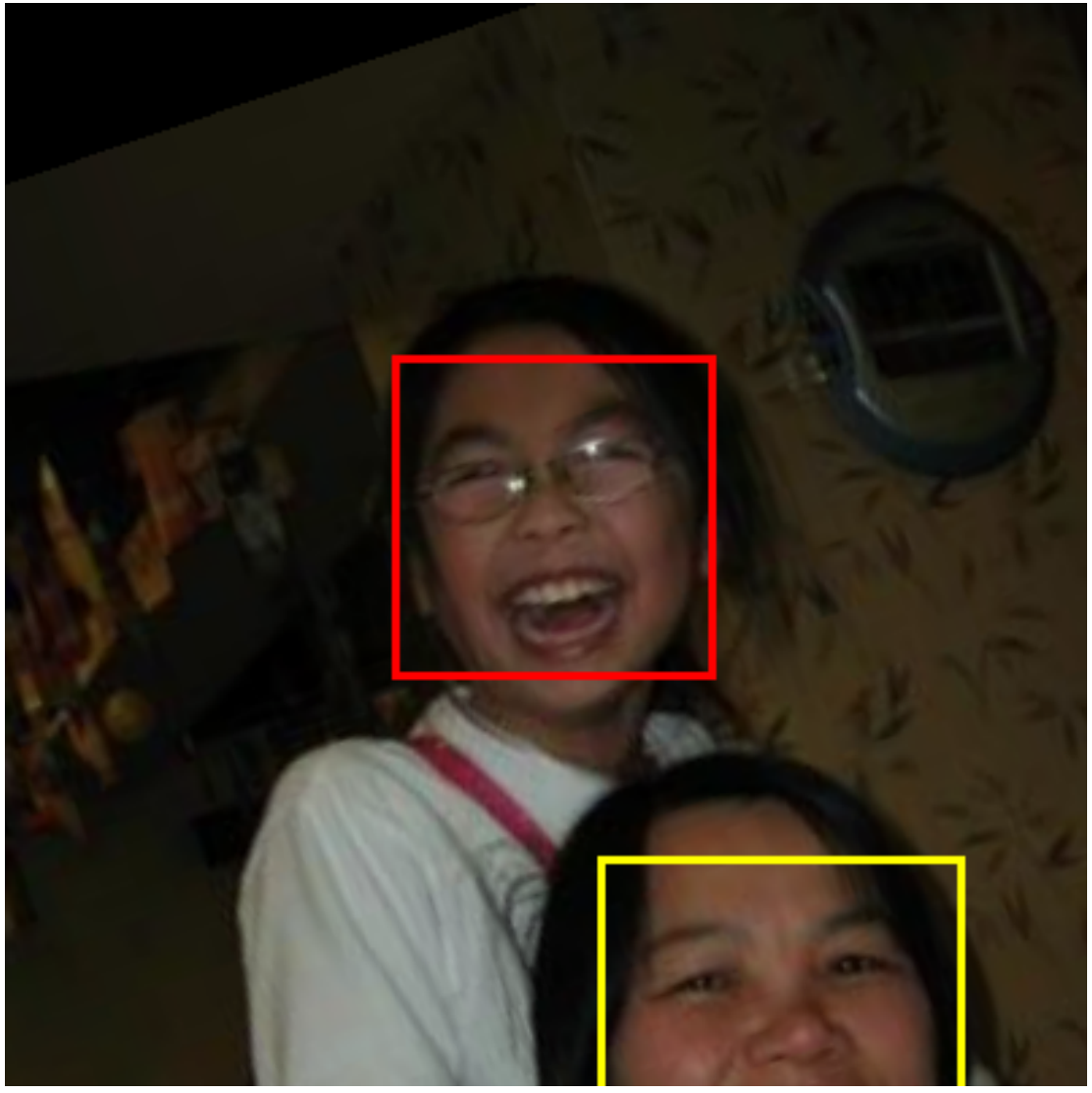
Data augmentation - Random scale,translation,gamma,mirroring,rotating



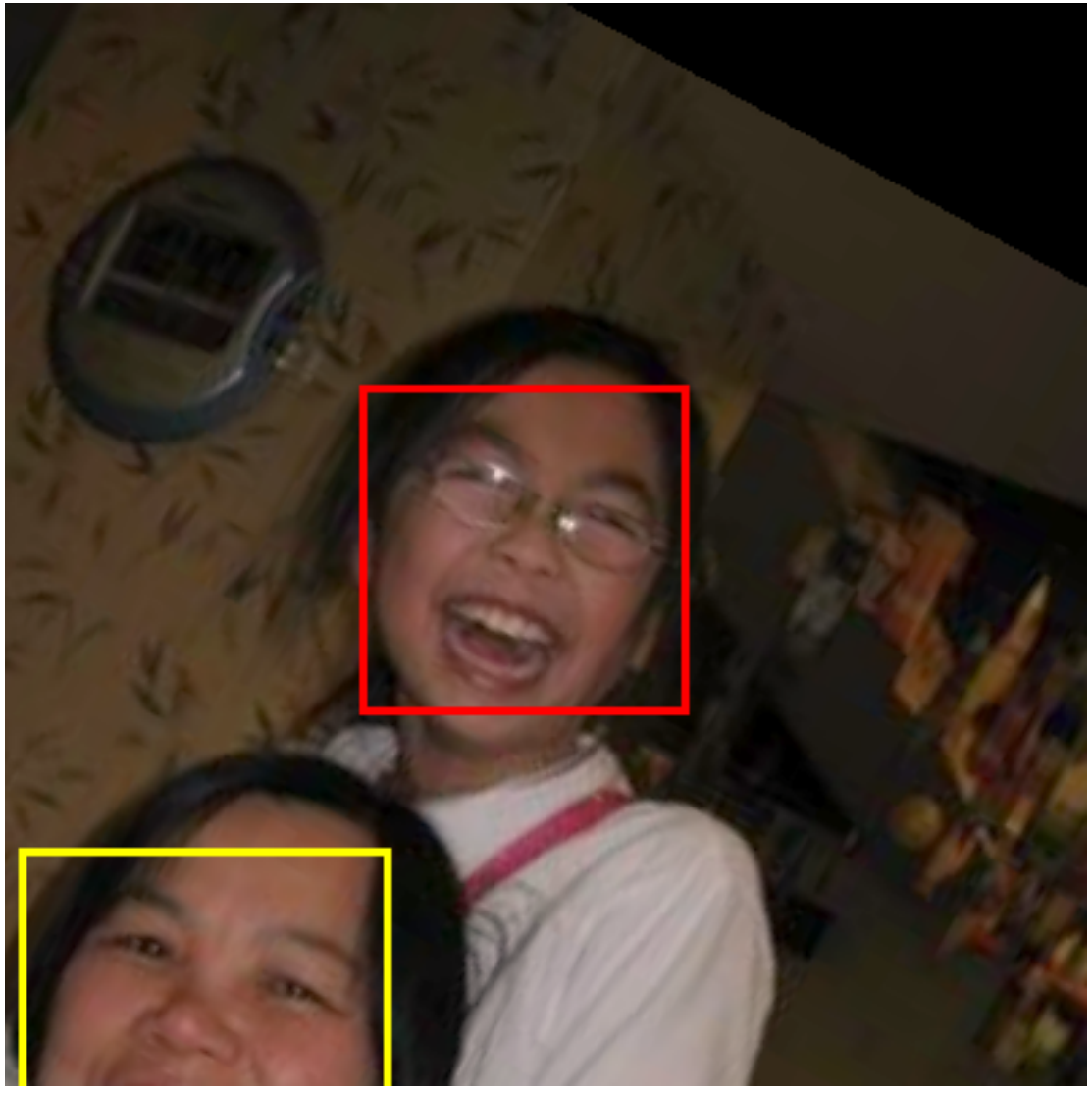
Data augmentation - Random scale,translation,gamma,mirroring,rotating



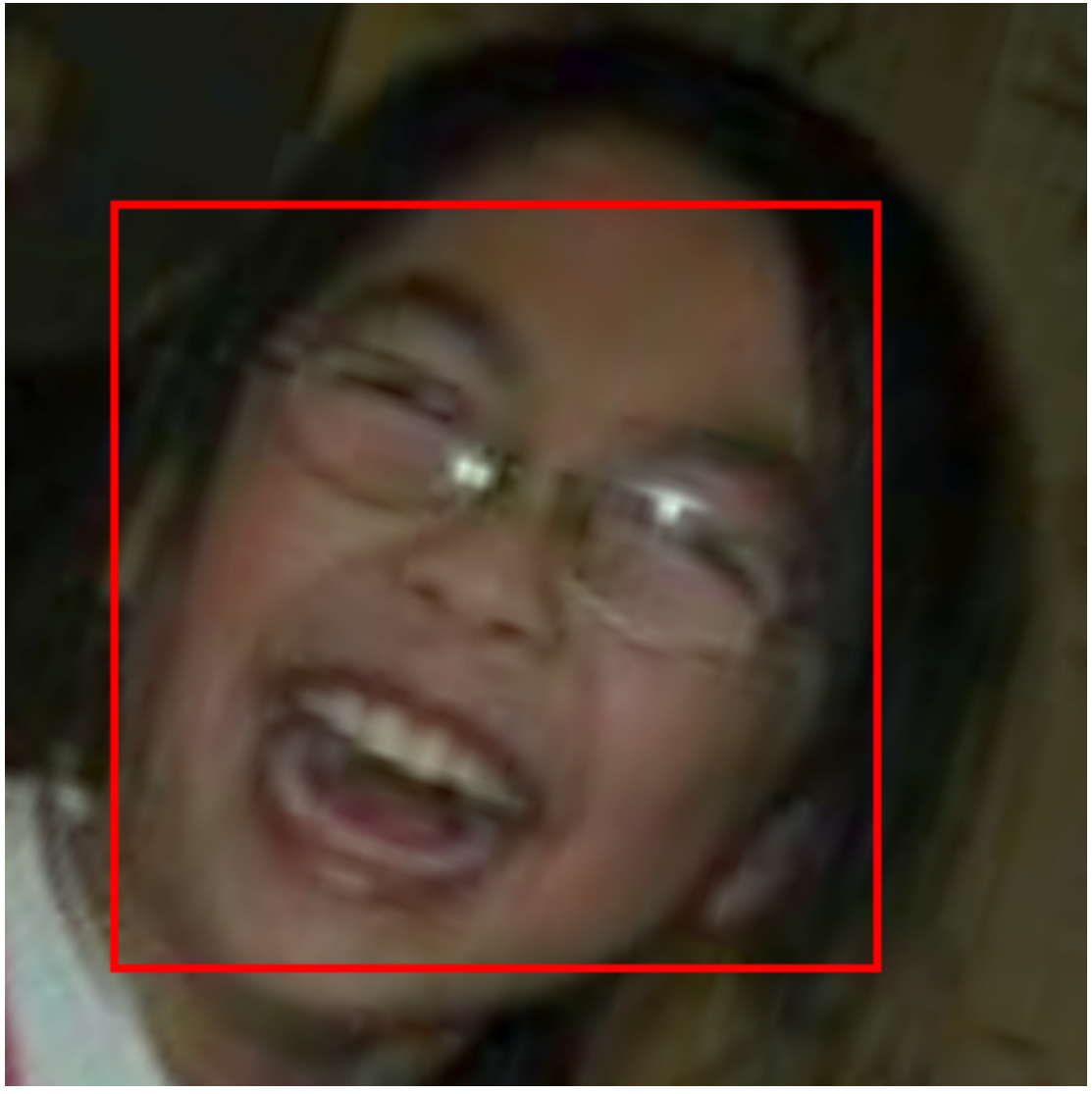
Data augmentation - Random scale,translation,gamma,mirroring,rotating



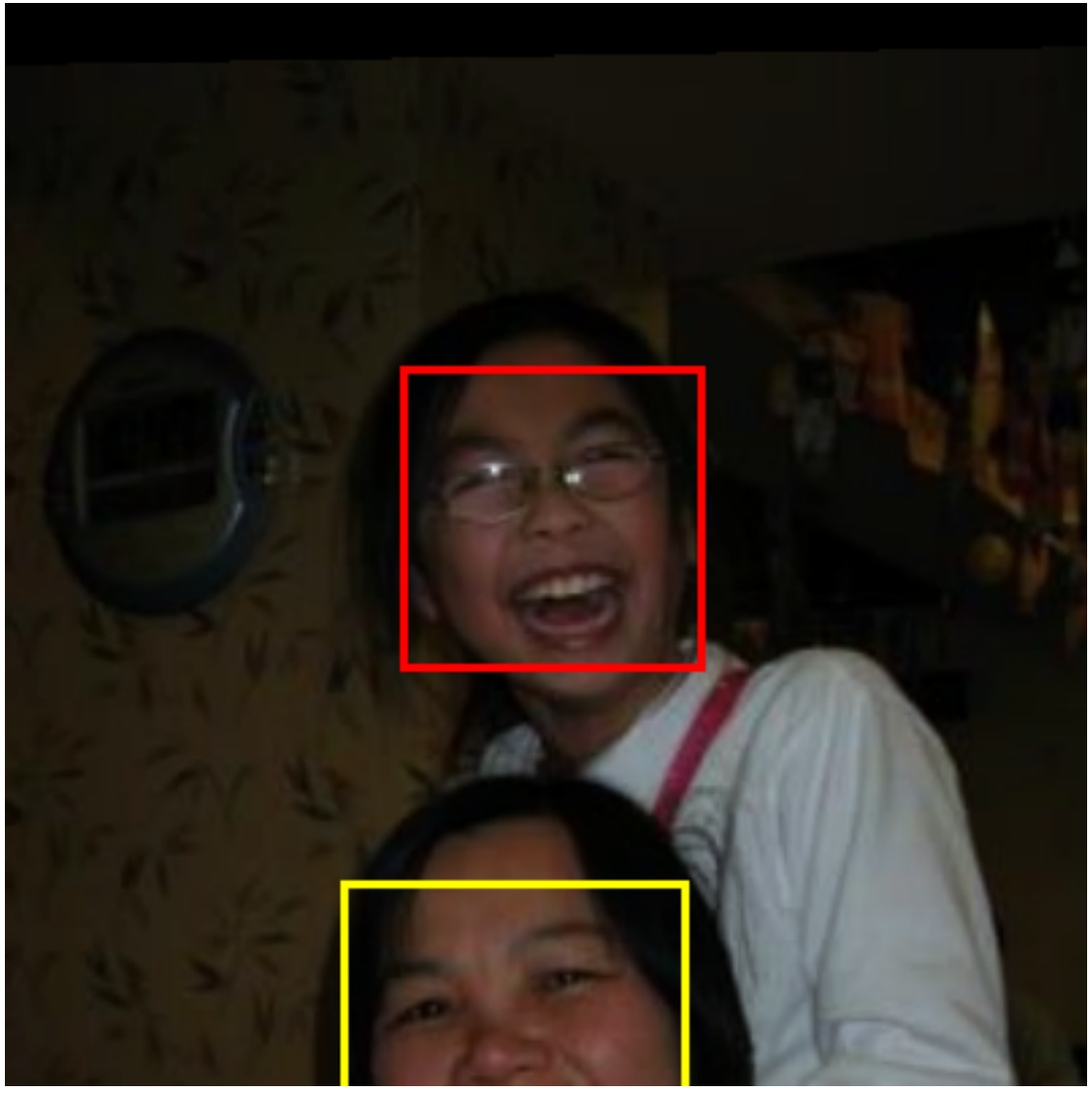
Data augmentation - Random scale,translation,gamma,mirroring,rotating



Data augmentation - Random scale,translation,gamma,mirroring,rotating

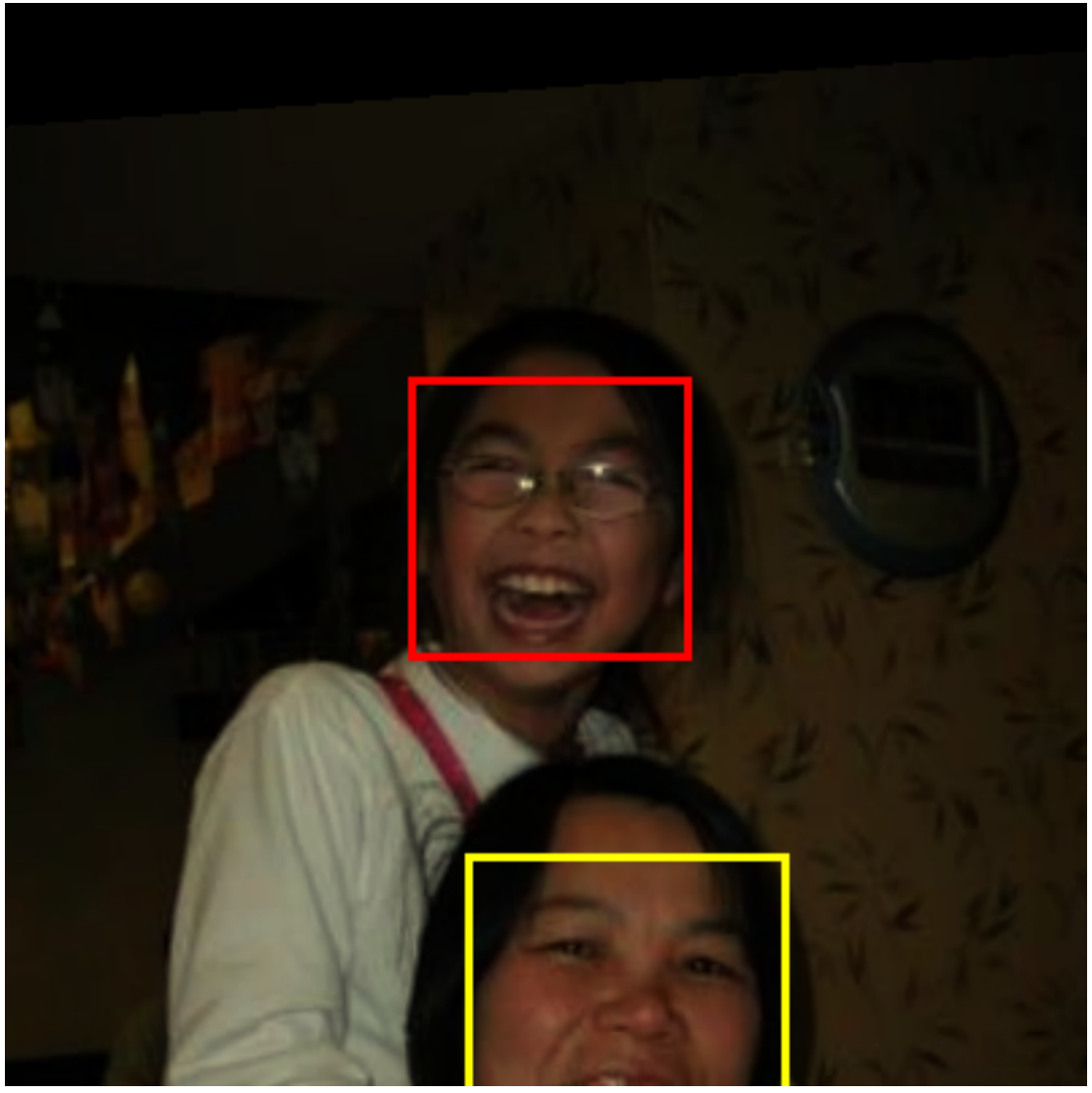


Data augmentation - Random scale,translation,gamma,mirroring,rotating



Data augmentation - Random scale,translation,gamma,mirroring,rotating

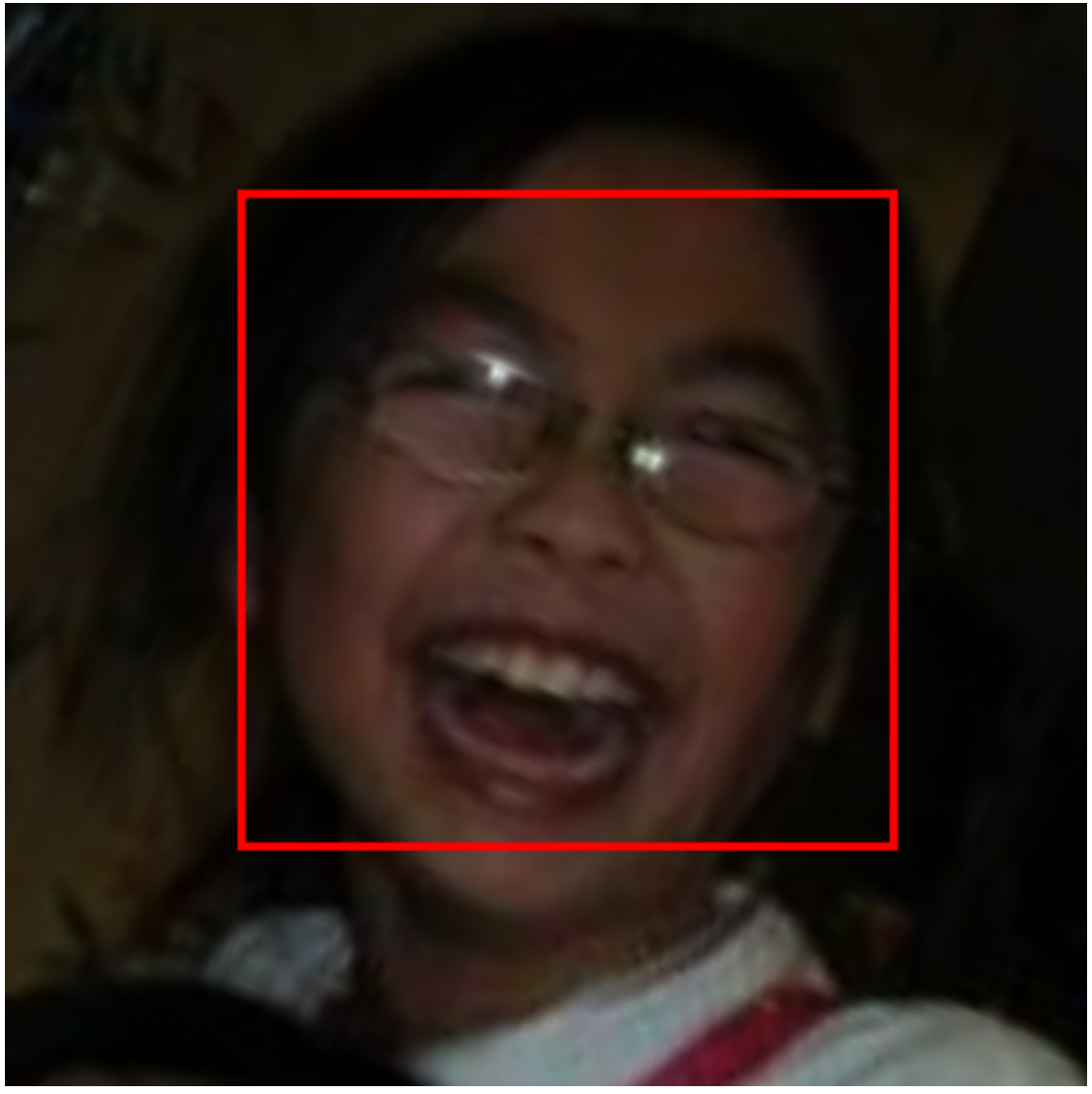
Data augmentation - Random scale,translation,gamma,mirroring,rotating



Data augmentation - Random scale,translation,gamma,mirroring,rotating

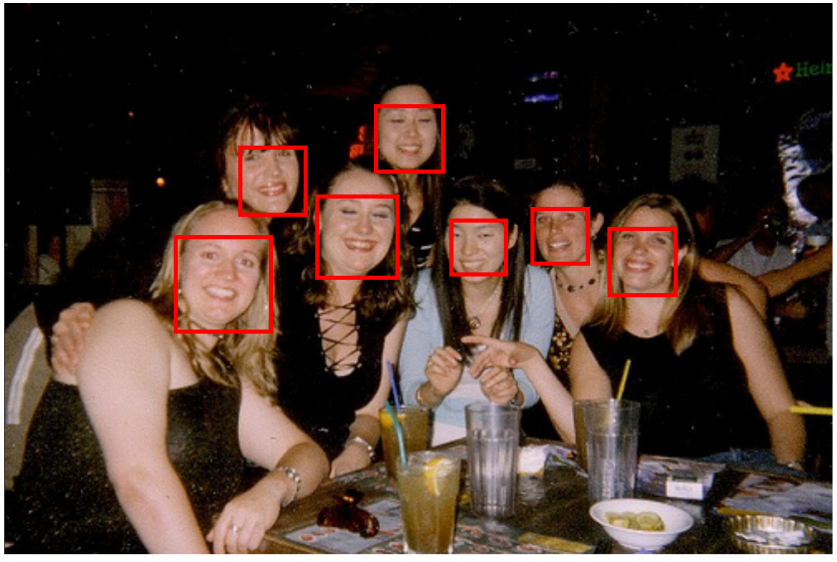
Data augmentation - Random scale,translation,gamma,mirroring,rotating

Data augmentation - Random scale,translation,gamma,mirroring,rotating



Data augmentation is really important

- CNN+MMOD face detector trained with and without data augmentation:




- Done by setting this in examples/dnn_mmod_ex.cpp:

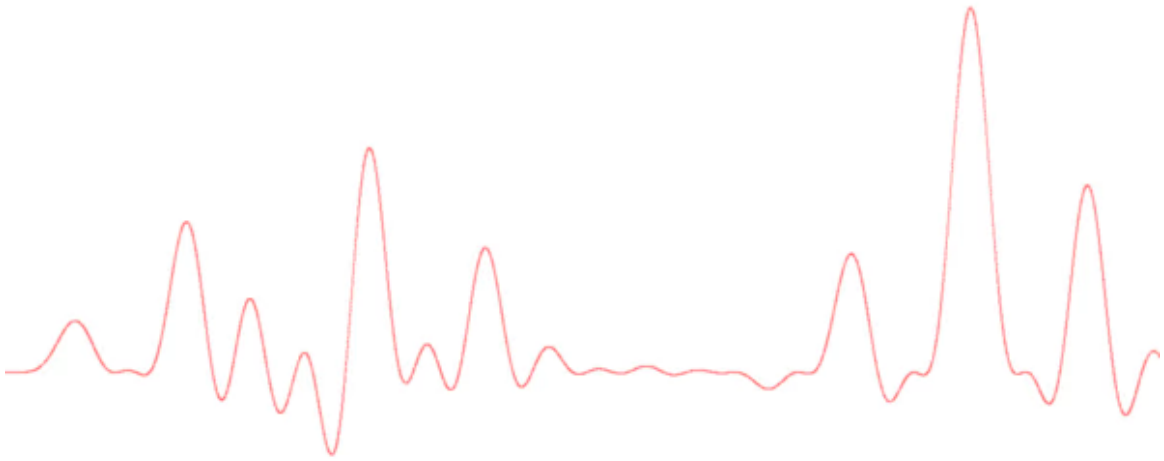
```
cropper.set_translate_amount(0);
cropper.set_randomly_flip(false);
cropper.set_max_object_size(1);
cropper.set_max_rotation_degrees(0);
```

Setting hyperparamters

- Hyperparameters are the parameters machine learning doesn't set for you. E.g. stopping conditions, learning rates, regularization strengths, etc.

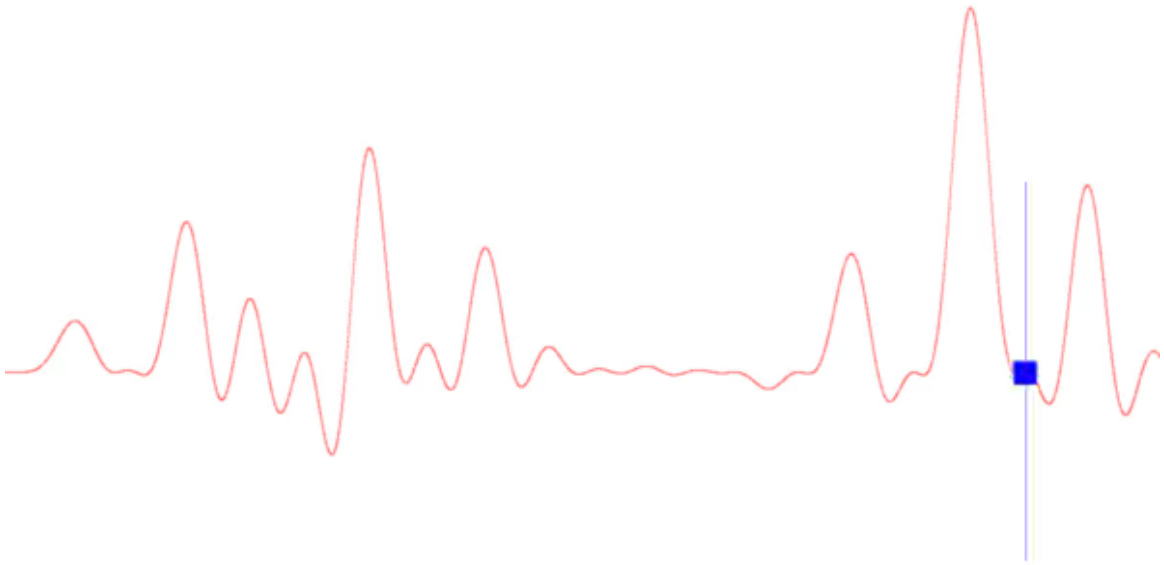
find_max_global() in action - 1

- Optimize this function:



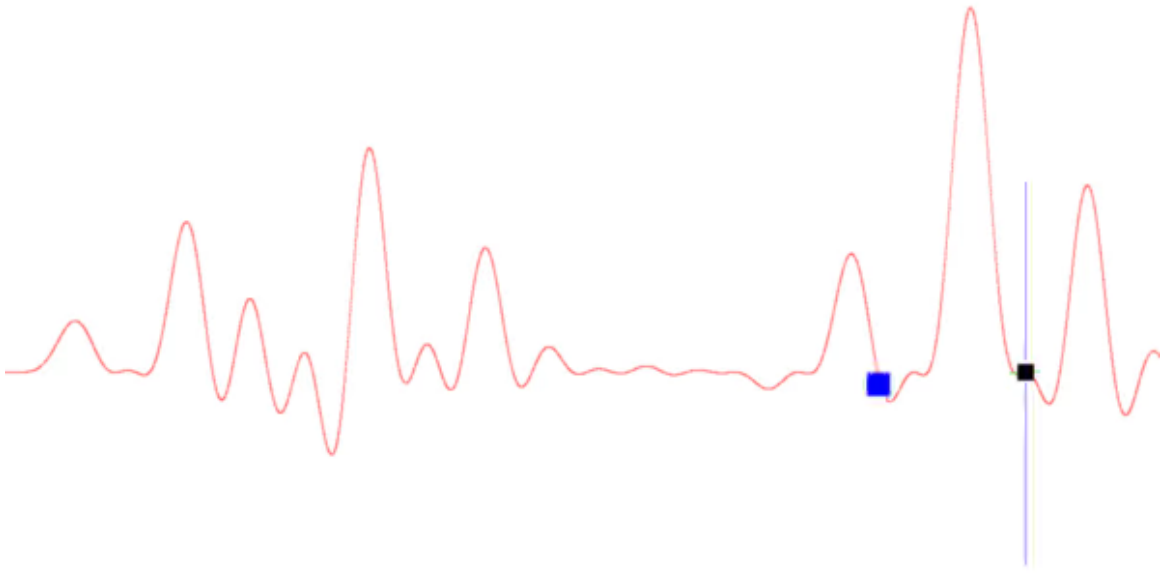
find_max_global() in action - 2

- Pick random point



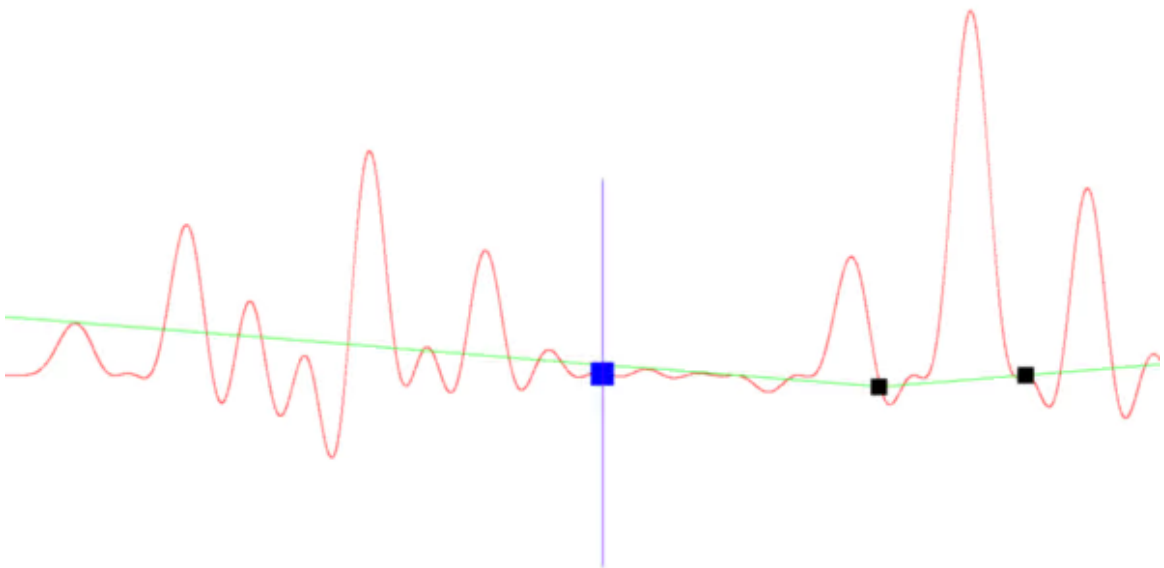
find_max_global() in action - 3

- Pick random point



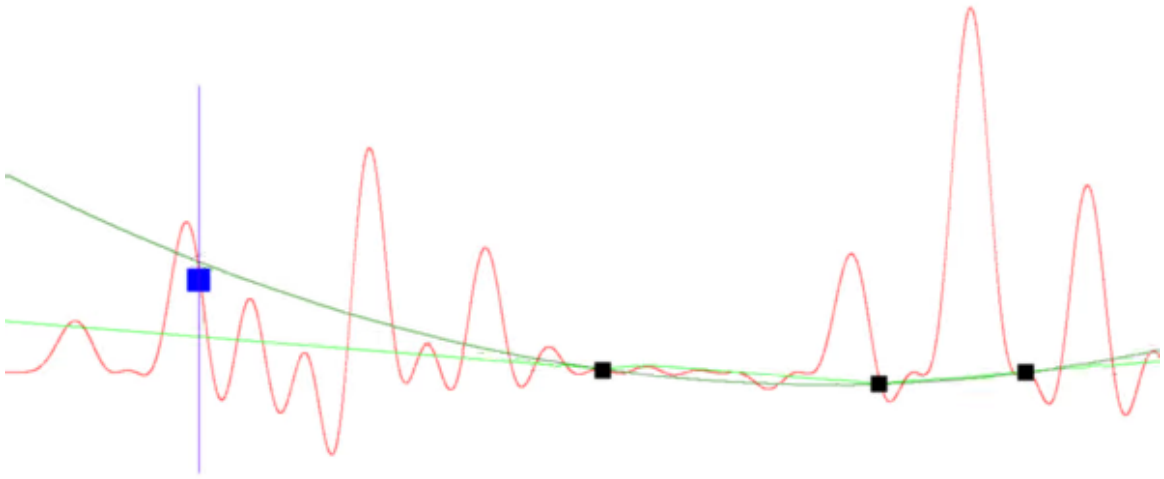
find_max_global() in action - 4

- Pick random point



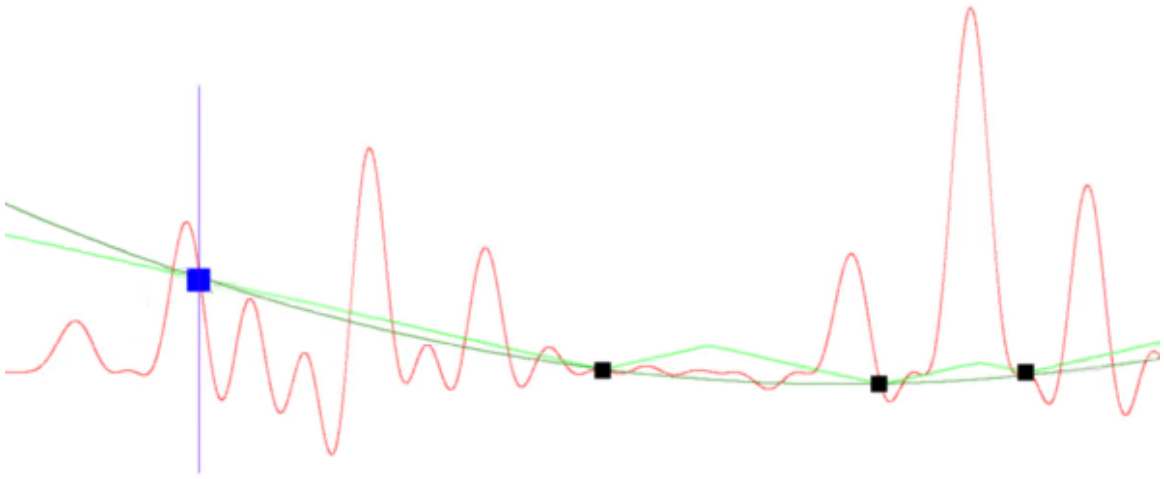
find_max_global() in action - 5

- Pick random point



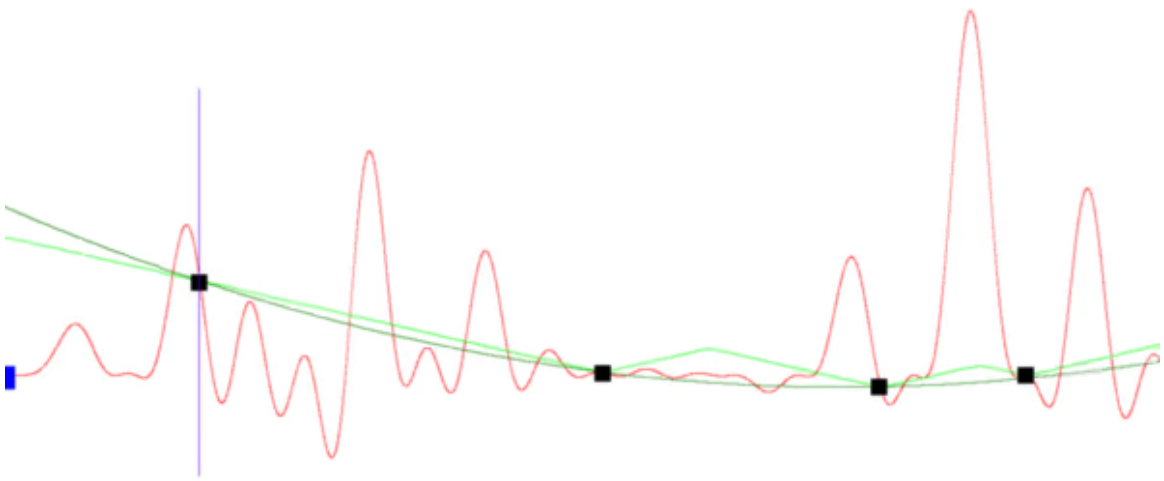
find_max_global() in action - 6

- Update models



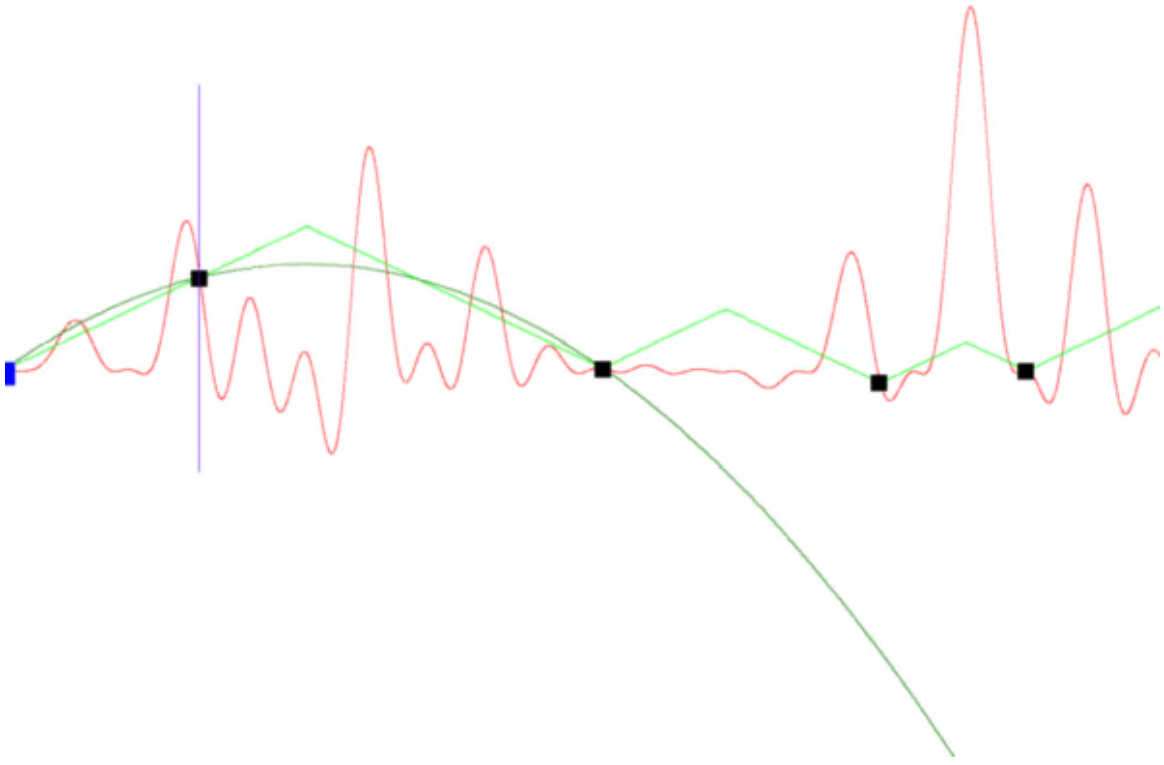
find_max_global() in action - 7

- Use upper bounding model



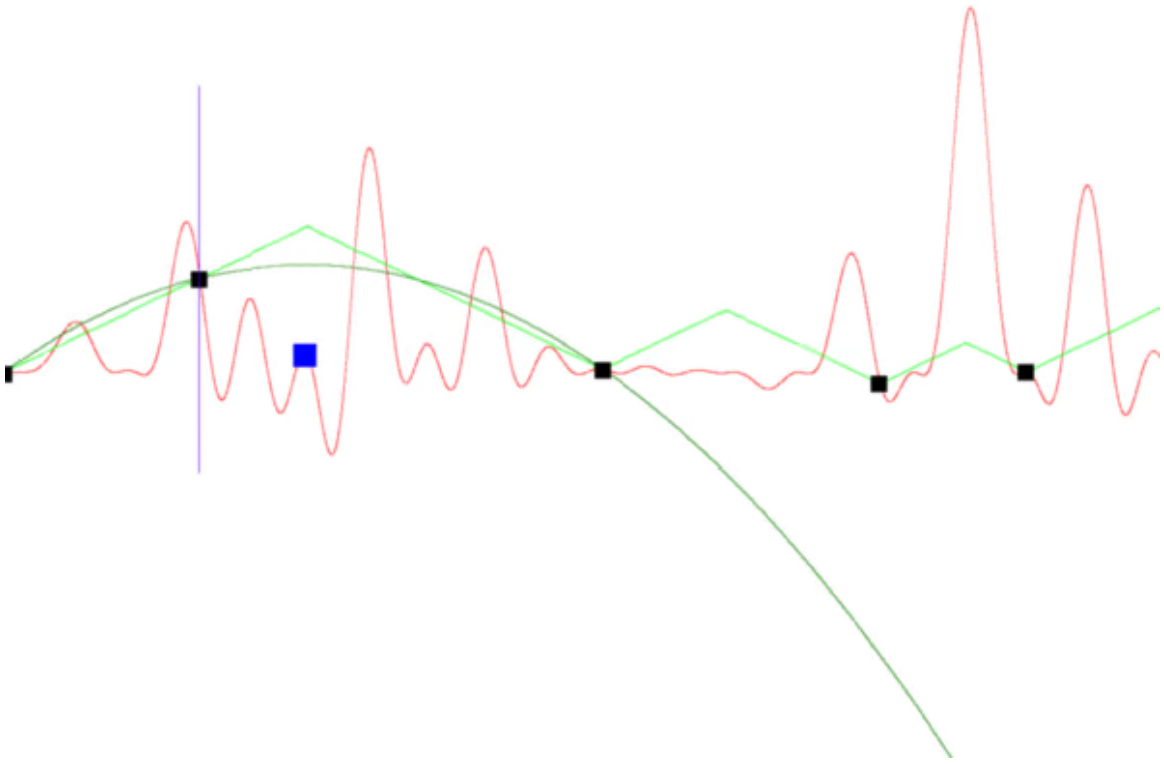
find_max_global() in action - 8

- Update models



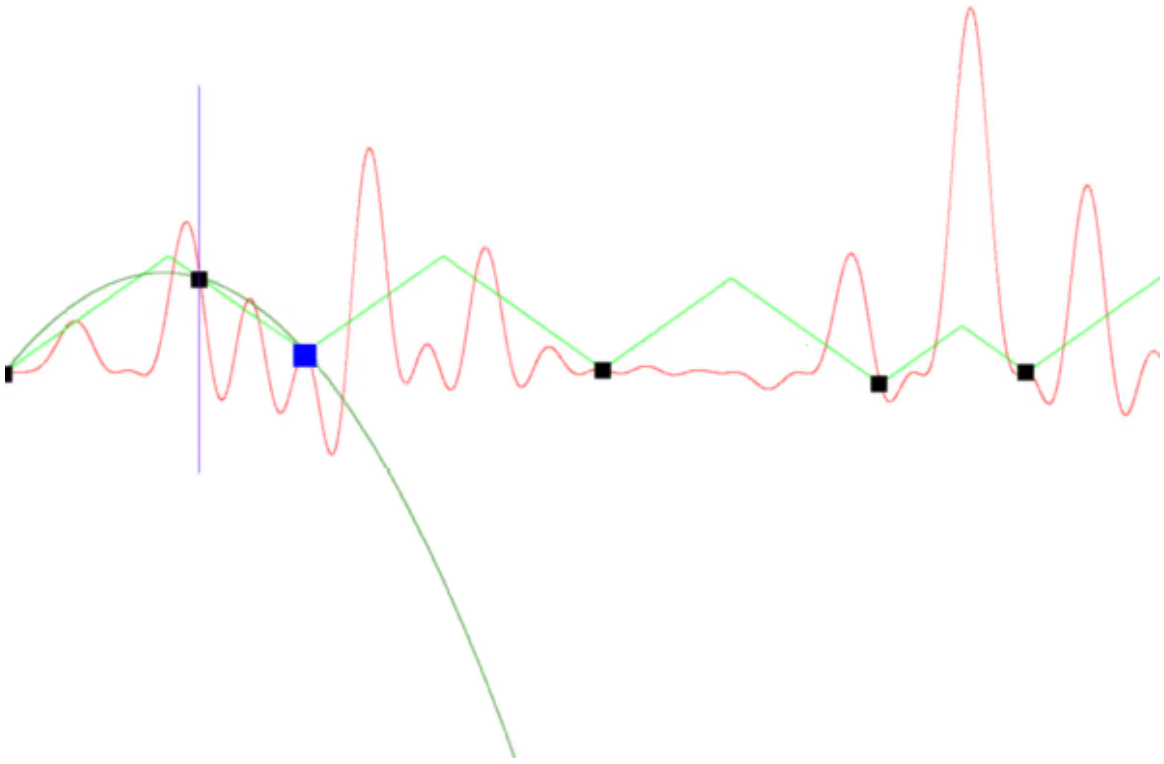
find_max_global() in action - 9

- Use trust region model



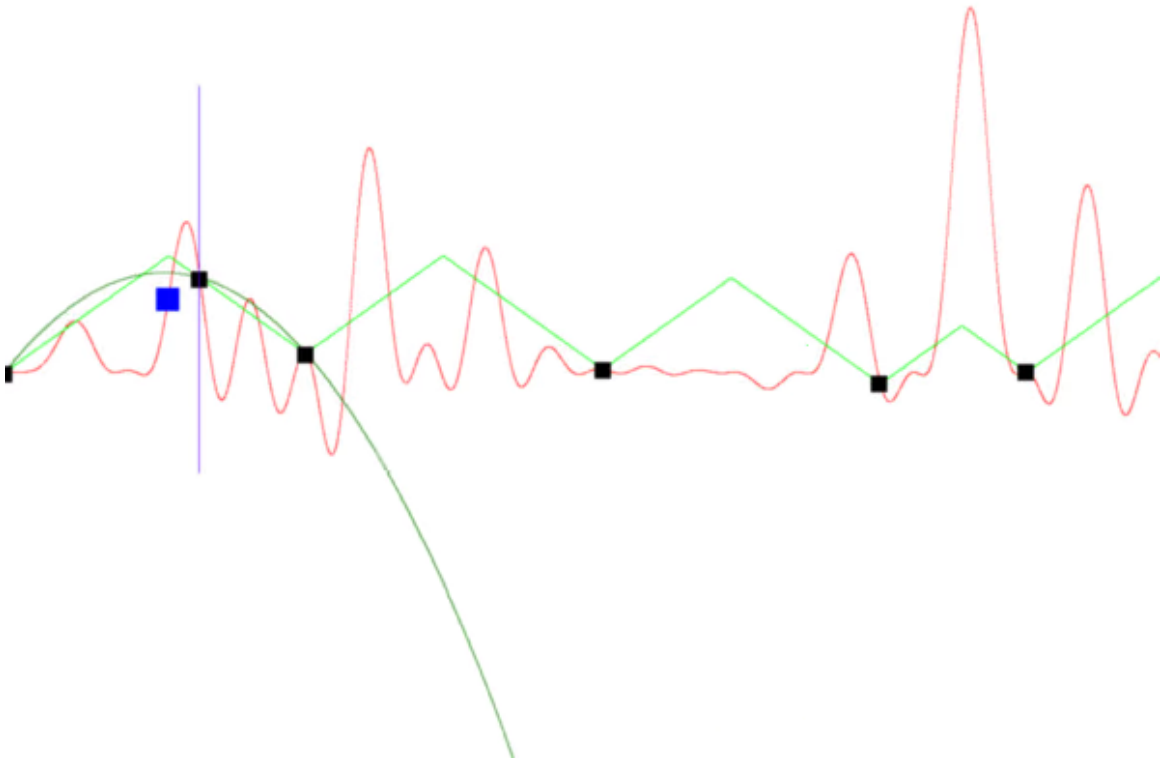
find_max_global() in action - 10

- Update models



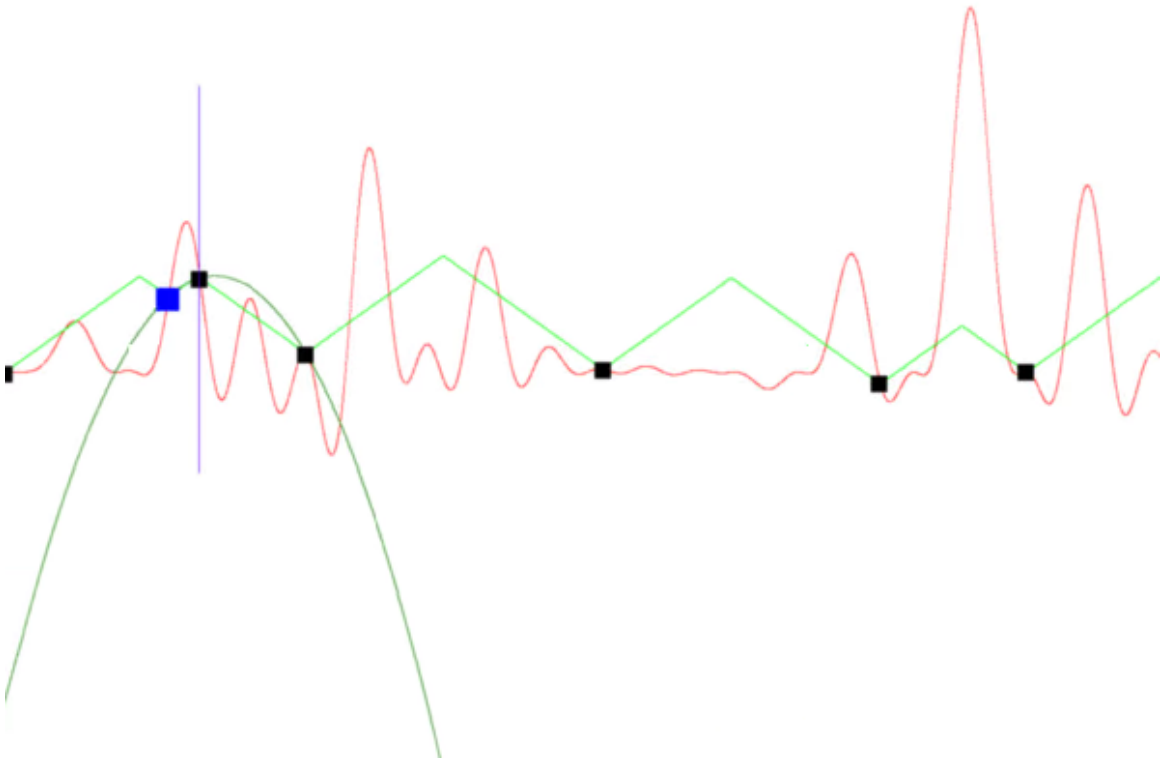
find_max_global() in action - 11

- Use upper bounding model



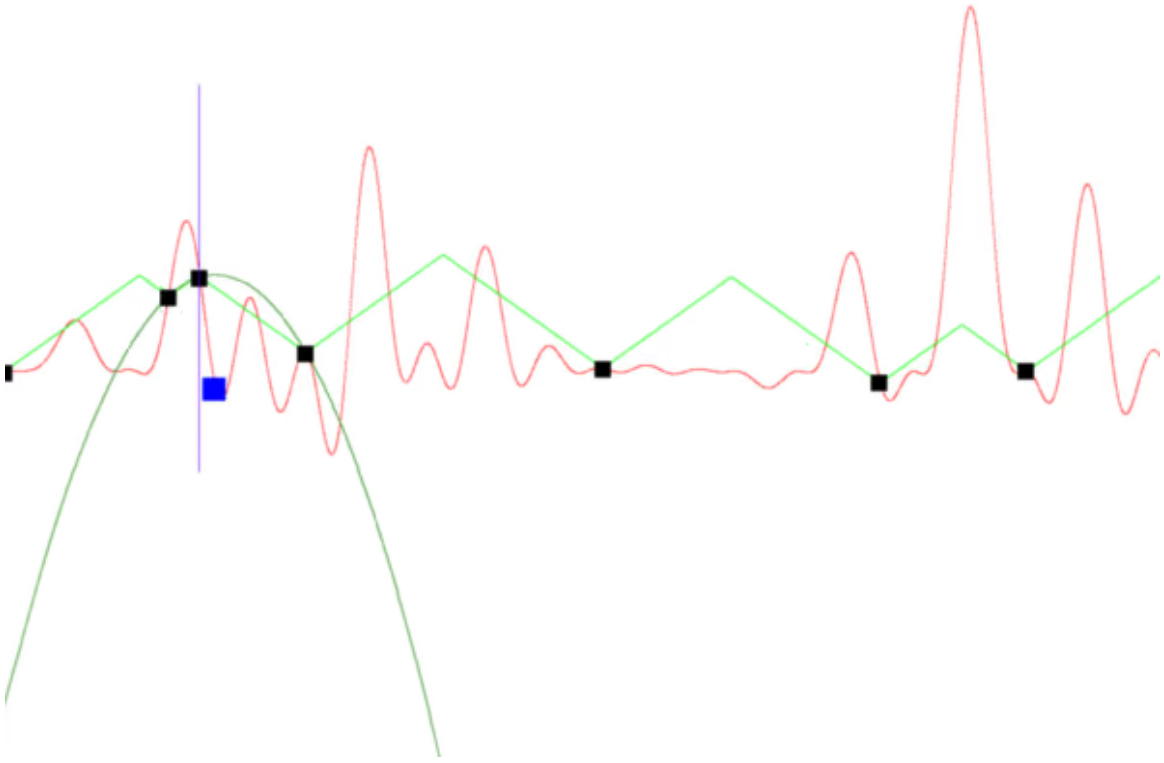
find_max_global() in action - 12

- Update models



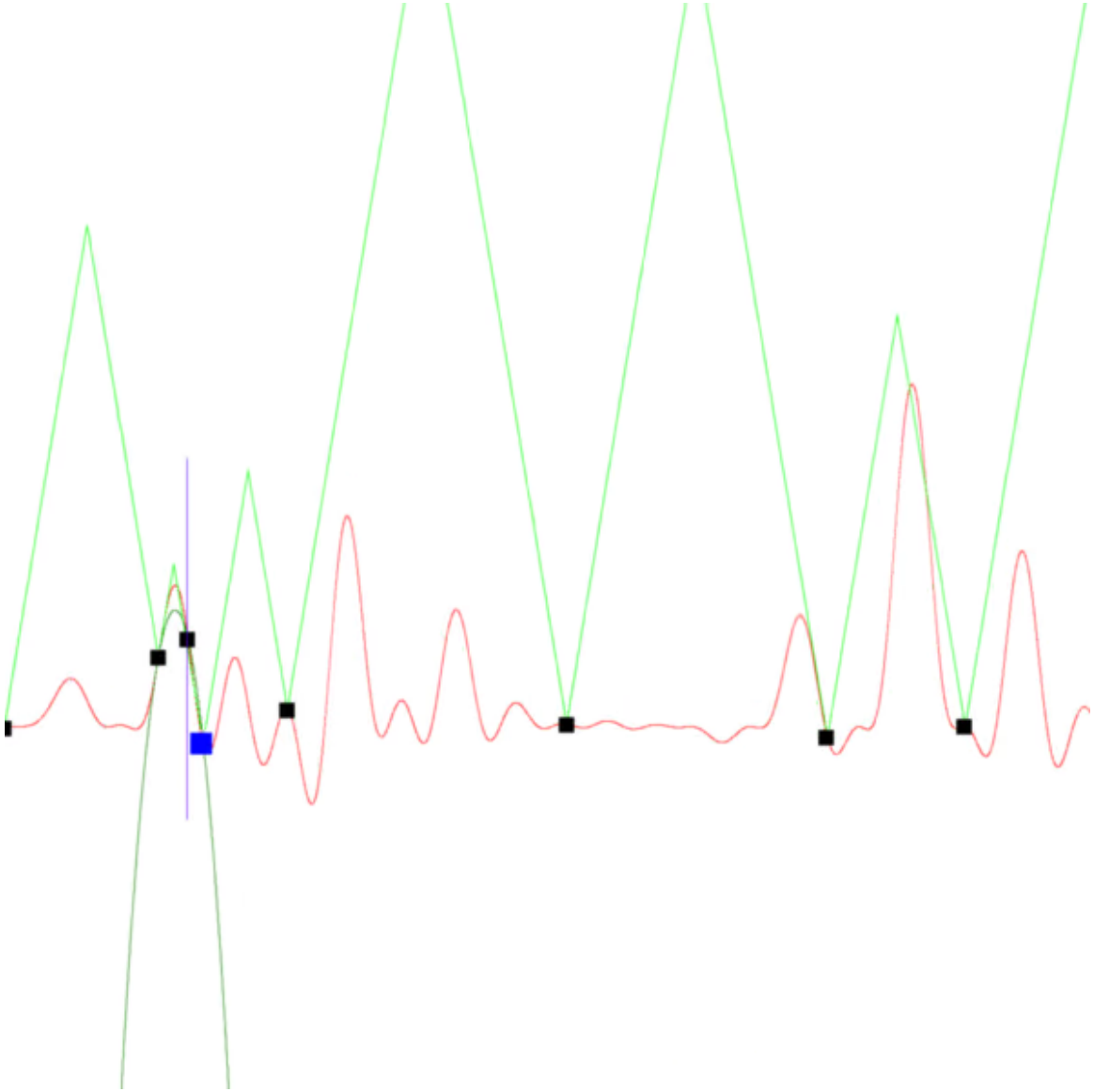
find_max_global() in action - 13

- Use trust region model



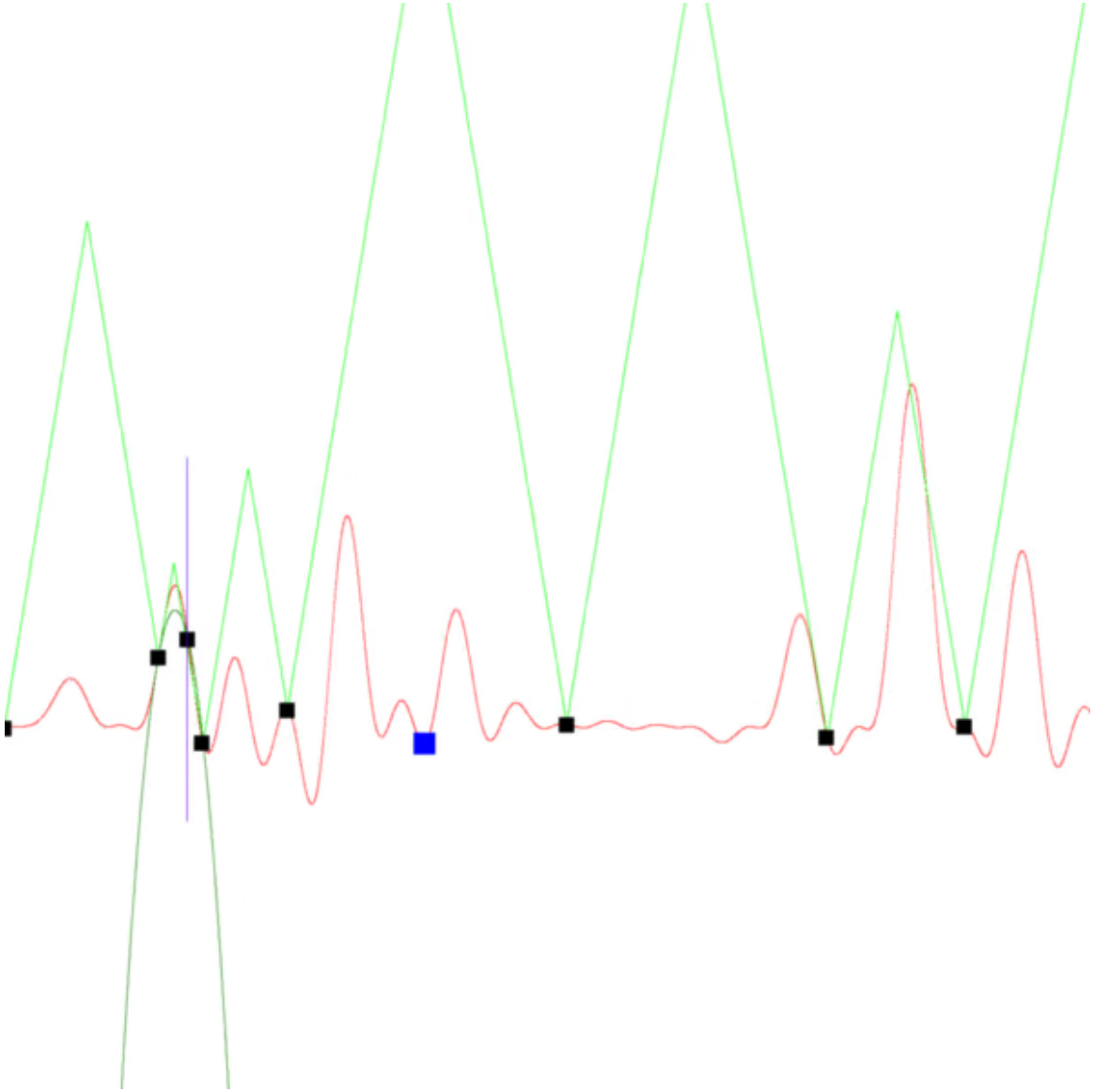
find_max_global() in action - 14

- Update models



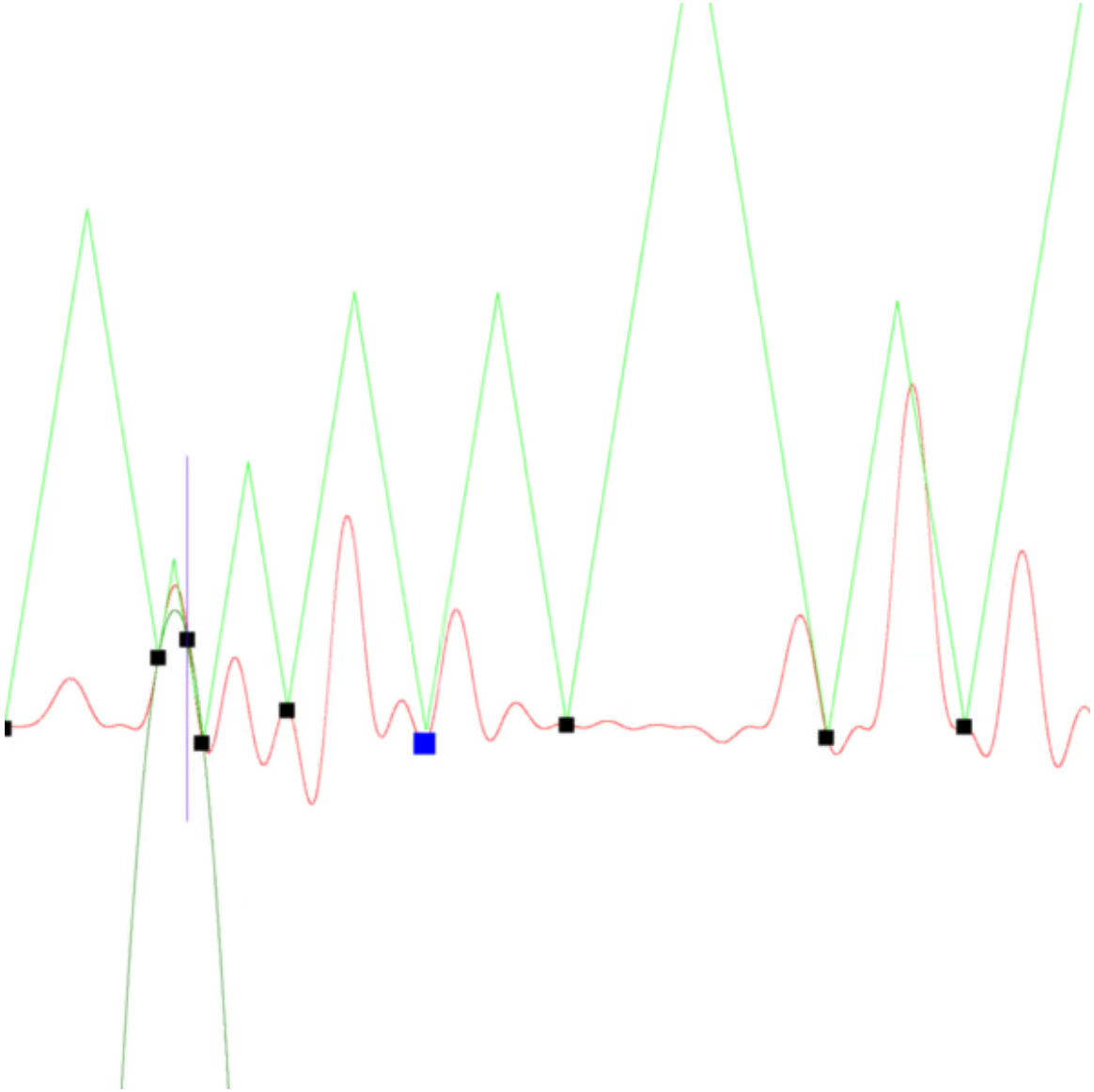
find_max_global() in action - 15

- Use upper bounding model



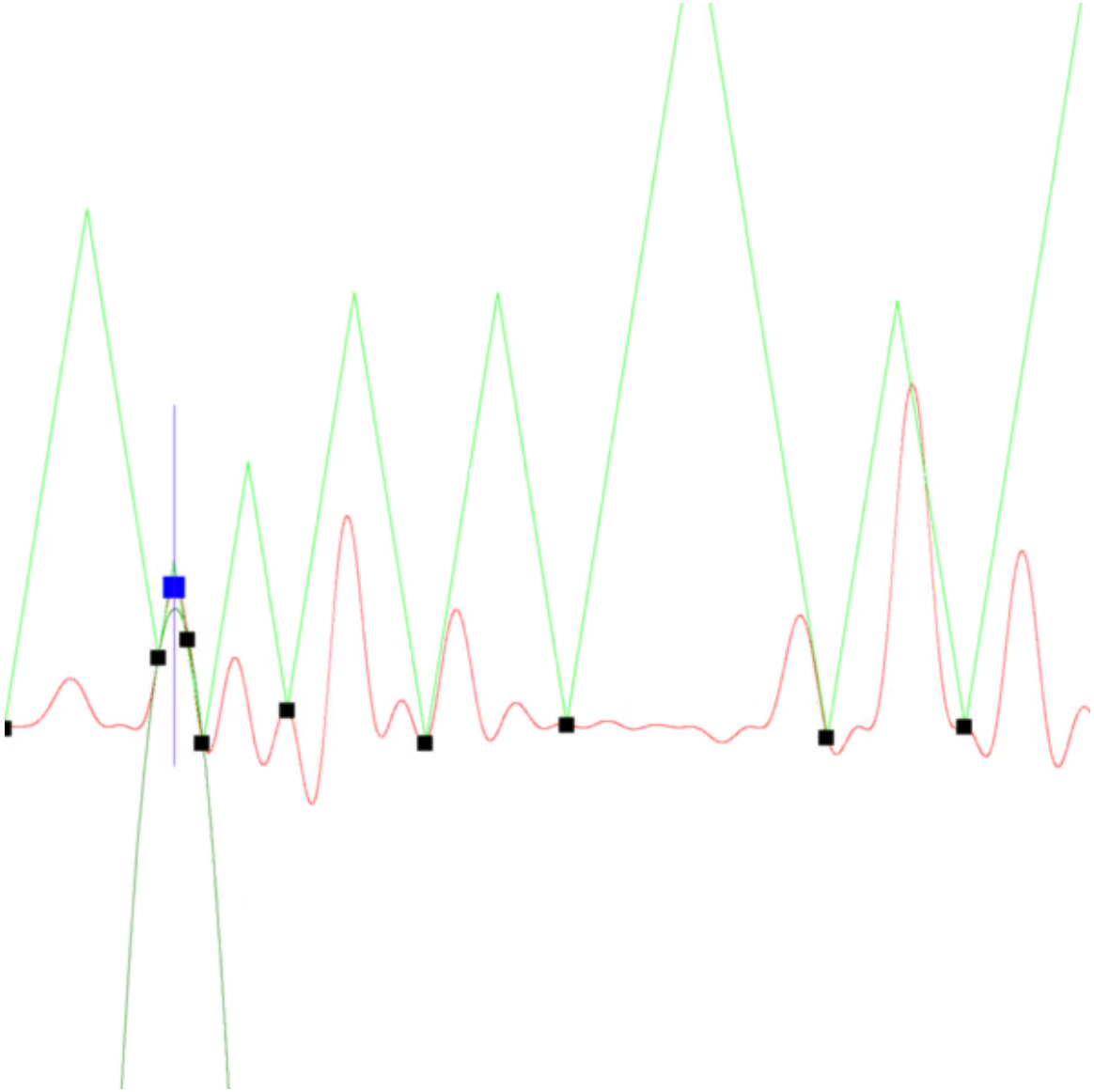
find_max_global() in action - 16

- Update models



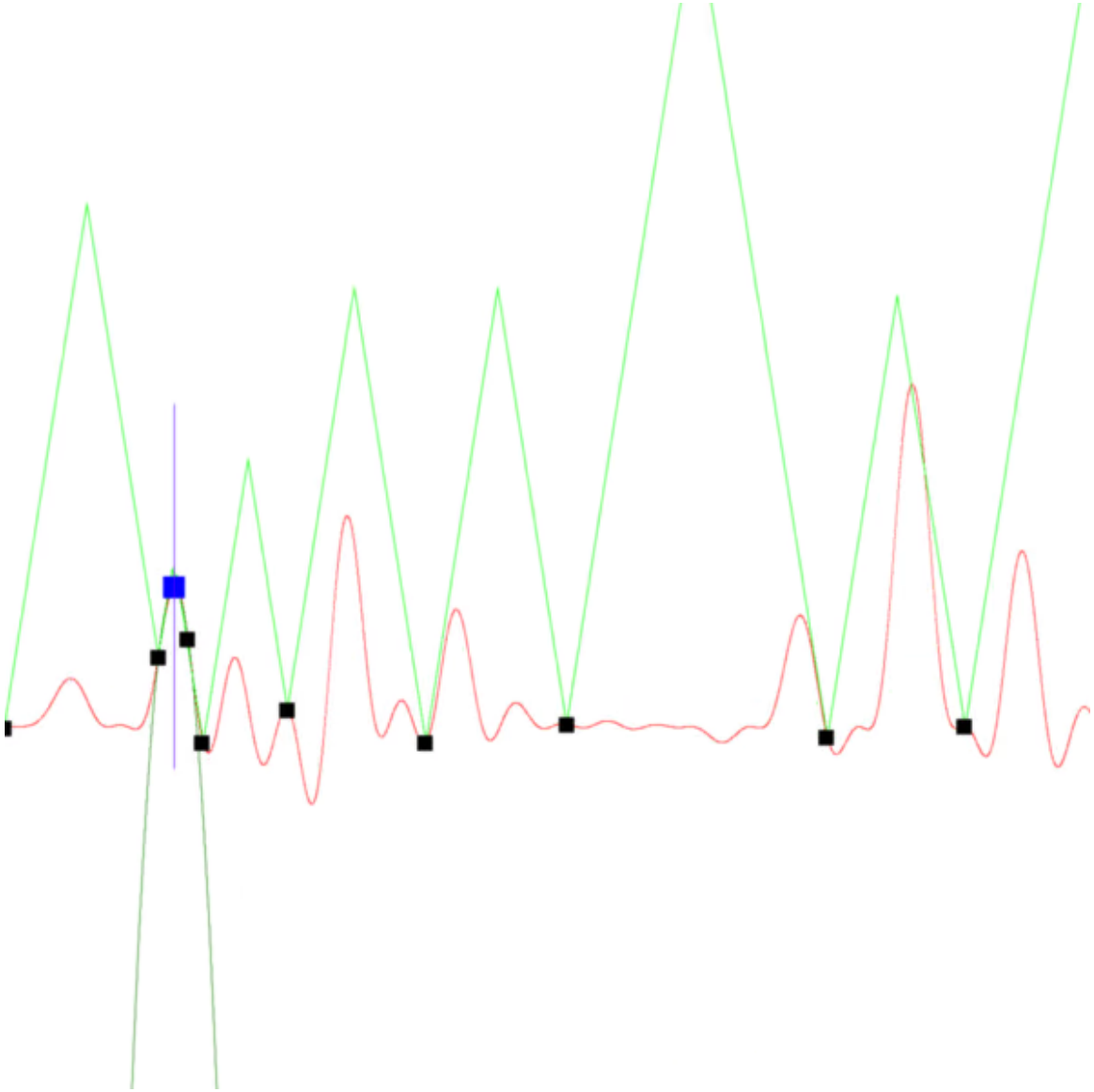
find_max_global() in action - 17

- Use trust region model



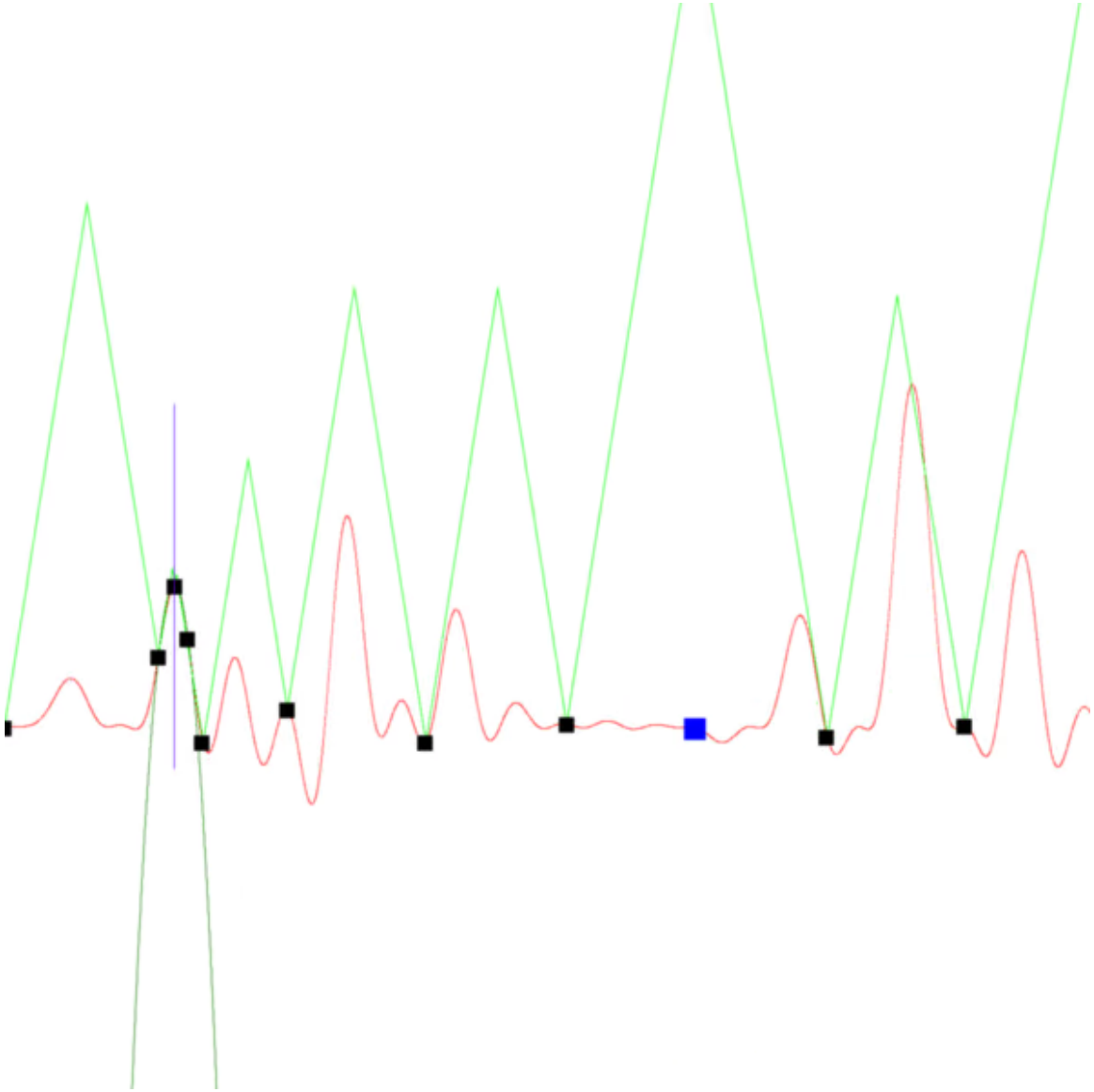
find_max_global() in action - 18

- Update models



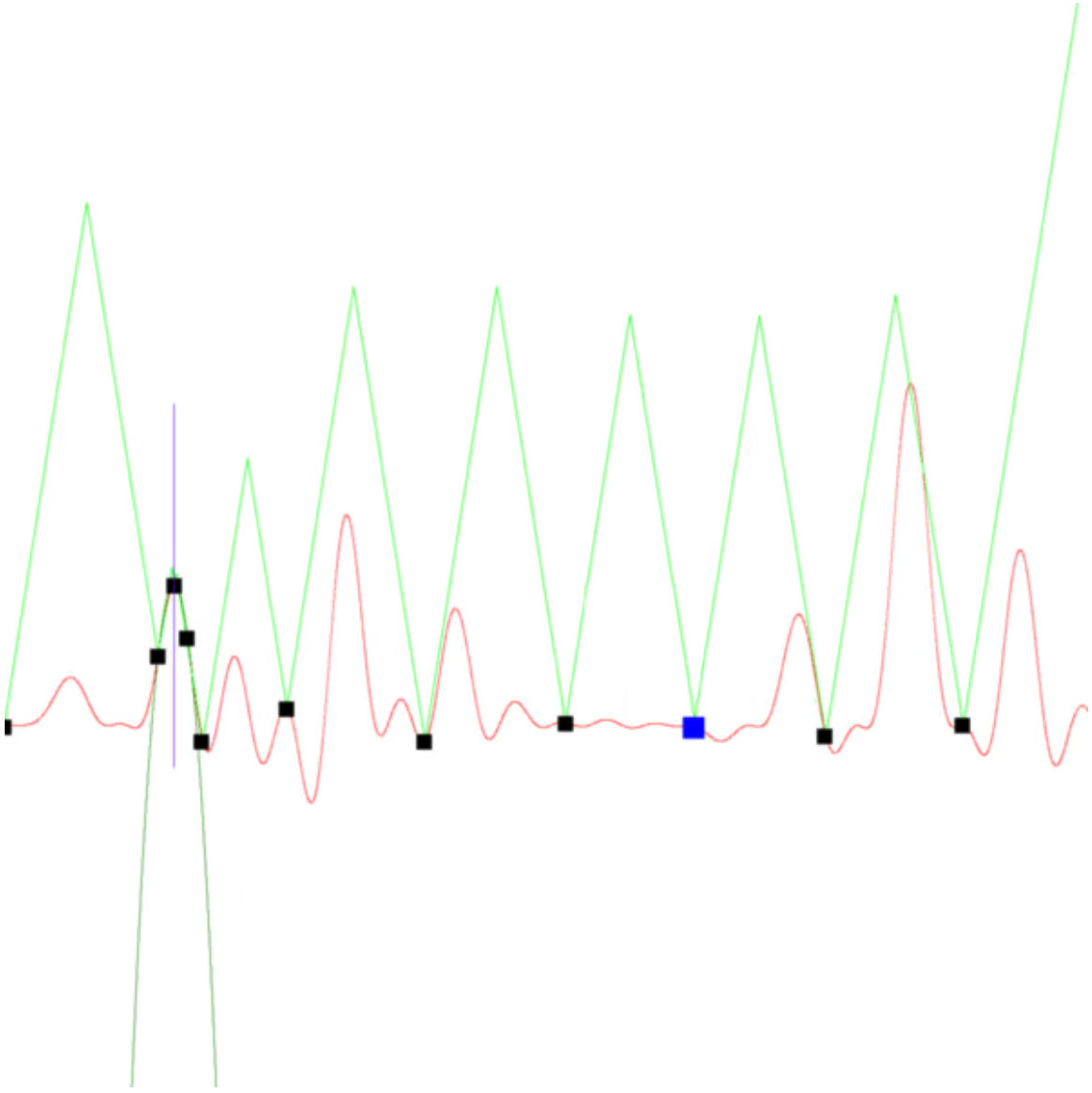
find_max_global() in action - 19

- Use upper bounding model



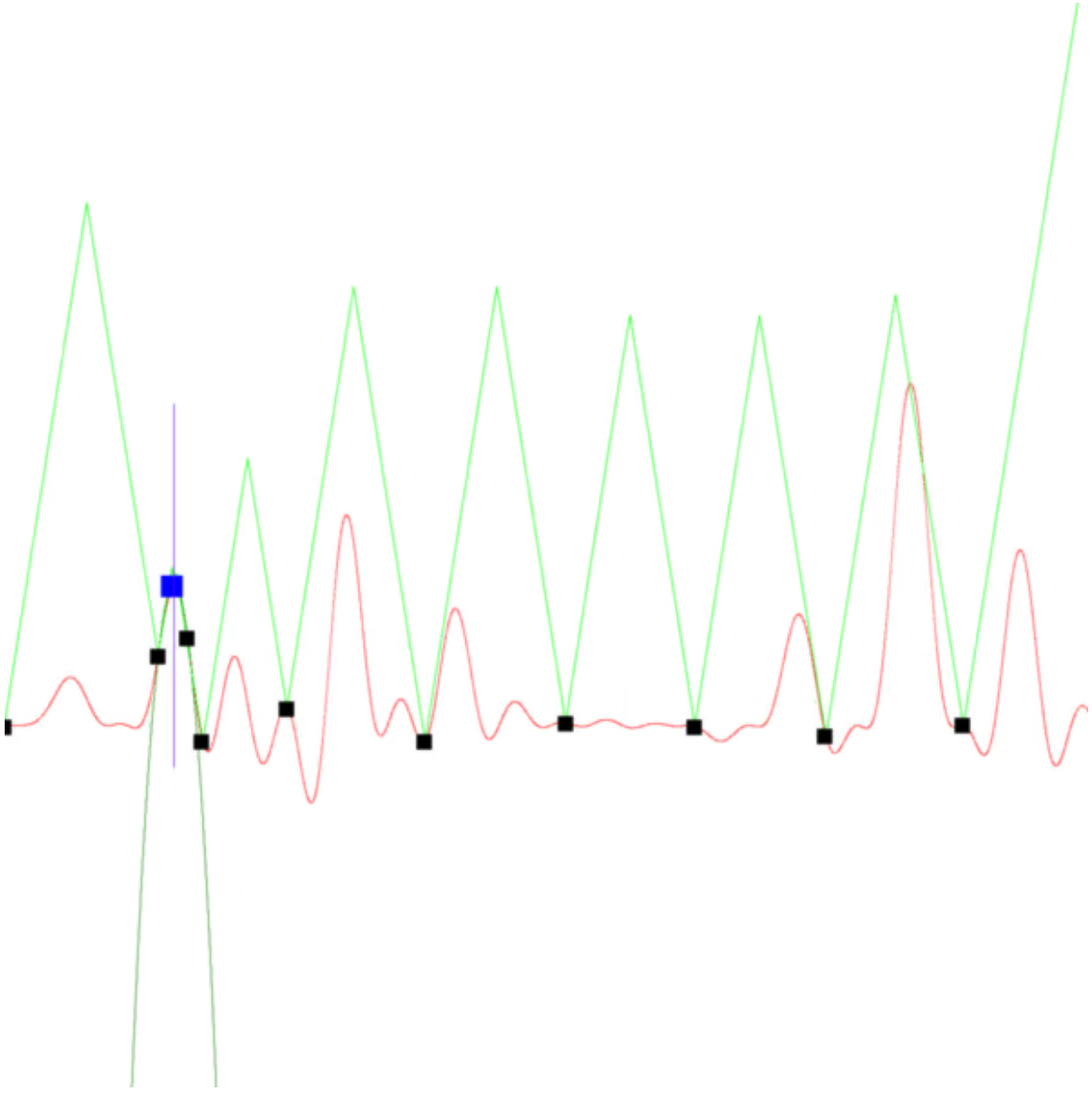
find_max_global() in action - 20

- Update models



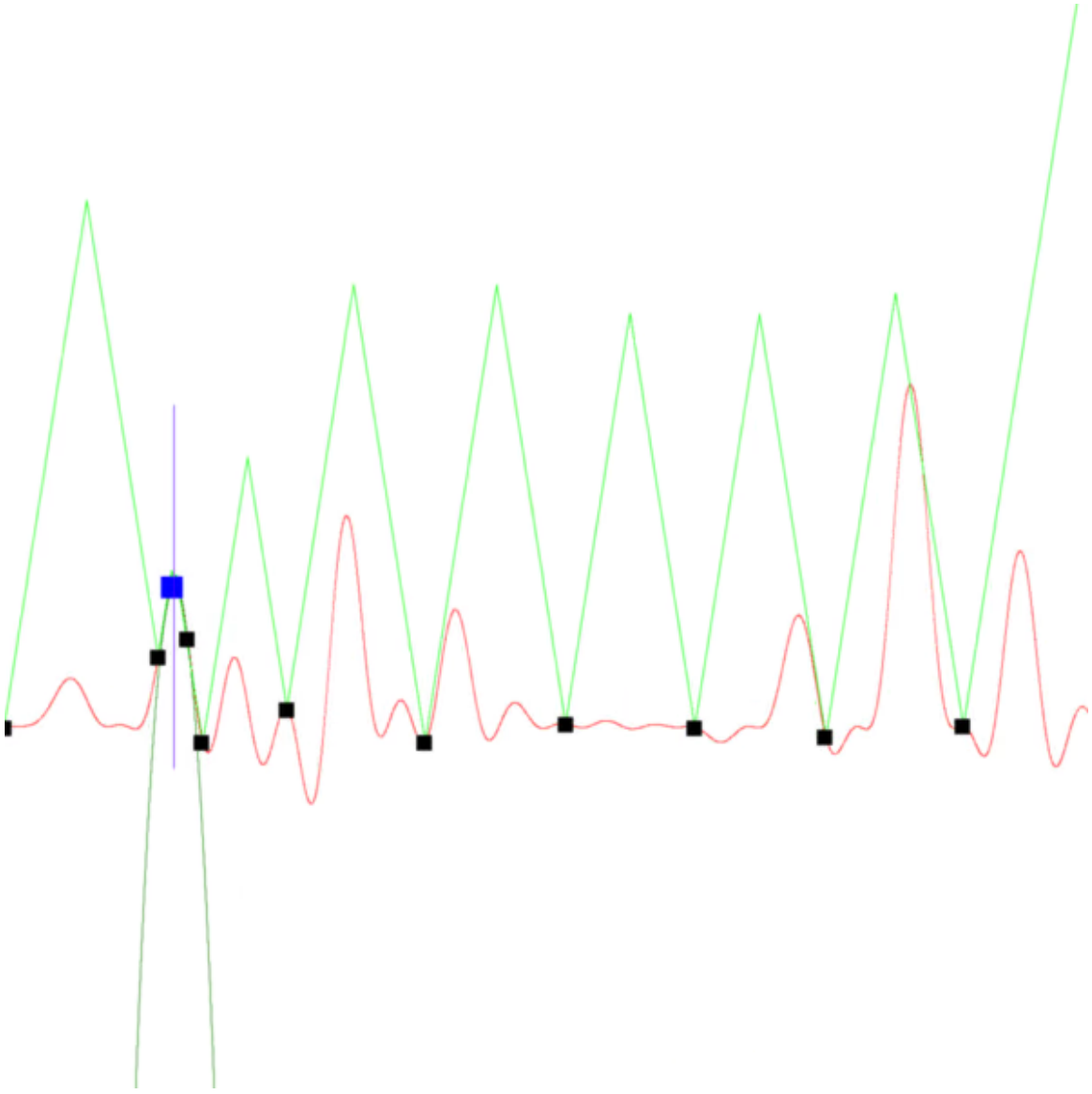
find_max_global() in action - 21

- Use trust region model



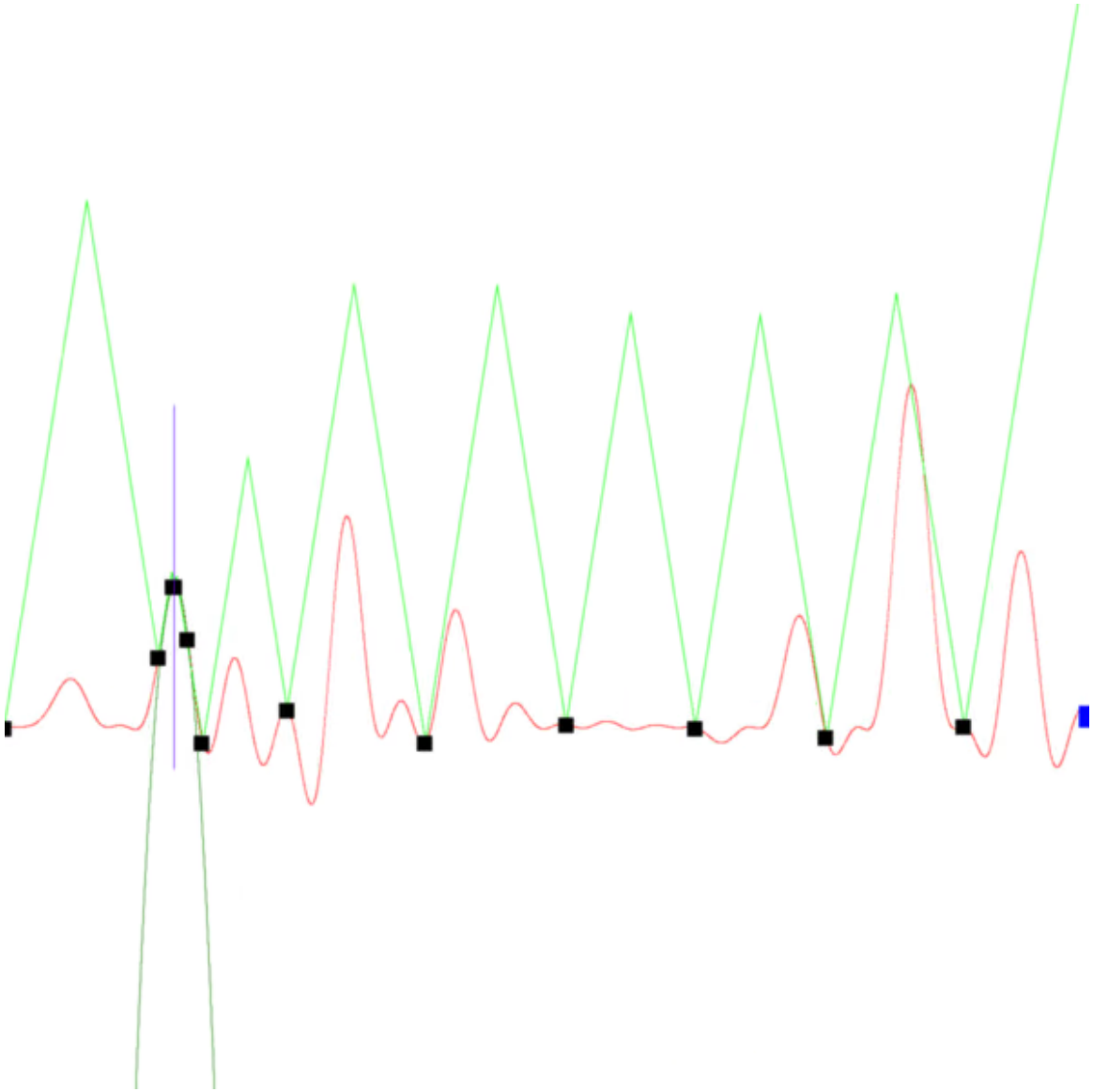
find_max_global() in action - 22

- Update models



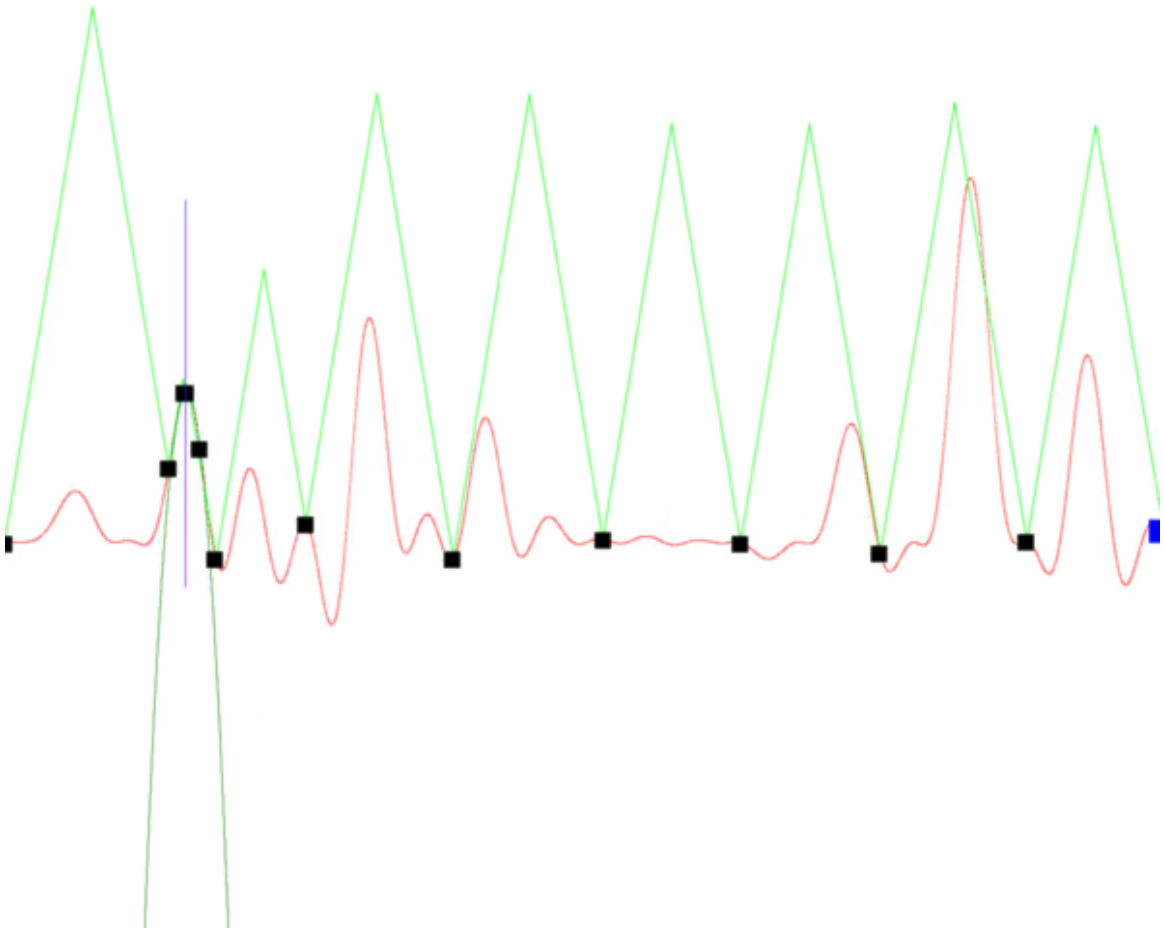
find_max_global() in action - 23

- Use upper bounding model



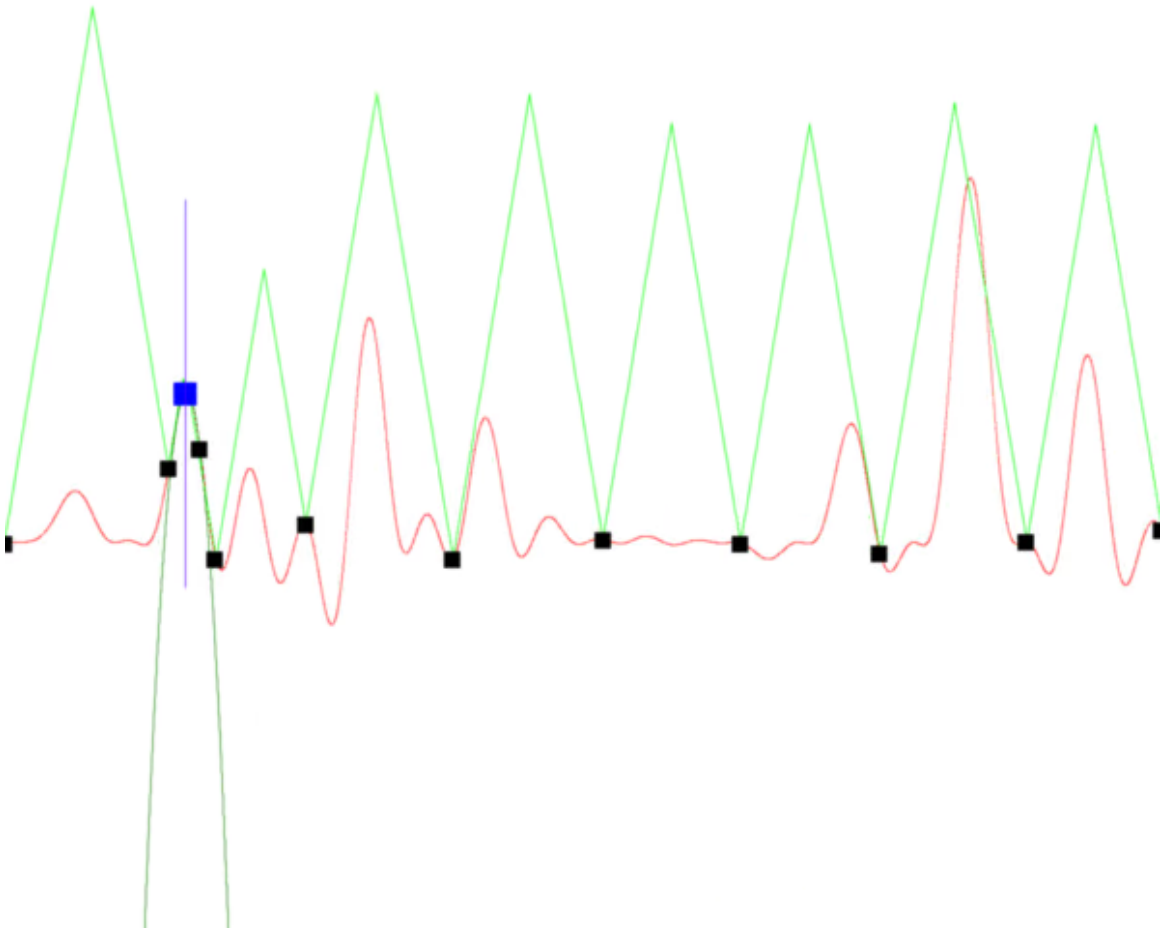
find_max_global() in action - 24

- Update models



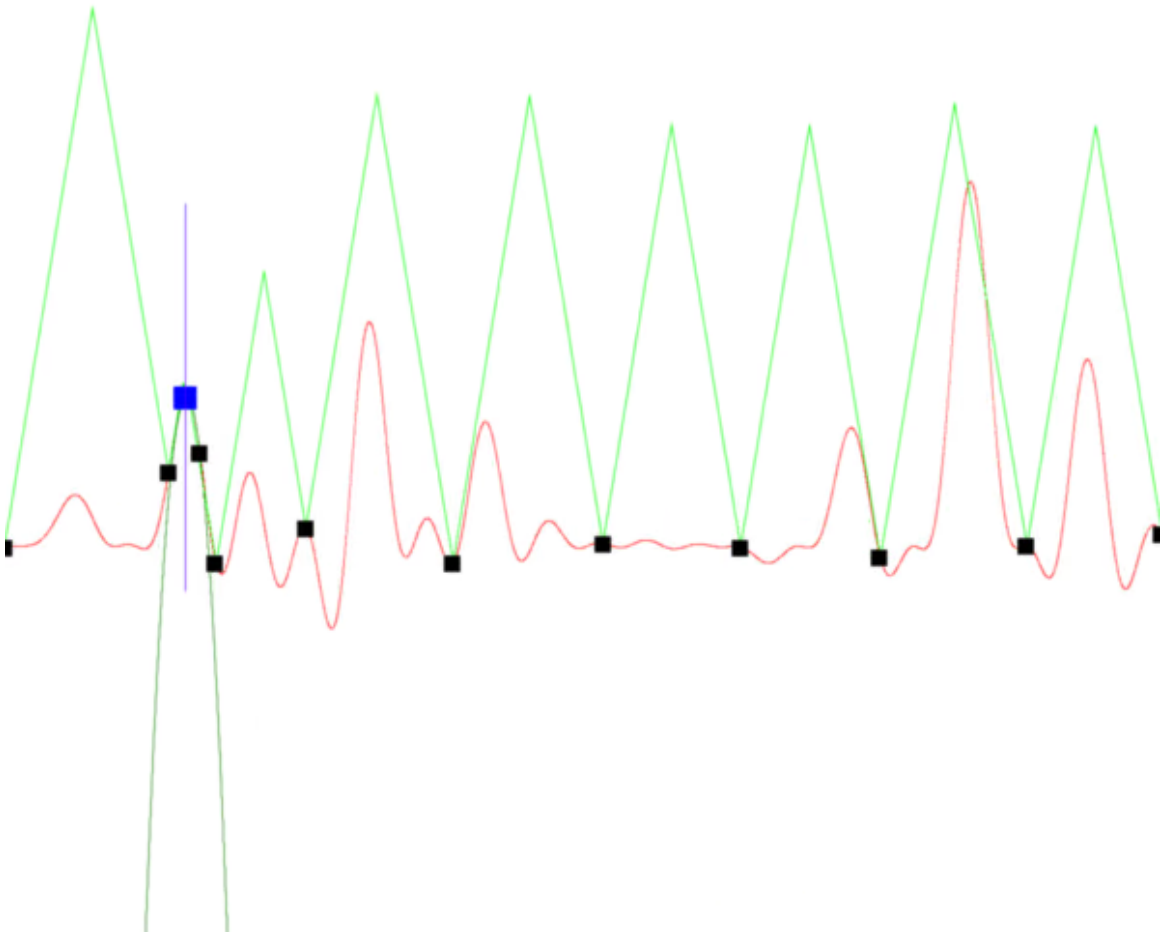
find_max_global() in action - 25

- Use trust region model



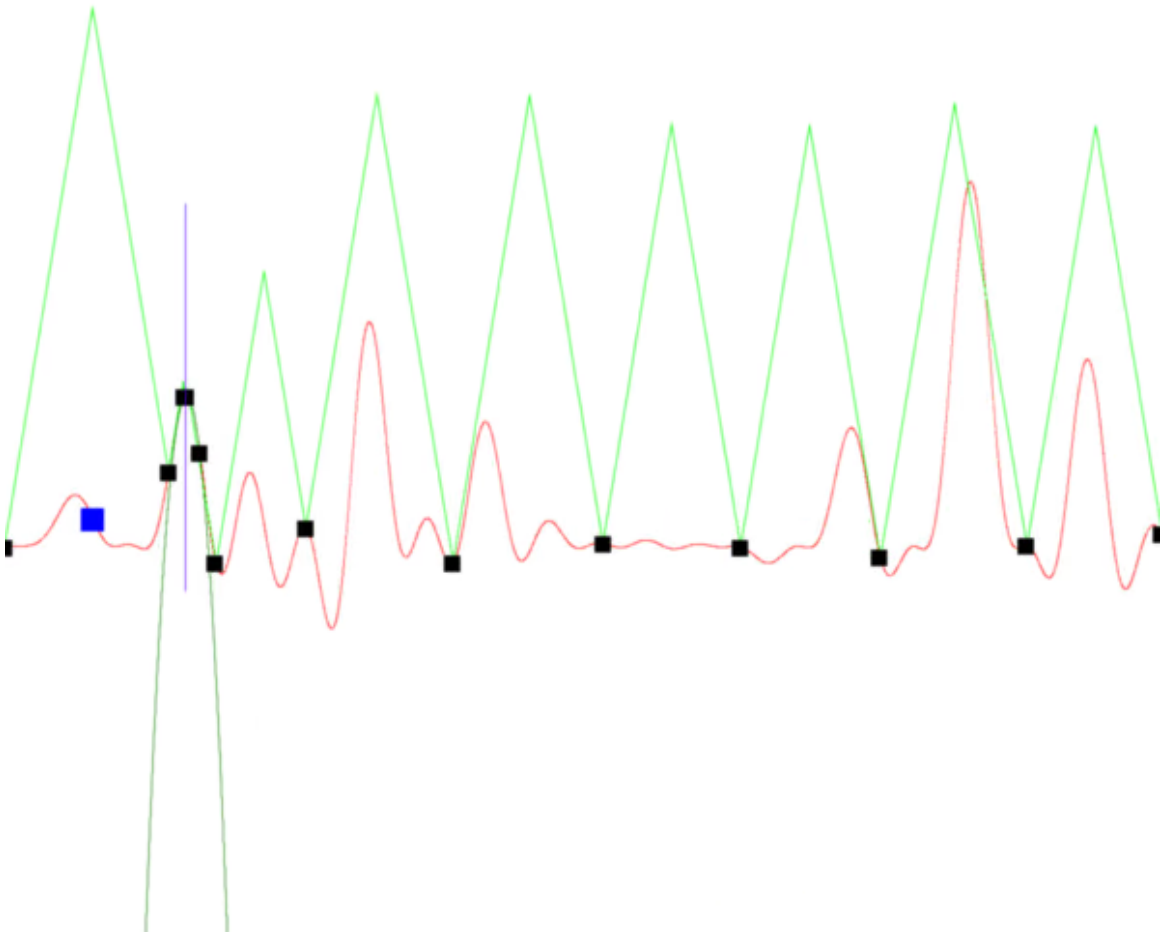
find_max_global() in action - 26

- Update models



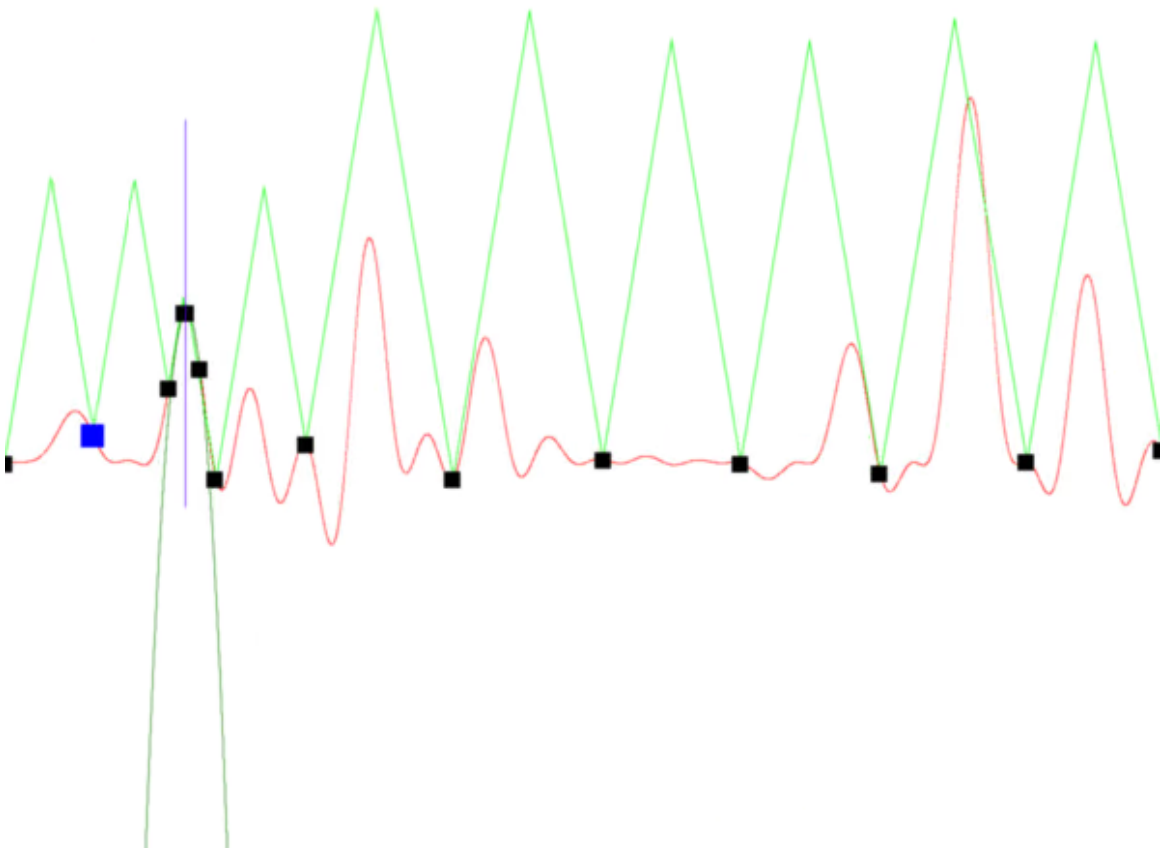
find_max_global() in action - 27

- Use upper bounding model



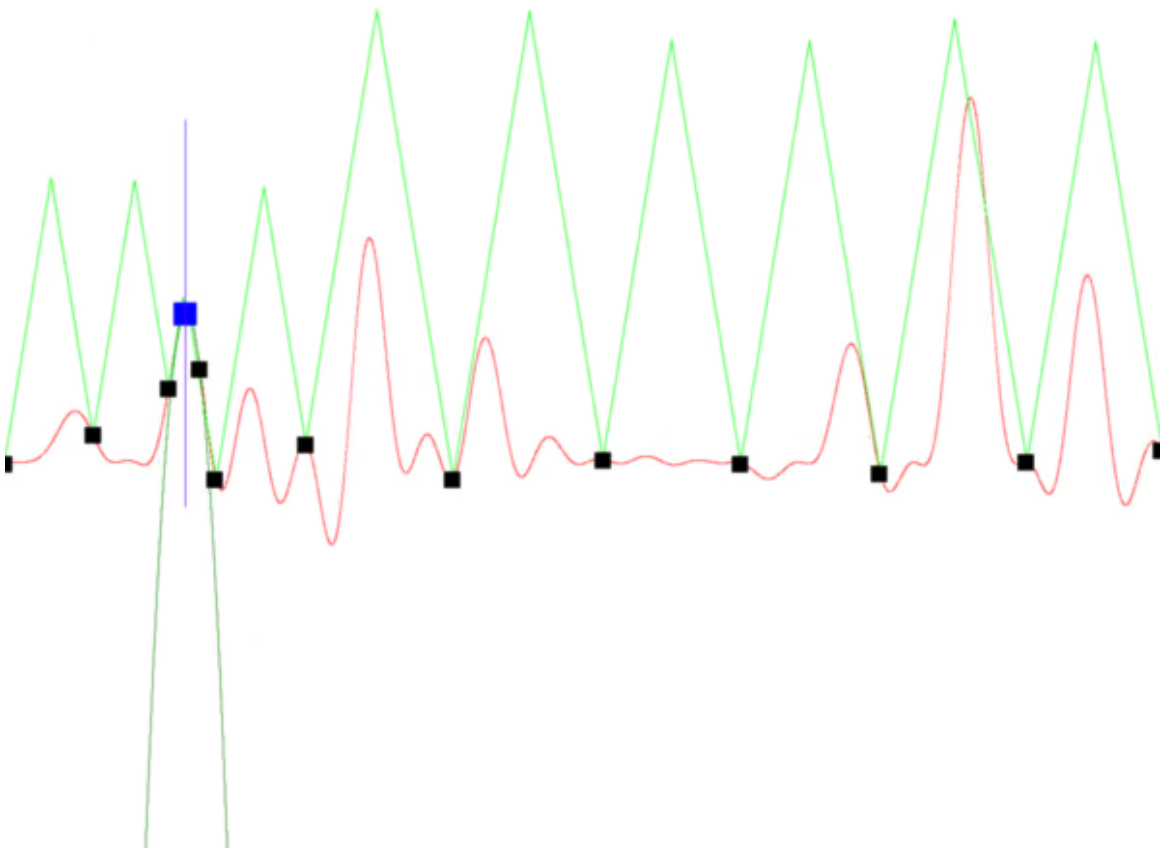
find_max_global() in action - 28

- Update models



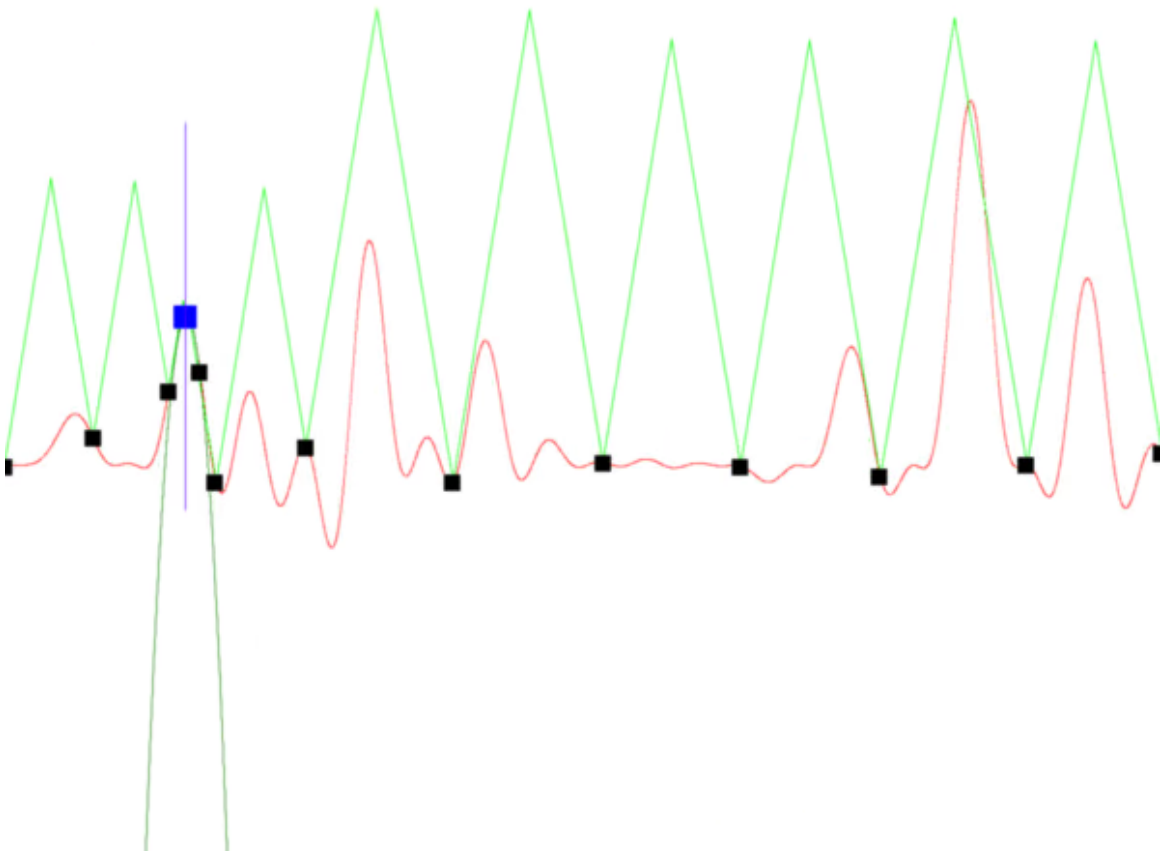
find_max_global() in action - 29

- Use trust region model



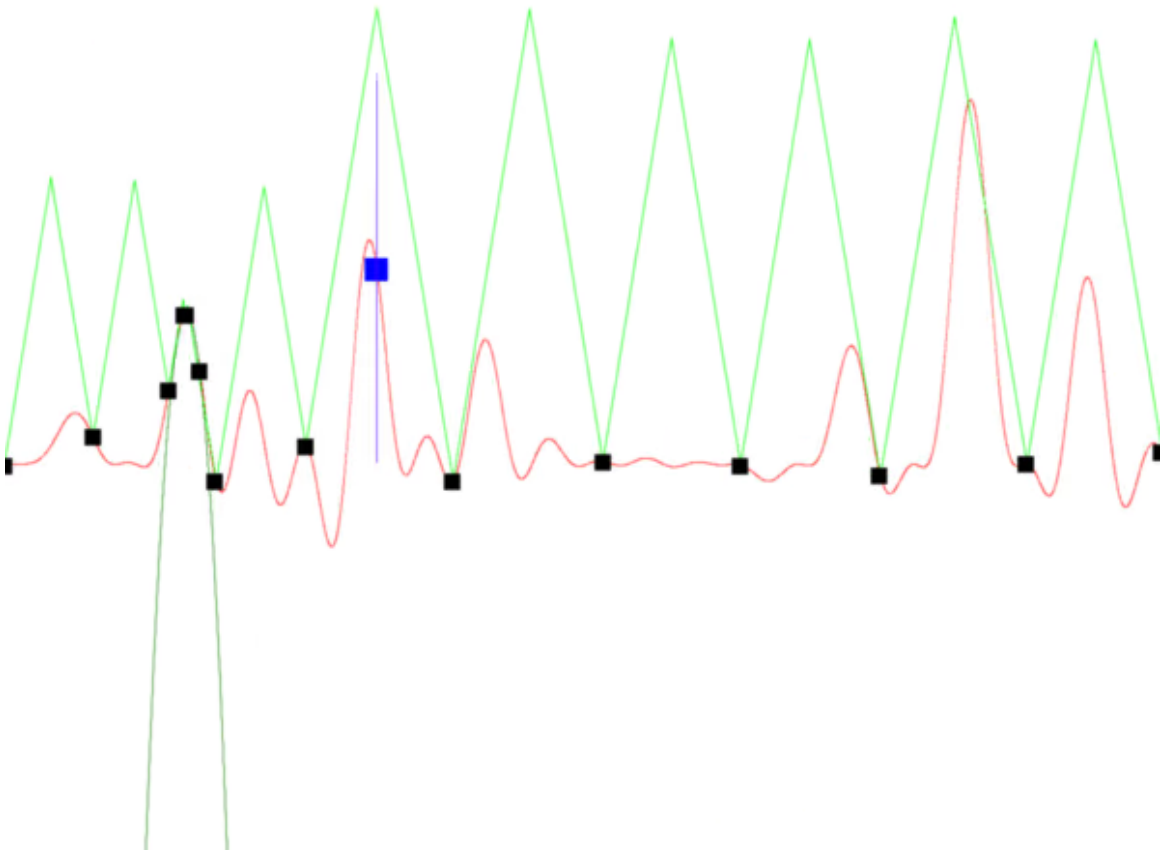
find_max_global() in action - 30

- Update models



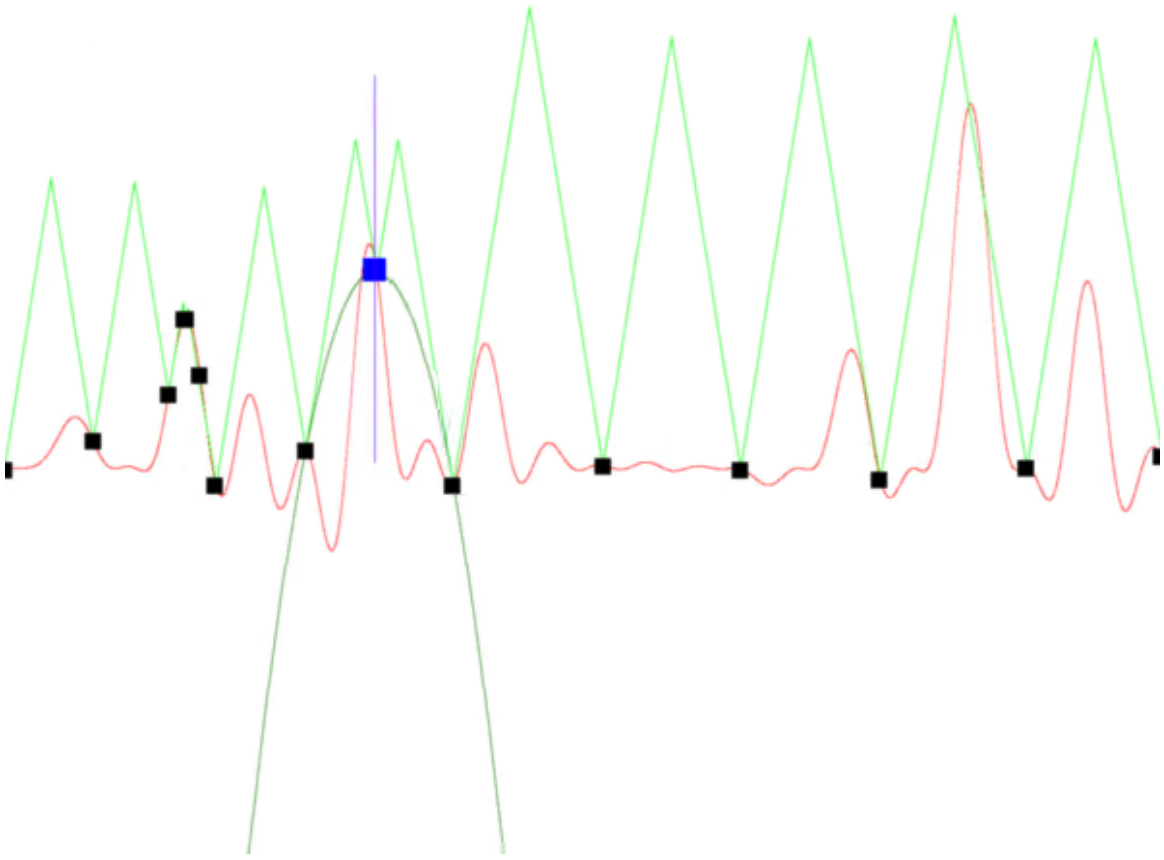
find_max_global() in action - 31

- Use upper bounding model



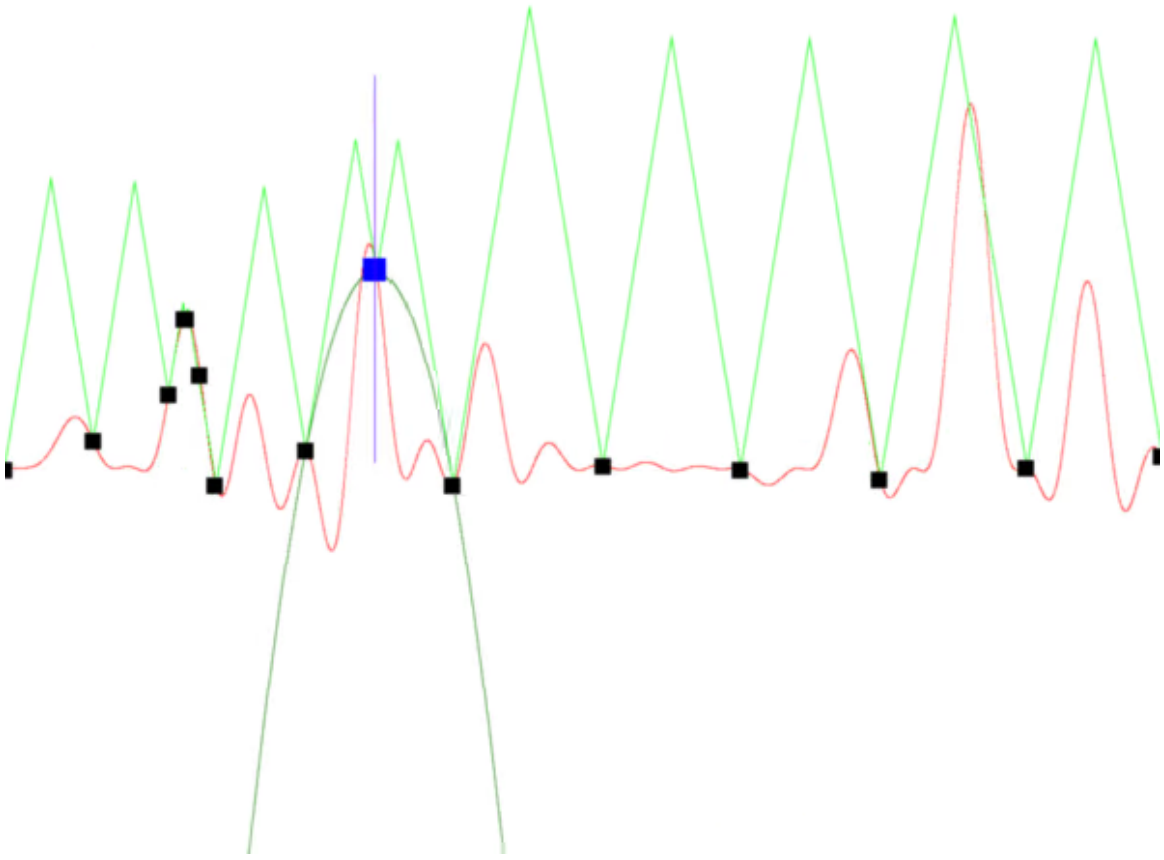
find_max_global() in action - 32

- Update models



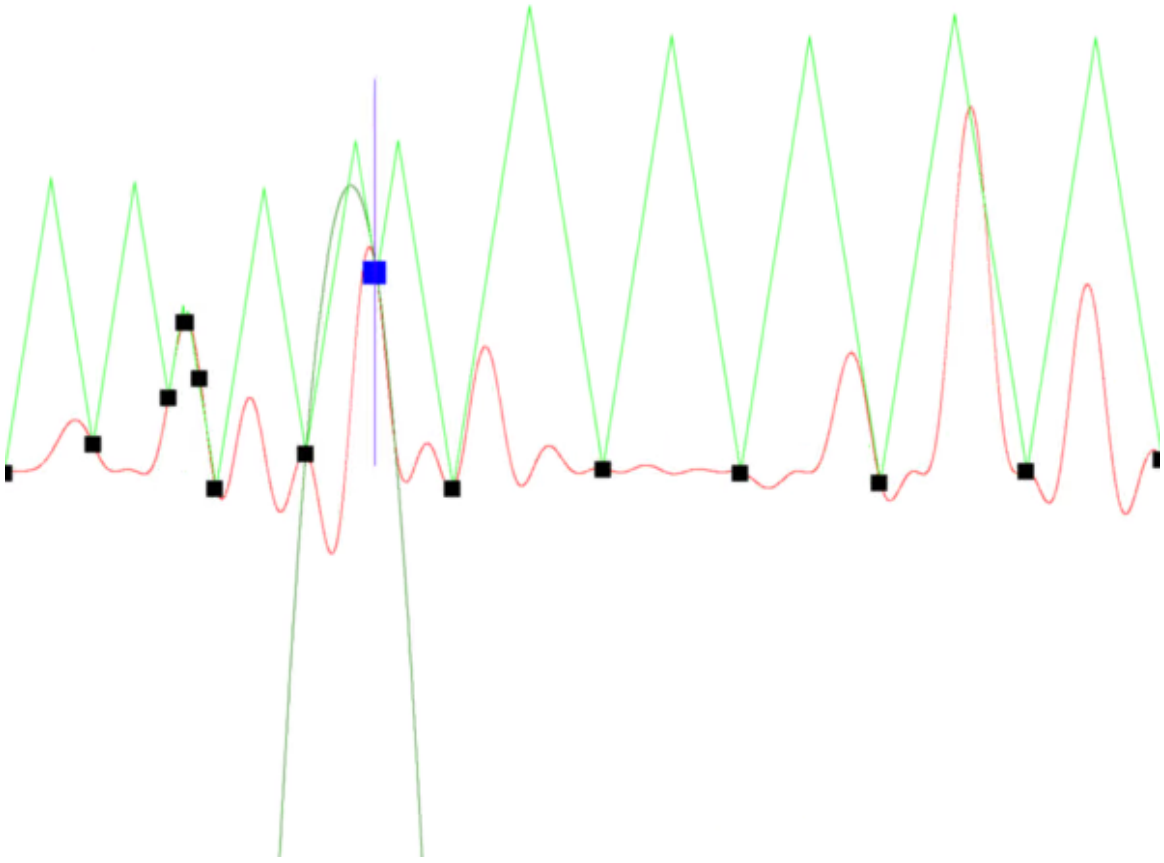
find_max_global() in action - 33

- Use trust region model



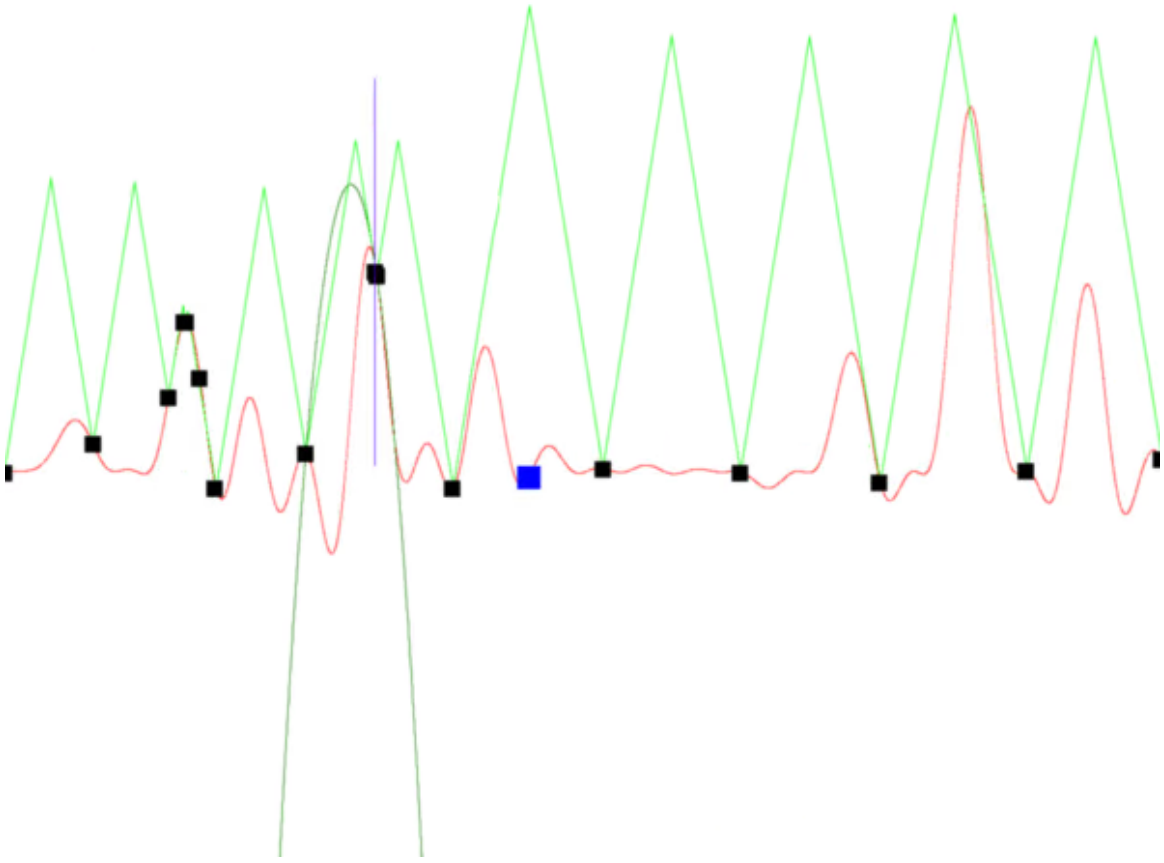
find_max_global() in action - 34

- Update models



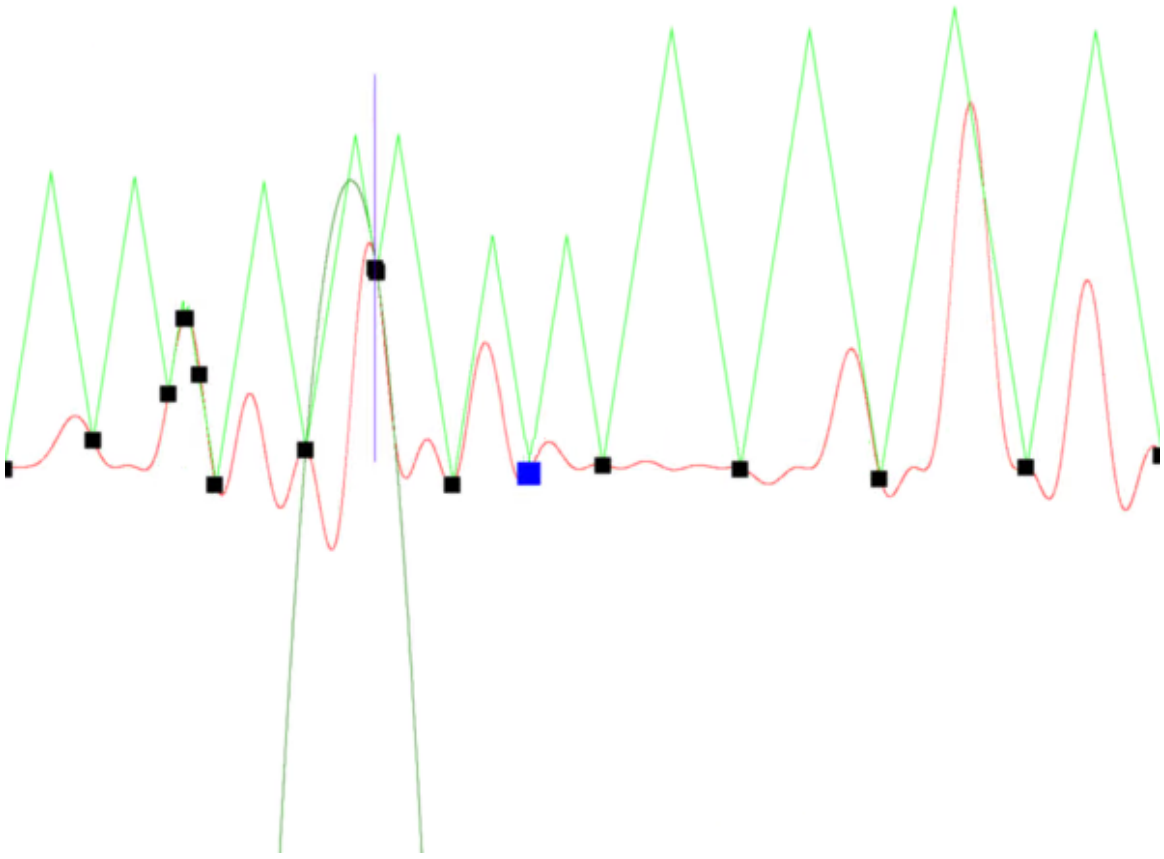
find_max_global() in action - 35

- Use upper bounding model



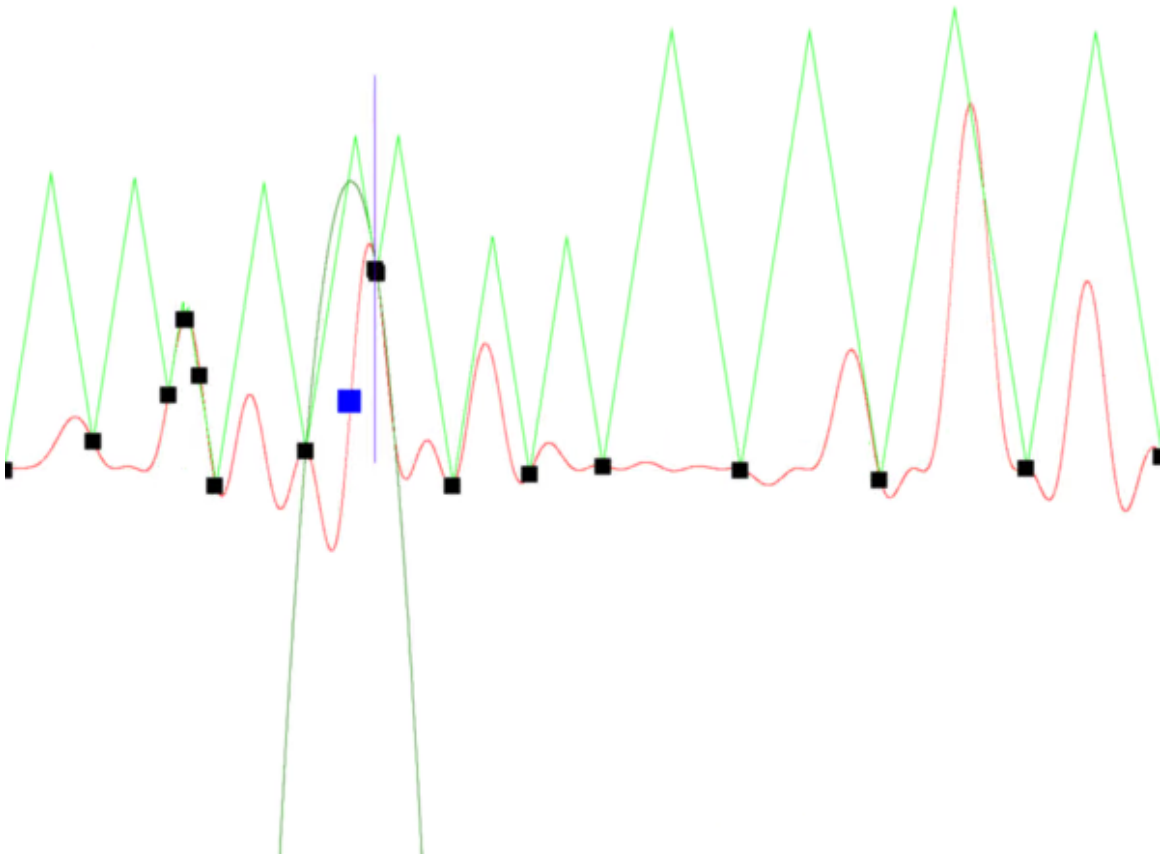
find_max_global() in action - 36

- Update models



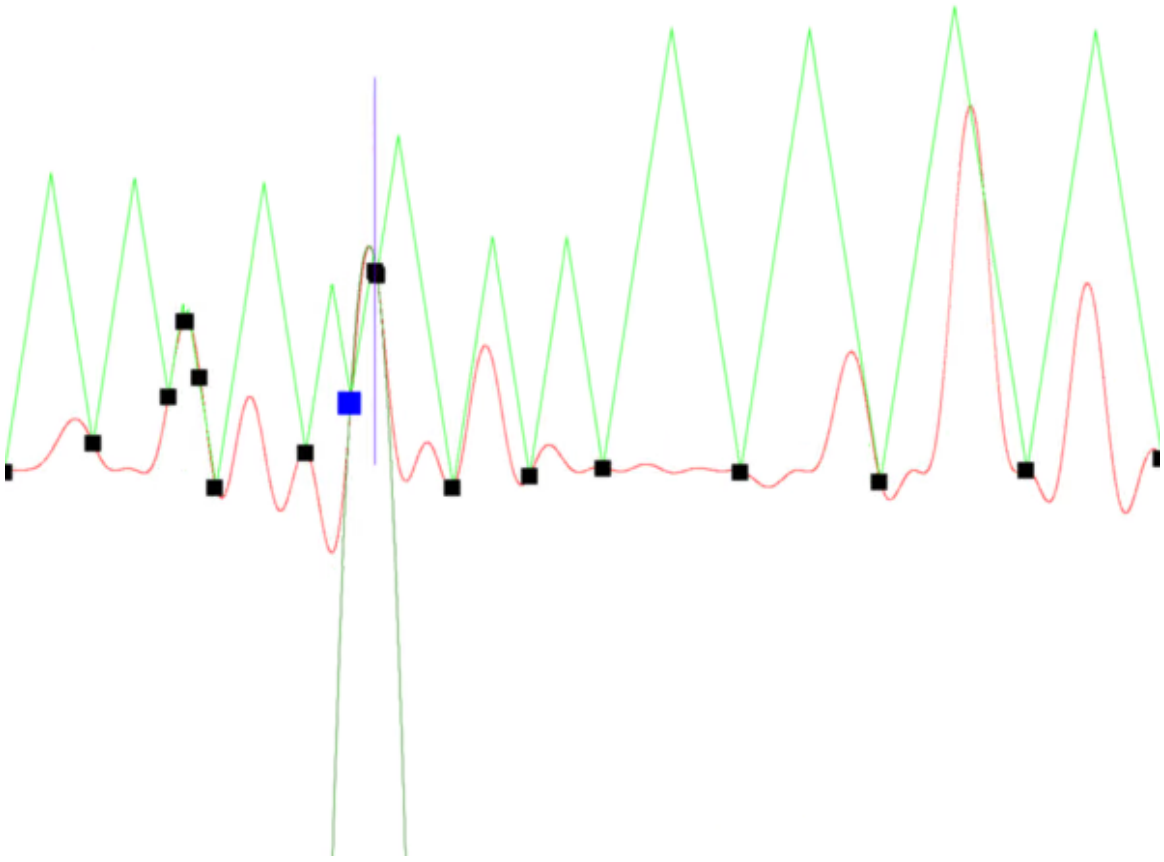
find_max_global() in action - 37

- Use trust region model



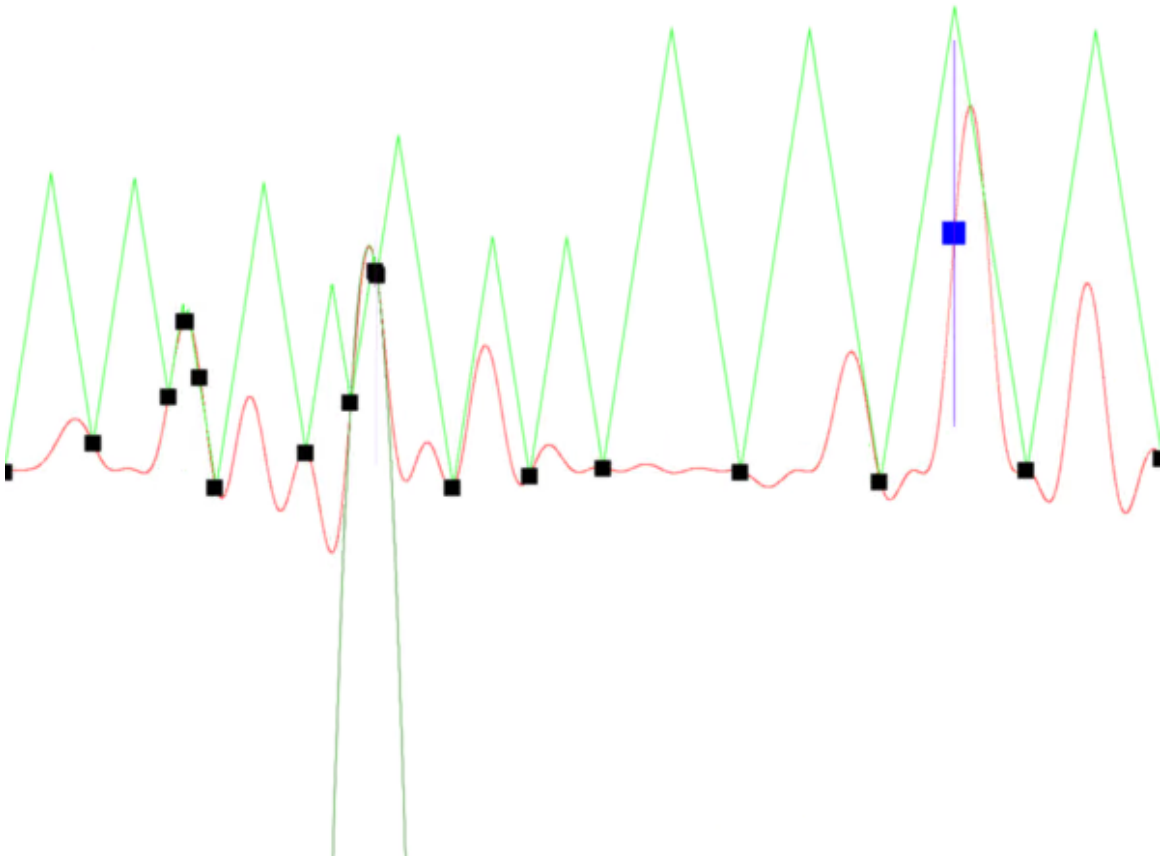
find_max_global() in action - 38

- Update models



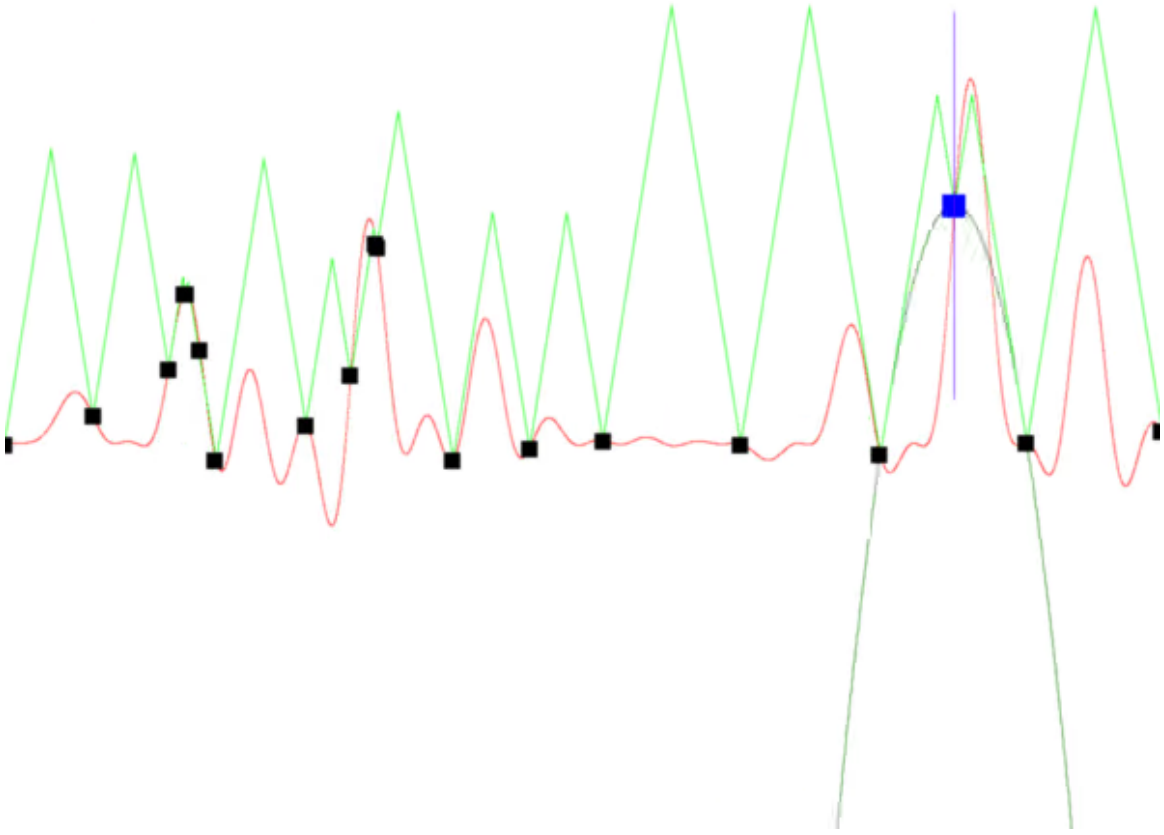
find_max_global() in action - 39

- Use upper bounding model



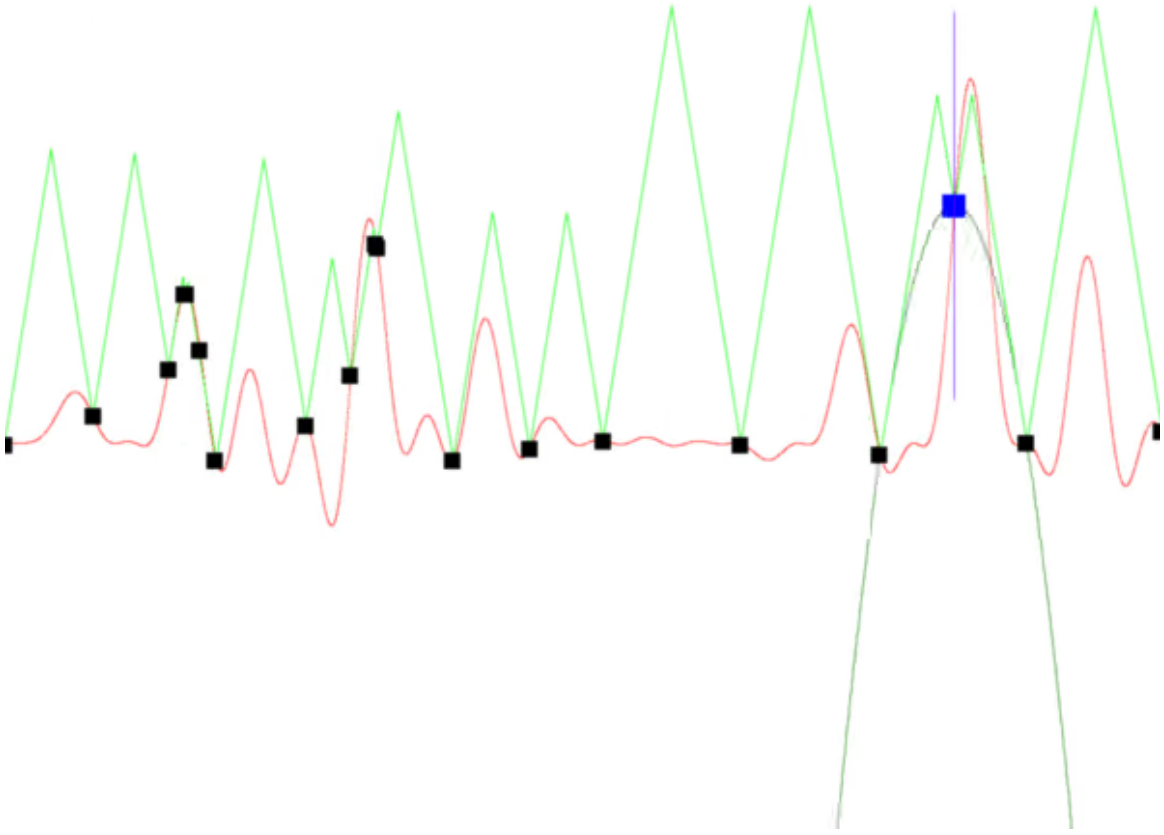
find_max_global() in action - 40

- Update models



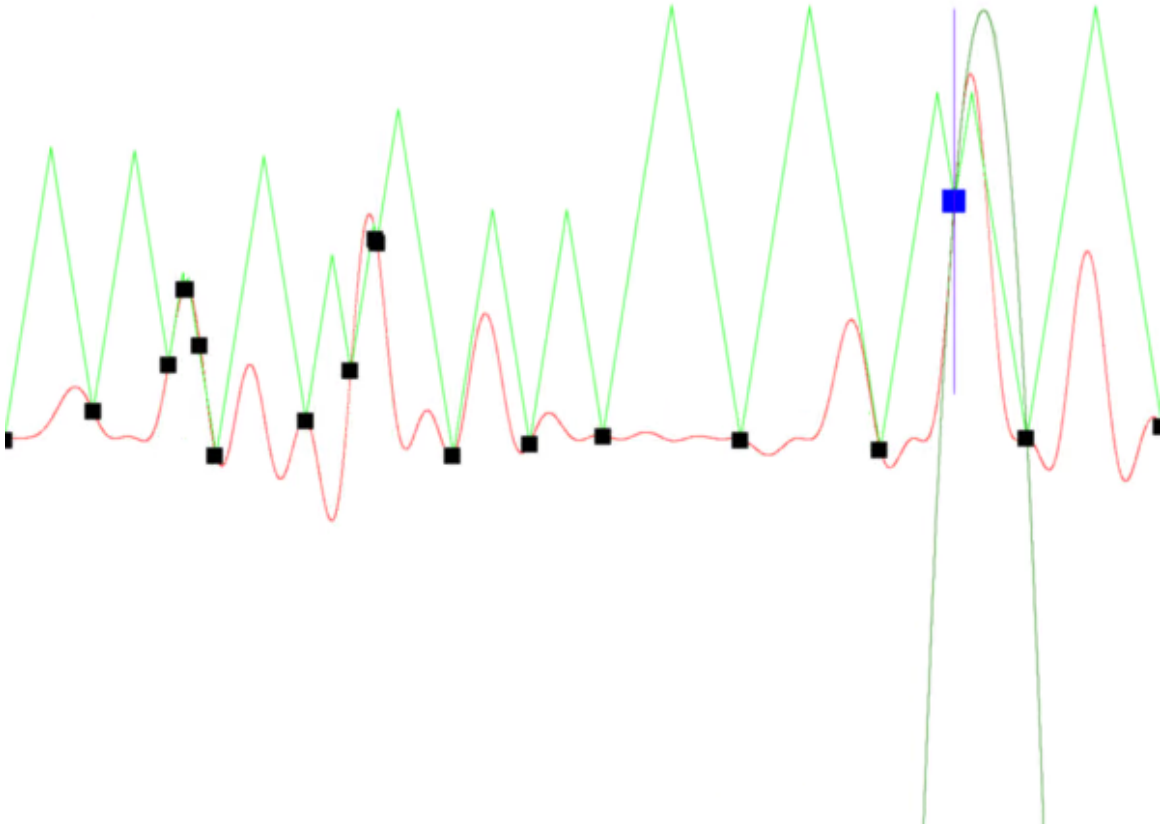
find_max_global() in action - 41

- Use trust region model



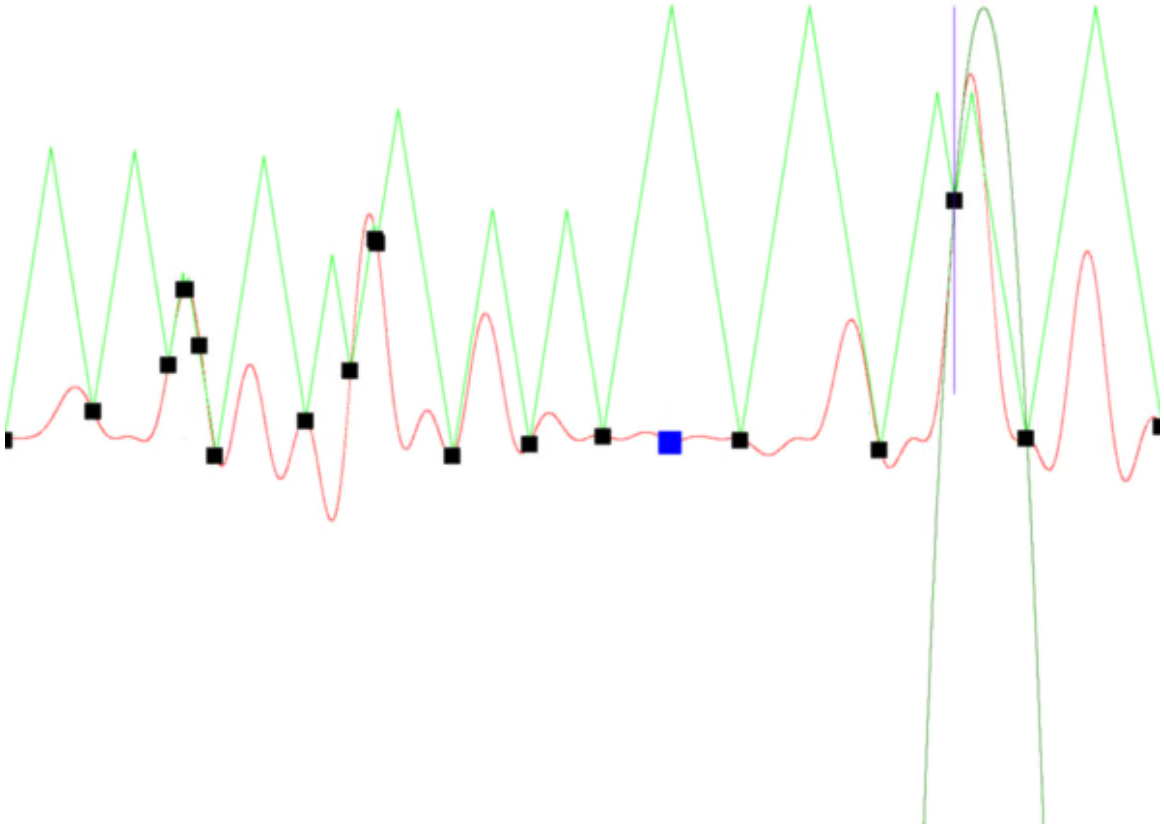
find_max_global() in action - 42

- Update models



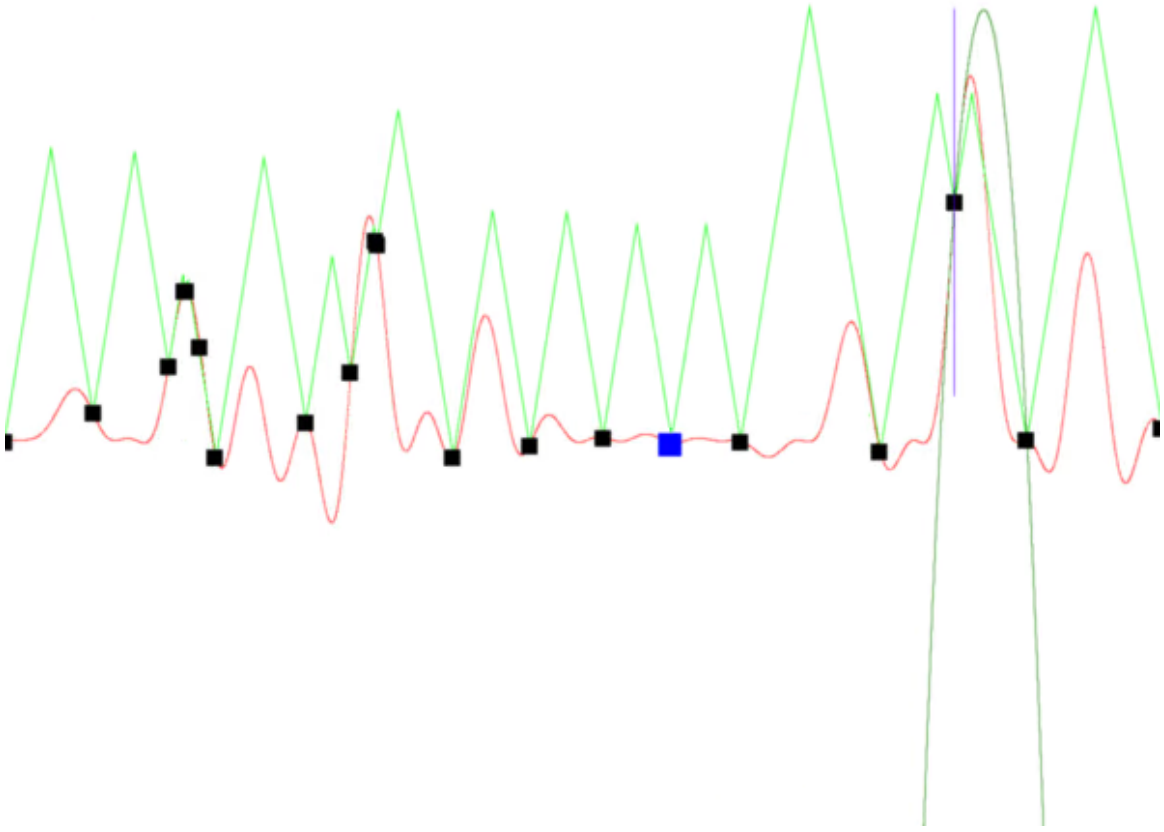
find_max_global() in action - 43

- Use upper bounding model



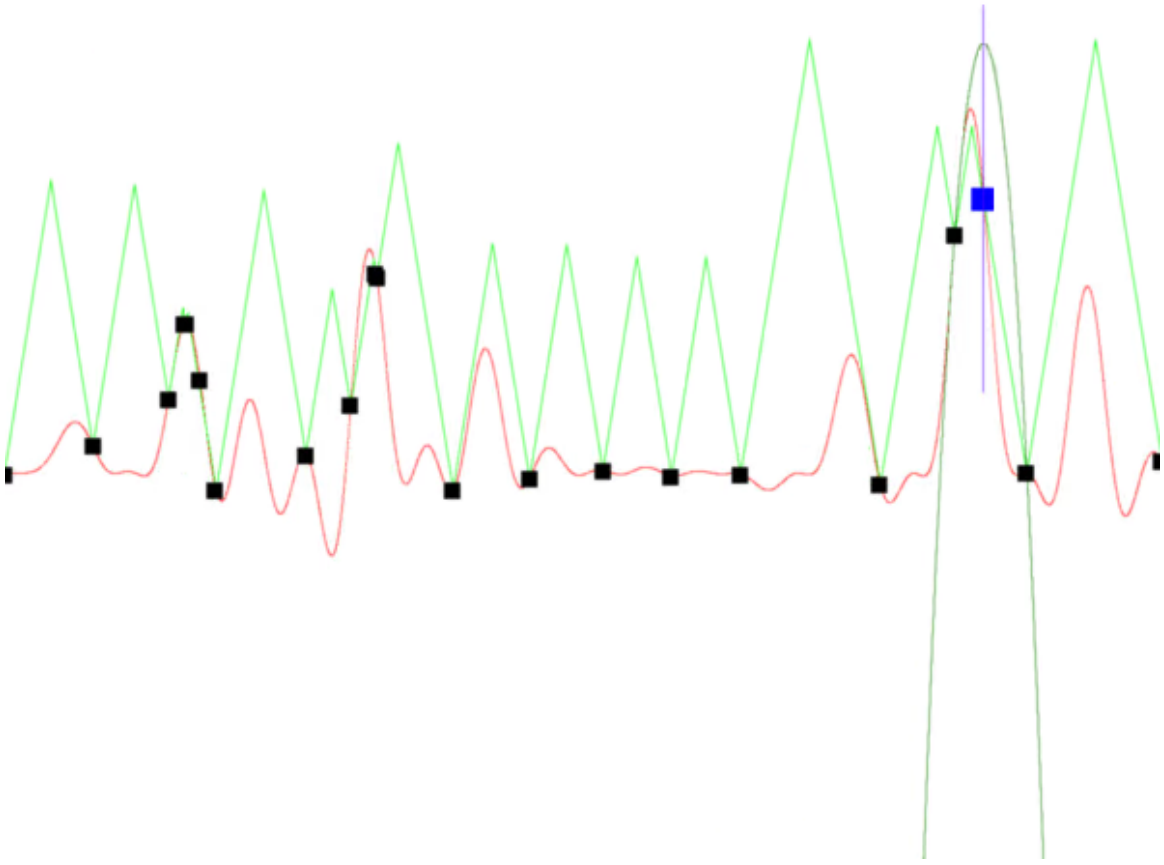
find_max_global() in action - 44

- Update models



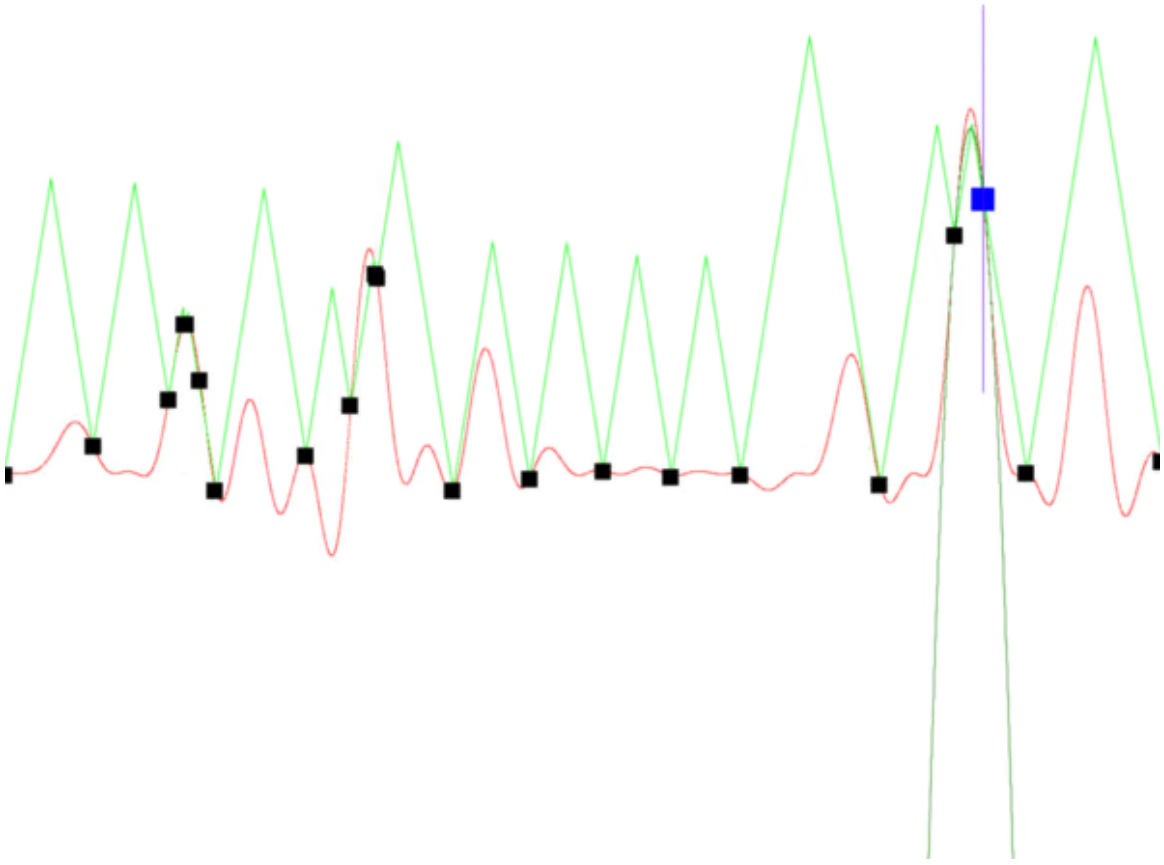
find_max_global() in action - 45

- Use trust region model



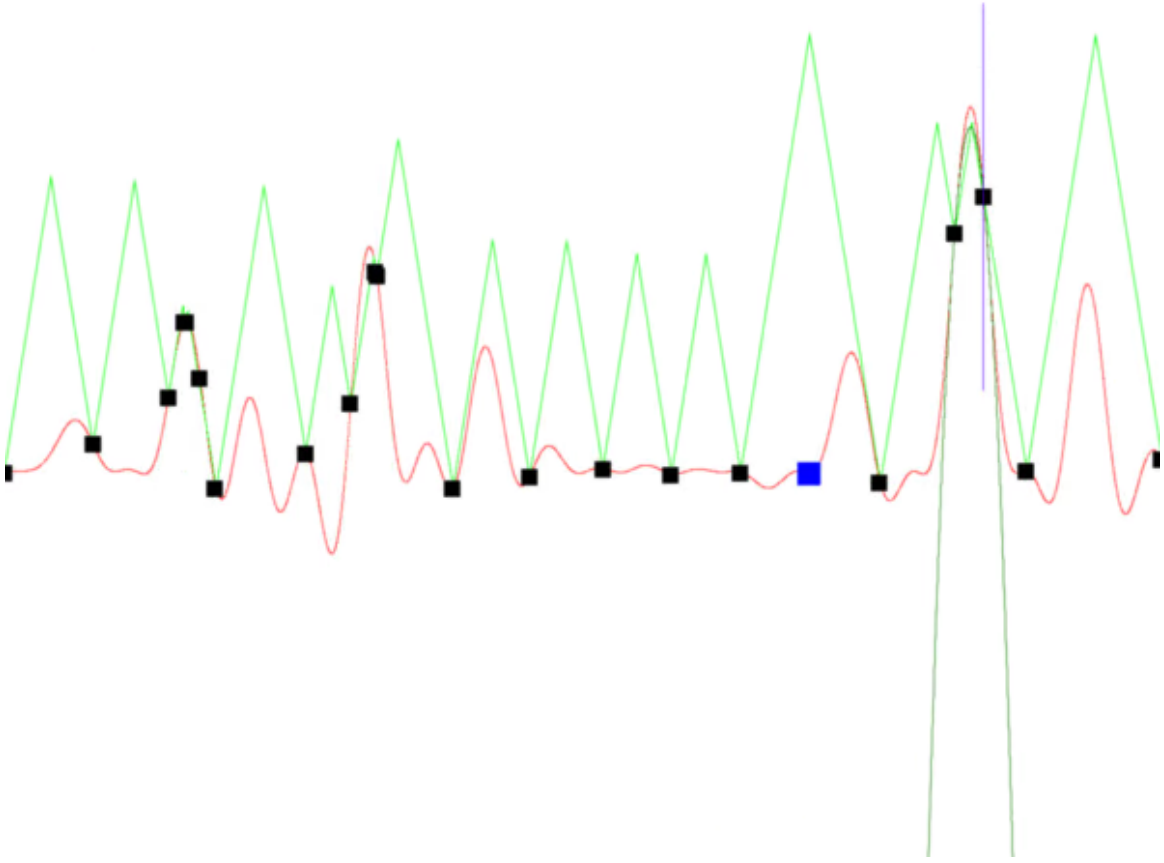
find_max_global() in action - 46

- Update models



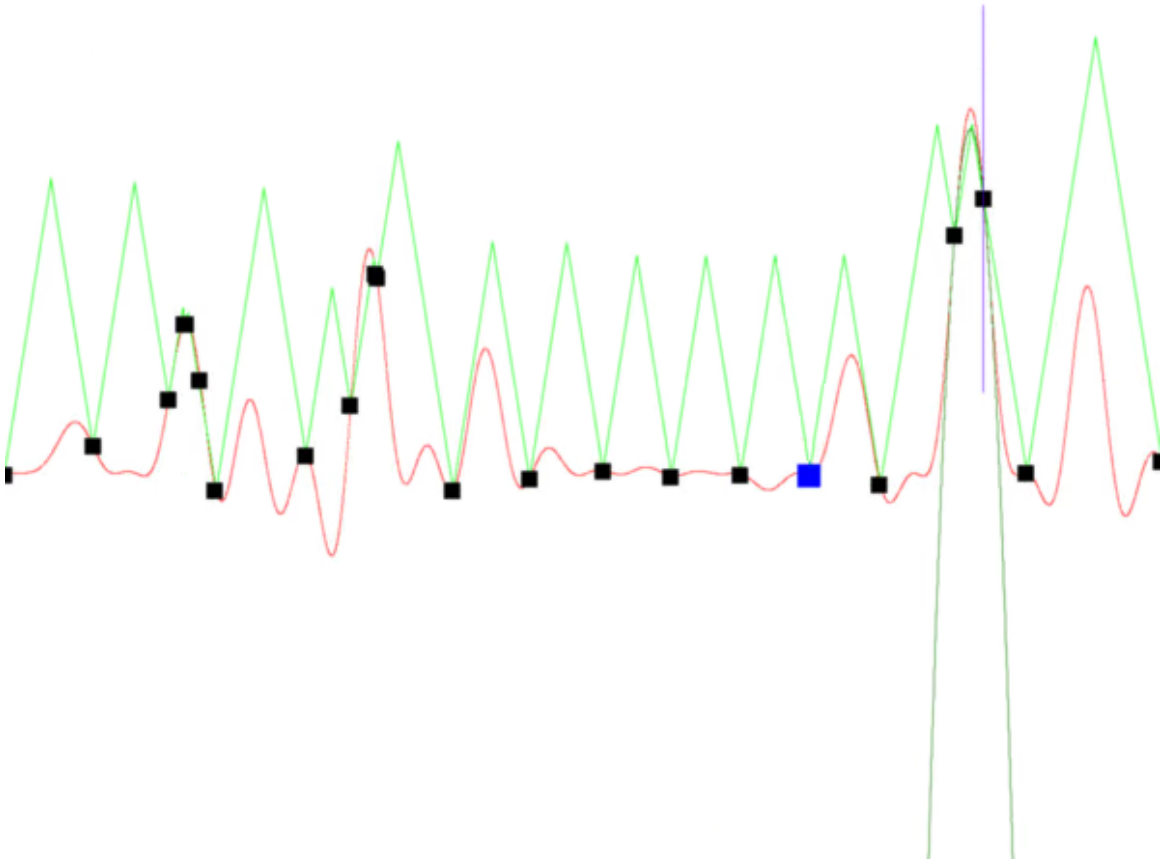
find_max_global() in action - 47

- Use upper bounding model



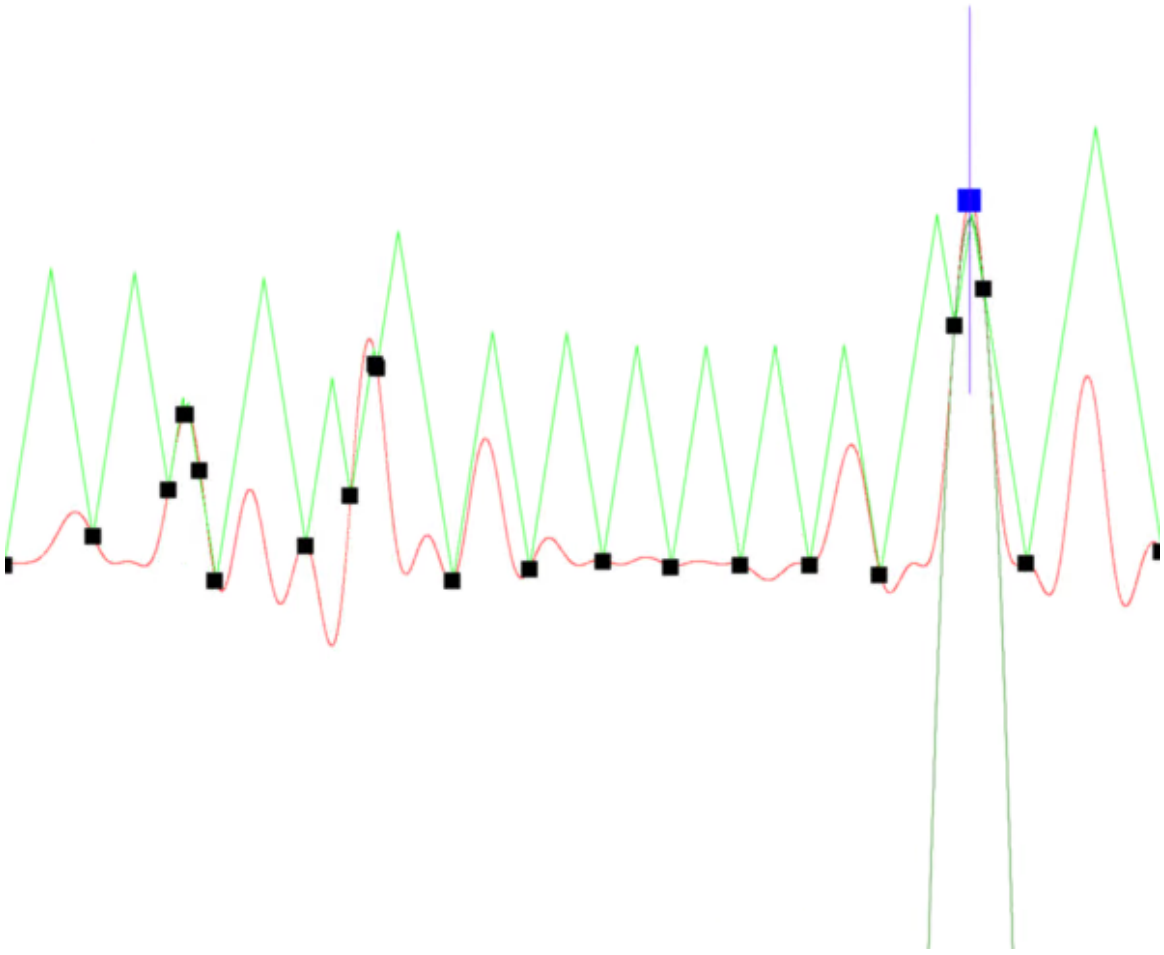
find_max_global() in action - 48

- Update models



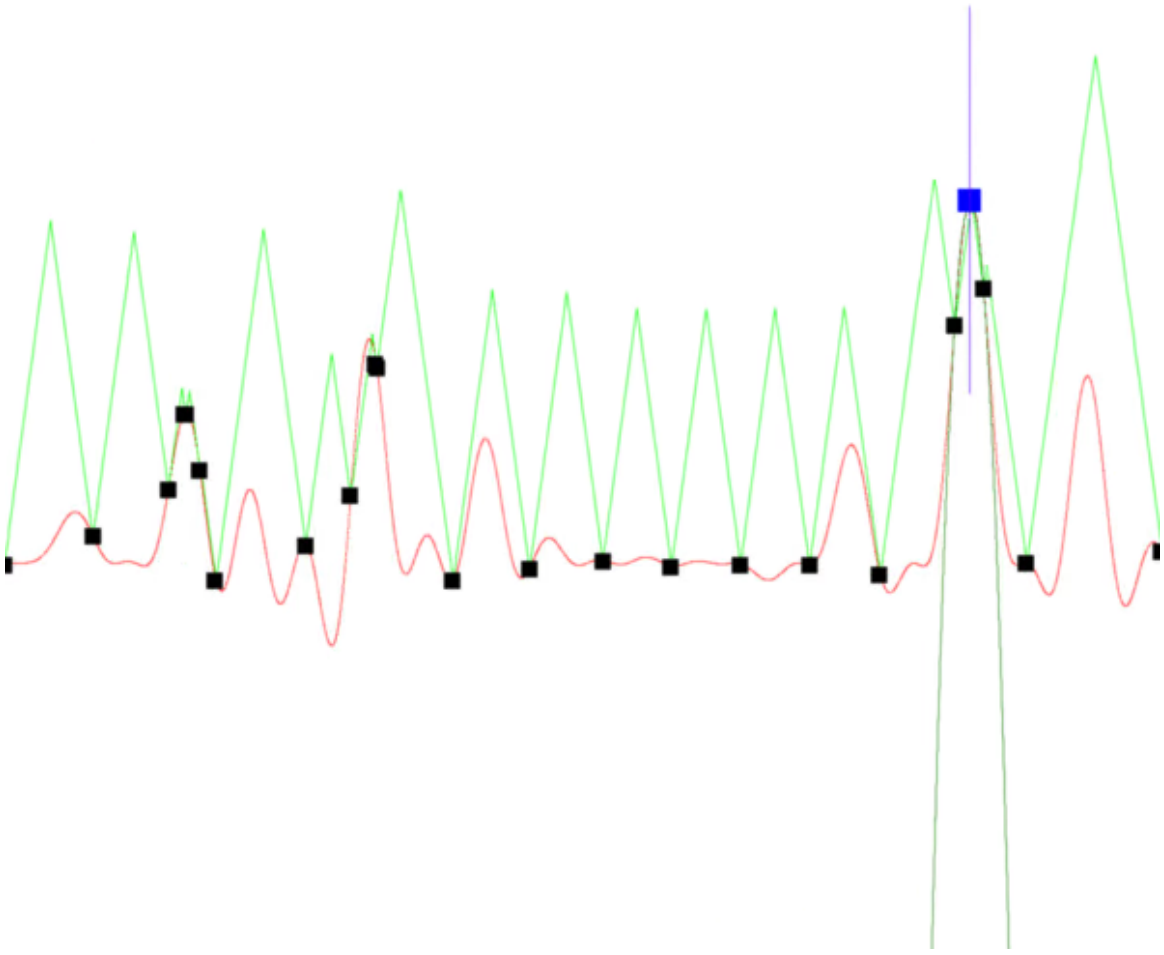
find_max_global() in action - 49

- Use trust region model



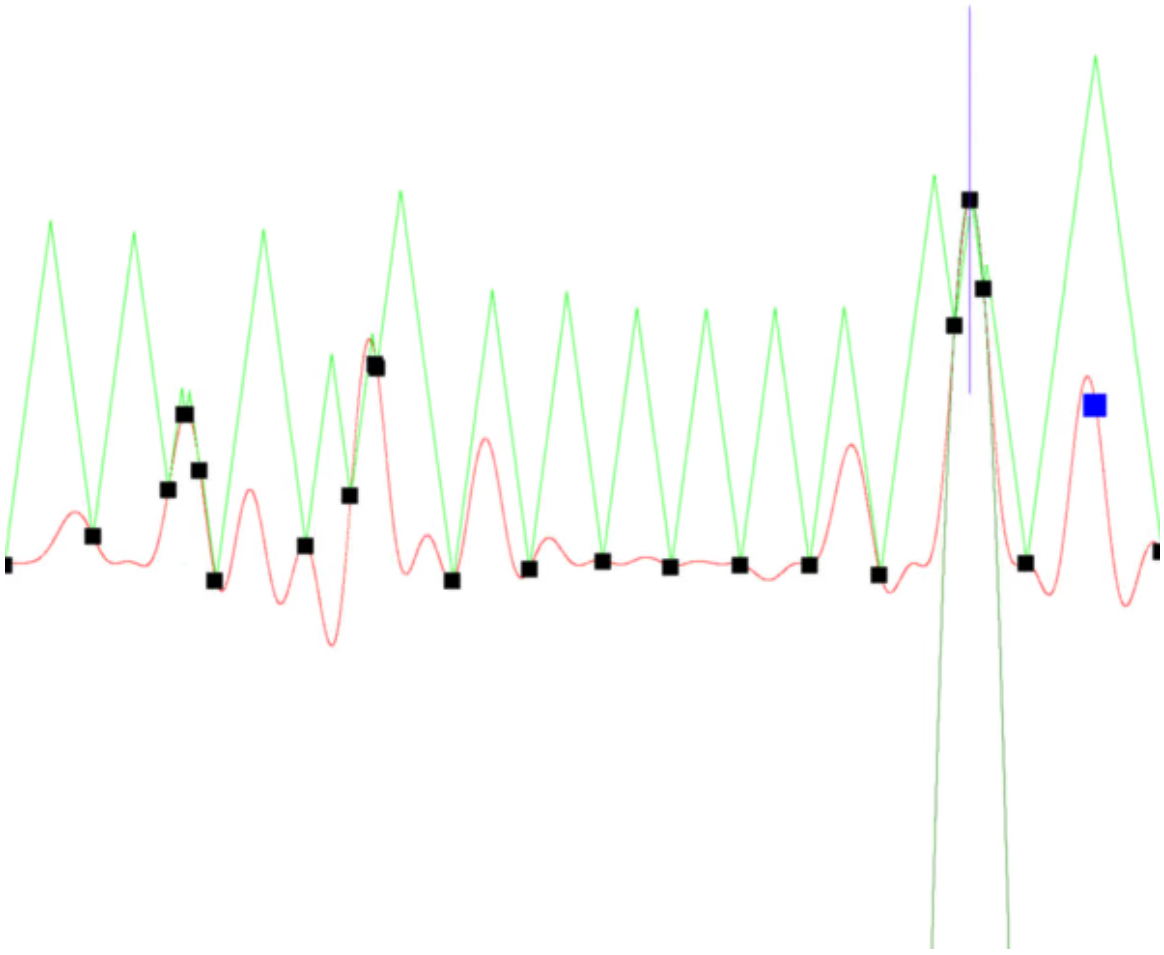
find_max_global() in action - 50

- Update models



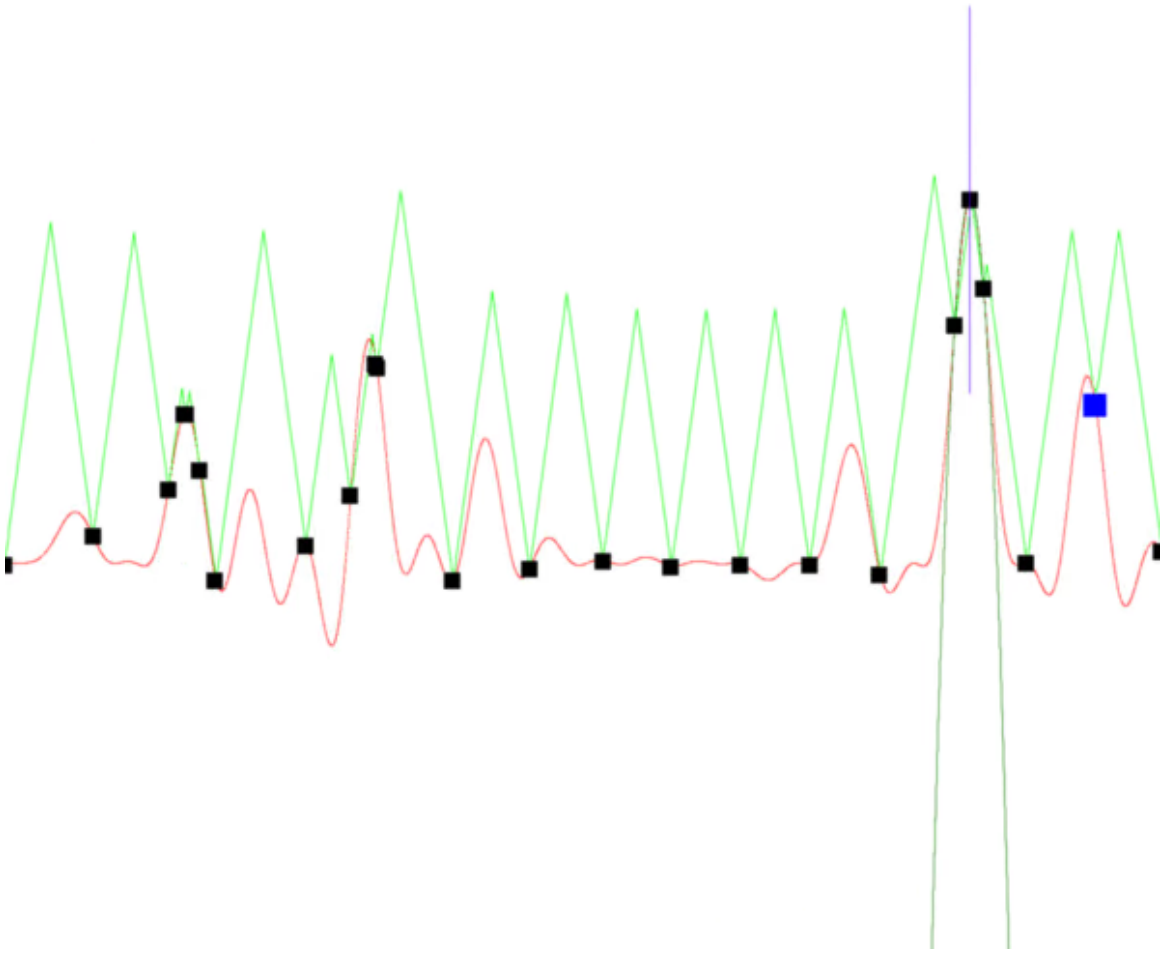
find_max_global() in action - 51

- Use upper bounding model



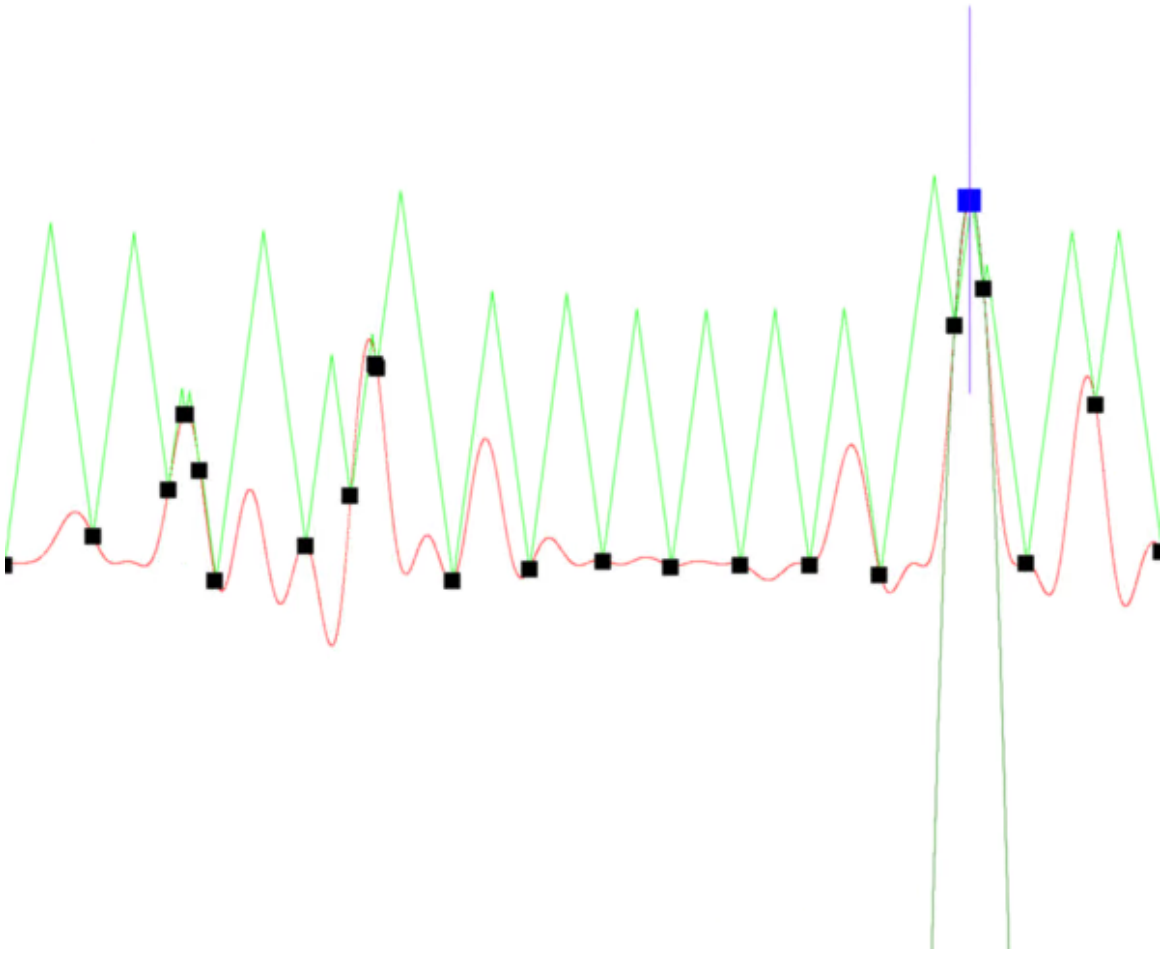
find_max_global() in action - 52

- Update models



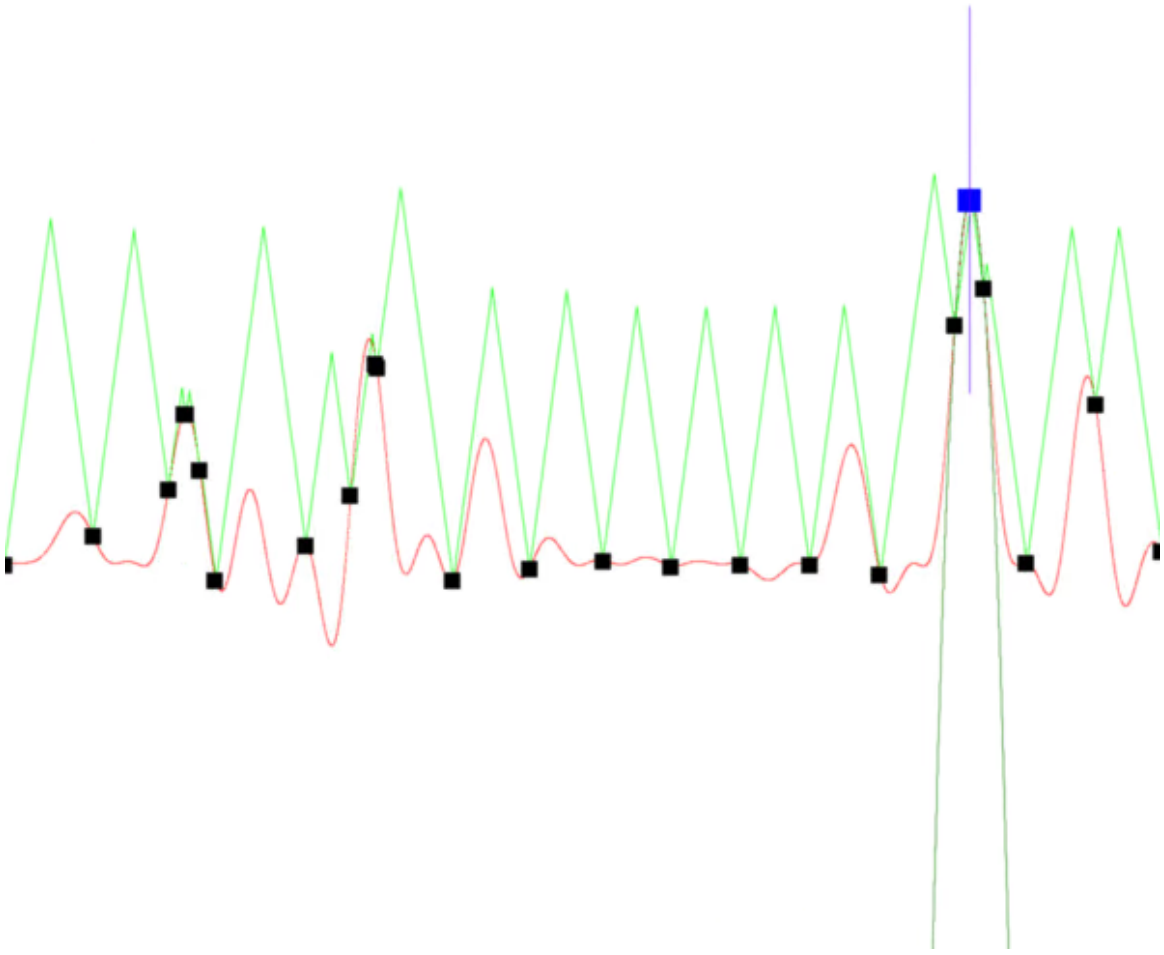
find_max_global() in action - 53

- Use trust region model



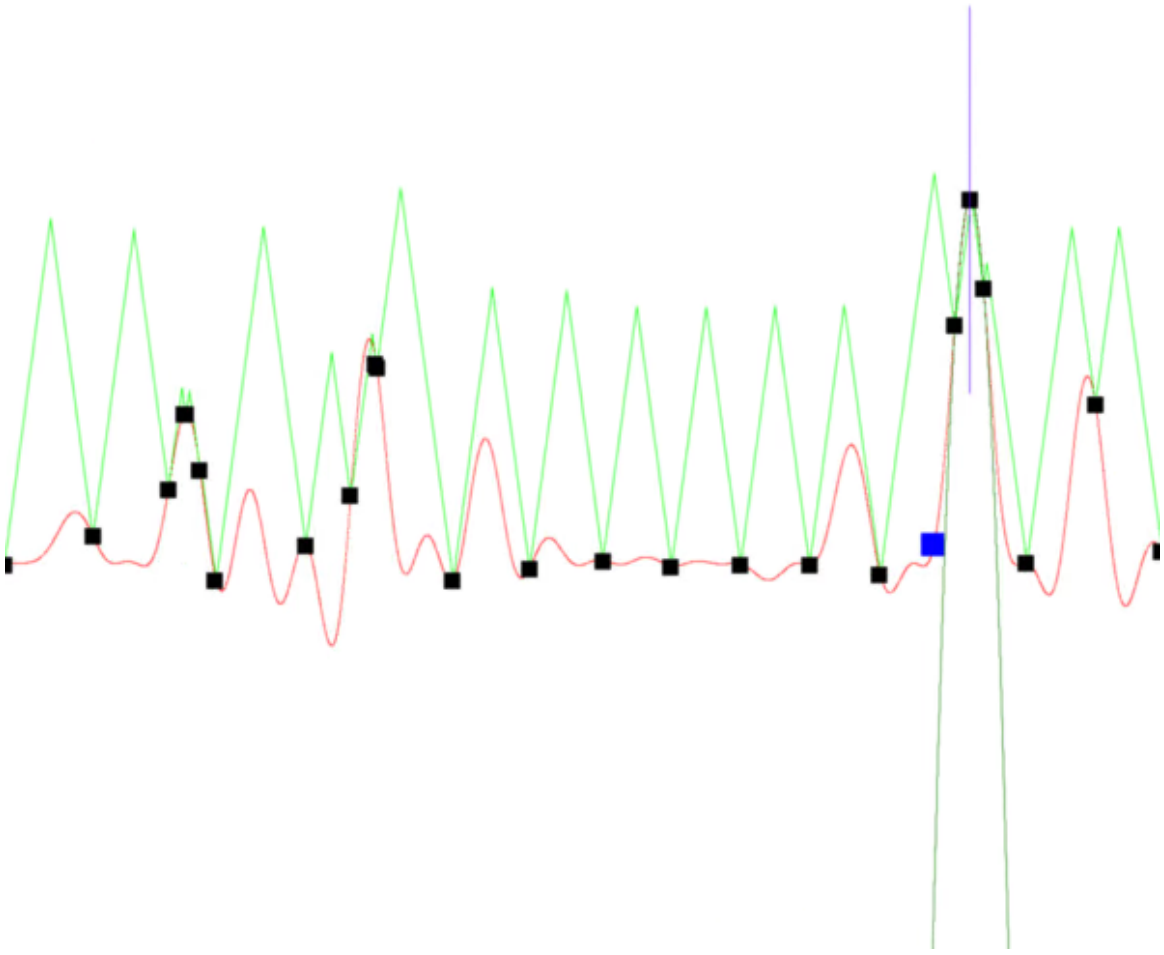
find_max_global() in action - 54

- Update models



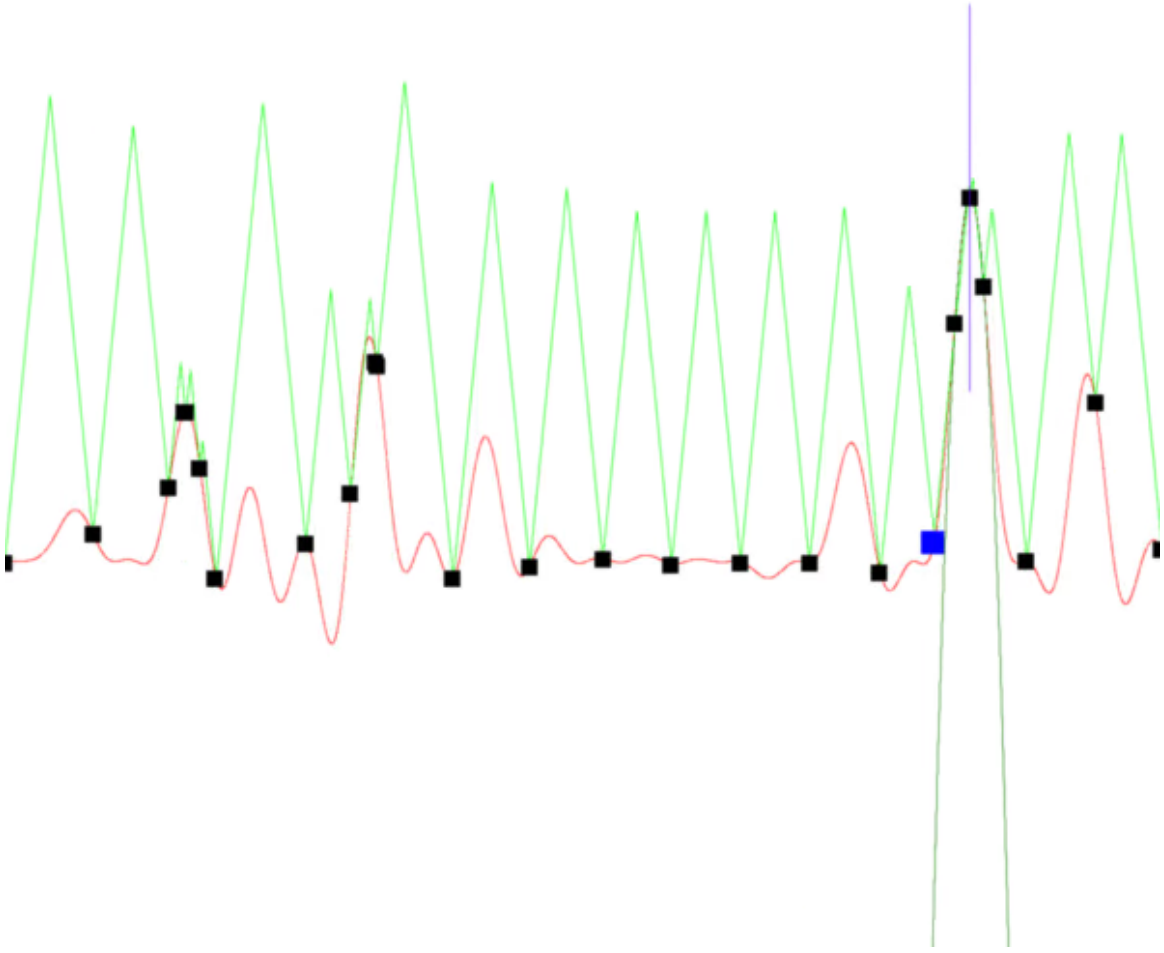
find_max_global() in action - 55

- Use upper bounding model



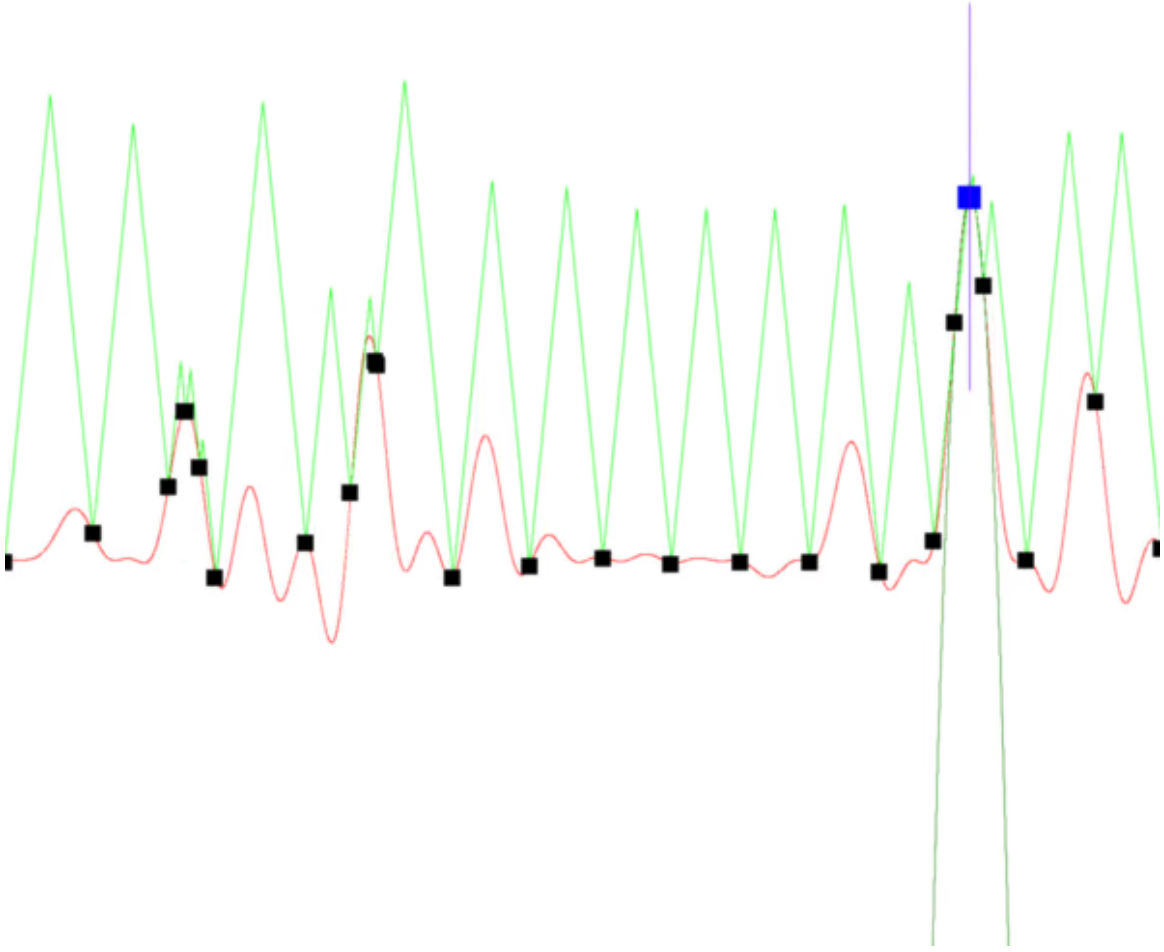
find_max_global() in action - 56

- Update models



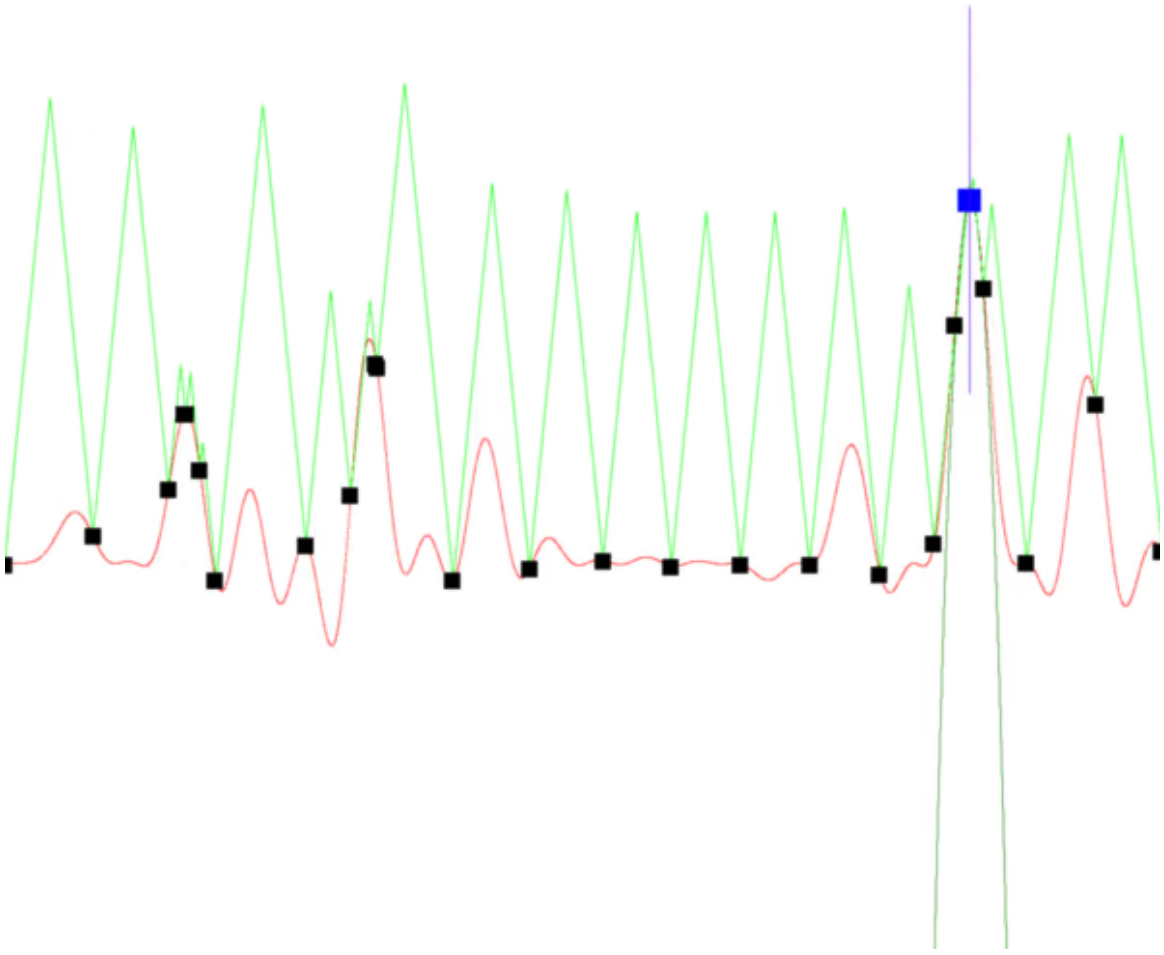
find_max_global() in action - 57

- Use trust region model



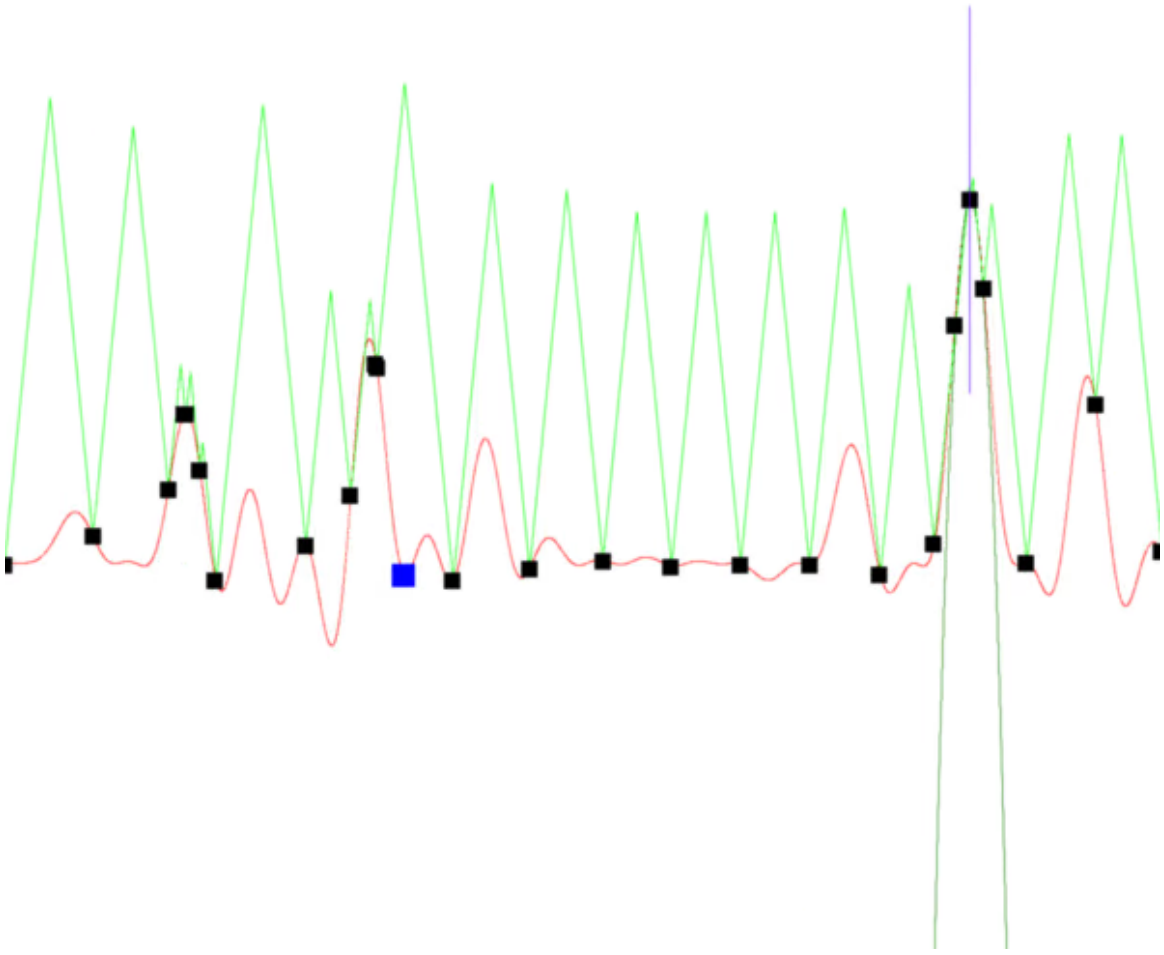
find_max_global() in action - 58

- Update models



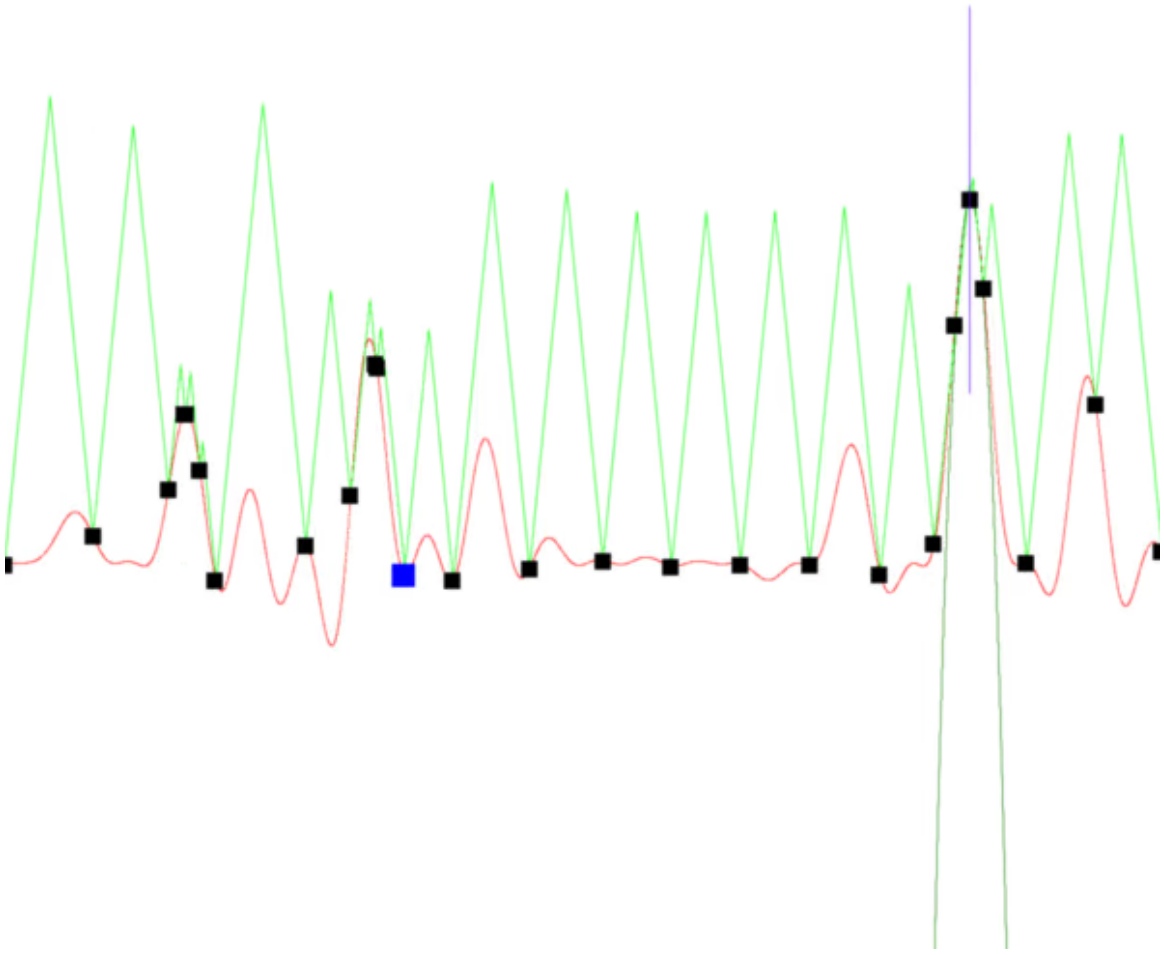
find_max_global() in action - 59

- Use upper bounding model



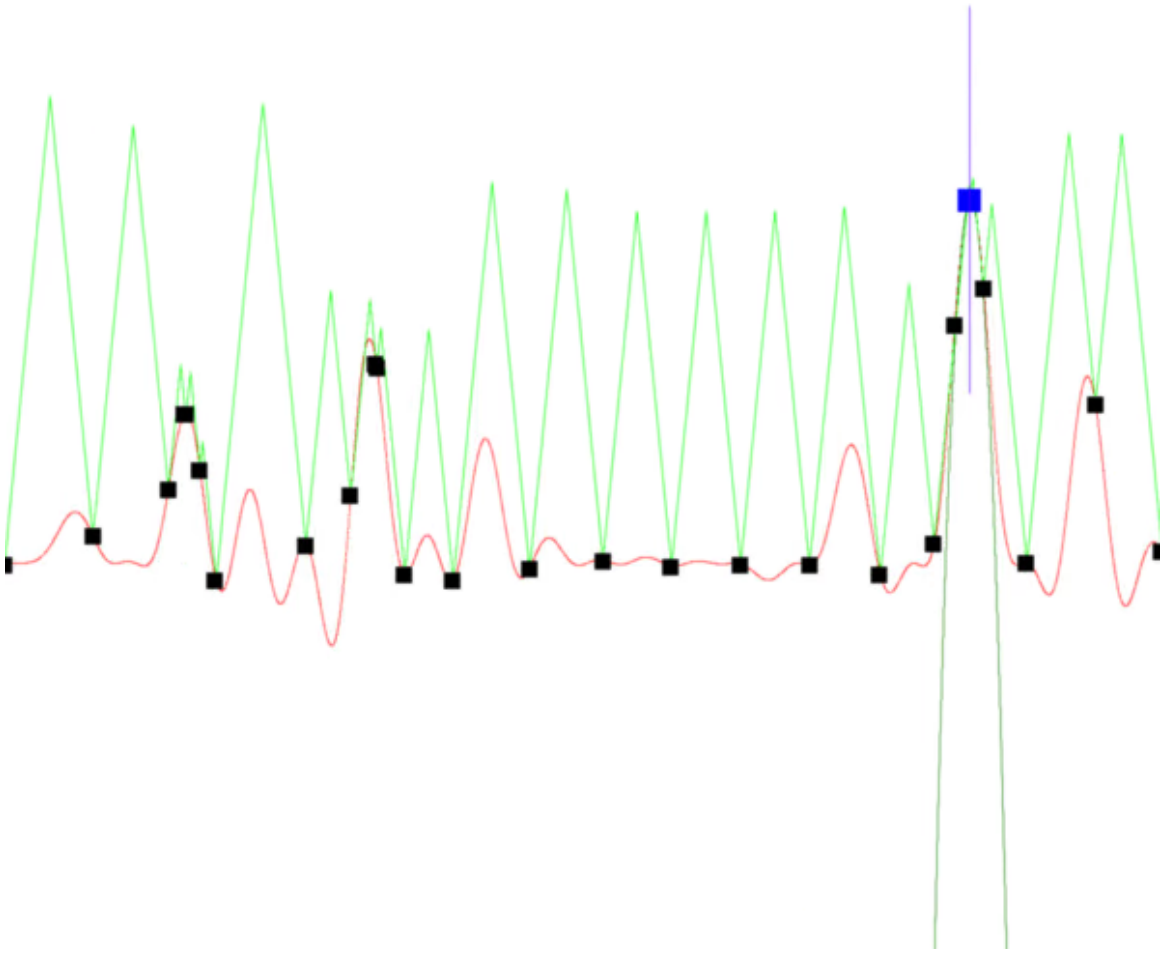
find_max_global() in action - 60

- Update models



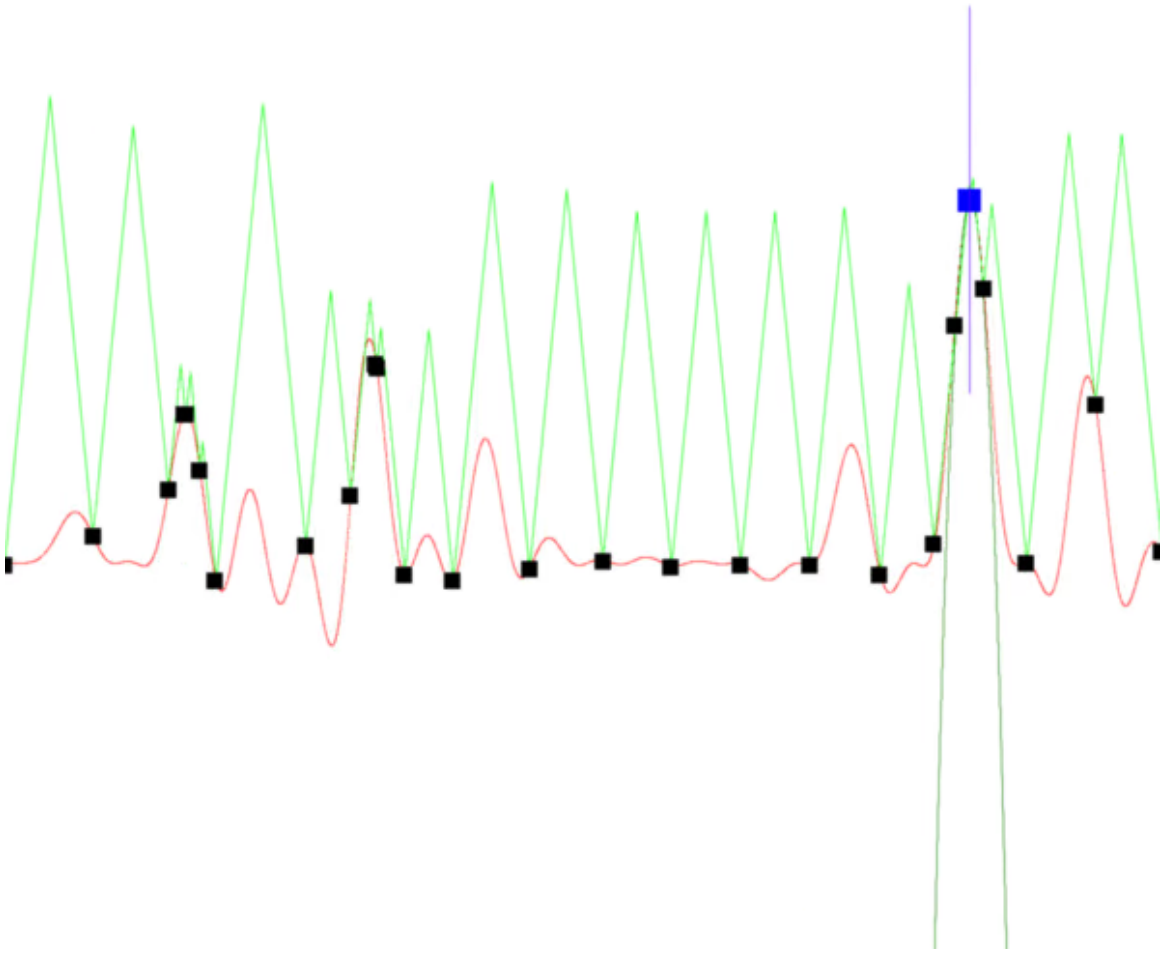
find_max_global() in action - 61

- Use trust region model



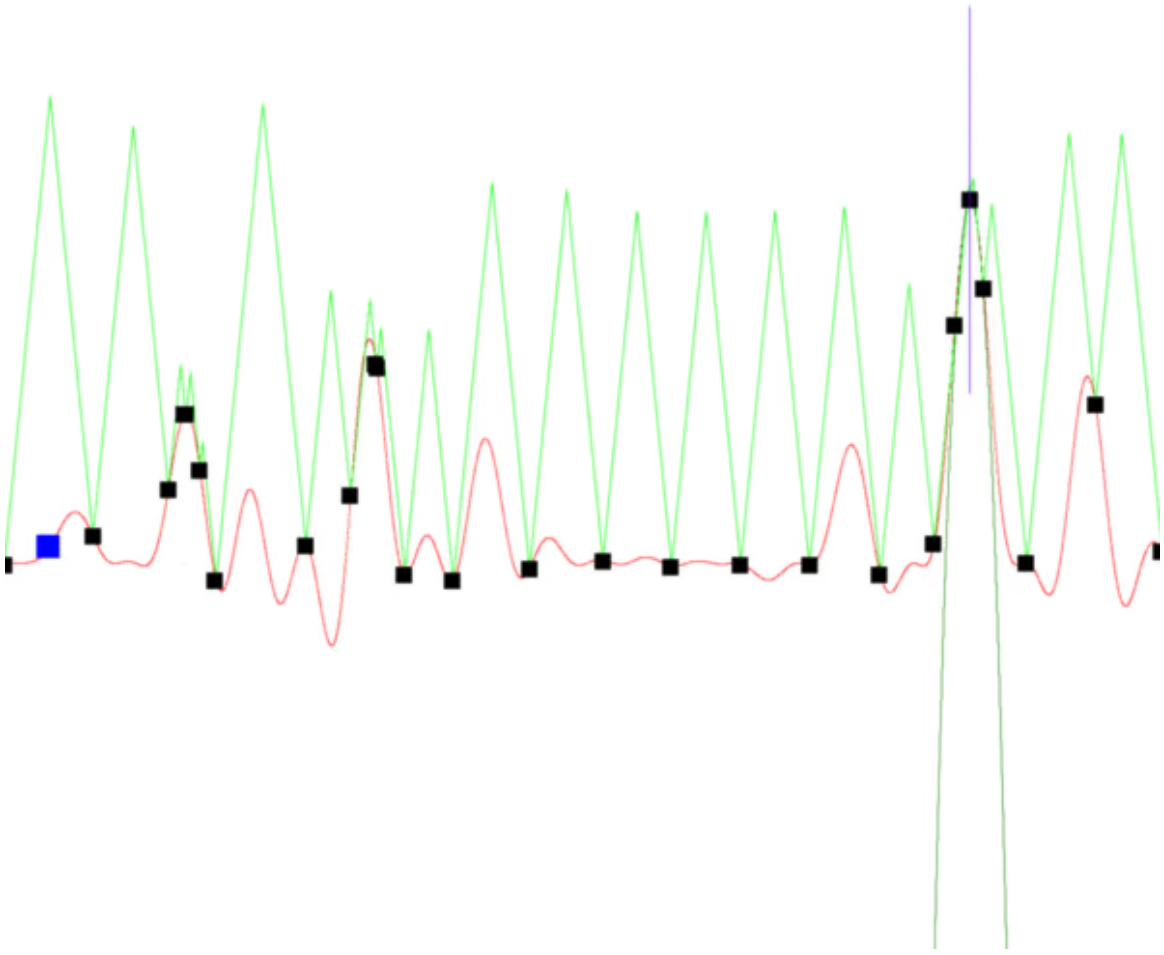
find_max_global() in action - 62

- Update models



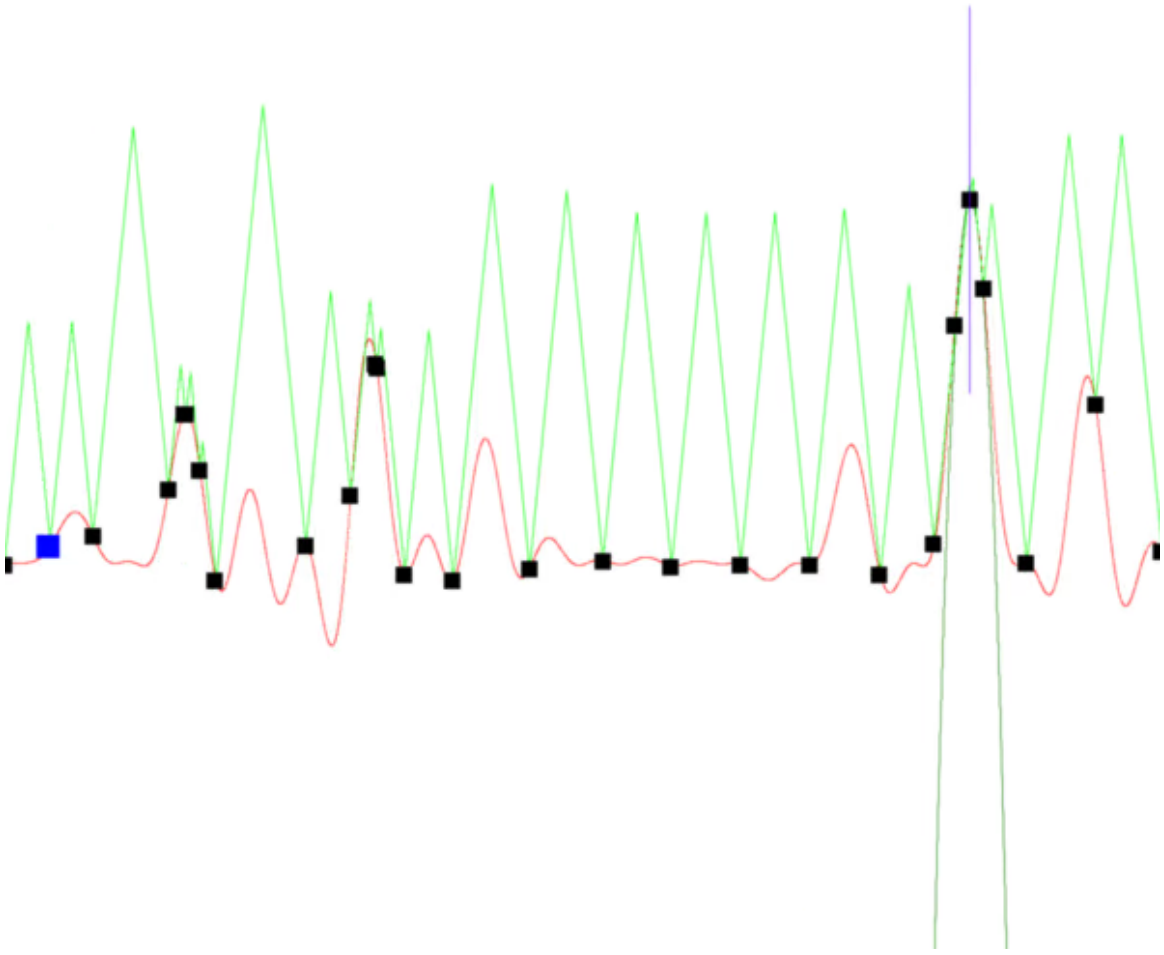
find_max_global() in action - 63

- Use upper bounding model



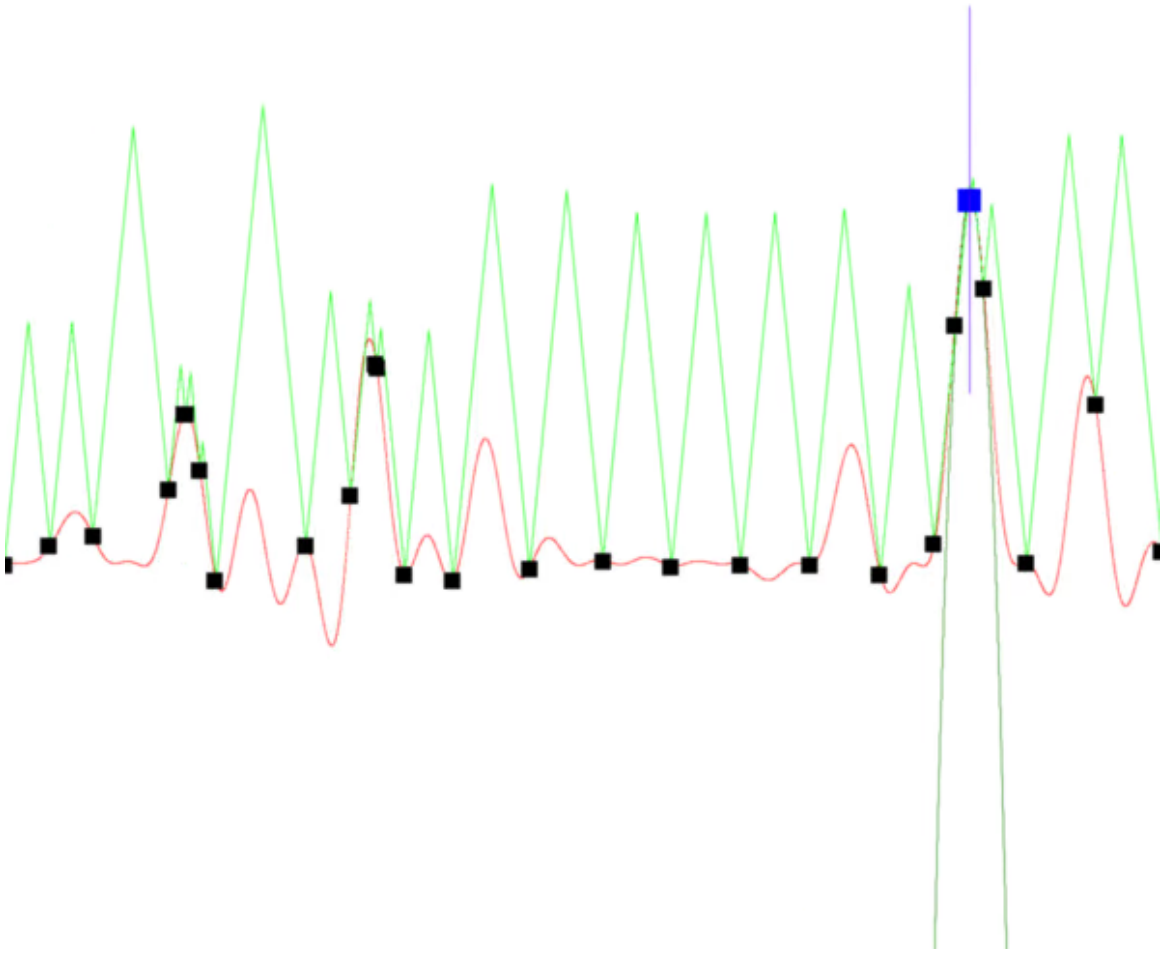
find_max_global() in action - 64

- Update models

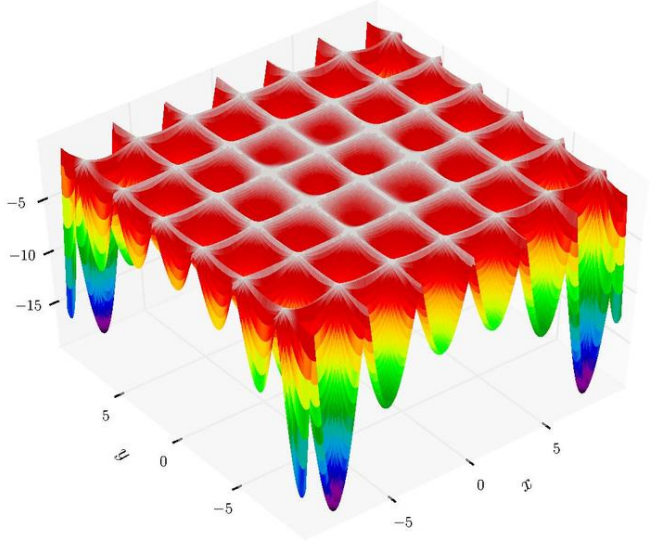


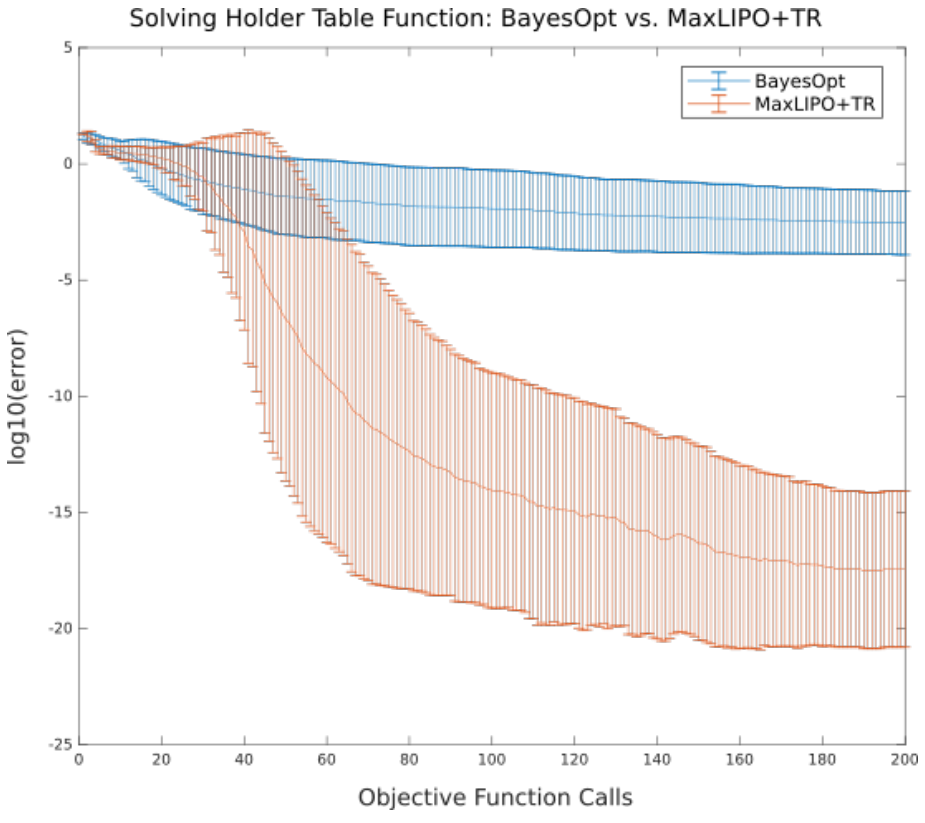
find_max_global() in action - 65

- Use trust region model

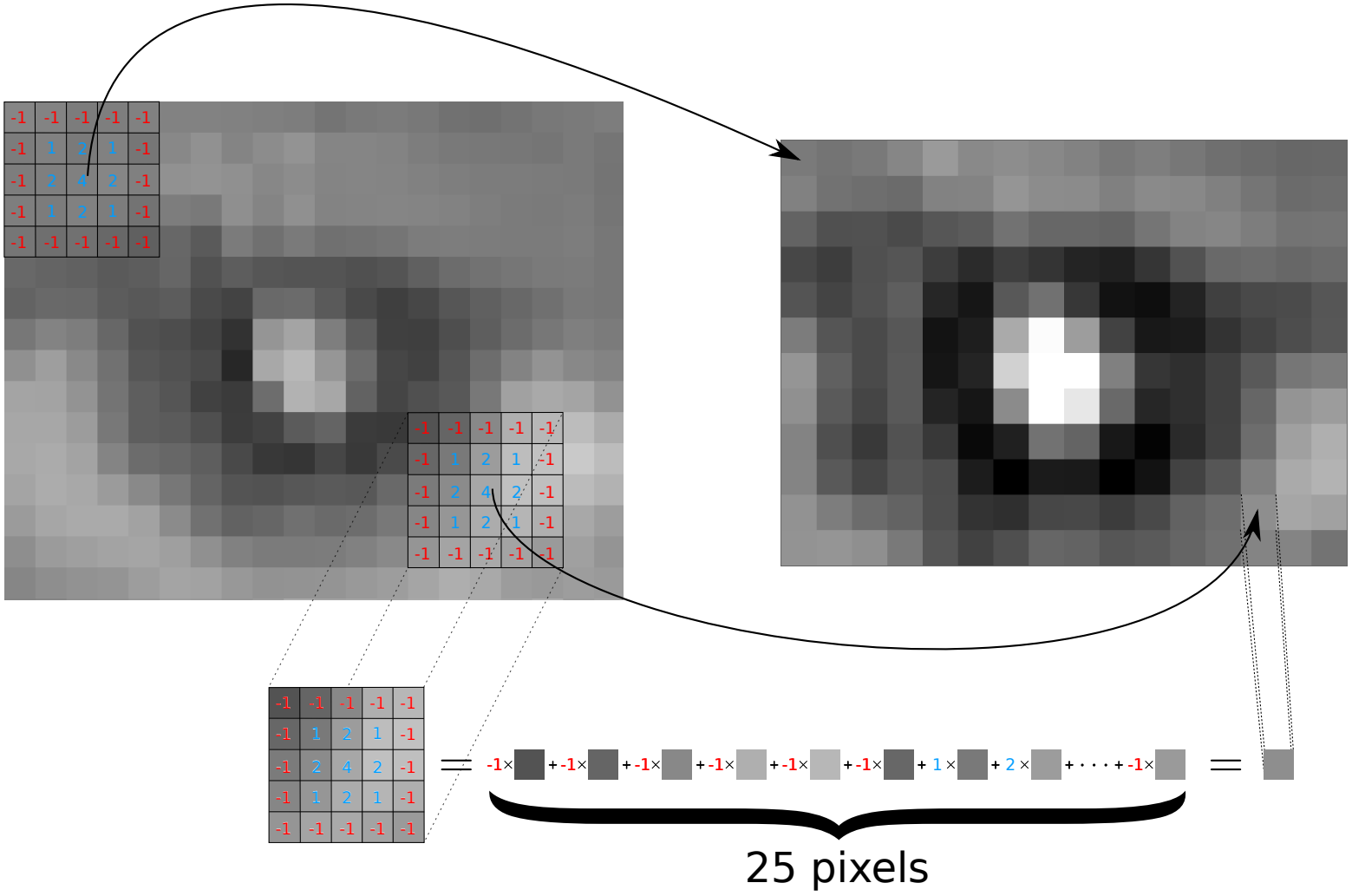


MaxLIPO+TR vs BayesOpt



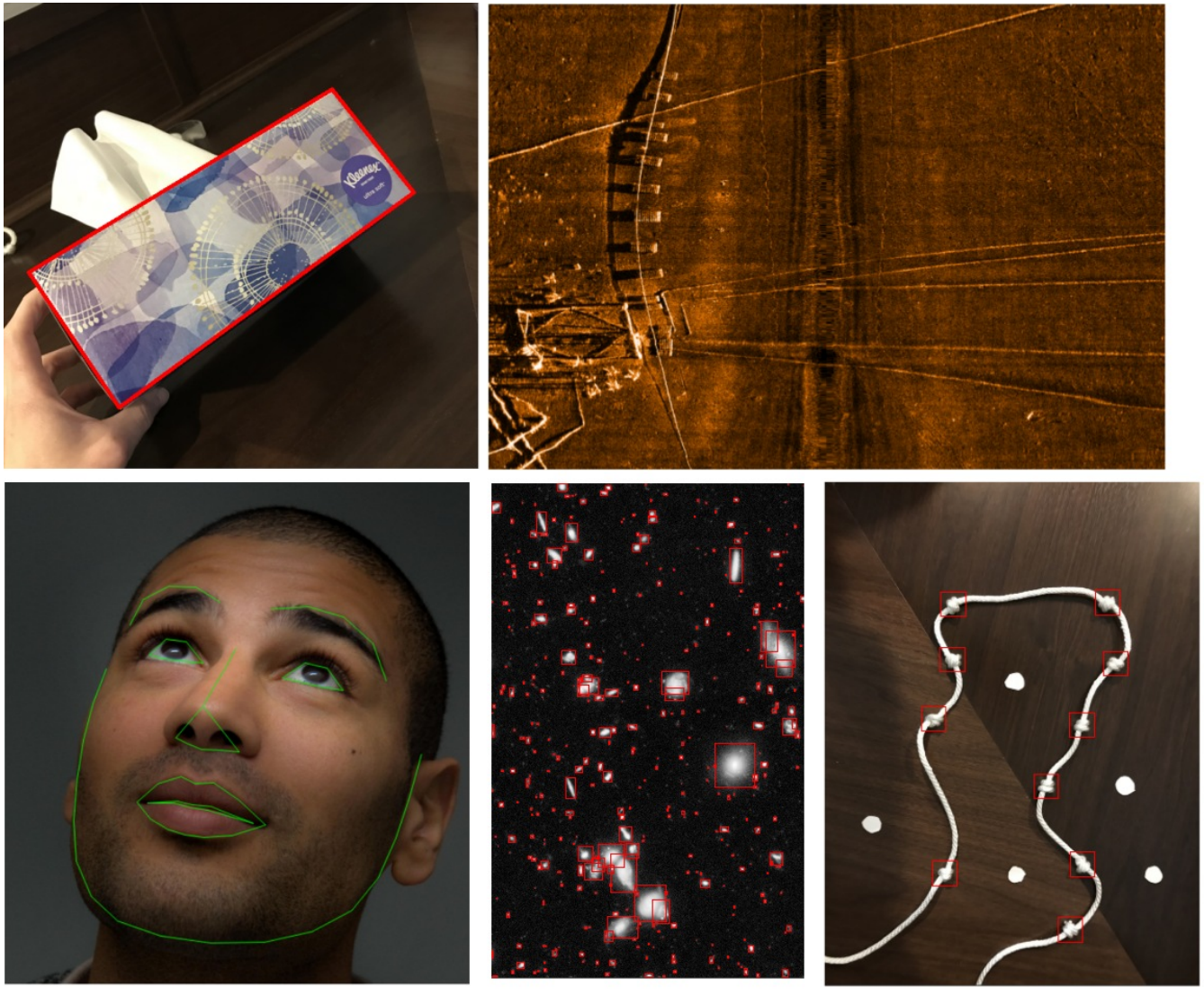


What is a spatial filter/convolution/cross-correlation?



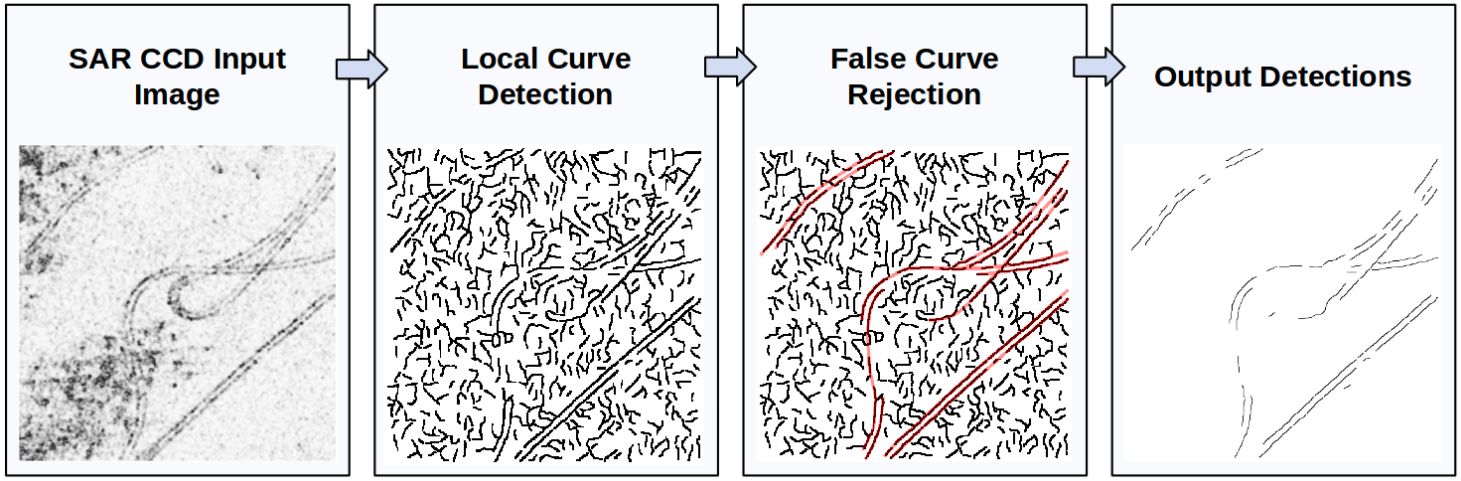
Used to detect or accentuate something in an image. In this case to detect the bright spot.

Something other than vanilla sliding window detection



Something other than vanilla sliding window detection

- Find tire tracks using synthetic aperture radar:



- Image segmentation:

The two fundamental parts of a detector

- Something that decides where to look
  - Predefined scanning patterns: sliding window, sliding window+pyramid, or multiple windows with different aspect ratios
  - Predefined + adjustment: fixed "anchor boxes" + bounding box regression
  - Simple image based: selective search, MSER, MBD
  - Learned: Region proposal networks, Haar cascades. Generally just some really fast sliding window detector.
- Something that decides if a location contains an object:
  - SVM, CNN, random forest, geometric rule, etc.

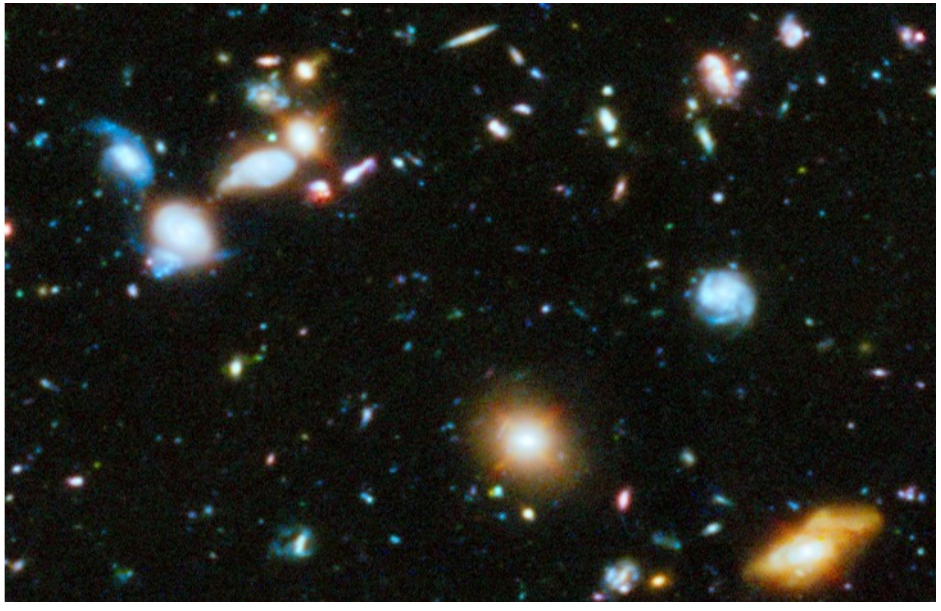
	SSD	RPN	Faster-RCNN	dlib-HOG	dlib-CNN
Search Strategy	Sliding window + BBR	Sliding window + BBR	RPN	Sliding window + pyramid	Sliding window + pyramid
Classifier	CNN	Small CNN	CNN	SVM+HOG	CNN

Deciding where to look, many options

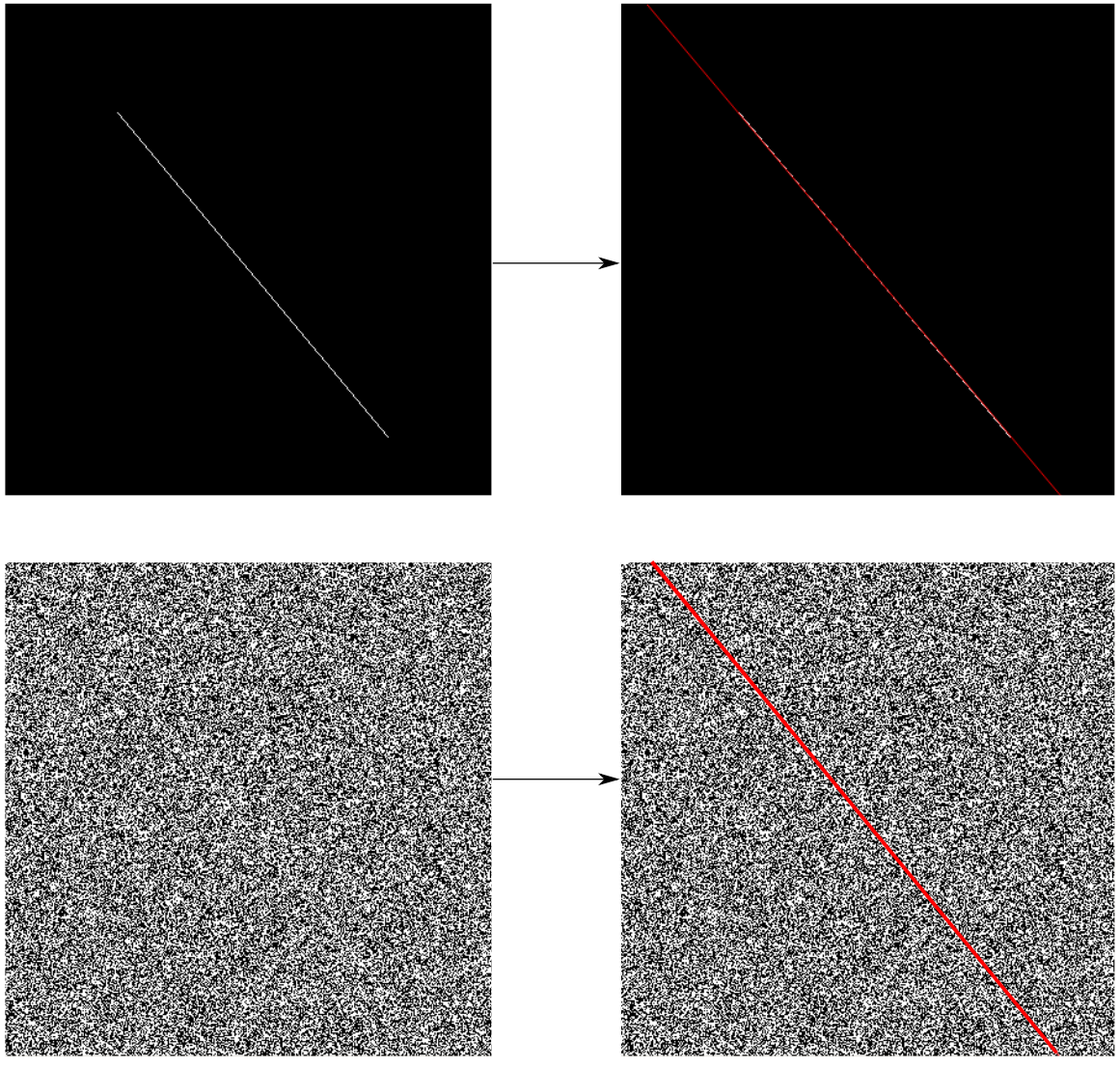
MSER

MBD

Deciding where to look, many options

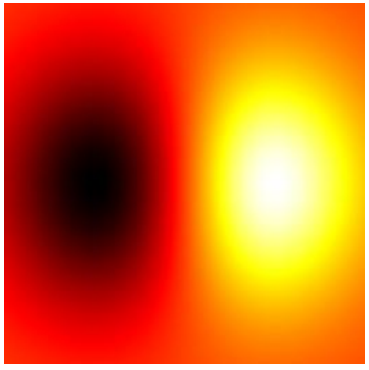


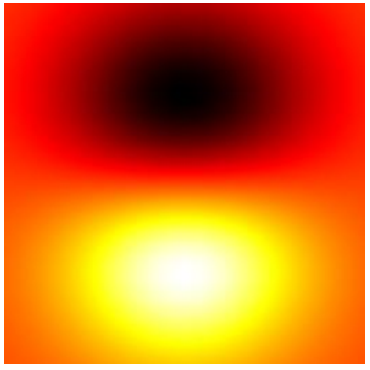
Deciding where to look, Hough transform

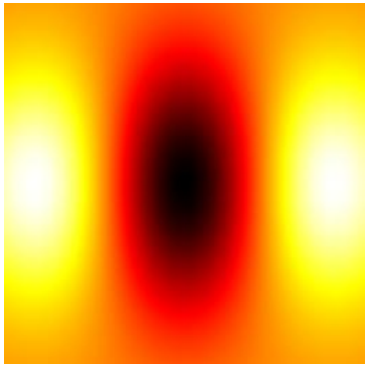


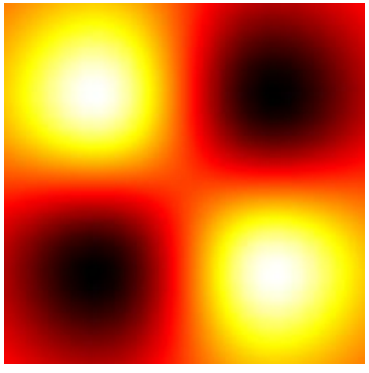
How Does image_gradients Work?

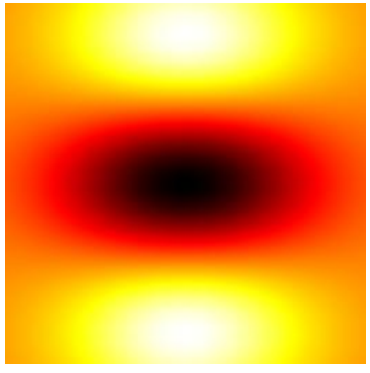
It convolves the image with these 5 filters:

$\frac{\partial I}{\partial x}$ 

$\frac{\partial I}{\partial y}$ 

$\frac{\partial^2 I}{\partial x^2}$ 

$\frac{\partial^2 I}{\partial x \partial y}$ 

$\frac{\partial^2 I}{\partial y^2}$ 

Why those filters?

How Does image_gradients Work?

Why those filters?
You have to ask yourself, what is an image gradient? Images are discrete grids, there is no gradient.

How Does image_gradients Work?

Why those filters?
You have to ask yourself, what is an image gradient? Images are discrete grids, there is no gradient.
Fit a quadratic surface to the neighborhood of a pixel located at $(\hat{x}, \hat{y})$:

$$\min_{w} \sum_{i=1}^{N} e^{\frac{-(x_i^2+y_i^2)}{2\sigma^2}} (f(x_i, y_i) - I(\hat{x}+x_i, \hat{y}+y_i))^2 \tag{1}$$

where $f(x, y) = w_1 x^2 + w_2 y^2 + w_3 xy + w_4 x + w_5 y + w_6$.
Then you can take the gradients of $f(x, y)$ as the "image gradient".

How Does image_gradients Work?

$$A = \begin{bmatrix} x_1^2 & y_1^2 & x_1 y_1 & x_1 & y_1 & 1 \\ x_2^2 & y_2^2 & x_2 y_2 & x_2 & y_2 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_N^2 & y_N^2 & x_N y_N & x_N & y_N & 1 \end{bmatrix} \quad P = \begin{bmatrix} I(\hat{x} + x_1, \hat{y} + y_1) \\ I(\hat{x} + x_2, \hat{y} + y_2) \\ \vdots \\ I(\hat{x} + x_N, \hat{y} + y_N) \end{bmatrix} \tag{2}$$

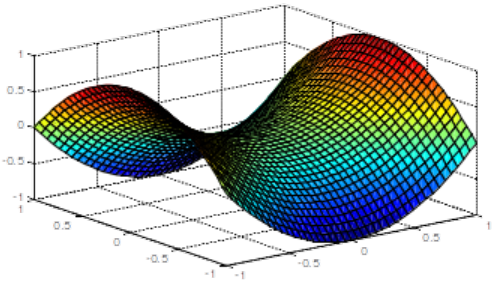
$$G = \begin{bmatrix} e^{\frac{-(x_1^2+y_1^2)}{2\sigma^2}} & 0 & \dots & 0 \\ 0 & e^{\frac{-(x_2^2+y_2^2)}{2\sigma^2}} & \dots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & 0 & e^{\frac{-(x_N^2+y_N^2)}{2\sigma^2}} \end{bmatrix} \tag{3}$$

Then the optimal $w = (\bar{A}^T \bar{A})^{-1} \bar{A}^T \bar{P}$ where $\bar{A} = GA$, $\bar{P} = GP$.

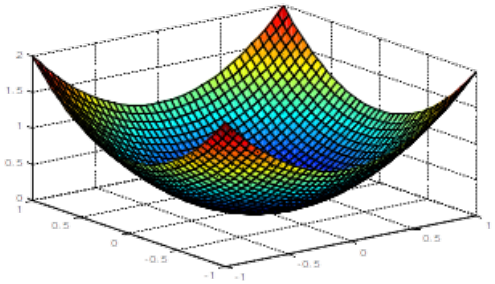
This means that the magic filters are just the rows of:
$(\bar{A}^T \bar{A})^{-1} \bar{A}^T G$

Why a quadratic?

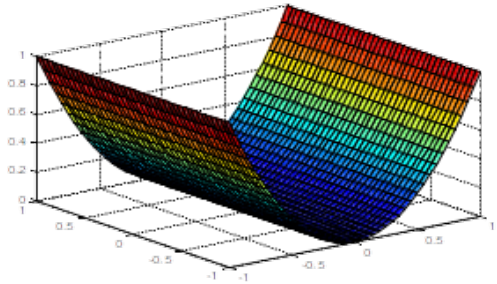
Saddle



Bowl



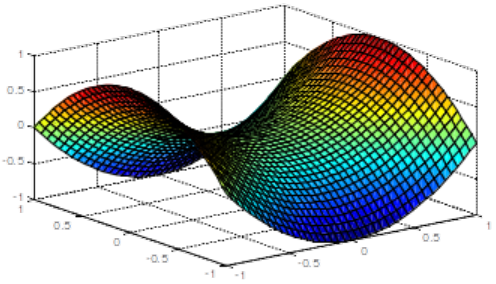
Trough



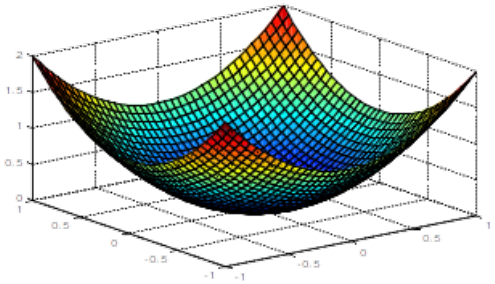
$$f(x,y) = \frac{1}{2}(w_1x^2 + w_2y^2 + 2w_3xy) + w_4x + w_5y + w_6 \qquad (4)$$

Why a quadratic?

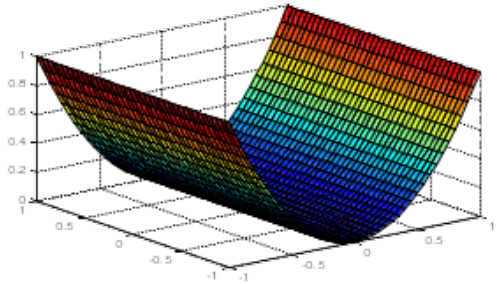
Saddle



Bowl



Trough



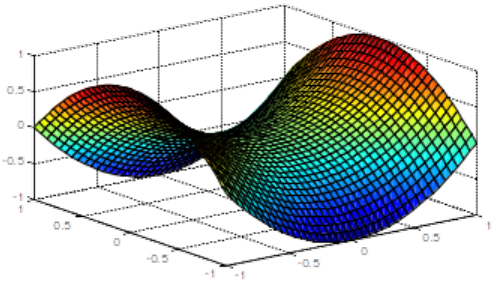
$$f(x,y) = \frac{1}{2}(w_1 x^2 + w_2 y^2 + 2w_3 xy) + w_4 x + w_5 y + w_6 \qquad (4)$$

or

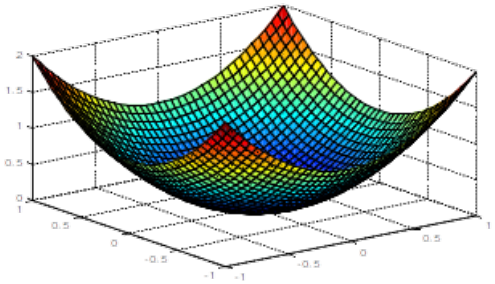
$$f(x,y) = \frac{1}{2}\begin{bmatrix} x & y \end{bmatrix}\begin{bmatrix} w_1 & w_3 \\ w_3 & w_2 \end{bmatrix}\begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} w_4 & w_5 \end{bmatrix}\begin{bmatrix} x \\ y \end{bmatrix} + w_6 \qquad (5)$$

Why a quadratic?

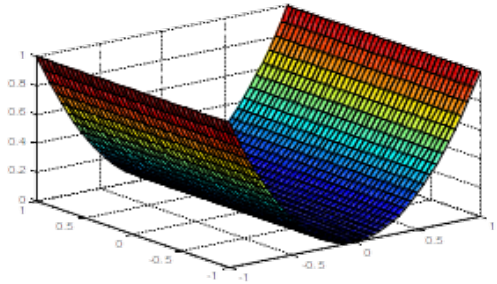
Saddle



Bowl



Trough



$$f(x,y) = \frac{1}{2}(w_1 x^2 + w_2 y^2 + 2w_3 xy) + w_4 x + w_5 y + w_6 \qquad (4)$$

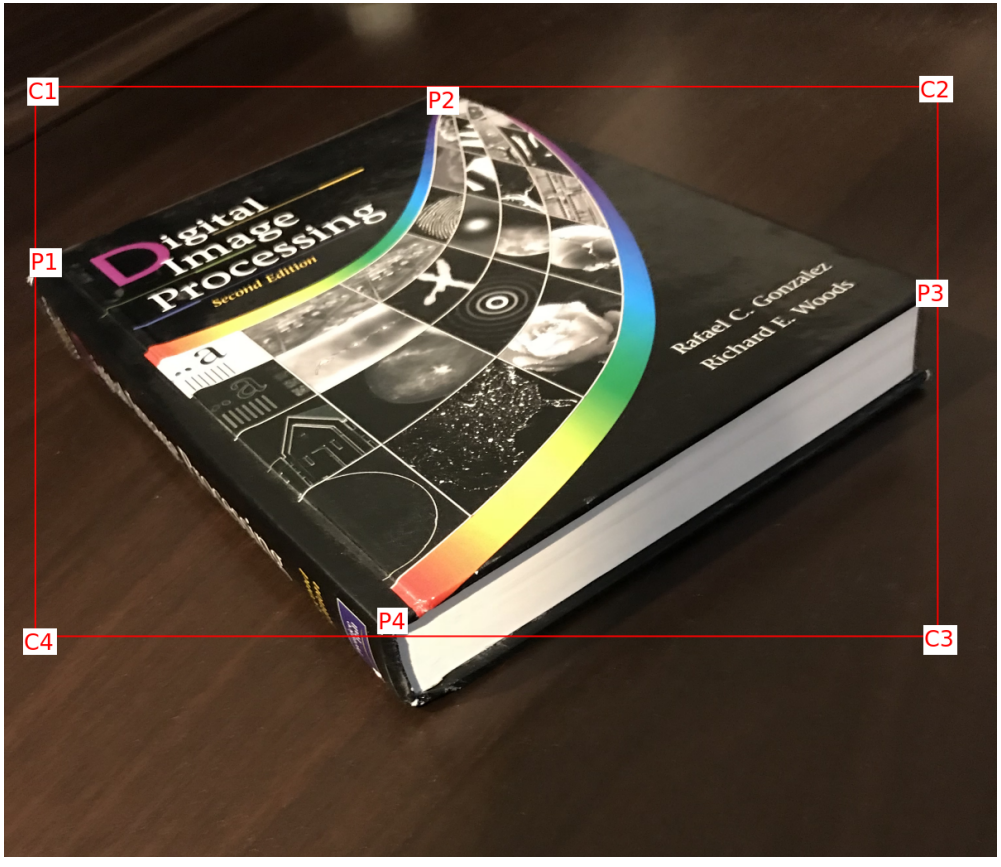
or

$$f(x,y) = \frac{1}{2}\begin{bmatrix} x & y \end{bmatrix}\begin{bmatrix} w_1 & w_3 \\ w_3 & w_2 \end{bmatrix}\begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} w_4 & w_5 \end{bmatrix}\begin{bmatrix} x \\ y \end{bmatrix} + w_6 \qquad (5)$$

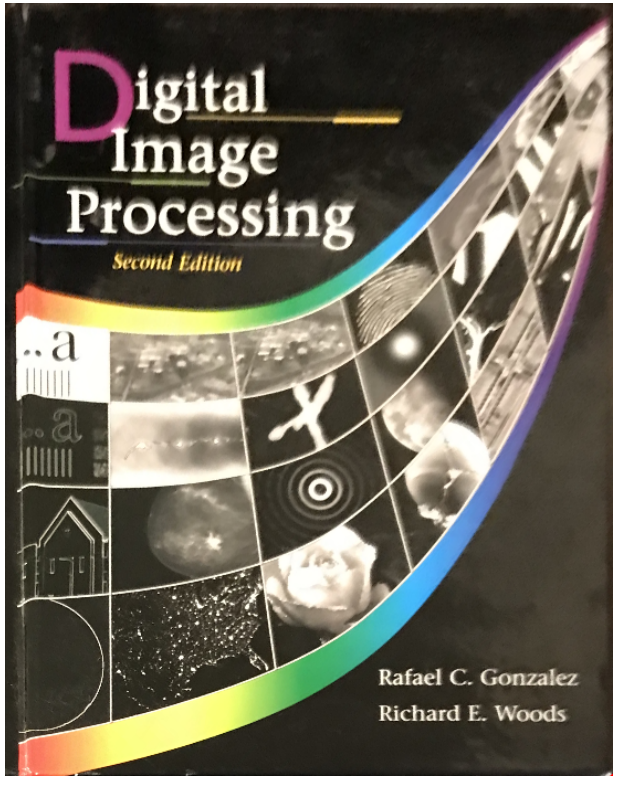
Second derivative matrix is $\begin{bmatrix} w_1 & w_3 \\ w_3 & w_2 \end{bmatrix}$

How Does extract_image_4points() Work?

Given an image and 4 points:



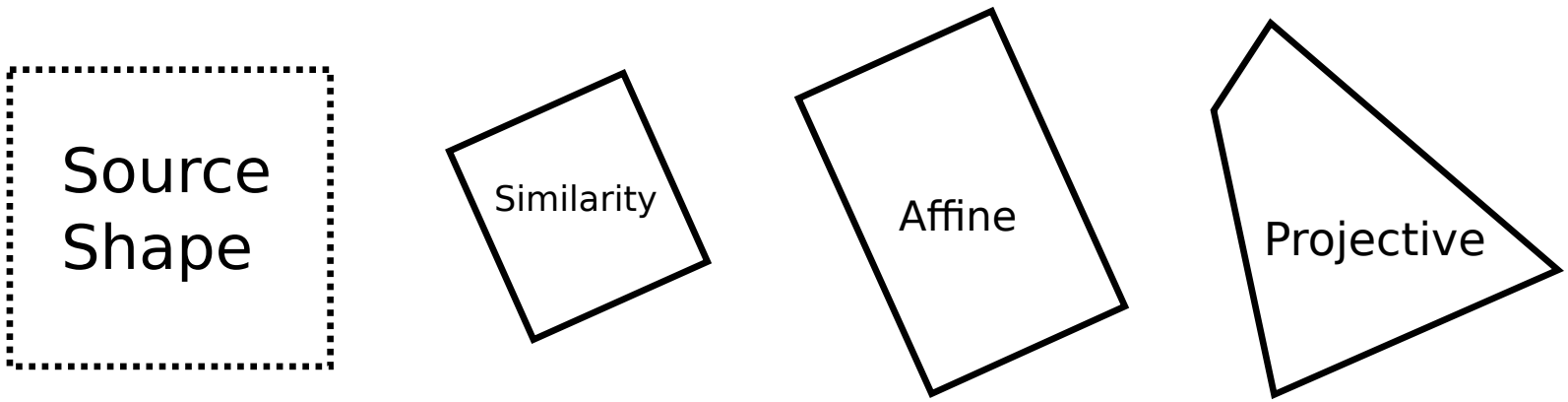
Produce:




```
chip = extract_image_4points(img, [P1, P2, P3, P4],
    rows, cols)
```

How Does extract_image_4points() Work?

- How to find a mapping function M between pixels in input and output image?
- Lots of different types of M
  - similarity transforms: scale, translation, and rotation
  - affine transforms: scale, translation, rotation, and skews
  - projective transforms: scale, translation, rotation, skews, and **out of plane rotations**



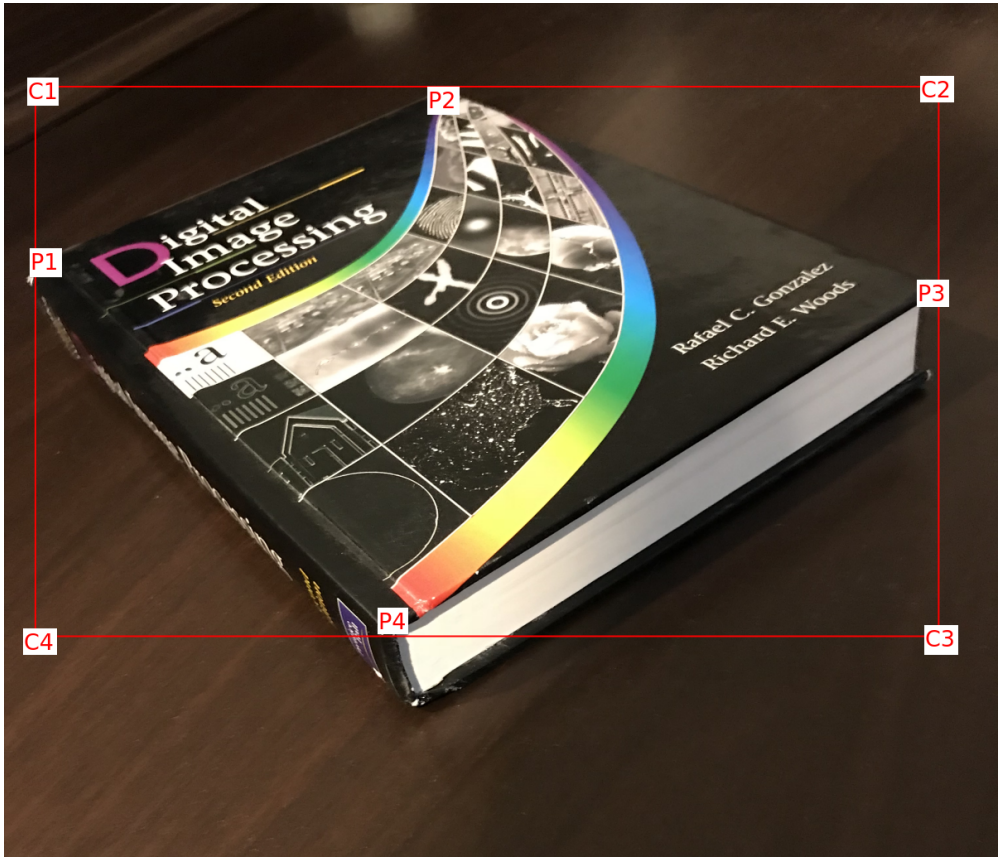
How Does extract_image_4points() Work?

- Step 1: find correspondence between box corners and object corners (e.g. $P_1 \rightarrow C_1$ or $P_1 \rightarrow C_3$?)
- Step 2: find a function, M, mapping from the output image pixels to the input image. e.g. $M(P_1) = C_1$, $M(P_2) = C_2$, etc.
- Step 3: copy the data

```
for r in range(rows):
    for c in range(cols):
        ir,ic = M(r,c)
        out_image[r][c] = in_image[ir][ic]
```

How Does extract_image_4points() Work?

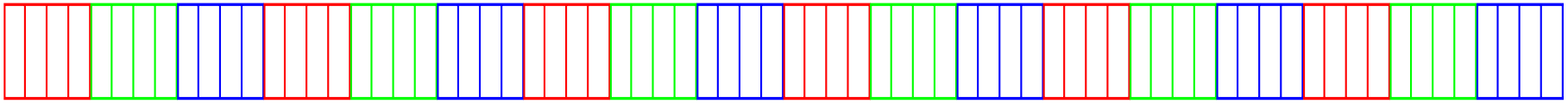
- How to find a correspondence between the corners?



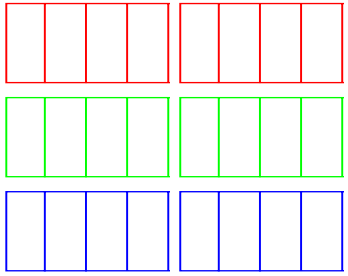
- Assign corners to nearest neighbors? Eh, not so good.
- Hungarian algorithm: find the assignment that globally minimizes the sum of distances.

What's up with memory caches?

MEMORY

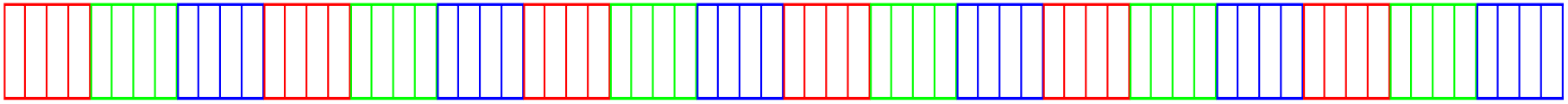


CACHE (24 byes)

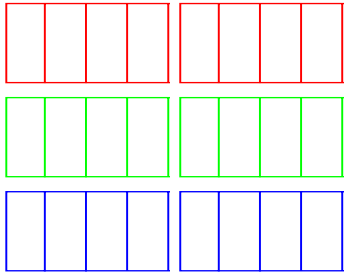


What's up with memory caches?

MEMORY



CACHE (24 byes)



Imagine an image with 3 rows and 12 columns: