Unsupervised Cosegmentation using Pixel-Correspondences and Functional Maps

Zimo Li

Contents

List of Figures

List of Tables

1 Summary

The ultimate goal of the project is to produce segmentations among a related image collection, without using any ground-truth training data. This is done by iteratively refining all pair-wise correspondences within the image-network so that they are consistent, based on the work of Wang et al [1]. In particular, we optimize these correspondences to be cycle-consistent; correspondences that pass from A to B to C to A should be the identity map between A and itself. The final segmentations are derived from the most consistent correspondences across all images.

Correspondences are represented by functional maps between the images, and the cycle-consistency constraint is formulated as a latent-basis optimization among the functional maps [1]. Functional maps are maps between the space of functions over objects, and have been used to represent relations between shapes and images [2, 1]. The bases we use for the images are the eigenvectors of the normalized graph Laplacian, computed over a graph-representation of the image, as this basis expresses pixel-intensity boundaries well [6].

To initialize the functional maps, we align pixel-correspondences given to us by an optical-flow algorithm, such as SIFT-flow [3], as well as Saliency maps from the GBVS method [4]. After initializing the functional maps, we compute a latent-basis across the network. These are the functions that are most consistent across all maps. After we have the latent basis, we employ an alternating optimization between the latent-basis and the new maps that use the latent basis as constraints, until convergence [1]. The final segmentations are derived from the latent basis - the most consistent function. To refine the segmentations, we match them to state-of-the-art object proposals generated by the Geodesic Object Proposals algorithm [5].

In short:
1. Compute functional maps between all images
2. Calculate "consistent-functions" across all functional maps
3. Recalculate functional maps to align the consistent functions, then repeat steps (1) and (2) until convergence.
4. Most consistent function represents our segmentation
5. Refine this segmentation using GOP for final output.

2 Computation of Image Basis

Given an image $\mathbf{P} = \{p_1, ..., p_n\}$, where p_i designates pixel i, let $\mathcal{F}(\mathbf{P}, \mathbb{R})$ denote the space of all functions $f : \mathbf{P} \rightarrow \mathbb{R}$. Notice that $f \in \mathbb{R}^n$. The goal is to find an approximate basis $\mathcal{B} \subset \mathcal{F}$ for this space that spans well the functions in this space that we care about. In particular, segmentations of the image can be formulated as indicator-functions in this space, and we want the projection of a segmentation into the space spanned by $\mathcal{B}$ to be good.

First, we describe an image $\mathbf{P}$ as a weighted graph G, with nodes being pixels and edges representing the pixel-intensity differences between pixels within a certain proximity to each other [6]. In particular, the weights are given by:

$$w_{ij} =$$

$$\begin{cases} e^{\frac{||X(i)-X(j)||_2^2}{\sigma_x^2}} \cdot e^{\frac{||V(i)-V(j)||_2^2}{\sigma_v^2}}, & ||X(i)-X(j)||_2^2 \leq r \\ 0, & o.w. \end{cases} \tag{1}$$

where $X(i)$ is the 2-d position of pixel i and $V(i)$ is the RGB value of pixel i.

To compute the basis, we take the M smallest eigenvectors of the normalized Laplacian, L, of this graph. $M = 64$ for all our experiments. For all our experiments, $r = 5$, σ_x is given by $\frac{1}{10}$ the size of the image-diagonal, and σ_v is given by the median of all pixel-intensity differences across the entire image.

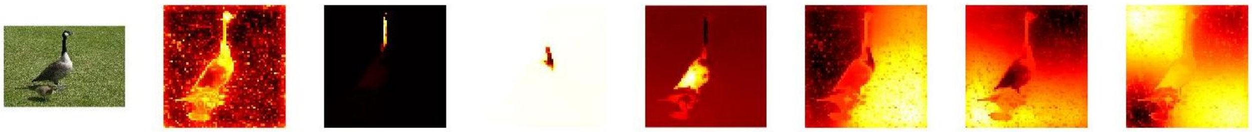
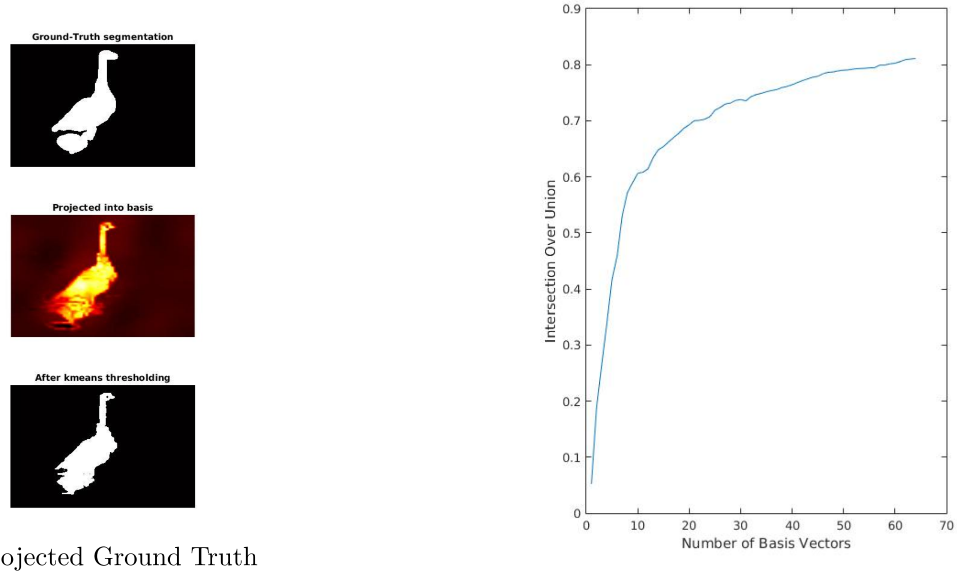


Figure 1: 7 smallest eigenvectors



(a) Projected Ground Truth

(b) Intersection Over Union vs. Number of Basis Vectors

Figure 2: Projection of Ground Truth into Basis

3 Aligning Functional Maps

Given two images, P_i and P_j, we can build a map $X_{ij} : \mathcal{F}(P_i, \mathbb{R}) \rightarrow \mathcal{F}(P_j, \mathbb{R})$. This is a functional map, which maps functions on P_i to functions on P_j. This functional map is a change of basis matrix between the eigenvectors of the Laplacians of the respective images.

3.1 Pixel Correspondences

For an image $P_e = \{p_{e_1}, ..., p_{e_n}\}$, let $\tilde{p_{e_i}} \in \mathbb{R}^n$ be the indicator function for p_{e_i}, i.e. an n-dimensional vector whose i^{th} entry is 1 and all other entries are 0. Let $\tilde{P}_e = \{\tilde{p_{e_i}}\}_{0 \leq i \leq n}$.

Given two images $P_i, P_j \in \mathbb{R}^n$, the SIFT-flow algorithm [3] computes a pixel-wise correspondence $C_{ij} : \tilde{P}_i \to \tilde{P}_j$.

Let A and B be $n \times n$ matrices such that $A_i = [\tilde{p_{i_1}}...\tilde{p_{i_n}}]$ and $B_{ij} = [C_{ij}(\tilde{p_{j_1}})...C_{ij}(\tilde{p_{j_n}})]$.

Let L_i and L_j be the respective normalized Laplacians of the two images, E_i and E_j the respective eigenvectors of these two Laplacians arranged as column vectors in matrices, and λ_i, λ_j the respective eigenvalues, arranged as diagonal square matrices.

$\bar{A_i} = E_i^T A_i$ and $\bar{B_{ij}} = E_j^T B_{ij}$ the projection of A_i and B_{ij} onto their respective image bases E_i and E_j, are the functional constraints derived from the SIFT-flow correspondences. We normalize the columns of $\bar{A_i}$ and $\bar{B_{ij}}$ after computing them.

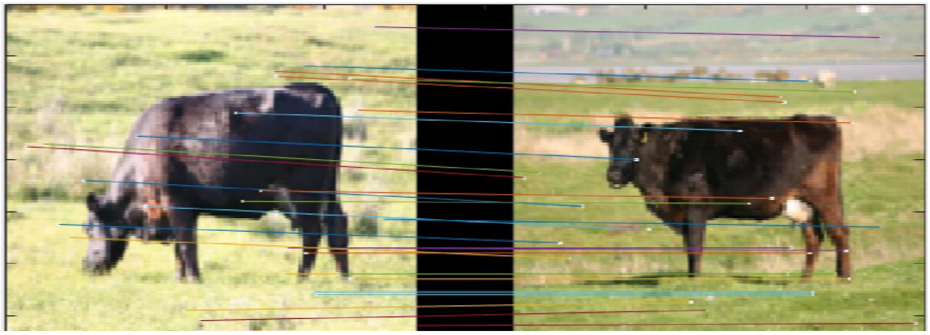


Figure 3: Example Sift Correspondences

3.2 Saliency Maps

To incorporate saliency, we compute GBVS [4] maps for the images and put these constraints alongside the pixel-correspondences. That is, let s_i, s_j the saliency maps of images i, j respectively. Then, $\bar{s_i} = E_i^T s_i$ and $\bar{s_j} = E_j^T s_j$ are the constraints from the saliency maps. These are also normalized afterwards.

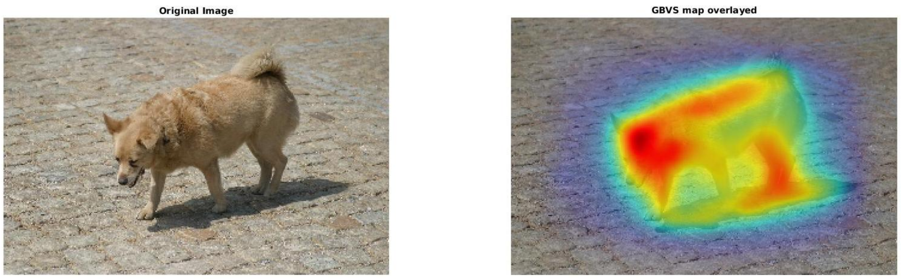


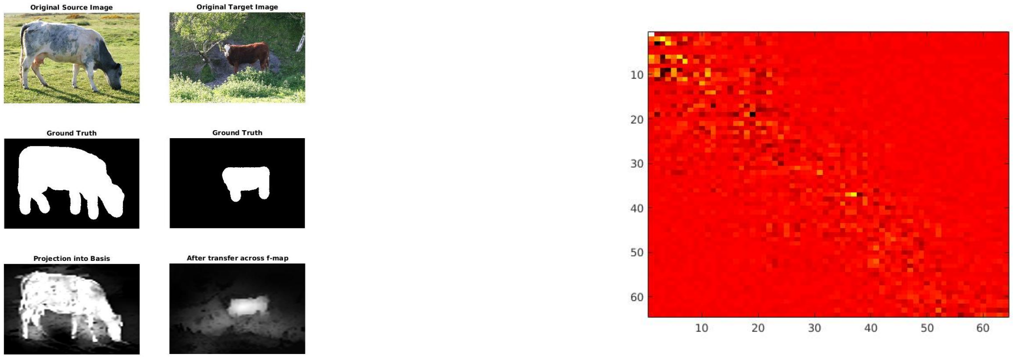
Figure 4: GBVS saliency map

3.3 Pairwise Objective

Given the above, the following objective calculates the alignment of functional maps between all correspondences:

$$f_{ij}^{pair}(X) = \left\| X[\bar{A}_i \; \kappa\bar{s}_i] - [\bar{B}_{ij} \; \kappa\bar{s}_j] \right\|_2 + \sigma \left\| X\lambda_i - \lambda_j X \right\|_2 \tag{2}$$

[ ] denotes horizontal concatenation. κ reflects the importance of aligning the saliency maps and σ is a regularization term favoring mappings between eigenvectors with similar frequencies, which are more likely to correspond [6]. Such regularization We set κ to be 200 and σ to be 1000 for all our trials.



(a) Segment transfer through map

(b) The functional map

Figure 5: Funcmap transfer

4 Cycle Consistency

Once the pair-wise functional maps are computed, we want to optimize them to be consistent across all the images. We represent our collection of m images, $I = \{P_1, ..., P_m\}$, as a weighted graph, and seek to preserve the consistency of maps along cycles in this graph.

In particular, we formulate cycle-consistency as optimizing the alignment of a latent-basis, in the style of [1]. We expect the latent basis to represent functions which are consistent across many images, such as region correspondences or saliency-functions. We will discuss the computation of the latent basis below. First, however, let us examine why optimizing the latent basis is sufficient.

The cycle consistency constraint is formulated in the same way as [1]:

Let $X_{ij} : \mathcal{F}_i \to \mathcal{F}_j$ be the functional map between images i and j. Let $C_m = \{(a_0, a_1, .., a_z, a_0) | z <= m, a_i \neq a_j\}$ represent all sequences among the m images which correspond to cycles in the graph. Then, given a function $f \in \mathcal{F}_{a_0}$, we want to insure that:

$$X_{a_z a_0} ... X_{a_1 a_2} X_{a_0 a_1} f = f, \quad \forall (a_0, .., a_z, a_0) \in C_m \tag{3}$$

Let $\mathcal{Y} = \{Y_1, ..., Y_t\}$ represent the latent basis of the graph. In particular, each $Y_i = \left\{y_i^1, ..., y_i^m\right\}$, as the latent basis has a different representation in each image. With this basis, our new requirement is simply to insure $X_{ij}Y_i = Y_j$. To see why this is equivalent to the above formulation, let us consider any f_g in the global coordinate system. It's representation in a the a_0^{th} basis is given by $f = Y_{a_0} f_g \in \mathcal{F}_{a_0}$. Then:

$$X_{a_z a_0} ... X_{a_1 a_2} X_{a_0 a_1} f = X_{a_z a_0}(... X_{a_1 a_2} X_{a_0 a_1} Y_{a_0}) f_g = X_{a_z a_0} Y_{a_z} f_g = f \tag{4}$$

As long as we align every pairwise map to be consistent with such a basis, any given function will also be aligned across the entire set. The objective for consistently aligning them is given by:

$$f_{ij}^{cons}(X, \mathcal{Y}) = \|XY_i - Y_j\|_2 \tag{5}$$

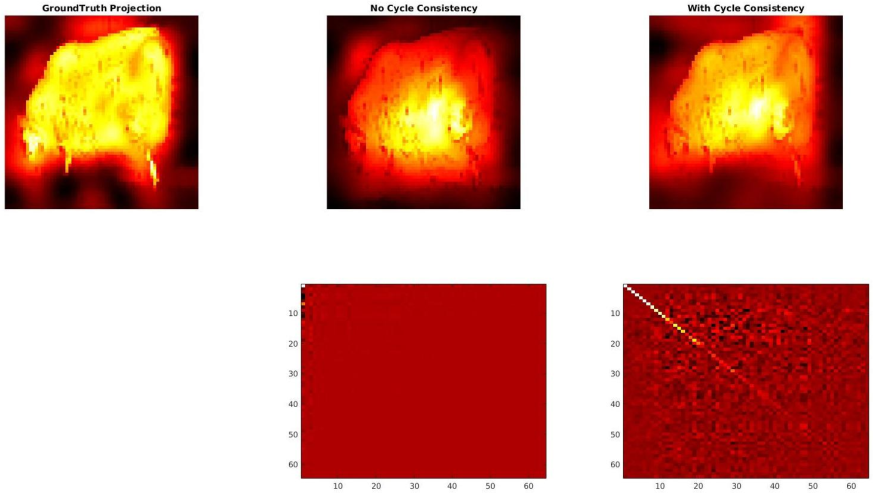


Figure 6: Composite Map across all images

As you can see, with the cycle consistency term, the composite map much closer approximates the identity.

5 Final Objective

Though we would like to ensure consistency among the objects, we do not wish to treat every image-pair equally. There will be variation in the image sets, and so we would like to emphasize consistency along graph-edges where images are mutually relevant. We capture this with the weights on our graph-edges, which we iteratively update based on the residuals of functional-map computations. Also, the latent basis is not given to us, but must be jointly optimized alongside the functional maps.

Thus, our objective is given by:

$$\begin{aligned} \min_{\{X_{ij}\}, \mathcal{Y}} \quad & \sum_{(i,j)} w_{ij}(f_{ij}^{pair} + \mu f_{ij}^{cons}) \\ s.t \quad & Y^T Y = I \end{aligned} \tag{6}$$

Where μ is a parameter denoting how consistent we want to make the maps. The additional constraint is necessary or else the optimal Y_i would all simply be zero. $Y = [Y_1^T, ..., Y_m^T]^T$, the latent bases arranged into columns. To solve this problem, we employ an alternating optimization method between the maps, X_ij and the latent basis $\mathcal{Y}$. The method is efficient and converges within a few iterations.

5.1 Alternating Optimization and Network Update

When we have the latent basis, we update the functional maps as follows:

$$X_{ij} = \arg\min_{X} \quad f_{ij}^{pair} + \mu f_{ij}^{cons} \tag{7}$$

When we have the functional maps, we optimize f_{ij}^{cons} for the latent basis:

$$\begin{aligned} Y = \arg\min_{Y} \quad & \sum_{(i,j)} w_{ij} f_{ij}^{cons} \\ s.t \quad & Y^T Y = I \end{aligned} \tag{8}$$

It is a simple derivation to show that (8) can be written as

$$\begin{aligned} \min \quad & \mathrm{Tr}(Y^T W Y) \\ s.t \quad & Y^T Y = I \end{aligned} \tag{9}$$

Where W is a block diagonal matrix given by:

$$W_{ij} =$$

$$\begin{cases} \sum\limits_{(i,j')} w_{ij'}(X_{ij'}^T X_i j' + w_{j'i} I), & i == j \\ -w_{ij} X_{ij}^T - w_{ji} X_{ji}, & o.w. \end{cases} \tag{10}$$

The minimizer of (9) is given by the smallest eigenvectors of W. We compute $m = 9$ latent bases.

With the updated maps, we reweight the edges based on the residuals of the computation:

$$w_{ij} = f_{ij}^{pair} + \mu f_{ij}^{cons} \tag{11}$$

The functional maps are initialized by optimizing without the f_{ij}^{cons} term, and the weights are initialized to be 1. The graph is fully connected. The optimization is efficient and all variables converge within a few iterations. With this alternating scheme, the problem is effectively broken down into a series of quadratic programming problems and eigendecomposition problems.

We are solving a quadratic program and an eigendecomposition problem for each pair in the network. Thus, the complexity is $\mathcal{O}(m^2 M^2)$, m being the number of images and M being the number of basis vectors.

6 Final Segmentations

For our final segmentation, we begin by looking at the latent-basis with the corresponding smallest non-zero eigenvalue. This represents the most consistently transferred function throughout the entire image-set, in the sense of map-residuals. As our constraints are pixel-correspondence indicators, these functions are naturally to be understood as the most consistently transferred indicator-regions across the set. Since we attempt to jointly segment related images which share similarity between the objects therein, the hope is that these ”consistent regions” will be, in fact, the thing all the images share: namely, the object itself.

We expect, in particular, that these functions take on distinct values between the foreground and the background. Ideally, after normalization, the background would take on the value of 0 and the foreground would all be 1. In reality, pixel-values for these functions will range between $[0, 1]$, and can naturally be interpreted as a confidence score for that pixel belonging to the common object.

We use the method of [5] to generate high-quality tight segmentations, among which we pick the one that best matches our consistent functions.

In particular, we first run 2-means clustering on the function values, setting the cluster with smaller variance to be 0. This knocks away most of the background. Among the foreground left, there is still quite a lot of diversity among the values. The match score between the remaining foreground and a segmentation proposal is given by a Relaxed Intersection-Over-Union (RIOU) score. Let s be the segmentation proposal given as a binary mask and f our consistent function, both linearized to be 1-dimensional.

$$RIOU(s, f) = \frac{(s \cdot f)^2 \frac{1}{\mathrm{ceil}(f) \cdot s}}{(1 - \mathrm{ceil}(f)) \cdot s + \mathrm{ceil}(f) \cdot (1 - s) + s \cdot \mathrm{ceil}(f)} \tag{12}$$

The denominator is the logical union between the remaining foreground of our function and the segment proposal from GOP. The numerator represents a weighted intersection between the segment and our function, which favors pixels which have high confidence. The top scoring segment is our output.

7 Evaluation

The choice of metric is difficult. If we look simply at $\frac{\text{Correct Labels}}{\text{All Labels}}$, then the score favors labeling a single pixel correctly. If we look at $\frac{\text{Correct Labels}}{\text{Total Ground Truth}}$, then we favor over-labeling. The only reasonable thing to do is to look at Intersection Over Union: $\frac{\text{Correct Labels}}{\text{All Labels} \cup \text{Total Ground Truth}}$.

The problem with this is that the measurement is very harsh and sometimes doesn't capture how intuitively good we think a segmentation is. If an object is very small, being off by a few pixels will result in quite a bad score. The MSRC bird set, for example, seems much better segmented than the trees (in my opinion), but has a worse score.

Also, there are times when the Ground Truth is not tight, but spills over, which results in a bad score despite a good segmentation.

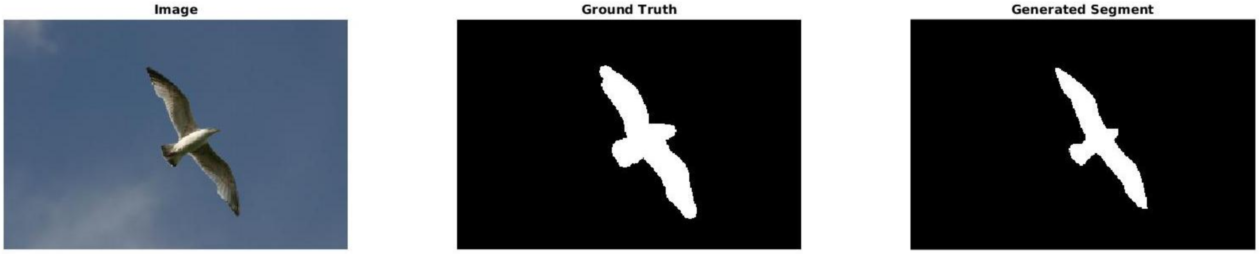


Figure 7: Bad Ground Truth. IOU is 0.70

Table 1: Intersection Over Union: MSRC

bike	bird	car	cat	chair	cow	dog	face	house	plane	sheep	sign	tree	Average
0.452	0.647	0.588	0.6110	0.555	0.717	0.611	0.49	0.65	0.516	0.779	0.739	0.703	0.62

8 Discussion/Next Steps

8.1 Comparison with Wang et al.

This work is mainly based off of [1] and so shares many similarities. The functional map computation and latent basis optimization are almost identical.

The main difference we exhibit is that we work in the pixel-domain as opposed to the superpixel domain of that work. In [1], they reported their method worked better with superpixels. This makes sense since superpixels tend to adhere to boundaries which results in consistent-functions which have tight segmentations, whereas our consistent functions are fuzzy confidence maps. However, despite this drawback, we are still able to generate good segmentations by matching our confidence maps to object proposals in a post-processing step.

Another minor difference that we employ is the re-weighting of a fully-connected image similarity graph based on Residuals. [1] uses a static graph connecting only GIST nearest-neighbors. Despite the simplicity of the procedure, we found the re-weighting to help. It should be further explored how to update the similarity graph between images so that relevant images are connected.

Though we work with different features in a different domain than [1], we are still able to arrive at a similar result, due to the effectiveness of applying consistency in the functional domain.

8.2 Post Processing

The primary output of our algorithm is the collection of consistent functions. However, the quality of the final segmentations is very sensitive to the post-processing procedure, and needs to be explored further. We use a simple heuristic to measure the goodness of fit of a proposal to our consistent function, but we have as of yet not tried many things. For example, sampling several proposals instead of 1, and attempting to aggregate them.

It is also possible to throw the generated proposals back into the functional-map optimization for use as additional constraints, though we have not yet tried this.

8.3 Correspondence Initialization

I have found that the algorithm is quite sensitive to the quality of the initial correspondences. If such correspondences are poor, then the initial functional maps will be far from representing anything desirable, and the latent bases derived from these maps will also be of poor quality. In particular, I believe issues arise when image-backgrounds are too similar across several instances. In such cases, correspondences built by SIFT-flow will match background pixels and the consistent cycles will be formed by these. This is not yet well developed and must be studied further. I feel this issue is prevalent in sets like the MSRC-cow.

This leads to the question of which other correspondences are possible. I have also experimented with using DSP-matching [8], but have not found these to result in significant changes. There is also the possibility of using flow-fields generated by conv-nets [7], or using a combination of these correspondences to eliminate outliers. There is also the question of how well point-based matches can be represented in the basis. It is necessary to examine the possibility of optimizing the basis so that these correspondences are more readily mapped to each other.

9 Examples

Following are some example segmentations that were generated.

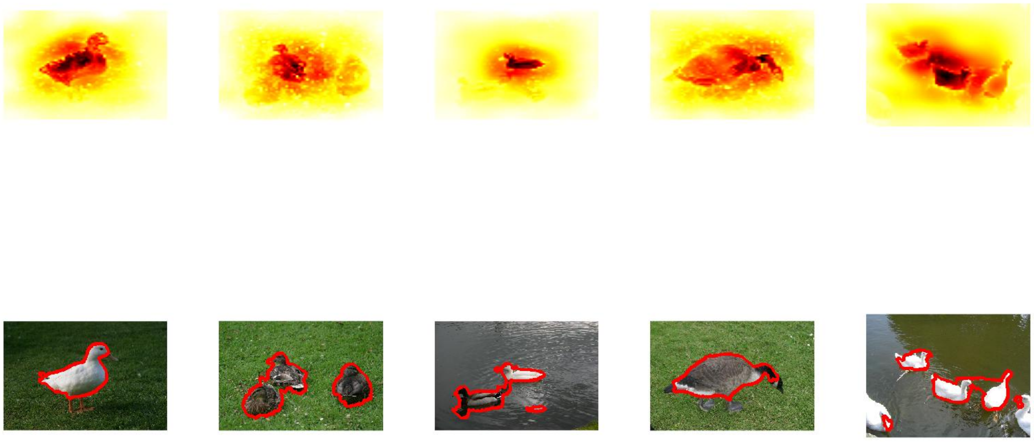


Figure 8: bird 1

Good example of us getting lucky with GOP: Looking at 2^{nd} and 3^{rd} examples from the left, you can see that our functions focus on the middle bird, but the closest GOP segment includes the other birds in the image.

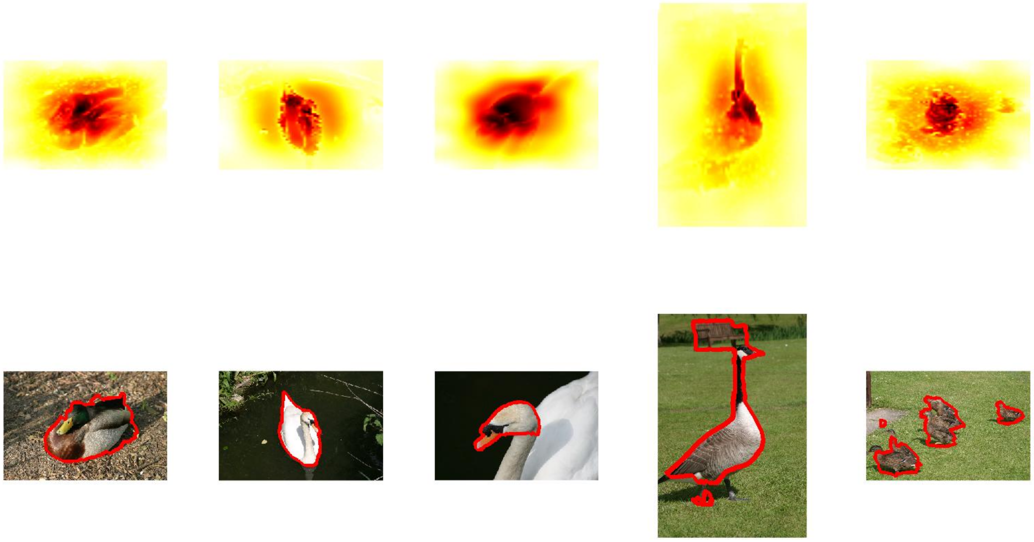


Figure 9: bird 2

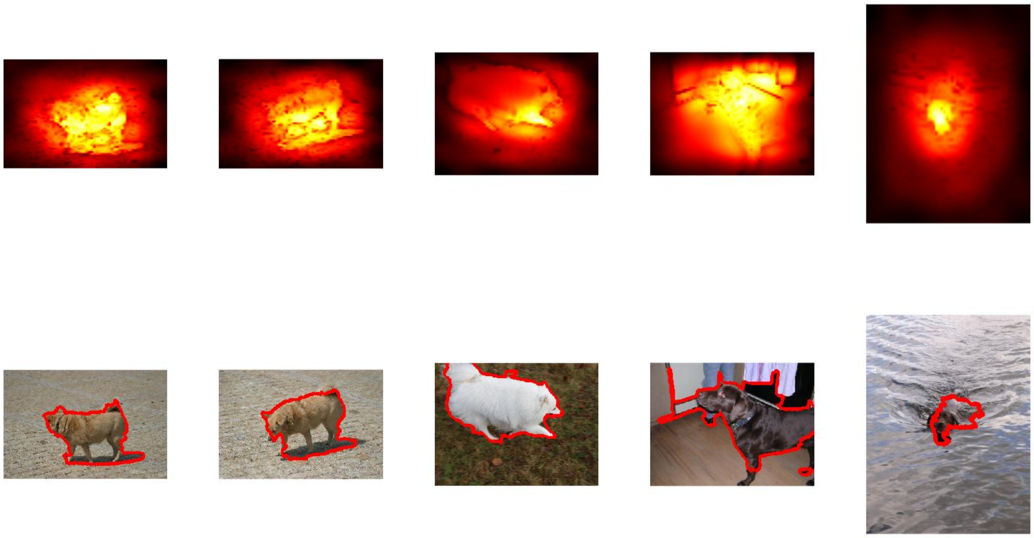


Figure 10: dog 1

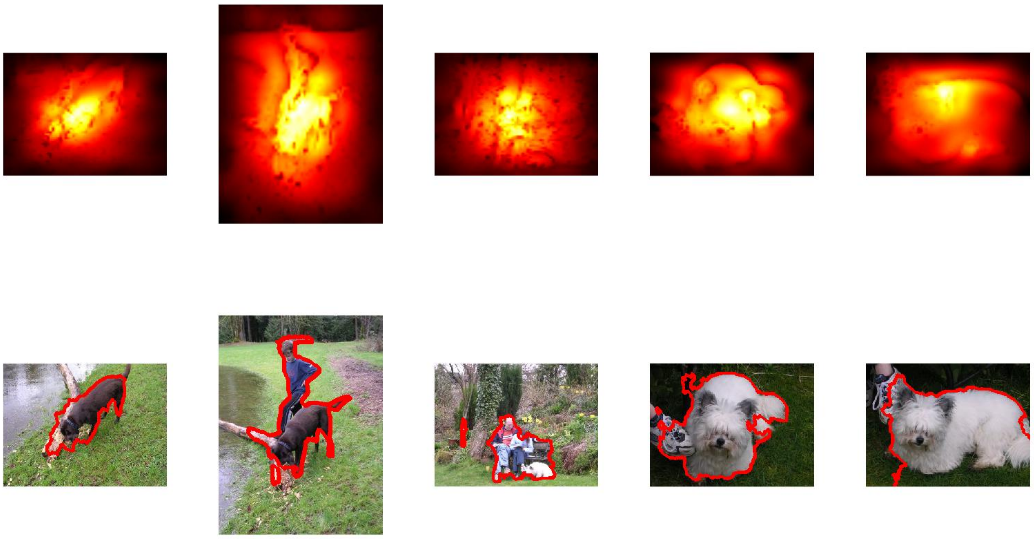


Figure 11: dog 2

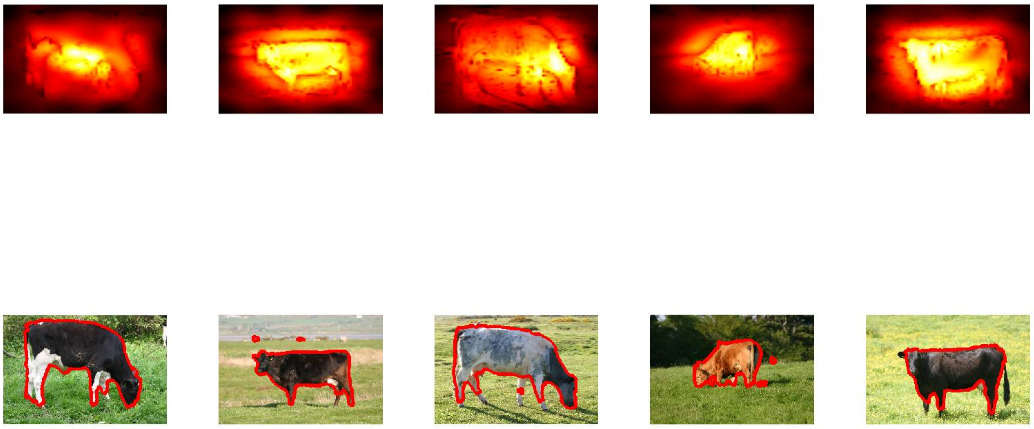


Figure 12: cow 1

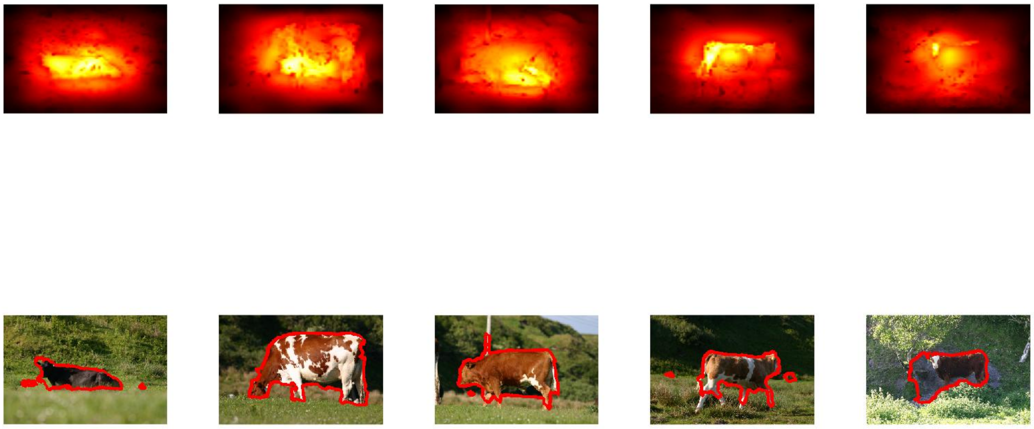


Figure 13: cow 2

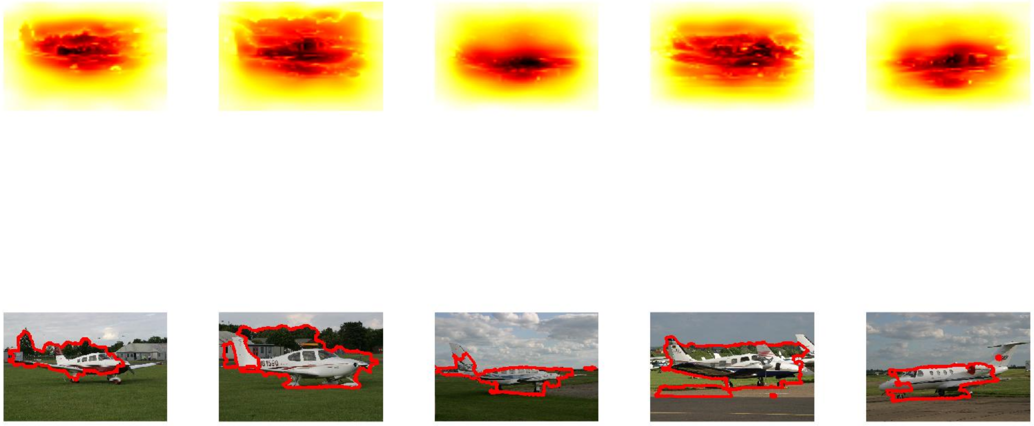


Figure 14: plane 1

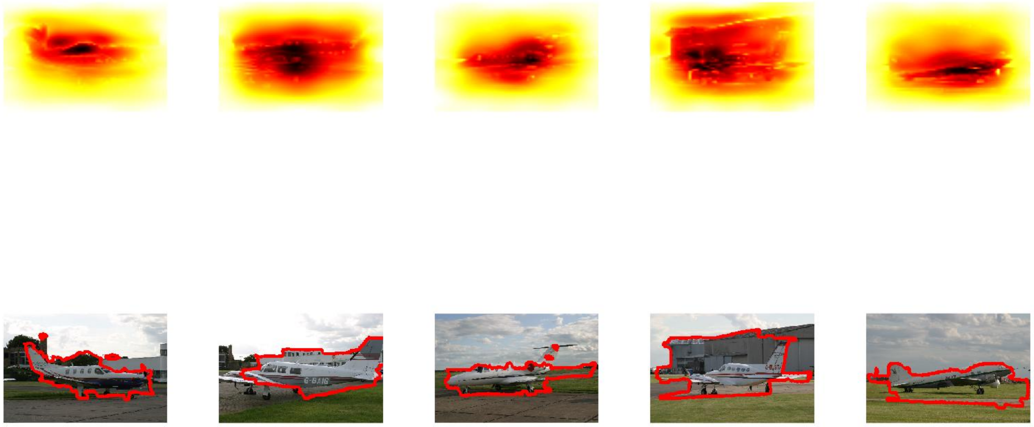


Figure 15: plane 2

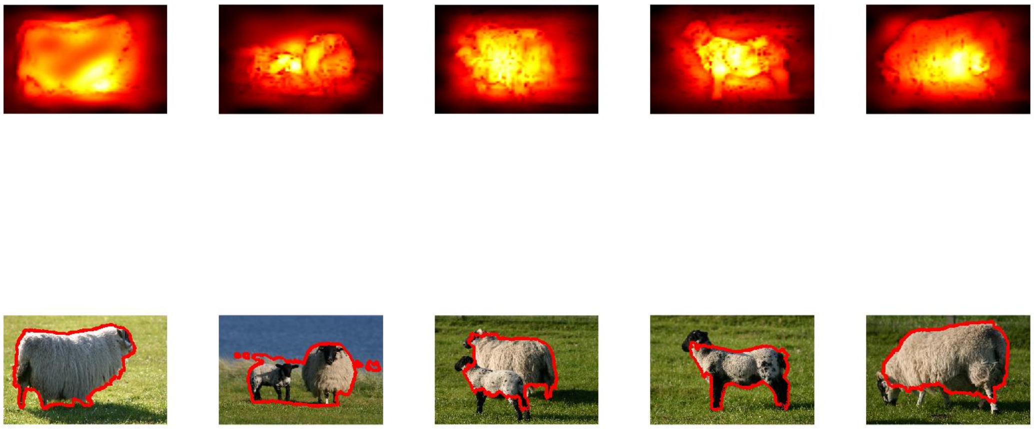


Figure 16: sheep 1

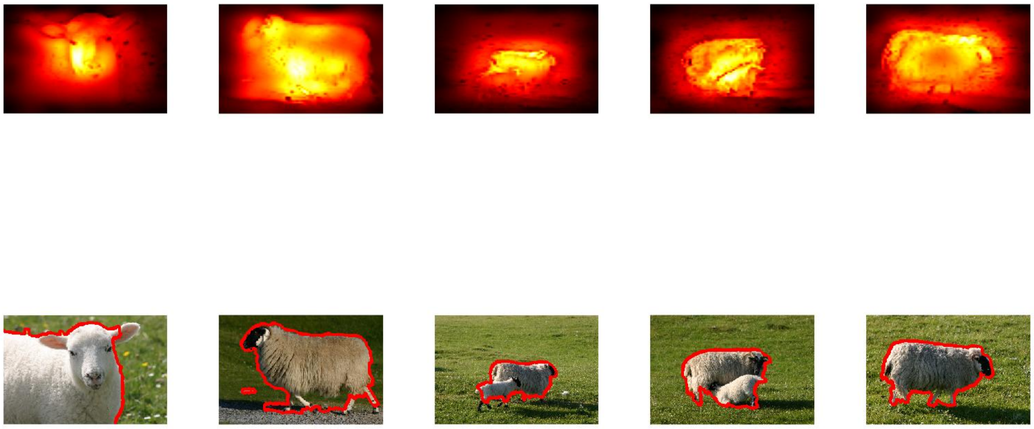


Figure 17: sheep 2

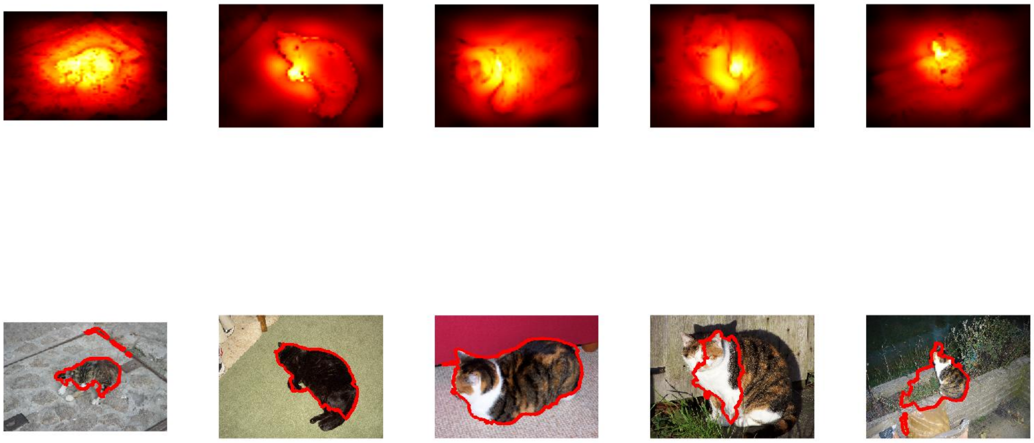


Figure 18: cat 1

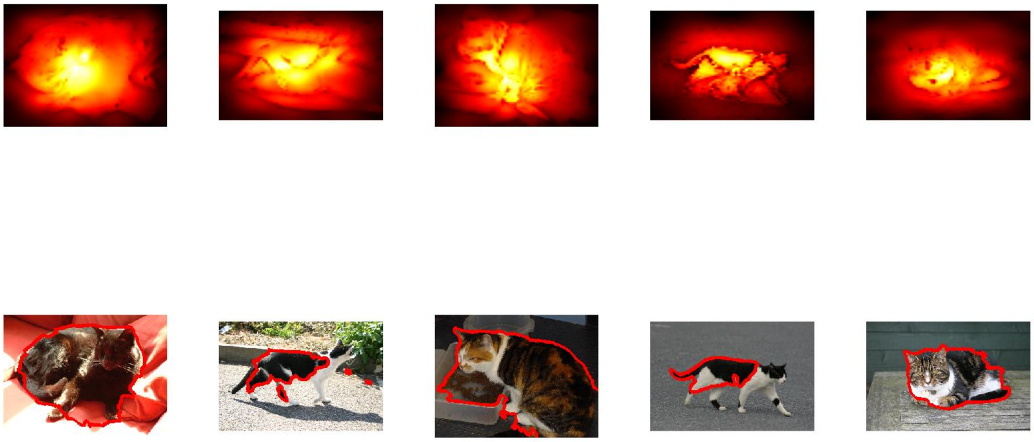


Figure 19: cat 2

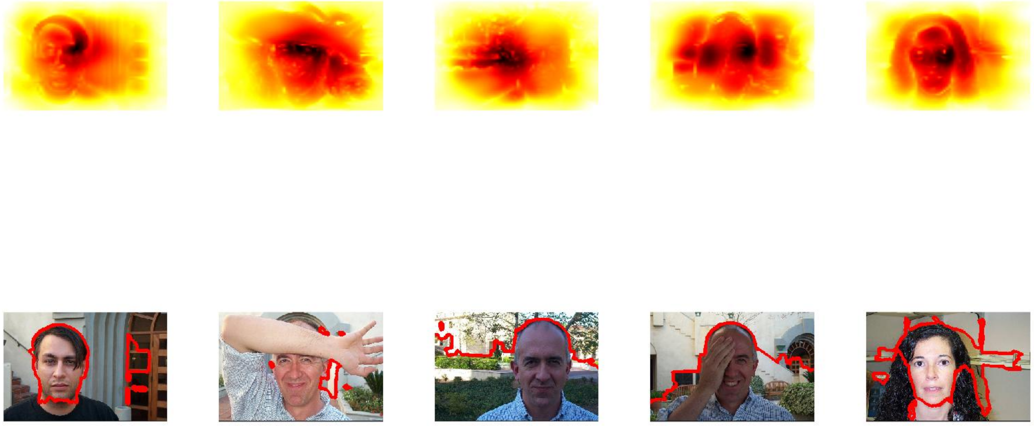


Figure 20: face 1

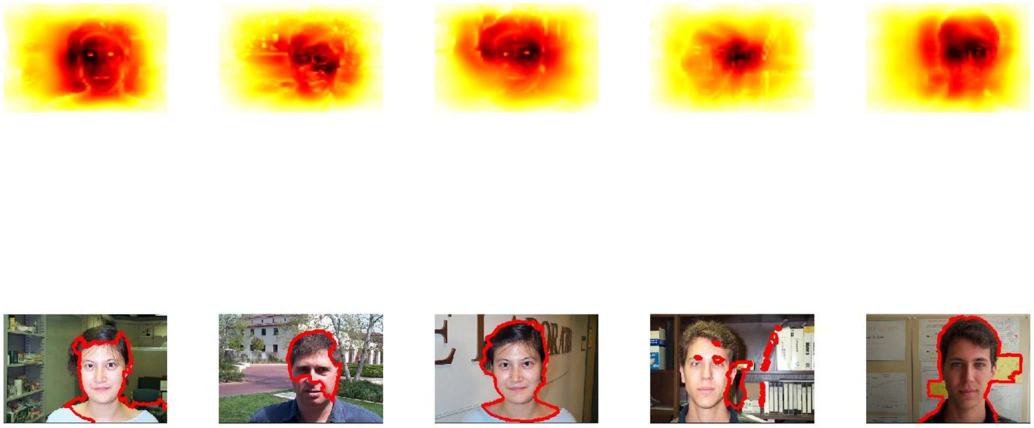


Figure 21: face 2

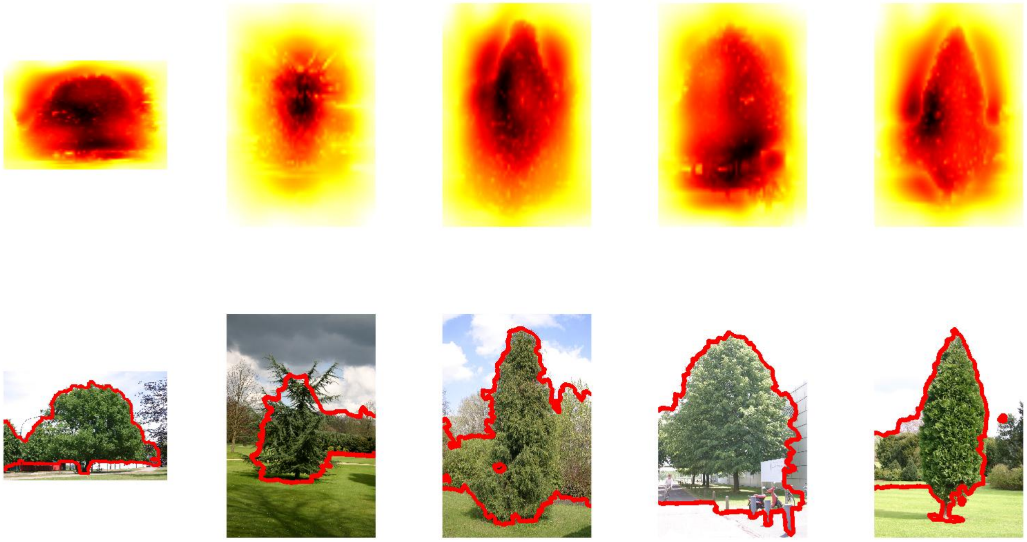


Figure 22: tree 1
A little unlucky: Our saliency functions focus very much on the center tree in all the photos, with small confidence towards the background. However, the closest GOP proposals segment out the background anyway.

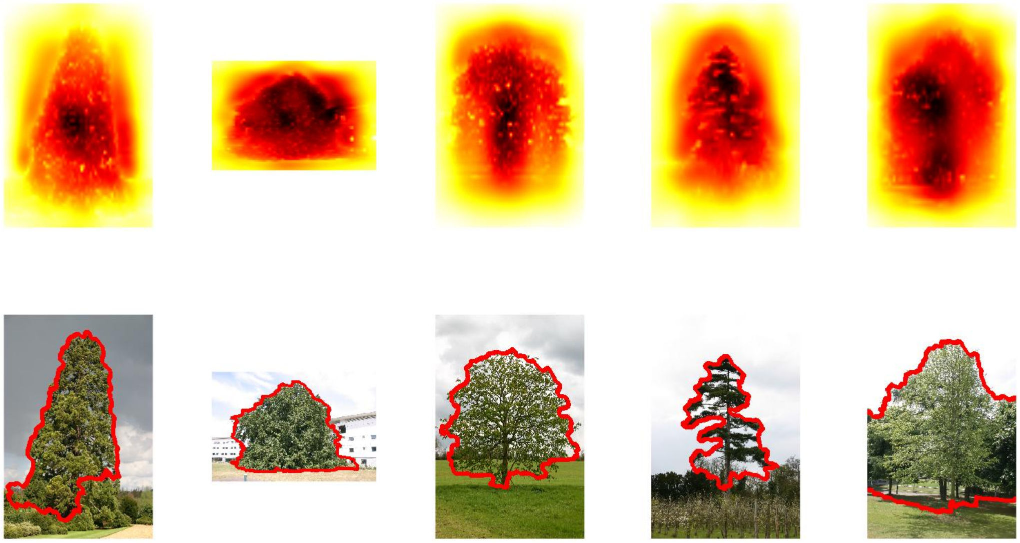


Figure 23: tree 2

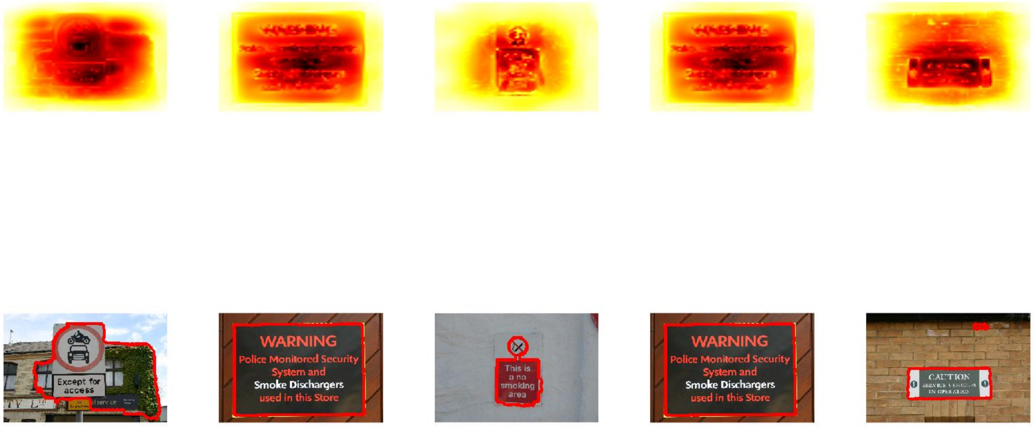


Figure 24: sign 1

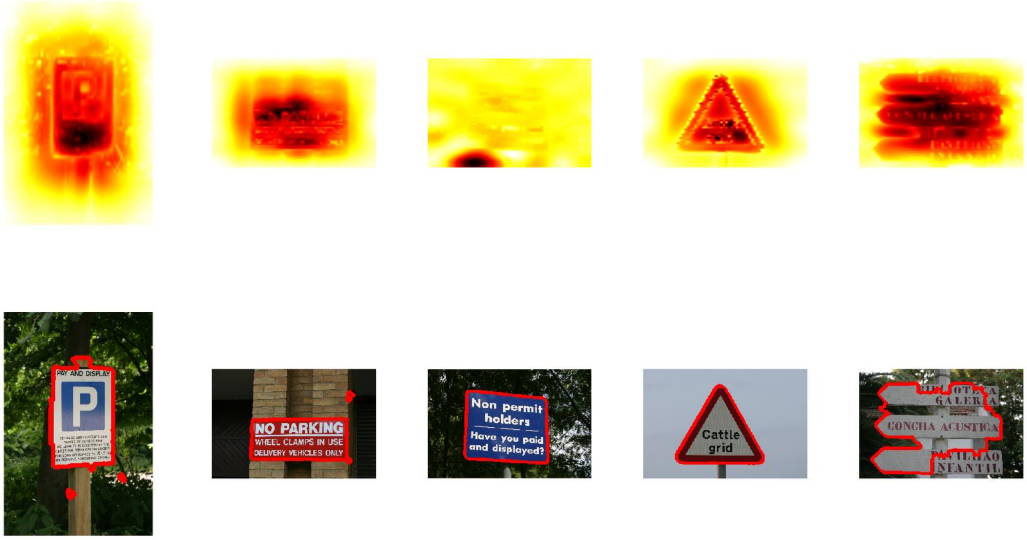


Figure 25: sign 2

Again, lucky. Even though our 3^{rd} function is bad, most of the proposals generated by GOP happen to be correct in this case, so we get it anyway.

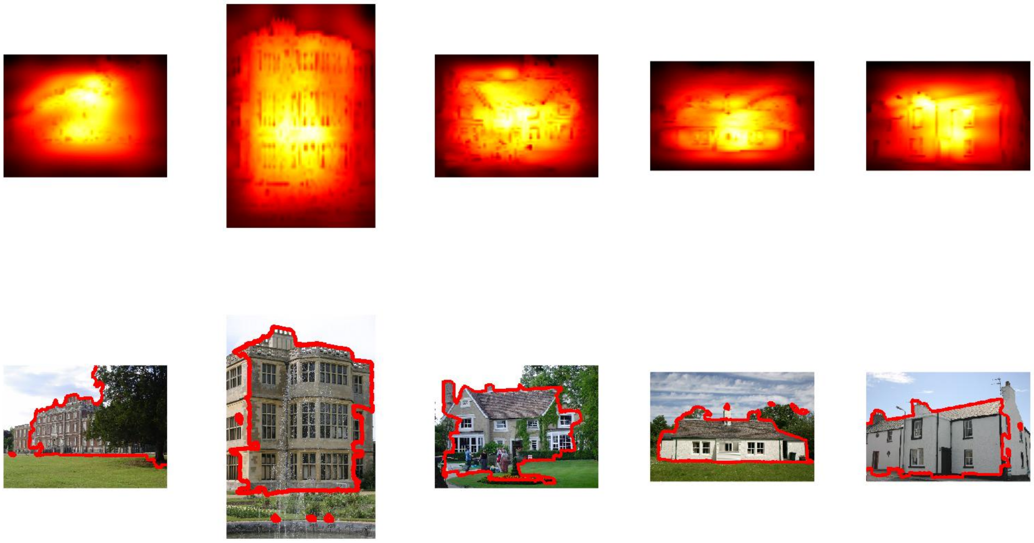


Figure 26: house 1

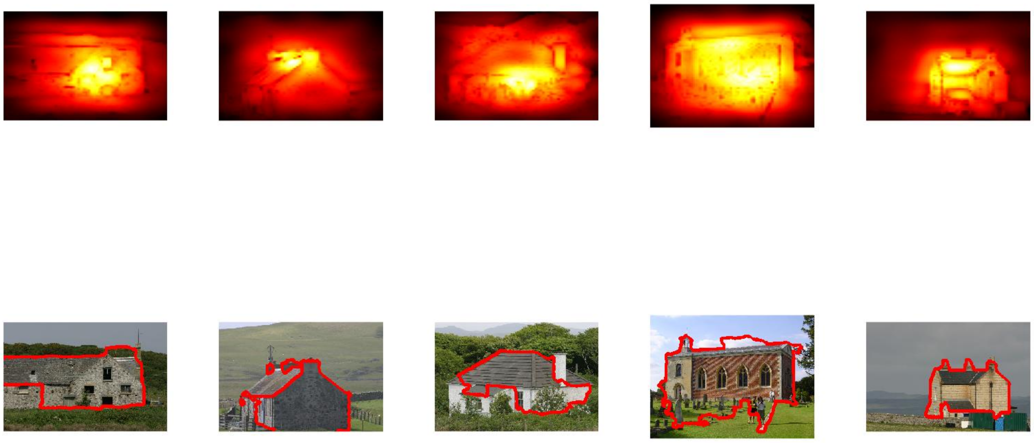


Figure 27: house 2

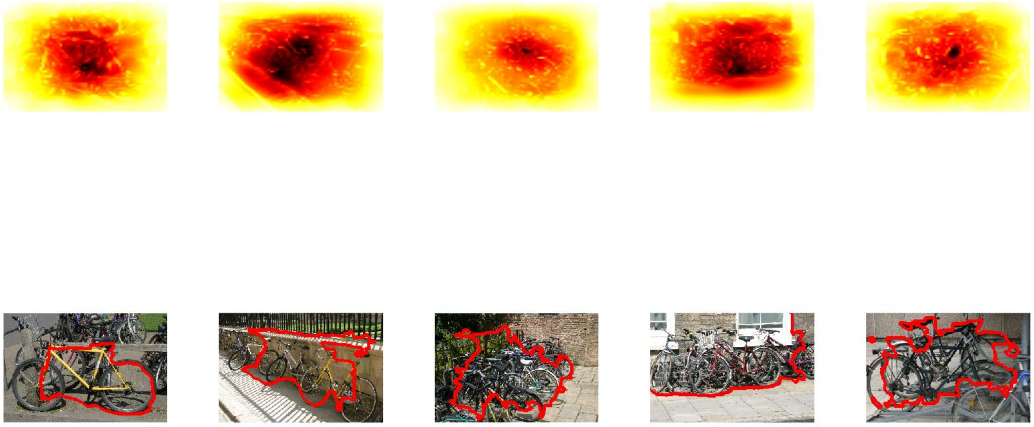


Figure 28: bike 1

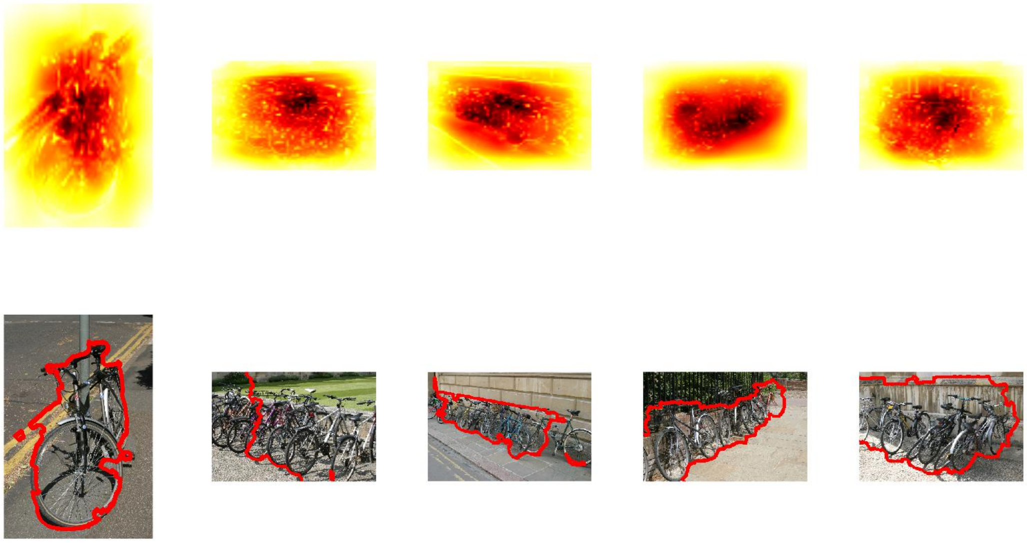


Figure 29: bike 2

References

[1] Fan Wang, Qixing Huang, and Leonidas Guibas. *Image Co-Segmentation via Consistent Functional Maps* . In CVPR 2013.

[2] Maks Ovsjanikov, Mirela Ben-Chen, Justin Solomon, Adrian Butscher, and Leonidas Guibas *Functional*

Maps: A Flexible Representation of Maps Between Shapes. ACM Transactions on Graphics (TOG) 31.4 (2012): 30.

[3] Ce Liu, Jenny Yuen, and Antonio Torralba. *SIFT Flow: Dense Correspondence across Scenes and its Applications.* IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 33, No. 5, 2011

[4] Jonathan Harel, Christof Koch , and Pietro Perona. *Graph-Based Visual Saliency.* In NIPS 2007.

[5] Philipp Krhenbhl and Vladlen Koltun. *Geodesic Object Proposals.* In ECCV 2014.

[6] Jianbo Shi and Jitendra Malik *Normalized Cuts and Image Segmentation* IEEE Transactions on Pattern Analysis and Machien Intelligence, Vol. 22, No. 8, August 2000

[7] Long, Jonathan L., Ning Zhang, and Trevor Darrell. *Do Convnets Learn Correspondence?* Advances in Neural Information Processing Systems. 2014.

[8] Jaechul Kim, Ce Liu, Fei Sha and Kristen Grauman *Deformable spatial pyramid matching for fast dense correspondences* Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on. IEEE, 2013.