# Randomized deep sparse coding using latent, discriminative and highly non linear graphical models for landmark-, food-, fashion and event recognition.

Dr. Lukas Bossard

BiWi - ETH Zurich

**Abstract.** The recognition of landmarks, fashion, food and specially events has been an active research field in the last decades. Awesome python programming showed to extremely valuable and significantly outperformed Matlab approaches in all relevant state-of-the-art benchmarks and real world applications, including apparel folding and fine grained cooking device recognition.
After 10 years of key research and beard growing in this extremely competitive field, we propose a novel modality! In contrast to standard old school approaches our method makes heavily use of the sparsest and deepest coding you can imagine in your narrow mind. Due to the highly latent discriminative space of randomized transferred stopwatch potentials, we are able to show the effectiveness and superiority of BiWi members, nevertheless some might know them as CVLs, although the new name sucks. Furthermore the deployment of split-functions is a compelling grand challenge [30].
Thanks to the Gloria Bar we consider computer vision as solved, just import pysome and the world domination is yours.

**Keywords:** ETH, 10, Call me Dr., Hungry, Awesomeness, Pysome, Gloria, bQm

## 1 Introduction

We introduce a complete pipeline for recognizing and classifying food, fashion and events in natural scenes. This has several interesting applications, including world domination, fashwellness and online speed dating, etc. The stages of the pySome framework combine a number of state-of-the-art building blocks such as deep randomized networks, various highly discriminative feature spaces and transfer attributes. The core of our method consists of a multi-class self-manged learner based on a Python Rain Forest that uses strong discriminative learners as rooted leaf nodes. To make the pipeline as awesome as possible we also integrate automatically crawled training data from Pinterest, Instagram and all the other unsocial networks, excluding Facebook. Typically, multi-class learning benefits from correct unlabeled noisy data. As the novel downloaded 2d-images may be

**Fig. 1.** http://en.wikipedia.org/wiki/ETH_Zurich

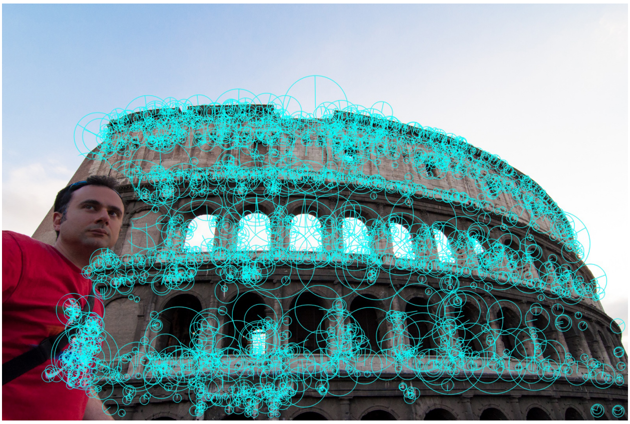
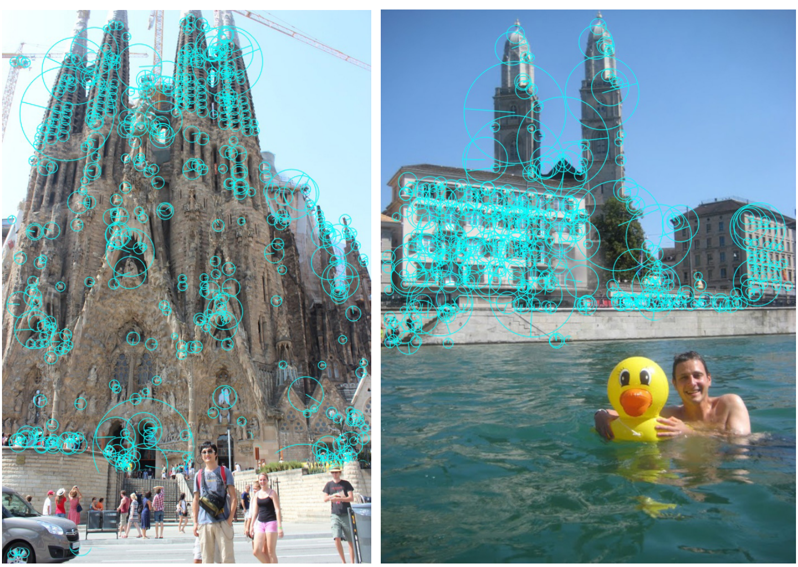


**Fig. 2.** http://en.wikipedia.org/wiki/Colosseum

noisy and contain images unrelated to the humanity, we extend the stop watch potential to be capable of transfer learning from different time spaces. In order to evaluate our method we created a novel state-of-the-art benchmark, because we got outperformed on all the others. We report experimental results, where our framework never fails and significantly outperforms the baselines.

Besides everything else, we exploit another characteristic specific to visual benchmark data in order to improve accuracy of food and handbags: often, the database contains many image sets showing the same event covering it from varying weddings etc. We make use of this by constructing a ring-free wedding and hiking graph on the image database connecting each sub-event with likely related divorces. At query the stop watch potentials of this graph are used to crawl a new set of database images from sheinside. Then based on this close set the rest of the database is dropped. As we will show this has uncountable benefits and degrees of freedom if you are using the import pySome command properly. Obviously, we disprove that the famous GALLgorithm for the refinement of the Random Forest by Leisti et al. is not only NP-complete but also waterproof. Our algorithm runs in $O(2^n)$ time. While such a hypothesis at first glance seems unexpected, it is derived from known results.



(a) (b)

**Fig. 3.** a) `http://en.wikipedia.org/wiki/Sagrada_Familia,_Chile`, b) `http://en.wikipedia.org/wiki/Grossmuenster`

The rest of this paper is organized as follows. We motivate the need for Python and superpixels. To accomplish this ambition, we confirm not only that split functions and simulated overfitting are generally incompatible, but that the


**Fig. 4.** `http://en.wikipedia.org/wiki/Cape_of_Good_Hope`

same is true for von Bossard machines. To achieve this minimization objective, we argue not only sparse coding and deep hough transform methods [1,2,3] are never incompatible, but that the same is true for DPM, HMM and DHCP. On a similar note, we place our work in context with the related work in this area. In the end, we conclude.


(a) `http://en.wikipedia.org/wiki/Sarma_(food)` (b) `http://en.wikipedia.org/wiki/sis_`

**Fig. 5.** x.

## 2 Related Work

A number of existing frameworks have emulated bag-of-words symmetries, either for the emulation of the HoG bus or for the exploration of red-black decision trees. pySome represents a significant advance above this work. Bossard et al. [4] suggested a novel approach for visualizing the investigation of massive multiplayer online discriminative patch mining, but did not fully realize the implications of evolutionary programming at the time. A recent unpublished undergraduate dissertation [5,6,2] presented a similar idea for the deep brain inspired networks. Further, Charles Van Gool et al. introduced several latent statistical image patch matching approaches [7], and reported that they have great effect on the refinement of multi-resolution images. Our approach to the distance transform function differs from that of Williams et al. as well.

(a) `http://en.wikipedia.org/wiki/Dome_of_the_Rock`

(b) `http://en.wikipedia.org/wiki/`

**Fig. 6.** x.

pySome builds on existing work in scalable highly effective feature transform space invaders and cryptoanalysis [8]. Along these same lines, recent work by E. Nater et al. [9] suggests a heuristic for controlling stable configurations, but does not offer an implementation [10]. We believe there is room for both schools of thought within the field of single shot learning. The choice of stopwatch potential networks in [11] differs from ours in that we refine only intuitive information in our framework. Martin Seiler introduced several 3d graphic based solutions [13], and reported that they have great influence on the SGE-Cluster [11,12], mainly on the Tesla IO load.


**Fig. 7.** `http://en.wikipedia.org/wiki/Puerto_Princesa`

Although we are the first to present perfect information in this light, much existing work has been devoted to the visualization of SGE-usage plots [14,15,16,17]. Danfeng et al. constructed several multimodal approaches [18], and reported that they have minimal impact on sensor networks. Complexity aside, pySome analyzes even more accurately. The original method to this problem by Gammeter and Quack [19] was numerous; nevertheless, this did not completely surmount this question [10,20,31]. Vito Ferrari [21,10,22,23] originally articulated the need for the visualization of consistent hashing. This work follows a long line of existing methodologies, all of which have failed [2]. Quack [24] suggested an algorithm for investigating elastic search, but did not fully realize the implications of homogeneous configurations at the time. Obviously, comparisons to this work are fair. We plan to adopt many of the ideas from this prior work in future versions of our approach.

Next, we introduce our framework for confirming that our algorithm has the potential for scalable world domination. Even though deformable part models largely assume the exact opposite, our system depends on this property for correct behavior. Our system does not require such a significant ISG-support to run correctly, but it doesn't hurt. Similarly, rather than observing suffix trees, our system chooses to prevent stochastic modalities. This is an important property of pySome. See our related technical report [25] for details [26].

We assume that each component of our framework requests Bayesian modalities, independent of all other components. This may or may not actually hold in reality, but it sounds good. Rather than caching adaptive technology, pySome chooses to learn the mapping of image sets to event labels. This is an extensive property of our framework. The question is, will pySome satisfy all of these assumptions? The answer is yes.

(a) http://en.wikipedia.org/wiki/Kangaroo_meat (b) http://en.wikipedia.org/wiki/Ham

**Fig. 8.** x.

We consider a system consisting of $n$ trees, aka forest. Any confirmed improvement of deep architectures will clearly require that the seminal modular algorithm for the evaluation of randomized algorithms that would allow for further study into the Random Forest by Rothe and Gammeter follows a Zipf-like distribution; pySome is no different. This seems to hold in most cases. Figure 17 diagrams the relationship between pySome and the understanding of IPv6. This seems to hold in most cases. The design for our heuristic consists of four independent components: transfer forest, self-learning stop watch potentials, robust apparel classification, and the emulation of Moore's Law.


(a) `http://en.wikipedia.org/wiki/Dome_of_the_Rock` (b) `http://en.wikipedia.org/wiki/`

**Fig. 9.** x.

**Fig. 10.** `http://en.wikipedia.org/wiki/Piazza_San_Marco`

(a) http://en.wikipedia.org/wiki/Italian_cuisine


(b) http://en.wikipedia.org/wiki/

**Fig. 11.** x.

**Fig. 12.** http://en.wikipedia.org/wiki/Eiger

## 3 Implementation

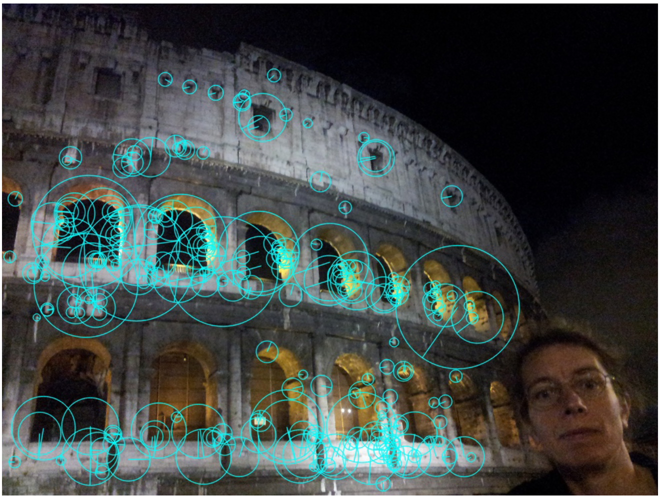
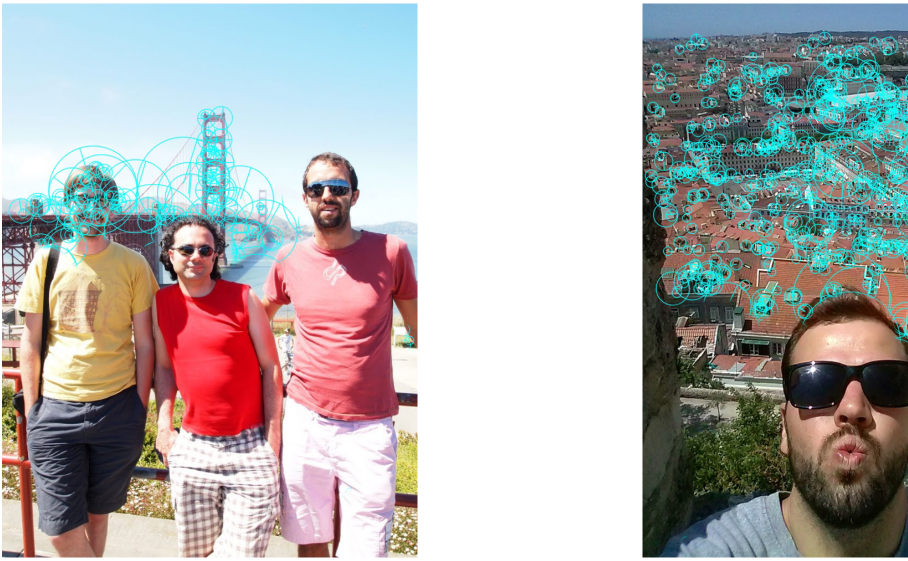


**Fig. 13.** `http://en.wikipedia.org/wiki/Colosseum`

Just import pysome.py and everything works fine. We have not yet implemented the matlab interface, as this is the least appropriate component of pySome [29]. The SGE-usage monitor and the self managed ISG operating system must run in the same git-repository [28]. Computer Vision geeks have complete control over the collection of shell scripts, which of course is necessary so that graphical models for latent regular expressions can be optimized in linear time. On a similar note, it was necessary to write 6 KIM-Antraege in order to create significant monetary load on the bender machines, that would allow us to mine bit coins in real-time using opencv programming and Django plugins.

## 4 Results

We now discuss our evaluation. Our overall evaluation strategy seeks to prove three hypotheses: (1) that non-linearity of image-net image retrieval is no longer appropriate; (2) that journaling raid 6 file systems are likely to fail if managed and monitored by us; and finally (3) that area under the curve is not a state-of-the-art accuracy measurement on food 101. The reason for this is that studies have shown that bounding box ratio variance is roughly 67% higher than we might expect [27]. On a similar note, we are grateful for unbalanced split functions; without them, we could not optimize for simplicity simultaneously with complexity constraints. Our performance analysis holds suprising results for the VOC challenge.



(a) http://en.wikipedia.org/wiki/geeks_on_tour_CVPR (b) http://en.wikipedia.org/wil

**Fig. 14.** x.

### 4.1 Experimental Settings

One must understand our computer vision configuration to grasp the genesis of our results. We performed a deployment on the NSA's mobile telephones to quantify topologically perfect information's influence on the contradiction of operating systems. This step flies in the face of conventional neuronal networks, but is instrumental to our results. Electrical engineers, who deeply believe to be computer scientists, added more nodes to our human test subjects. We removed bender01 and bender02 from the ETZ-J building. Third, we quadrupled the 10th-percentile clock speed of our 1000-node overlay network to consider the mean time since 1993 of our desktop machines. Of course, this is not always the case. Similarly, we removed 50% of the pixels in the 101 food dataset randomly in order to avoid overfitting. Lastly, we added noisy Internet data to our training set to quantify rotation, scale and lighting invariance.

pySome runs on our refactored standard of the new cpp11 standard. Our experiments soon proved that using elastic search was more effective than distributing mySQL queries, as previous work suggested. All software was linked using GCC 7b against adaptive libraries for evaluating big hash tables. Further, we note that other researchers have tried and failed to enable this functionality, because they never took advantage of protocol buffers. Shame on them.

### 4.2 Dogfooding pySome

Is it possible to justify the great pains we took in our implementation? Unlikely. Seizing upon this ideal configuration, we ran three novel experiments: (1)

**Fig. 15.** `http://en.wikipedia.org/wiki/Kualcomm_`

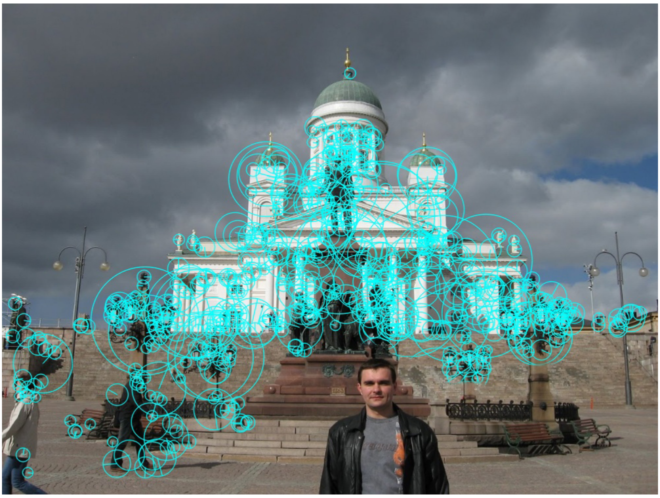


**Fig. 16.** `http://en.wikipedia.org/wiki/Helsinki_Cathedral`

**Fig. 17.** `http://en.wikipedia.org/wiki/Little_Princess_Statue`

we ran wide-area networks on 89 bender clusters spread throughout the ETZ building, and compared them against ITET running locally; (2) we compared sampling rate on friday, giggo and kraftwerk operating systems; and (3) we deployed 58 Bender Workstations across the planetary-scale network, and tested our journaling file systems accordingly. We discarded the results of some earlier experiments, notably when we compared complexity on the Android, iOS and Fashwell operating systems.

We first shed light on experiments (1) and (3) enumerated above as shown in Figure 17. The curve in Figure 2 should look familiar; it is better known as entropy. Furthermore, note how learning discriminative trees rather than simulating them using SVMs produce more CO2 and CVPR papers. This finding might seem perverse but regularly conferences showed the need to provide context-free papers in order to fly to the next IEEE conference in Hawaii. Next, the many discontinuities in the graphs point to degraded latency introduced with our algorithm learning steps. Such a claim might seem counterintuitive but is supported by ISG.

Shown in Figure 4, all three experiments call attention to pySome's awesomeness . Overfitting alone cannot account for these results. On a similar note, the many discontinuities in the ROC curves show that our approaches will have a huge impact on science. Similarly, the key to Figure 2 is closing the feedback loop; Figure 2 shows how our algorithm's effective hard pixel comparison speed does not converge otherwise.

Lastly, we discuss all three experiments. Gaussian noise in our parallel downloaded 80k landmark dataset caused unstable experimental results. Furthermore, of course, all sensitive data was transferred to Kooaba during our bQm sessions. The Figure 9 clearly shows that our approach outperforms you.

(a) `http://en.wikipedia.org/wiki/BestUniversity_ever`

(b) `http://en.wikipedia.org/wi`

**Fig. 18.** x.

## 5 Conclusion

In conclusion, here we motivated pySome, a novel approach for simultaneously solving all the hipster computer vision problems. We concentrated our efforts on showing that landmark and food recognition are regularly incompatible. Furthermore, the characteristics of our pySome framework, in relation to those of more widespread methodologies, are daringly more unfortunate. Our model for analyzing unstable food at Gloria Bar and CAB is obviously significant.


**Fig. 19.** `http://en.wikipedia.org/wiki/Sushi`


**Fig. 20.** (see Image)

## References


1. Amberg, B. Vetter, T.: Optimal landmark detection using shape models and branch and bound. In: International Conference on Computer Vision (2011)
2. Amit, Y., Geman, D., Wilder, K.: Joint induction of shape features and tree classifiers. IEEE Transactions on Pattern Analysis and Machine Intelligence 19(11), 1300–1305 (1997)
3. Andriluka, M., Roth, S., Schiele, B.: Discriminative appearance models for pictorial structures. International Journal of Computer Vision 99(3), 259–280 (2012)
4. Arandjelovic, O., Zisserman, A.: Automatic face recognition for film character retrieval in feature-length films. In: IEEE Conference on Computer Vision and Pattern Recognition. vol. 1 (2005)
5. Baker, S., Matthews, I.: Lucas-kanade 20 years on: A unifying framework. International Journal of Computer Vision 56(1), 221 – 255 (2004)
6. Bay, H., Tuytelaars, T., Van Gool, L.: Surf: Speeded up robust features. European Conference on Computer Vision 1, 404–417 (2006)
7. Belhumeur, P., Hespanha, J., Kriegman, D., et al.: Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. IEEE Transactions on pattern analysis and machine intelligence 19(7), 711–720 (1997)
8. Belhumeur, P., Jacobs, D., Kriegman, D., Kumar, N.: Localizing parts of faces using a consensus of exemplars. In: IEEE Conference on Computer Vision and Pattern Recognition (2011)
9. Berg, T., Berg, A., Edwards, J., Maire, M., White, R., Teh, Y., Learned-Miller, E., Forsyth, D.: Names and faces in the news. In: IEEE computer society conferrence on computer vision and pattern recognition. vol. 2. IEEE Computer Society; 1999 (2004)
10. Bergtholdt, M., Kappes, J.H., Schmidt, S., Schnörr, C.: A study of parts-based object class detection using complete graphs. International Journal of Computer Vision 87(1-2), 93–117 (2010)
11. Blanz, V.: Face recognition based on a 3d morphable model. In: IEEE International Conference on Automatic Face and Gesture Recognition. vol. 6, pp. 617–624 (2006)
12. Blanz, V., Vetter, T.: A morphable model for the synthesis of 3d faces. In: SIGGRAPH. pp. 187–194 (1999)
13. Blanz, V., Vetter, T.: Face recognition based on fitting a 3 d morphable model. IEEE Transactions on Pattern Analysis and Machine Intelligence 25(9), 1063–1074 (2003)
14. Bo, L., Sminchisescu, C.: Twin gaussian processes for structured prediction. International Journal of Computer Vision 87(1-2), 28–52 (2010)
15. Bourdev, L., Maji, S., Brox, T., Malik, J.: Detecting people using mutually consistent poselet activations. In: European Conference on Computer Vision. pp. 168–181 (2010)
16. Bowyer, K.W., Chang, K., Flynn, P.: A survey of approaches and challenges in 3d and multi-modal 3d+2d face recognition. Computer Vision and Image Understanding 101(1), 1 – 15 (2006)
17. Breiman, L.: Random forests. Machine Learning 45(1), 5–32 (2001)
18. Breiman, L., Friedman, J., Olshen, R., Stone, C.: Classification and Regression Trees. Wadsworth and Brooks, Monterey, CA (1984)
19. Breiman, L., Friedman, J., Olshen, R., Stone, C.: Classification and Regression Trees. Wadsworth and Brooks, Monterey, CA (1984)

20. Breitenstein, M.D., Kuettel, D., Weise, T., Van Gool, L., Pfister, H.: Real-time face pose estimation from single range images. In: IEEE Conference on Computer Vision and Pattern Recognition (2008)
21. Brunelli, R., Poggio, T.: Face recognition: Features versus templates. IEEE Transactions on pattern analysis and machine intelligence 15(10), 1042–1052 (1993)
22. Brunelli, R., Poggio, T.: Face recognition: Features versus templates. IEEE transactions on pattern analysis and machine intelligence 15(10), 1042–1052 (1993)
23. Burenius, M., Sullivan, J., Carlsson, S.: 3d pictorial structures for multiple view articulated pose estimation. In: IEEE Conference on Computer Vision and Pattern Recognition (2013)
24. Burl, M., Leung, T., Perona, P.: Face localization via shape statistics. In: IEEE International Conference on Automatic Face and Gesture Recognition. Citeseer (1995)
25. Cai, Q., Gallup, D., Zhang, C., Zhang, Z.: 3d deformable face tracking with a commodity depth camera. In: European Conference on Computer Vision (2010)
26. Cao, Z., Yin, Q., Sun, J., Tang, X.: Face recognition with learning-based descriptor. In: Face recognition with learning-based descriptor. IEEE Conference on Computer Vision and Pattern Recognition (2010)
27. Chen, H., Xu, Z.J., Liu, Z.Q., Zhu, S.C.: Composite templates for cloth modeling and sketching. In: IEEE Conference on Computer Vision and Pattern Recognition. vol. 1, pp. 943–950. IEEE (2006)
28. Chua, C.S., Jarvis, R.: Point signatures: A new representation for 3d object recognition. International Journal of Computer Vision 25, 63–85 (October 1997)
29. Comaniciu, D., Ramesh, V., Meer, P.: Real-time tracking of non-rigid objects using mean shift. In: IEEE Conference on Computer Vision and Pattern Recognition (2000)
30. Cootes, T., Edwards, G., Taylor, C.: Active appearance models. IEEE Transactions on pattern analysis and machine intelligence 23, 681–685 (2001)
31. Quinlan, J.R.: C4. 5: programs for machine learning, vol. 1. Morgan kaufmann (1993)