Sketch Classification and Classification-driven Analysis using Fisher Vectors

Rosália G. Schneider*
ESAT-PSI-VISICS, iMinds MMT
Katholieke Universiteit Leuven, Belgium

Tinne Tuytelaars†
ESAT-PSI-VISICS, iMinds MMT
Katholieke Universiteit Leuven, Belgium

Figure 1: *Sketches are a high-level representation that does not always convey enough information to distinguish between different categories of objects. Ambiguous sketches - as the ones above - pose a challenge in evaluating computer methods for sketch recognition.*

Abstract

We introduce an approach for sketch classification based on Fisher vectors that significantly outperforms existing techniques. For the TU-Berlin sketch benchmark [Eitz et al. 2012a], our recognition rate is close to human performance on the same task. Motivated by these results, we propose a different benchmark for the evaluation of sketch classification algorithms. Our key idea is that the relevant aspect when recognizing a sketch is not the intention of the person who made the drawing, but the information that was effectively expressed. We modify the original benchmark to capture this concept more precisely and, as such, to provide a more adequate tool for the evaluation of sketch classification techniques. Finally, we perform a classification-driven analysis which is able to recover semantic aspects of the individual sketches, such as the quality of the drawing and the importance of each part of the sketch for the recognition.

CR Categories: I.4.8 [Computer Vision]: Scene Analysis—;

Keywords: sketches, classification, Fisher vectors

Links: DL PDF

1 Introduction

Sketching is a natural way of expressing some types of ideas. It conveys information that can be really hard to explain using text, and at the same time it does not require a tremendous amount of effort. It is also a suitable communication tool for children or illiterate people. As human-computer interaction moves towards easier and more high level languages, sketching will certainly continue to have its place in all sorts of applications, including image [Eitz et al. 2011] and shape [Eitz et al. 2012b] retrieval, shape modeling [Olsen et al. 2009] [Schmidt et al. 2006] and character animation [Davis et al. 2003].

In a different perspective, sketching is possibly the most high-level and sparse type of visual media that can be understood by humans, which makes it an interesting object of study in computer vision. Why we can understand sketches so well and whether we can teach computers to do the same are research questions still in need of an answer.

We present a technique for sketch classification that performs significantly better than the state-of-art. Using the TU-Berlin sketch benchmark [Eitz et al. 2012a], we achieve a recognition rate of 68.9%, which is an absolute improvement of 13% over their results. Also, with these results, we come close to the accuracy achieved by humans, which is 73%. Unfortunately, it might be too soon to say computers are performing comparably to humans in this task.

Before looking into how humans understand sketches, we need to determine when it is possible for humans to understand sketches. More specifically, when does a sketch contain enough information to allow it to be unmistakably put into a specific category? As can be seen in Figure 1, this is not always the case. We discuss the specific reasons for the low performance achieved by humans in the TU-Berlin benchmark, and modify it to make it less sensitive to the types of problems we found.

Finally, we perform a data-driven analysis of sketches based on the classification method. We get sound results when determining which sketches are good/poor representatives of a class, performing consistently better than [Eitz et al. 2012a]. Then, we analyze which parts of the sketch are most important for recognition. These results usually describe the most basic features of a sketch, and provide intuition on how the computer understands the different categories.

The **contributions** of our work are:

- State-of-the-art results in sketch classification that are comparable to human performance in an existing benchmark;
- A modified benchmark that is more suitable for the evaluation of classification algorithms;
- A classification-driven analysis that can extract semantic information from sketches.

The rest of this paper is structured as follows. Section 2 reviews existing work that is related to ours. In Section 3, we explain the

*e-mail:rgaliazz@esat.kuleuven.be
†e-mail:tinne.tuytelaars@esat.kuleuven.be

application of Fisher vectors to sketches and discuss the results obtained on the existing benchmark. An analysis of the benchmark is performed in Section 4, and we introduce a new benchmark for the evaluation of sketch classification algorithms in Section 5. Finally, we present a classification-driven strategy for the analysis of sketches in Section 6. Section 7 concludes the paper.

2 Related Work

Sketch classification is closely connected to image classification. We will first review the existing work in image classification that is most related to ours, and then provide an overview of existing methods that are specific for sketches.

2.1 Image Classification

The problem of image classification is defined as follows: Given a number of classes and a set of images for each class (training set), find the class labels for a disjoint set of images (testing set). The first step in this process is deciding which image representation to use. Once we have a reasonable way to describe the image, we can decide which category the image belongs to by simply finding the nearest neighbor in the training set, or by using a standard classifier (Support Vector Machines [Hearst et al. 1998] are a popular choice). We discuss here the different image representations commonly used.

Bag-of-visual-words. One of the most widely used techniques for image classification is bag-of-visual-words [Sivic and Zisserman 2003] [Csurka et al. 2004]. It consists of calculating a dictionary of visual words (that can be any type of descriptor of a patch, *e.g.*, SIFT [Lowe 2004]) and representing the image using this dictionary. More specifically, a number of representative SIFTs are selected as the words of the dictionary - this is usually done by k-means clustering. Then, we calculate the SIFTs of the image and assign them to the nearest word. The final image representation is given by the histogram of the visual words.

Spatial pyramids. One of the major drawbacks with bag-of-visual-words is the lack of spatial information. Spatial pyramids [Lazebnik et al. 2006] were introduced to deal with this problem. The method consists of repeatedly subdividing the image and applying bag-of-words to each subdivision. At each level, a histogram of the visual words is calculated. The concatenation of these histograms, weighted by the size of the pyramid cells, form the image representation.

Fisher vectors. While bag-of-words are very powerful, it is unclear why this should be an optimal representation of the image. Alternatively, an approach based on Fisher Kernels [Jaakkola and Haussler 1998] has been proposed [Perronnin et al. 2010] [Sánchez et al. 2013]. It consists of characterizing a sample from a distribution by its deviation from the generative model. More specifically in the field of image classification, we define a Gaussian Mixture Model as the generative model for the SIFTs, and then represent each image by the deviation of its patches from this model.

2.2 Sketch Classification

Research in sketching goes back to the beginning of Computer Graphics as a field, with the development of SketchPad [Sutherland 1964]. Since then, much work has been devoted to the understanding of sketches, first constrained to specific low-level types, such as arcs and circles, and more recently extended to general objects.

Domain specific sketch recognition. One of the earliest works on sketch recognition is due to [Rubine 1991]. They achieved good accuracy when recognizing some types of symbols, like digits and letters. [Sezgin 2001] tries to represent sketches in a more formal way by fitting geometry to them, allowing beautification algorithms to be applied. Later, [Hammond and Davis 2007] discussed the problem that sketch recognition worked only in specific domains. They proposed a language to describe these domains and a technique to automatically generate the corresponding specific sketch recognition algorithm. [LaViola and Zeleznik 2004] developed a system to allow recognition of sketched math expressions, and [Ouyang and Davis 2011] introduced the same sort of application in the domain of chemistry diagrams. Recently, [Donmez and Singh 2012] introduced a gesture recognition framework that allows learning repetitive patterns by example, allowing automatic synthesis of this type of sketches.

General sketch classification using Bag of Visual Words. Based on the successful bag-of-visual-words framework, [Eitz et al. 2012a] introduced a sketch classification method for general objects: they created a benchmark of 250 categories, covering many objects we encounter every day. Technically, their sketch classification does not differ much from the respective image classification technique - using bag-of-visual-words and Support Vector Machines. Their patch descriptors, however, are a modification of SIFT specially tailored for sketches - taking into account that sketches do not have smooth gradients and are also much sparser than images. This is the work most closely related to ours.

General sketch classification using Structure-based Descriptors. Recently, [Cao et al. 2013] introduced a symmetry-aware flip-invariant descriptor for sketches. Drawings often appear flipped, and this kind of descriptor could improve the classification results, specially in settings where not much training data is available. For the classification task, the reported recognition rate is close to the one reported by [Eitz et al. 2012a], while using only 24 categories, instead of 250. Following a similar line, [Li et al. 2013] represented the sketch by a star-graph and performed a specific matching strategy for this structure. While they achieve 61.5% accuracy, their 4-cross-fold validation results in bigger training sets than [Eitz et al. 2012a], such that the comparison is not fair.

3 Sketch Classification with Fisher Vectors

In this Section, we describe the application of Fisher Vectors to the sketch classification problem - the settings we used and some small modifications that were essential for achieving good results on the TU-Berlin benchmark. In our implementation, we used the library VLFeat [Vedaldi and Fulkerson 2008], both for Fisher Vectors, SIFT, and for the Support Vector Machines.

We start from the rasterized version of the sketches (resized to 480×480 pixels) and describe them by extracting the Fisher Vector representation. We use a Gaussian Mixture Model with 256 Gaussians as the generative model of the patch descriptors. To estimate the parameters of this GMM, we obtain 25600 sample descriptors by applying dense SIFT in the images of the training set and reduce them from 128 to 80 dimensions using PCA. Finally, we use Expectation Maximization [Sánchez et al. 2013] to estimate the mean, variance and weight of each Gaussian. With the GMM in place, we can now encode each image in the training set using the deviation from the distribution (using Fisher Vectors).

The final step is to train the Support Vector Machines. For every class, we train a SVM where the positive set are the sketches of the class, and the negative set are all other sketches (one vs. rest). This gives us a score for the object, denoting the confidence that the object belongs to the class. To extend this to multi-class categorization, we set the category of the object as the one which obtained the highest score.

Differently from what was reported by [Eitz et al. 2012a], our results were improved by the usage of spatial pyramids. For all tests we use only two levels: 1×1 and 2×2 grids.

3.1 Larger SIFT Patches

Rasterized sketches are a very specific subset of the image category. A main difference is that they are extremely sparse - a small patch in this kind of drawing will usually be composed by at most one line segment, which is not very informative. To account for that fact, we increased the size of our SIFT bins from 4×4 pixels, which would be the common size for image classification, to 8×8, 16×16, and 24×24 pixels. Note that SIFT descriptors use a 4×4 grid, so the actual size of the patches is 32×32, 64×64, and 96×96, respectively.

Our results show that having bigger SIFT patches improves the classification rate (see Figure 2). Changing the parameters of SIFTs, instead of performing modifications on the structure of the descriptor, has the advantage of being compatible with the standard Computer Vision framework. Using the specially tailored features from [Eitz et al. 2012a] could improve our results even further.

3.2 Classification Results

We now discuss the results we obtained applying the technique described above to the TU-Berlin sketch benchmark.

Benchmark. The benchmark consists of 250 object categories, with 80 sketches each. The category of each sketch was defined by the person drawing it. After the sketch creation, there was a phase where humans tried to recognize the sketches - achieving an accuracy of 73%.

Results. The test settings were chosen to be consistent with the setup from [Eitz et al. 2012a]. We test 3 different patch sizes, with and without using spatial pyramids, and 10 subset sizes (the subset is the part of the dataset that will be used in each test). We divide the subset in 3 parts: 2 to be used as the training set and 1 as the testing set. The results reported are the average accuracy of three runs, one with each part being used as the testing set. Figure 2 shows the results we obtained for patch sizes of 16×16 and 24×24. We omitted the inferior results (8×8) for better clarity.

The results demonstrate that Fisher vectors significantly improve over the state-of-art. Also, with enough training samples, accuracy is now close to human performance. Our usage of bigger SIFT patches is also responsible for an important amount of the improvement (the best results for FV with 8×8 patches was 63.1%, on the subset with 80 images). The usage of spatial pyramids also gave us better accuracy - differently from what was reported by [Eitz et al. 2012a]. We suppose their features are big enough to already encode most of the spatial information. Finally, note that they show in their paper the difference between soft and hard assignment. In our experiments, we only used soft assignment.

4 Benchmark Analysis

In Section 3, we demonstrate that our approach achieves results that are close to human performance when classifying sketches. These results are somewhat counter-intuitive, since object recognition is usually a very hard task for computers and one where humans perform amazingly well.

As discussed in [Hoiem et al. 2012], average accuracy over thousands of images does not always reveal the true difference between two algorithms, *e.g*, when and why a technique outperforms the other. In their paper, they analyze the performance of two methods

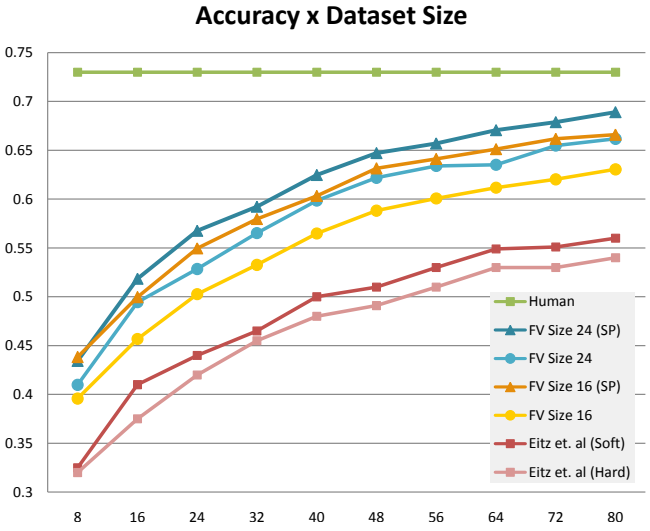

Figure 2: *Classification results on the benchmark from [Eitz et al. 2012a] (accuracy as a function of subset size). (FV) denotes Fisher Vector, (SP) denotes Spatial Pyramid. The results from [Eitz et al. 2012a] are taken from their paper, for soft and hard assignment using SVMs.*

on the PASCAL VOC 2007 dataset [Everingham et al. 2010] by examining the effects of different obstacles (occlusion, confusion with similar objects, localization error, ...) on the classification results. In this Section, we use the same sort of ideas to understand the real differences between computer and human performance, and propose a benchmark that is more in accordance with the sketch classification task.

4.1 Analyzing human misclassifications

To understand why computers perform almost as well as humans, we need to understand when humans make mistakes. By analyzing the results of [Eitz et al. 2012a], we identify three sources of problems: categories that are too similar or contain each other, poor sketches and, finally, simply poor recognition.

Similar categories. Almost 40% of human errors can be explained by confusion between 0.3% of the pairs of categories. For example, the category **chair** is confused with the category **armchair** 39% of the time. One reason for this problem is that human sketches are high level representations and often do not contain enough detail to distinguish between very similar categories (see Figure 3). In some cases, the categories chosen by [Eitz et al. 2012a] also contain each other or strongly overlap, like **bear** and **teddy bear**, or **sea gull** and **flying bird**. This shows that a high percentage of the mistakes made by humans are semantically meaningful and cannot be seen as real misinterpretations of the sketch.

Poor sketches. A different source of errors are sketches that do not really represent their category. This type of error is much harder to quantify, because we cannot distinguish between a good sketch that was poorly classified, and a bad sketch. Figure 4 shows some examples of sketches that are, in our judgement, not a good representation of their categories. Penalizing human/computer performance using this kind of sketch is misleading.

Poor recognition. Finally, there are what we consider to be the real mistakes made by humans. Again, it is challenging to identify them, because they are intrinsically confused with the problem of having unrecognizable sketches.

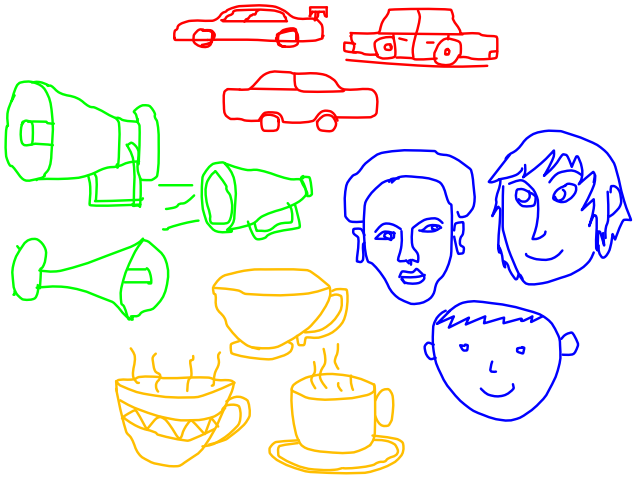

Figure 3: *Some categories of the benchmark proposed by [Eitz et al. 2012a] are very hard, if not impossible, to distinguish through the coarse representation given by sketches. The figure shows examples of images from different classes. (red) Categories **car (sedan)** and **race-car**. (green) Categories **megaphone** and **loudspeaker**. (blue) Categories **face** and **head**. (yellow) Categories **cup** and **tea cup**.*

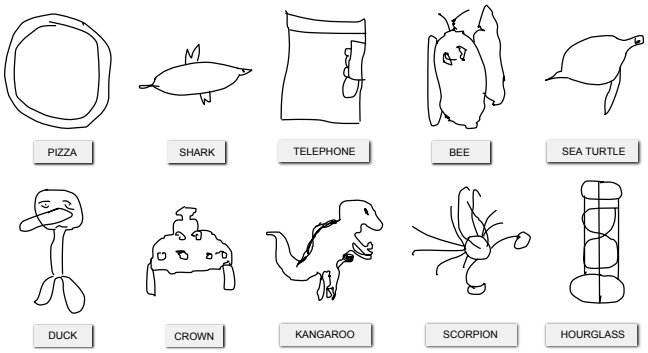

Figure 4: *Some sketches are drawn so poorly that it is not possible to recognize the intention of the author. Penalizing recognition performance using these sketches leads to unintuitive results.*

5 Improved Benchmark

From the analysis of human results, we identify two problems: **similar categories**, that are hard to distinguish from each other using sketches, and **poor sketches**, whose category cannot be identified. Based on these issues, we propose a different way to think about this problem, shifting the question from "how do humans sketch objects?" to "how do humans understand sketches?". The key idea is that a sketch is only adequate if it expresses what it was meant to express, to the extent that it can be understood by other humans.

Ideally, one would create a benchmark of sketches that were consistently recognized by humans as belonging to one class. This would solve the two problems described earlier, because both sketches that are not recognizable and sketches that belong to undistinguishable classes are unlikely to be classified consistently. As discussed in [Eitz et al. 2012a], however, the amount of work required for creating such a benchmark is huge. Instead, we use only the correctly classified sketches from this existing benchmark. By having at least two humans who agree about the class of the sketch (the one drawing it, and the one recognizing it) we move considerably in the direction of the kind of benchmark we would like to have, without needing additional annotation effort.

Procedure. We create our benchmark by first excluding all sketches that were misclassified by humans. From the remaining set, we select only those classes that have at least 56 samples, which is 70% of the original amount. This results in 160 classes of objects, each one with at least 56 objects. For categories where there were more than 56 correctly classified sketches, we did not remove the exceeding ones. We randomly select a subset when performing experiments to avoid bias towards bigger classes. Note that the procedure we use to select which classes and sketches belong to the new benchmark is systematic and totally based on the results obtained by humans. The benchmark is by no means biased towards the classification method we used.

5.1 Classification Results in the new Benchmark

We applied the exact same algorithm and parameters as explained in Section 3 to the modified benchmark. Since not all categories have more than 56 samples, we perform the experiments up to this dataset size. While the accuracy is higher than the results reported on the original benchmark, there is more room for improvement - here, human performance defines the ground truth, *i.e.*, 100%. Note that in this case the problem of a better technique getting worse results because of poor human drawing is attenuated, so that improvements in technique should also lead to better scores. Figure 5 shows the average precision for our Fisher Vector representation using different patch sizes.

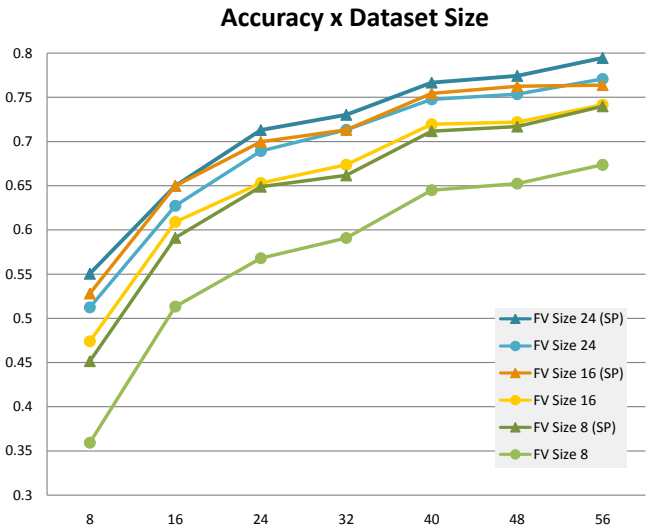

Figure 5: *Results obtained on the new benchmark. We expect that future improvement in techniques will translate more naturally into better performance.*

5.2 Discussion

From the 0.3% pairs of categories that were responsible for 40% of human misclassifications, this procedure was able to remove 85% (only 16 pairs were left, from an initial 107). Note that the policy we used for removing the sketches from the benchmark was not specially tailored for removing confusing pairs - although, of course, confusing pairs of categories tend to have less correctly classified sketches.

Analyzing which confusing pairs of categories were kept was also

very interesting - see Figure 6. They depict, in all cases, pairs of categories that do have some similarities, but still allow users to recognize a majority of the sketches. As such, we think it is a good thing that these classes are still part of the benchmark.

We believe the confusing classes problem was practically solved by our procedure. Note that this step is essential for every technique trying to understand sketches and is not a trivial one (it would be hard to justify that people can draw owls distinguishably, but not seagulls, in a non-data-driven manner).

The poor sketches problem was not totally solved by our procedure, since humans can sometimes correctly guess the category of low quality drawings. Having more human evaluations would help solve this problem. We show next that human answers are more consistent in the modified benchmark.

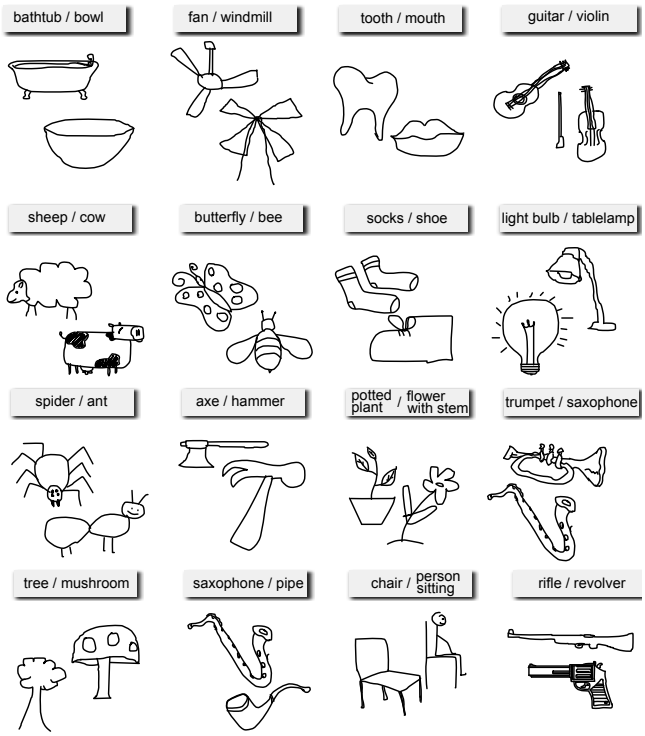

Figure 6: *Some pairs of categories that are classified as confusing were kept in the new benchmark. They express classes that are similar, but still distinct enough that most sketches will be correctly recognized.*

Our benchmark creation relies on the assumption that the agreement of two humans, together with the removal of some confusing categories, will lead to a dataset that can be more consistently recognized by humans. To confirm this hypothesis, we performed additional evaluation of 1000 sketches - 500 from the old dataset and 500 from the new one. We asked 10 users, which evaluated approximately 100 sketches each. Users achieved 93% of accuracy in the new dataset, against 75.8% in the old one. This improvement is significantly bigger than just the effect of a smaller number of classes. This demonstrates that our dataset already makes a big step towards what we understand to be the ideal benchmark. We will make the trimmed TU-Berlin benchmark available for future research.

As an additional note, useful for future annotation, many users complained about the way the hierarchy of categories was chosen and would have preferred to see all categories at once - we used the same structure as [Eitz et al. 2012a].

6 Classification-driven analysis

In this Section, we use classification results to investigate how our method understands sketches. Semantic understanding of sketches could improve tasks like recognition, beautification, animation and automatic synthesis. Also, because this type of representation is extremely sparse, it could give insight on which are the most basic features that define a category of objects - something that could be useful even in other types of visual classification.

6.1 Representative sketches

The first analysis we performed is whether we can use the data we have to assess the quality of a sketch, *i.e.*, how well it represents the category it belongs to. We define as most representative sketch the one that got the highest classification score for that class. In Figure 7, we compare our results with the representative sketches from [Eitz et al. 2012a]. For a fair comparison, we use the old dataset introduced by them, and we compare only to classes for which they have a single representative sketch. We believe our approach achieves superior results.

Also meaningful are the sketches that get the worst classification score for the class they are meant to represent. They are always poor sketches, or sketches that differ too much from the usual sketch in the category. In Figure 8, we show the sketches with lowest classification scores for some categories.

The quality of a sketch is very difficult to evaluate in a quantified manner. We provide the whole set of iconic sketches in the supplementary material.

6.2 Stroke importance

A different question when analyzing sketches is the importance of each individual stroke made by the user. The representation using Fisher Vectors does not directly translate into scores for the different parts of the image. Instead, we used the following method: for each stroke, remove it from the image and see how it affects the classification score. Our results show that this technique can provide insight of which strokes are important for the sketch - as can be seen in Figure 9.

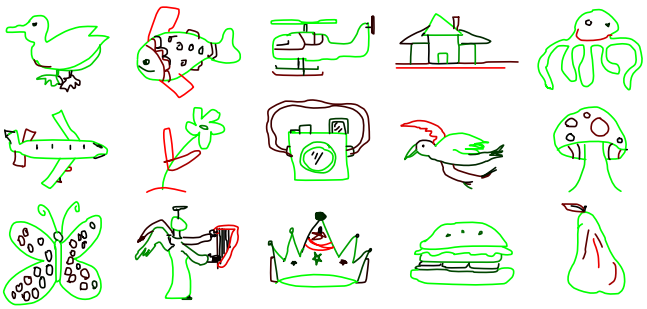

Figure 9: *Visualization of stroke importance in sketches. Strokes that contribute positively to the recognition are shown in green, while strokes that our approach perceives as confusing are shown in red. The saturation shows the magnitude, i.e., black strokes have small or no influence on the ability to recognize the sketch.*

Again, it is very hard to evaluate objectively whether a stroke was really important for the sketch. To demonstrate that we perform consistently across the whole dataset, we provide all results in the supplementary material. The results here are constrained to 56 sketches per class (while our dataset has bigger classes) because

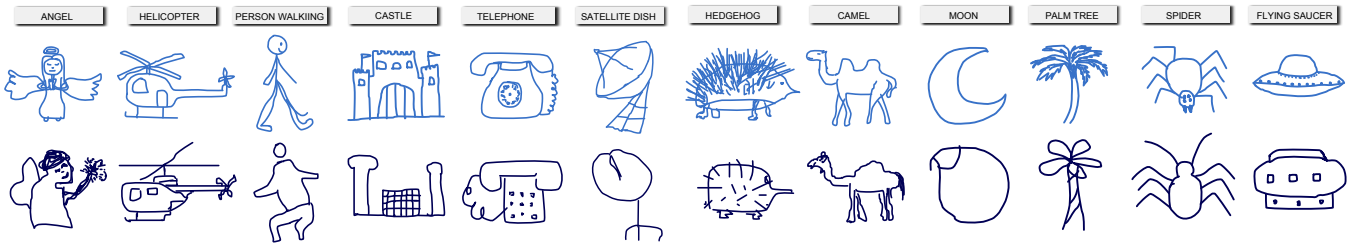

Figure 7: *Classification scores are a good measure of sketch quality. (top) Our best score. (bottom) Eitz et al.*

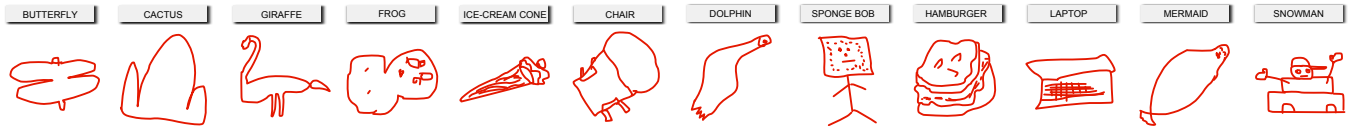

Figure 8: *Low classification scores identify very poorly drawn sketches or sketches that differ very much from the rest of the category.*

those were the 56 randomly chosen sketches used in the classification task.

For some cases where stroke importance was not intuitive, we investigated the reasons behind the results further. An interesting observation is that SVMs are discriminative models, which means that the negative set has an influence on the results, as well as the positive [Torralba and Efros 2011]. This means that increasing the classification score for one class may be more related to decreasing the score for other classes.

In Figure 10, we can see some examples of sketches with unexpected stroke importances that can be better understood by looking into the negative set. The airplane, for example, was originally classified as a submarine, so that making it look less like a submarine plays a main role in making its confidence as an airplane go up.

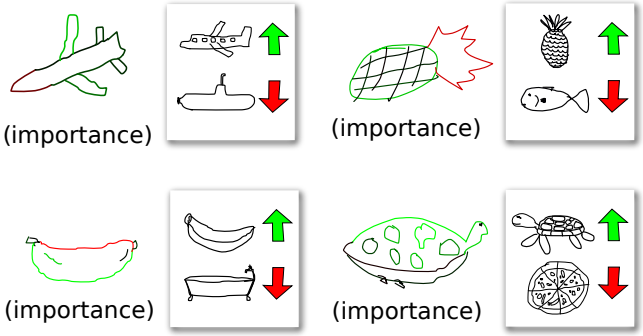

Figure 10: *The importance our method assigns to some strokes is unintuitive if we do not observe the negative set. (top-left) airplane, originally classified as submarine. (top-right) pineapple, originally classified as fish. (bottom-left) banana, originally classified as bathtub. (bottom-right) sea turtle, originally classified as pizza.*

It is important to make the disclaimer that this analysis is not exact - the removal of strokes also has effects on other categories, and it is not always easy to infer why. This analysis does suggest some further directions for exploration though, which we discuss next.

6.2.1 Pairwise SVMs

Since classification results are influenced by both the positive and the negative set, we performed some experiments using only one class as the negative set. We would expect that the stroke importance, according to these classification scores, would highlight the strokes that differentiate most between two classes. The pairwise SVMs we trained were the confusing categories from Figure 6.

The results we got were often meaningful - see Figure 11. Note that, differently from the previous experiments, green denotes features that are distinctive for this class, while red shows features that are distinctive for the other one.

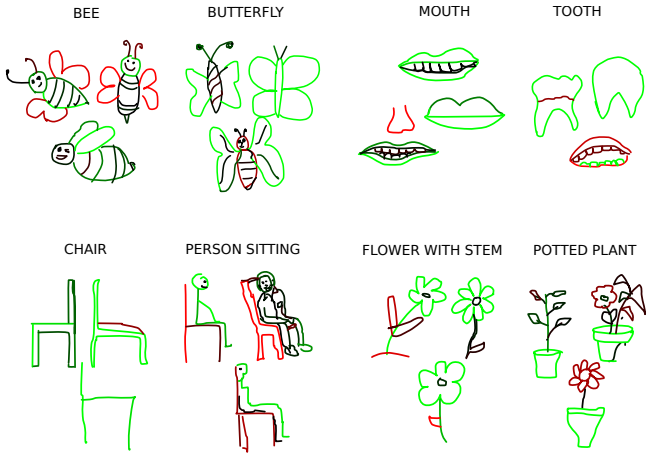

Figure 11: *The stroke importance according to pairwise SVMs highlights parts of the object that make it distinct from the negative set.*

6.2.2 Pruned Negative Set

The final experiment we made was to prune the dataset to avoid having similar classes in the negative set. The reasoning behind this is that the legs of a cow, for example, are an important part of the sketch, even if they look more like horse legs. While there was improvement in some sketches, it was not significant for the majority of cases. We believe this happens because sketches are such a coarse representation that the same stroke can be a leg of a cow or a part of a chair - making it hard to predict which categories would have overlapping subsets.

6.3 Most Recognizable Subset of Strokes

Going further into the analysis of stroke importance, one could try to obtain the subset of the sketch that would be mostly recognizable for the computer. Finding the set of strokes that would provide the best classification results, however, is a combinatorial problem.

The strategy we used was a greedy approach that iteratively removes the most confusing stroke. For each step, we compute the scores for the image without each of its strokes, and get the maximum for the next step. We stop if there is no stroke removal that would increase the classification score anymore. This approach is of course not robust against local minima. We implemented an alternative scheme using genetic algorithms, but it only rarely outperformed the greedy version. In Figure 12, we show some results achieved with our method.

The binary nature of optimal subsets (**kept** or **removed**) gives a less informative visualization than the one given by stroke importance. Since our optimal subset calculation is a straightforward application of the stroke importance, we preferred to add the stroke importance results to the supplementary material.

6.3.1 Discussion

There exists a body of research dedicated to line drawing simplification [Barla et al. 2005] [Shesh and Chen 2008]. These methods are, however, mostly dedicated to a different type of drawing. For example, they perform simplification by removing clutter. Our sketches are very simple already and do not contain that many overlapping strokes to be removed.

The goal we are trying to achieve here is not simplification by itself. By finding the optimal subset of the sketch, we intend to isolate the features that differentiate a category from the others. This is an essential step towards automatic semantic understanding of sketches.

7 Conclusions and Future Work

We introduced an approach for sketch classification based on Fisher Vectors that performs significantly better than other techniques, achieving results close to human outcomes. Even if the technique is not novel, the application of Fisher Vectors to sketches is new, and the results we achieved are clearly the new state-of-the-art in sketch classification.

Our modified benchmark is an essential improvement in the evaluation of sketching. The previous benchmark is saturated and performing a fair comparison between techniques in the presence of the artifacts we discussed can be difficult. Our view that sketch recognition systems should be evaluated by their ability to mimic human performance in this task, by itself, is a step towards better benchmarks and techniques.

Finally, we introduce a data-driven approach for the understanding of sketches. Our results demonstrate that classification scores are a good criterion for measuring the quality of a sketch. We also achieved good results when determining the contribution of parts of the sketch to the overall recognizability.

We believe our work opens many possibilities for further exploration. The stroke importance results are still not perfect, specially for classes with many strokes, and one drawback is the greedy nature of our method. Pre-processing the sketch to identify which strokes belong together could improve this analysis.

Following on our stroke importance analysis, it would be interesting to investigate which were the most important strokes for a category as a whole, and to formalize which kind of geometric shapes define a class. A more geometrical/topological description of sketches could also improve robustness, in the sense of being less disturbed by small differences in strokes.

Acknowledgements

We thank all participants of our experiments for their evaluations and the anonymous reviewers for their constructive comments. This work was supported by the ERC grand Cognimund and the PARIS project (IWT-SBO-Nr. 110067).

References

BARLA, P., THOLLOT, J., AND SILLION, F. X. 2005. Geometric clustering for line drawing simplification. In *ACM SIGGRAPH 2005 Sketches*, ACM, New York, NY, USA, SIGGRAPH '05.

CAO, X., ZHANG, H., LIU, S., GUO, X., AND LIN, L. 2013. Sym-fish: A symmetry-aware flip invariant sketch histogram shape descriptor. In *IEEE International Conference on Computer Vision (ICCV)*.

CSURKA, G., DANCE, C. R., FAN, L., WILLAMOWSKI, J., AND BRAY, C. 2004. Visual categorization with bags of keypoints. In *In Workshop on Statistical Learning in Computer Vision, ECCV*, 1–22.

DAVIS, J., AGRAWALA, M., CHUANG, E., POPOVIĆ, Z., AND SALESIN, D. 2003. A sketching interface for articulated figure animation. In *Proceedings of the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, SCA '03, 320–328.

DONMEZ, N., AND SINGH, K. 2012. Concepture: A regular language based framework for recognizing gestures with varying and repetitive patterns. In *Proceedings of the International Symposium on Sketch-Based Interfaces and Modeling*, Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, SBIM '12, 29–37.

EITZ, M., HILDEBRAND, K., BOUBEKEUR, T., AND ALEXA, M. 2011. Sketch-based image retrieval: Benchmark and bag-of-features descriptors. *IEEE Transactions on Visualization and Computer Graphics 17*, 11 (Nov.), 1624–1636.

EITZ, M., HAYS, J., AND ALEXA, M. 2012. How do humans sketch objects? *ACM Trans. Graph. (Proc. SIGGRAPH) 31*, 4, 44:1–44:10.

EITZ, M., RICHTER, R., BOUBEKEUR, T., HILDEBRAND, K., AND ALEXA, M. 2012. Sketch-based shape retrieval. *ACM Trans. Graph. 31*, 4 (July), 31:1–31:10.

EVERINGHAM, M., VAN GOOL, L., WILLIAMS, C. K. I., WINN, J., AND ZISSERMAN, A. 2010. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision 88*, 2 (June), 303–338.

HAMMOND, T., AND DAVIS, R. 2007. Ladder, a sketching language for user interface developers. In *ACM SIGGRAPH 2007 Courses*, ACM, New York, NY, USA, SIGGRAPH '07.

HEARST, M. A., DUMAIS, S., OSMAN, E., PLATT, J., AND SCHOLKOPF, B. 1998. Support vector machines. *Intelligent Systems and their Applications, IEEE 13*, 4, 18–28.

HOIEM, D., CHODPATHUMWAN, Y., AND DAI, Q. 2012. Diagnosing error in object detectors. In *Proceedings of the 12th*

European Conference on Computer Vision - Volume Part III, Springer-Verlag, Berlin, Heidelberg, ECCV'12, 340–353.

JAAKKOLA, T., AND HAUSSLER, D. 1998. Exploiting generative models in discriminative classifiers. In *In Advances in Neural Information Processing Systems 11*, MIT Press, 487–493.

LAVIOLA, JR., J. J., AND ZELEZNIK, R. C. 2004. Mathpad2: A system for the creation and exploration of mathematical sketches. *ACM Trans. Graph. 23*, 3 (Aug.), 432–440.

LAZEBNIK, S., SCHMID, C., AND PONCE, J. 2006. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, vol. 2, 2169–2178.

LI, Y., SONG, Y.-Z., AND GONG, S. 2013. Sketch recognition by ensemble matching of structured features. In *In British Machine Vision Conference (BMVC)*.

LOWE, D. G. 2004. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision 60*, 91–110.

OLSEN, L., SAMAVATI, F. F., SOUSA, M. C., AND JORGE, J. A. 2009. Sketch-based modeling: A survey. *Computers & Graphics 33*, 1, 85 – 103.

OUYANG, T. Y., AND DAVIS, R. 2011. Chemink: A natural real-time recognition system for chemical drawings. In *Proceedings of the 16th International Conference on Intelligent User Interfaces*, ACM, New York, NY, USA, IUI '11, 267–276.

PERRONNIN, F., LIU, Y., SANCHEZ, J., AND POIRIER, H. 2010. Large-scale image retrieval with compressed fisher vectors. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, 3384–3391.

RUBINE, D. 1991. Specifying gestures by example. In *Proceedings of the 18th Annual Conference on Computer Graphics and Interactive Techniques*, ACM, New York, NY, USA, SIGGRAPH '91, 329–337.

SÁNCHEZ, J., PERRONNIN, F., MENSINK, T., AND VERBEEK, J. 2013. Image classification with the fisher vector: Theory and practice. *International Journal of Computer Vision 105*, 3, 222–245.

SCHMIDT, R., WYVILL, B., SOUSA, M. C., AND JORGE, J. A. 2006. Shapeshop: Sketch-based solid modeling with blobtrees. In *ACM SIGGRAPH 2006 Courses*, ACM, New York, NY, USA, SIGGRAPH '06.

SEZGIN, T. M. 2001. Sketch based interfaces: Early processing for sketch understanding. In *Proceedings of PUI-2001. NY*, ACM Press.

SHESH, A., AND CHEN, B. 2008. Efficient and dynamic simplification of line drawings. *Comput. Graph. Forum 27*, 2, 537–545.

SIVIC, J., AND ZISSERMAN, A. 2003. Video Google: A text retrieval approach to object matching in videos. In *Proceedings of the International Conference on Computer Vision*, vol. 2, 1470–1477.

SUTHERLAND, I. E. 1964. Sketch pad a man-machine graphical communication system. In *Proceedings of the SHARE Design Automation Workshop*, ACM, New York, NY, USA, DAC '64, 6.329–6.346.

TORRALBA, A., AND EFROS, A. A. 2011. Unbiased look at dataset bias. In *Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition*, IEEE Computer Society, Washington, DC, USA, CVPR '11, 1521–1528.

VEDALDI, A., AND FULKERSON, B., 2008. VLFeat: An open and portable library of computer vision algorithms. `http://www.vlfeat.org/`.

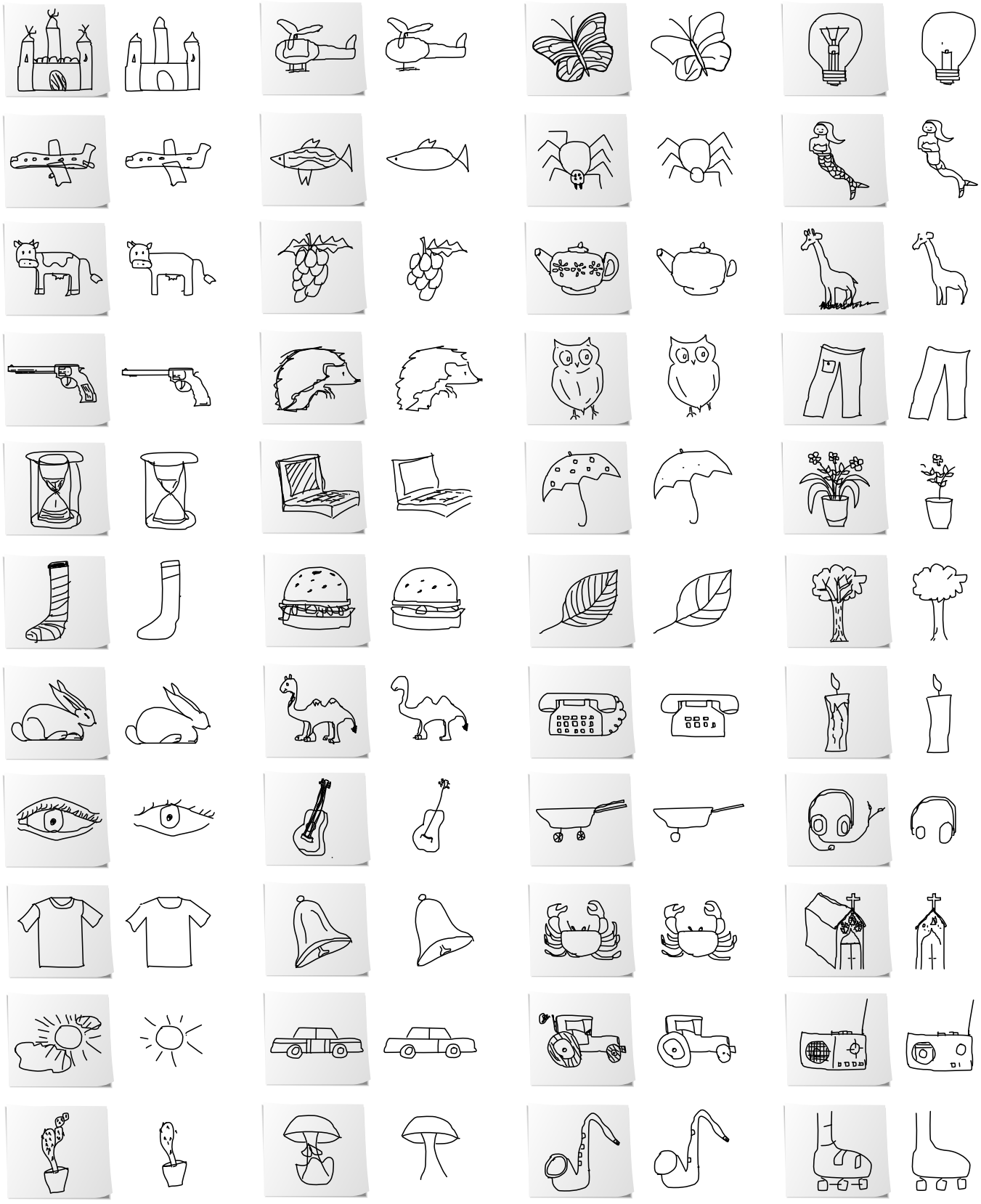

Figure 12: *Most recognizable subset of the sketch. Determining which features are important for a category is crucial for semantic understanding of sketches. (left) Original. (right) Our method.*