Unsupervised generation of context-relevant training-sets for visual object recognition employing multilinguality

Markus Schoeler Florentin Wörgötter Jeremie Papon
Tomas Kulvicius
III. Physikalisches Institut - Biophysik, Georg-August University of Göttingen
{mschoeler, worgott, jpapon, tkulvic}@gwdg.de

Abstract

Image based object classification requires clean training data sets. Gathering such sets is usually done manually by humans, which is time-consuming and laborious. On the other hand, directly using images from search engines creates very noisy data due to ambiguous noun-focused indexing. However, in daily speech nouns and verbs are always coupled. We use this for the automatic generation of clean data sets by the here-presented TRANSCLEAN *algorithm, which — through the use of multiple languages — also solves the problem of polysemes (a single spelling with multiple meanings). Thus, we use the implicit knowledge contained in verbs,* e.g. *in an imperative such as "hit the nail", implicating a metal nail and not the fingernail. One type of reference application where this method can automatically operate is human-robot collaboration based on discourse. A second is the generation of clean image data sets, where tedious manual cleaning can be replaced by the much simpler manual generation of a single relevant verb-noun tuple. Here we show the impact of our improved training sets for several widely used and state-of-the-art classifiers including Multipath Hierarchical Matching Pursuit. All tested classifiers show a substantial boost of about +20 % in recognition performance.*

1. Introduction

Classifiers are ubiquitous in modern vision applications, spanning various areas including autonomous vehicles, photo classification on image hosting websites or robotic systems in unstructured environments. As such, there has been a significant effort to improve classification pipelines, using more discriminative image features (*e.g.* SIFT [21]) and image signatures (*e.g.*, Bag of Words [8], Fisher Vectors [25]) or by using better machine learning algorithms (*e.g.*, Support-Vector-Machines). Additionally, new approaches like Deep Belief Networks [16] and sparse coding [6] have improved recognition performance in recent years.

Nevertheless, all of these approaches have one thing in common: They are supervised methods which heavily depend on the quality and size of the training set used. Compiling such a dataset typically involves humans in time consuming and tedious procedures [23, 26, 28]. To avoid this, some researchers use the highest ranking images returned by image search engines (*e.g.*, Google) using the class name as a search term [10, 11]. Unfortunately, Griffin *et al.* [12] discovered that only 33 % of the top-100 images from Google are relevant, when using the class name as a search term. This makes it unfeasible for collecting large datasets. One reason for such a low relevance are polysemes (a single spelling with multiple meanings). "Nail", for example could refer to the piece of anatomy or the object one hits with a hammer. Clearly, a command like "hit nail" provides a context to instantly disambiguate the meaning of "nail", and while indexed images for "hit nail" will be more relevant, they also contain a lot of clutter or irrelevant objects. This is a consequence of actions like "tighten bolt" or "hit nail" involving tools in addition to the actually searched for "bolt" and "nail". Figure 8 shows this problem on some examples for plain Google search without context (*GP*) and for Google searches with action context (*GC*).

To address these issues, we have developed TRANSCLEAN (*TC*), which uses verb-noun tuples from sentences. As verb-noun tuples are part of all sentences, it specifically provides a generic solution to the problem of command-disambiguation. This arises in human-robot-interaction applications such as [7, 9, 13, 17] or applications which execute commands from instruction sheets like [3, 19, 29]. On the other hand, TRANSCLEAN can be also be used for improving the relevance of Google Image Search results by manually providing context. While we show results for noun-verb tuples, the algorithm will work with descriptive adjective-noun combinations as well.

The algorithm accomplishes this by creating relevant noun translations for the provided context using the available Google services Google Translate, Google Text Search and Google Image Search. We should emphasize at this point that we could use other translation services (*e.g.* dict.leo.org, dict.cc) or search engines which index documents and images by word occurrences (*e.g.* bing.com, yahoo.com).

2. Related work

TRANSCLEAN combines word sense disambiguation with content-based image retrieval. Word sense disambiguation is the task aimed at discovering the meaning of single- and multi-words in texts and mapping occurrences to entries in a reference knowledge database [1, 22]. Image retrieval is the task of retrieving query relevant images from a database. This query can either be a phrase, or (as is the case with content-based image retrieval) another image, which requires computer-vision algorithms to be employed. Multilinguality has been used in various ways, *e.g.* by using parallel text corpora to build multilingual contexts [2, 15] or by exploiting complementary sense evidence from translations in different languages [24]. While these approaches stay in the text domain and require a semantic knowledge base like BabelNet, we exploit multilinguality to eliminate polysemy by applying image retrieval techniques to a superset of images created by multilingual searches. We should note that we are not interested in the semantic meaning of the classes, as is the goal in word sense disambiguation, but rather in an unambiguous training set which can be fed into existing classifier pipelines respecting the context given by *e.g.* a verb. A related approach was proposed by Kulvicius *et al.* [20], who used search cues generated from domain specific text-corporas to create a superset of images which are then merged into one relevant dataset. In contrast to this we do not need to know the context explicitly or to create a text-corpora beforehand. Instead, we deduce the context implicitly from the verb. Other works also make use of visual and textual cues. Either implicitly, by using the first results of text-based image search engines [10, 11], by constructing their own image search engine [5, 27], or explicitly, by making use of image tags and labels as found in photo-sharing websites like Flickr [14]. None of these methods can automatically cope with the problem of polysemes.

This paper is organized as follows: First, in Sec. 3 we present the outline of our algorithm. Section 4 gives an overview of the methods we used for performance evaluation. In Sec. 5 we present quantitative results showing the superiority of our method compared to plain Google search without context (*GP*) as well as Google search together with the context (*GC*). Additionally, we give results for an image classification experiment using several popular and state-of-the-art classifiers including the Multipath Hierarchical Matching Pursuit (M-HMP) proposed by Bo *et al.* [6]. Finally, we summarize and discuss our approach in Sec. 6.

3. Proposed algorithm

The proposed algorithm, depicted in Fig. 1, consists of five sequential parts which we describe in this Section: 3.1 Noun and verb translation, 3.2 Context check, 3.3 Image retrieval, 3.4 Subset matching, and 3.5 Duplicate and clutter removal. The input for the TRANSCLEAN algorithm consists of an object/noun (like apple, orange or saw) and an action context (like cut, fill or prick). For clarity, we adopt the notation *class (context)* to denote a class in a given context, *e.g.*, nail (hit).

3.1. Noun and verb translation

In the first step we translate the noun and verb separately into multiple languages (we use French, Spanish and German unless noted otherwise). Having “cup (fill)” as input, this step would retrieve all translations for cup and fill. We will use https://translate.google.co.uk/ throughout this paper. For washer (clean) the translations in French are: *washer: rondella, machine à laver; clean: nettoyer, éplucher, faire nettoyage, ratiboiser, ravaler, vider*.

3.2. Context check

This step determines the most relevant noun-translation for each language using Google Text Search https://www.google.co.uk/. For each verb-noun combination we perform two exact searches: “noun verb” and “verb noun”. For both searches we parse the number of results and take the maximum as the relevance score for that combination. The noun which gets the most matches combined with any verb is then selected as the relevant translation. We use the Google search parameter *lr* to only retrieve results from documents in a specific language.

In the washer (clean) example shown in Fig. 2 the translations which get selected are *machine à laver* in French, *Waschmaschine* in German and *lavadora* in Spanish.

3.3. Image retrieval

This step downloads the first 300 images for the translations which passed the context check (one per language) as well as for the English search. Again, we set the parameter *lr* to the respective language. For instance, to download images for “Waschmaschine” we use https://www.google.co.uk/search?q=Waschmaschine&tbm=isch&lr=lang_de.

3.4. Subset matching

In this step we are going to merge the different language subsets into one relevant dataset. We do this by pair-wise image comparison across the different subsets. To calculate similarity between two images we generate a histogram

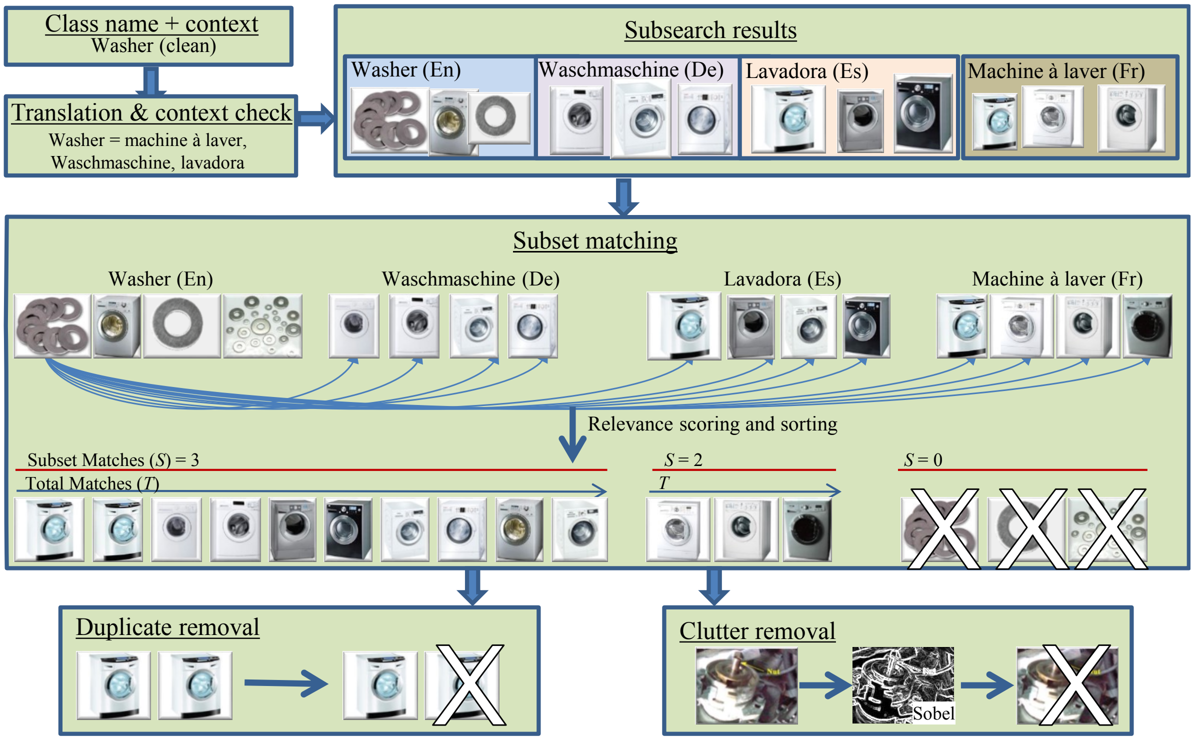


Figure 1. Flow diagram of the TRANSCLEAN algorithm exemplified on the class "washer" in the context of "clean". Subset Matches ($\mathcal{S}$) counts the total number of subsets in which a match has been found. Total Matches ($\mathcal{T}$) counts the total number of matches. $\mathcal{S}$ is our first order and $\mathcal{T}$ our second order relevance sorting criteria. Only images with $\mathcal{S} > 0$ are considered further. We used ISO639-1 language codes.

French	nettoyer	éplucher	faire nettoyage	ratiboiser	ravaler	vider
rondella	0	0	0	0	0	0
machine à laver	30400	5	0	0	1	313

German	reinigen	säubern	waschen	sauber machen	abwischen
Waschmaschine	39400	1430	126000	827	194
Scheibe	5390	2380	2530	7300	1200
Geschirrspülmaschine	3940	646	2270	7	8

Spanish	limpiar	asear	mondar	hacer una limpieza
lavadora	2140	2	0	0
arandela	185	5	0	0

Figure 2. Context check for washer (clean). Rows: Translations for the noun; columns: Translations for the verb. Each cell shows the number of exact Google search results (relevance score) for the noun-verb combination. The noun with the highest response gets selected (marked with red).

for each image using SIFT features [21]. Since we reduce color images to gray-scale for the SIFT features, we are able to compare color to gray-scale images as well. The features are sampled on a 14 × 14 grid on the full image using four scales. Bag-of-words [8] with $\mathcal{N} = 100$ clusters is used to generate normalized image signatures. The histogram intersection (min-kernel) over all bins is used as a match score. We require the score $\mathcal{M}$ to exceed a empirically determined threshold $\theta_{\text{th}} = 70\,\%$ to count as a match:

$$\mathcal{M}(x,y) = \sum_{i}^{\mathcal{N}} \min(x_i, y_i) \geq \theta_{\text{th}}. \tag{1}$$

A procedure similar to that of Kulvicius *et al.* [20] is used to generate an overall image relevance score, improving upon their work by using an additional relevance score. Fig. 1 shows how each image is scored: The score consists of 1) the number of other subsets where at least one match has been found $\mathcal{S}$ and 2) the total number of matches $\mathcal{T}$. Images without subset matches $\mathcal{S} = 0$ are pruned. Relevance of images is determined first by $\mathcal{S}$ and second by $\mathcal{T}$.

Given a fixed number of SIFT features per image and for the vocabulary generation, the complexity of the histogram generation algorithm is linear in the number of images $\mathcal{O}(kn)$ with k being the number of languages considered and n denoting the number of images per set. The complexity of the matching is $\mathcal{O}(k^2n^2)$. While the matching scales much worse, it is the fastest part of the algorithm when using four languages and 300 images per language, as it only consists of simple min operations. For $n = 300$ and $k = 4$ the running time of the image-to-image matching without parallelization is about 13 seconds on a 3.2 GHz processor.

Object	Given Context	Meanings
apple	cut	**food**, brand
axe	chop	**tool**, brand
bolt	tighten	**hardware**, athlete, movie
cup	fill	**drinking**, trophy, bra
fork	prick	**cutlery**, bike-part
glass	fill	**drinking**, material
hammer	hit	**tool**, brand
nail	hit	**hardware**, finger
nut	tighten	**hardware**, food
oil	eat	**food**, mineral-oil
orange	cut	**food**, color
pan	fry	**kitchenware**, movie, god
peach	fry	**food**, computer character
pot	cook	**kitchenware**, drug
saw	cut	**tool**, movie

Table 1. All classes together with their given context used for the experiments. Possible meanings as well as the relevant meaning (marked in bold) are shown.

3.5. Duplicate and clutter removal

Finally, in order to clean the result set of duplicate images and images with cluttered scenes we perform a duplicate and clutter search. To do this, we scale all images to exactly 150×150 pixels ignoring the aspect ratio and generate gradient magnitude images g_i using the sobel filter (the values g_i are in the range of 0 to 1). The similarity between image i and image j is calculated by L1-normalizing the gradient images g_i and g_j and calculating the histogram intersection. The duplicate threshold was empirically determined and set to 0.85 throughout all experiments. When a duplicate is found the image with lower relevance score is deleted. Additionally, we remove cluttered images by calculating the mean gradient magnitude within a five pixel image border of g_i. This value ranges between 0 (no clutter) and 1 (heavy clutter). Using a clutter threshold of 0.1 effectively removed all images which were recorded in cluttered scenes and therefore considered bad for the training (*e.g.*, an apple on a tree in a garden).

4. Evaluation methods

For the evaluation of our algorithm we benchmarked on the classes shown in Table 1. In all cases the noun itself is ambiguous and could refer to multiple meanings. In addition, we also provide context, in the form of verbs, which can be used for disambiguation. We used four languages: English, German, Spanish and French. Additionally, we used Portuguese for orange as the word is the same in German, French and English. We evaluate the proposed algorithm (*TC*) against plain Google search (*GP*) as well as Google search including the context (*GC*). For *GP* we conduct searches using the noun without the provided context to retrieve images. For *GC* we also provide the context together with the class label for the search. For example, with "pan" in the context of "fry", we retrieve images for *pan* (*GP*) as well as *fry pan* (*GC*). Image searches for *GC* are always conducted without quotes.

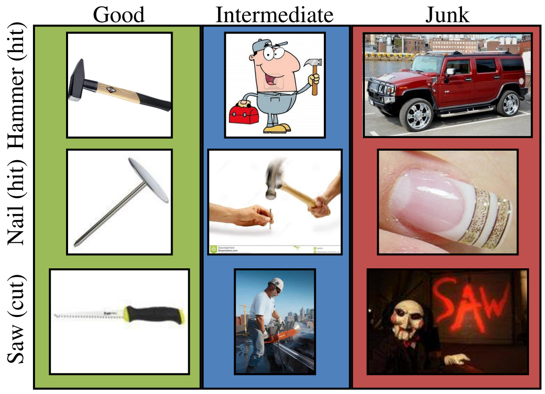


Figure 3. Three images showing the quality grading as introduced by [10] for three classes.

4.1. Quality of retrieved training-sets

Both our algorithm and Google provide results in an ordered list. We therefore first investigate how image-quality changes depending on list length. This is important as larger training image datasets should generally improve classifier performance. However, using more images may negatively affect the overall relevance of the set if more non-relevant images are added as the set expands, unnecessarily increasing intra-class variance. To quantify this effect we let a human grade each image retrieved by *GP*, *GC* and *TC*. We followed the categorization scheme introduced in [10]:

Good image Image containing the relevant object without major occlusions although there may be a variety of viewpoints, scalings and orientations.

Intermediate image Shows the relevant object, but may contain extensive occlusion, substantial image noise or the object is insignificant in the image.

Junk image Not relevant.

Stereotypic images for the three quality levels can be seen in Fig. 3. We use precision as well as quality distribution to assess the quality of the retrieved image sets. Precision is calculated counting only *Good* images as positives. Here we are especially interested in how precision changes depending on the number of images considered. This resembles the measure introduced by [12] in the Caltech-256 benchmark generation. The quality distribution, on the other hand, measures the ratio of the three quality categories, allowing deeper analysis of the properties of the methods.

4.2. Image Classification

To assess the performance dependency of classifiers on the training set we used three state-of-the-art object classification pipelines: Multipath Hierarchical Matching Pursuit (M-HMP)[6], the SIFT and Bag-of-Words based approach of Iravani *et al.* [18] combined with a Support-Vector-Machine (SIFT_SVM) and AdaBoost.MH (SIFT_BOOST). We decided on SIFT and Bag-of-Words as it has been and still is a very popular classification pipeline in the scientific community. Similar to the work of [18], SIFT features were extracted on a dense grid and 300 cluster centers were used to generate the signatures. We tested different kernels (χ^2, RBF and Histogram-Intersection) for the C-Support-Vector-Machine and found similar performance changes in all of them when trained with the 5 sets. For AdaBoost we used the multiclass capable Adaboost.MH from the MultiBoost library [4]. M-HMP achieved state-of-the-art results in many standard benchmarks by combining a collection of hierarchical sparse features to capture discriminative structures in the images[1]. For testing we use a manually cleaned image set containing only *Good* quality images which are disjoint from the sets used in the training.

5. Evaluation

5.1. Context check

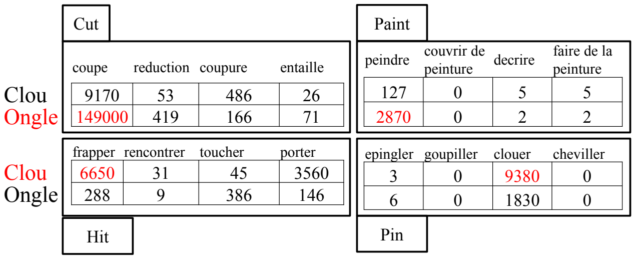
We conducted the context check with the contexts given in Table 1. Additionally, we investigated more closely how stable the translations of "nail" and "nut" are against changing contexts.

For "nail" we chose four contexts: hit, pin, paint and cut. For the first two we expect our algorithm to retrieve hardware-nail images and for the last two fingernail images. Figure 4 shows the context check scores we retrieved from Google Text Search using the French translations provided by the Google translation service. All actions can be used to correctly disambiguate the meaning of "nail".

"Nut" shows an interesting case when using the context "eat" (see Fig. 5). We expected the algorithm to select only the food-nuts. Unluckily one translation in German for "nut" is "Mutter", which means hardware nut, but also mother. Since mothers do also eat (and there is a lot written about this in the documents indexed by Google) the context check found this to be the relevant translation for the German subset. The same polyseme, however, does not exist in French, Spanish or Portuguese. Therefore the overall performance of the algorithm is stable as long as one or more unaffected languages are used.

Table 2 shows all the translations of the 15 classes used in the following sections.

[1] We used the publicly available code http://homes.cs.washington.edu/~lfb/software/hmp/index.htm distributed by the authors.



Cut

	coupe	reduction	coupure	entaille
Clou	9170	53	486	26
Ongle	149000	419	166	71

Paint

	peindre	couvrir de peinture	decrire	faire de la peinture
Clou	127	0	5	5
Ongle	2870	0	2	2

Hit

	frapper	rencontrer	toucher	porter
Clou	6650	31	45	3560
Ongle	288	9	386	146

Pin

	epingler	goupiller	clouer	cheviller
Clou	3	0	9380	0
Ongle	6	0	1830	0

Figure 4. Context check for the class "nail" in different contexts. Shown are the number of matches from Google Text Search for the French set. Top rows correspond to responses for the hardware-nail (clou) and bottom rows refer to the fingernail (ongle). Using the noun which gets the most matches (highlighted in red) the algorithm manages to retrieve the fingernail in the context of "cut" and "paint", whereas the contexts "hit" and "pin" lead to the hardware nail being selected.

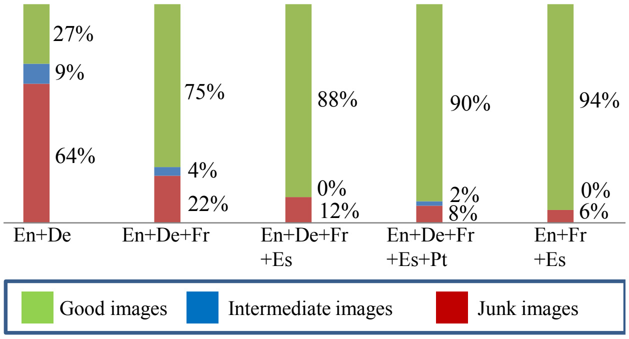


Figure 5. Quality distribution using the first 80 images of the *TC* generated sets for "nut (eat)" considering different languages depicted by their ISO639-1 language codes. Although the context check decided for the wrong translation in German, the algorithm only fails if no additional languages are used (En+De).

Class	Context	Translations
apple	cut	manzana, pomme, apfel
axe	chop	hacha, hache, axt
bolt	tighten	tornillo, boulon, schraube
cup	fill	taza, tasse
fork	prick	tenedor, fourchette, gabel
glass	fill	vaso, verre, glas
hammer	hit	martillo, marteau
nail	hit	clavo, clou, nagel
nut	tighten	tuerca, ecrou, mutter
oil	eat	aceite, huile, oel
orange	cut	laranja, naranja
pan	fry	sarten, poele, pfanne
peach	fry	molocoton, peche, pfirsich
pot	cook	cacerola, casserole, topf
saw	cut	sierra, scie, saege

Table 2. The final translations retrieved by the context check.

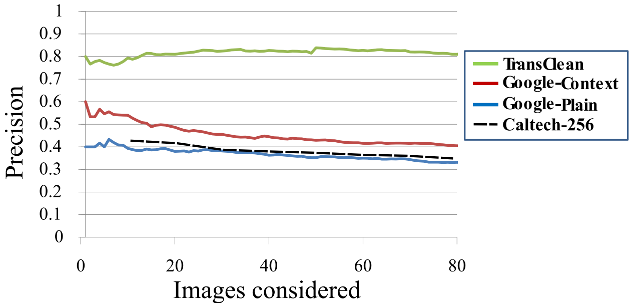


Figure 6. Average precision of the retrieved images (counting only *Good*) depending on the number of images considered. *TC* significantly outperforms the other methods. We included the Caltech-256 dataset curve [12] (dashed line, which closely resembles our results for Google-Plain) showing that our selected classes are representative of typical data.

5.2. Quality measure

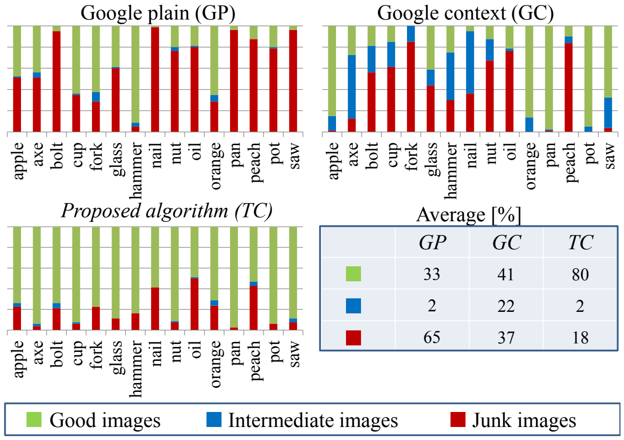
Figure 6 quantifies how precision of the sets changes depending on the number of images considered. On average, our algorithm improves Google results significantly, doubling the precision to about 80 % (in contrast to 41 %) for the first 80 images. While the average precision of our algorithm stays constant, *GP* drops by 10 % and *GC* even by 15 % when increasing the list length from 15 to 80. Remarkably, the averaged curve for Google-Plain resembles the curve found when creating the Caltech-256 dataset [12]. This confirms our notion that the 15 classes selected as a demonstration are representative of typical performance of unfiltered Google Image Searches.

Interestingly, for some classes like "hammer", plain Google Search without context outperforms Google Search with context. The reason for this is that using the context in a Google Search often retrieves images which are more related to the action than to the object itself. That is, while these images show the relevant object they often have a lot of clutter or the object is not visible at all (they will consequently be rated as *Intermediate* or *Junk*) as shown in Fig. 7. Searches for nouns without the context verb show a small fraction of *Intermediate* results, since images show either the correct object or not. Examples of results from the three methods can be seen in Fig. 8.

Our method retrieves on average 80 % *Good* images compared to 41 % returned by Google. Even the worst class "oil" shows 50 % *Good* images, which is 9 % better than the average quality of the Google searches. We shall show in the next section how much classifier performance can be boosted using training images returned by our TransClean algorithm.

5.3. Image classification

As a demonstration of the effectiveness of TransClean, we conducted an experiment to show the effect of different training sets on classifiers. We demonstrate that classifiers (see Sec. 4.2) are heavily dependent on the quality of the training sets and that our approach boosts them, resulting in better performance. To show this, we created five training sets: One training set generated by our algorithm (*TC*) and four sets consisting of plain Google search as well as context Google search results using the first 30 or 300 images (*GP30, GP300, GC30, GC300*).



Average [%]

	GP	*GC*	*TC*
🟩	33	41	80
🟦	2	22	2
🟥	65	37	18

Figure 7. Quality distribution for all classes in the contexts given in Table 1. We evaluated the 80 highest ranking images using the measures introduced by [10] (see Sec. 4.1). Full bars correspond to a 100 %.

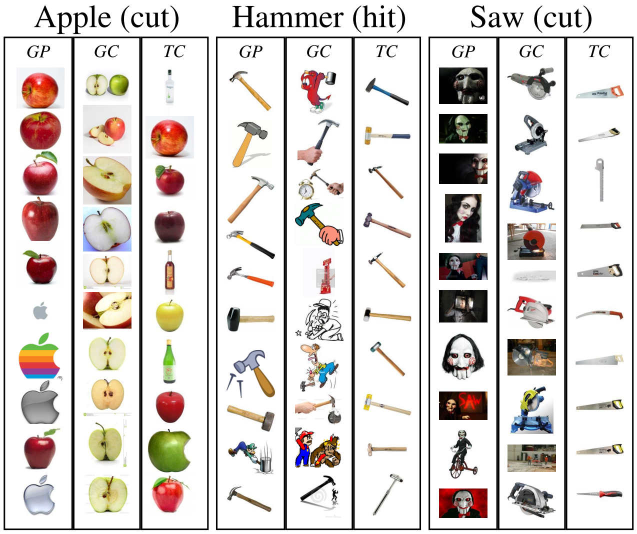


Figure 8. The top 10 images retrieved by Google Image Search (*GP*), Google Image Search with context (*GC*) and proposed TransClean algorithm (*TC*).

As can be deduced from Figure 9, the state-of-the-art classifier M-HMP is far superior to the classifiers based on SIFT and Bag-of-Words showing about 20 % better accuracy for the same training set. The M-HMP classifier achieved 61 % accuracy when using Google search images

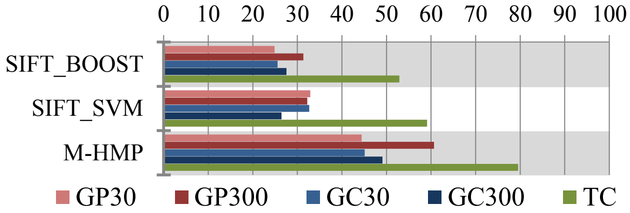


Figure 9. Performance impact of the training-sets on different classifier pipelines. All classifiers improve by roughly 20 % in accuracy using the *TC* set.

for only the noun (*GP*), 49 % when committing searches together with the verb (*GC*) and 80 % using images retrieved by our proposed method. Both Support-Vector-Machines and AdaBoost show poor performance across the Google search datasets. Nevertheless, using the TRANSCLEAN generated training sets improves performance across all classifiers by 20 %.

Figure 10 shows the per class recall and mean classification accuracy for the M-HMP. Remarkably, using the context for the search in general yields worse results than using the plain word for the search. The reason for this is the high percentage of *Intermediate* quality images in the *GC* sets (see Fig. 7). These images not only show the desired object in a bad way but typically also show other objects related to the action. This is far more destructive than *Junk* images, as the decision boundaries for one class may spread into the area of another class, *e.g.*, a hammer may be classified as a nail since it was shown in many nail (hit) images. Our method, in contrast, provides a high fraction of *Good* images and a low fraction of *Intermediate* images. This is clearly visible in the 19 % accuracy boost compared to the second best performance achieved (using *GP300*).

6. Conclusion

In this paper we presented an approach for unsupervised generation of training-sets for image classification algorithms. We proved using several experiments that the method outputs not only high quality sets of images when providing the class name and an action context but can also cope with polysemous classes. We also showed that while Google Image Search disambiguates quite well when provided the action context (*GC*), it suffers from a high fraction of images showing the object in a cluttered scene (*Intermediate* images). *GP* on the other hand shows a high fraction of images unrelated to the desired meaning. This is in agreement with the results reported by Griffin *et al.* [12]. Remarkably, *Intermediate* images are often far more destructive for a classifier's performance than *Junk* images, as wrong objects shown consistently with the right object may be learned instead. Our approach, however, yields few *Intermediate* images (2 %) and a high fraction of *Good* images (80 %).

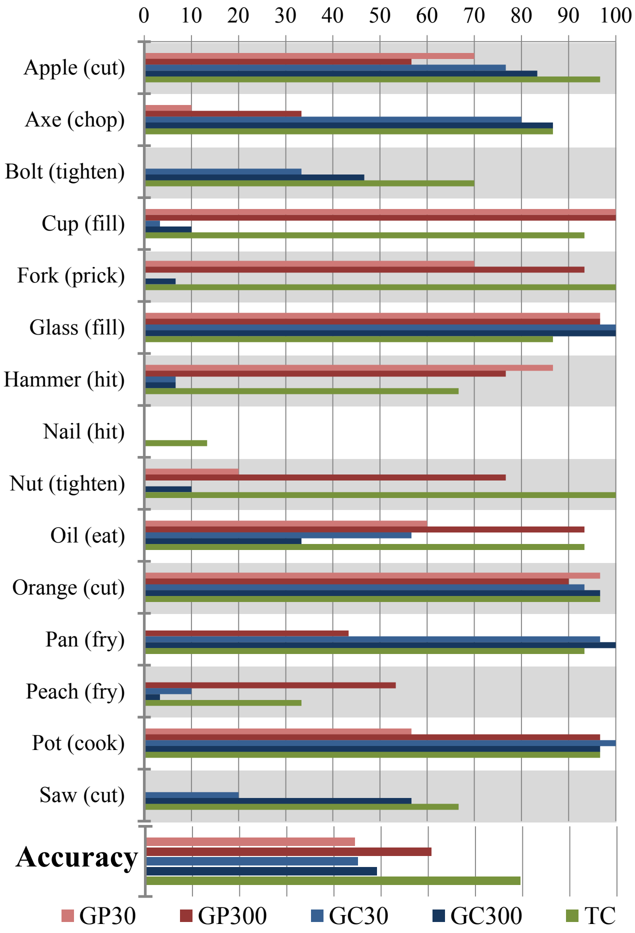


Figure 10. Per class recall and accuracy in percent of the M-HMP classifier using different training sets.

For this paper we provided manually entered search terms (which is one of the reference applications), but this algorithm can be adapted to extract needed noun+verb (or noun+adjective) tuples from autonomous and/or interactive robotic systems, since a context is usually available as part of a spoken or written command. Since we combine multiple languages the algorithm is robust against incorrect translations or context check failure in single languages as shown in Fig. 5. Using very class-specific actions will result in better context check and therefore algorithm performance. Conversely, one potential pitfall is that very general actions like "put", "take" and "place" may not show an improvement due to their context being applicable to many objects. This is a general problem however, as even humans are unable to disambiguate a class based on very general context.

Due to the importance of image classifiers for many modern applications and their need for large, high quality training sets we hope that this method will be useful for researchers in various fields.

Acknowledgment

The research leading to these results has received funding from the European Community's Seventh Framework Programme FP7/2007-2013 (Programme and Theme: ICT-2011.2.1, Cognitive Systems and Robotics) under grant agreement no. 600578, ACAT.

References

[1] E. Agirre, O. López de Lacalle, and A. Soroa. Random walks for knowledge-based word sense disambiguation. *Comput. Linguist.*, 40(1):57–84, Mar. 2014. 2

[2] C. Banea and R. Mihalcea. Word sense disambiguation with multilingual features. In *Int. Conf. on Computational Semantics*, pages 25–34, 2011. 2

[3] M. Beetz, U. Klank, I. Kresse, A. Maldonado, L. Mösenlechner, D. Pangercic, T. Rühr, and M. Tenorth. Robotic Roommates Making Pancakes. In *IEEE-RAS Int. Conf. on Humanoid Robots*, pages 529–536, 2011. 1

[4] D. Benbouzid, R. Busa-Fekete, N. Casagrande, F.-D. Collin, and B. Kégl. Multiboost: A multi-purpose boosting package. *Journal of Machine Learning Research*, 13:549–553, 2012. 5

[5] T. L. Berg and D. A. Forsyth. Animals on the web. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 1463–1470, 2006. 2

[6] L. Bo, X. Ren, and D. Fox. Multipath sparse coding using hierarchical matching pursuit. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 660–667, 2013. 1, 2, 5

[7] M. Bollini, S. Tellex, T. Thompson, N. Roy, and D. Rus. Interpreting and executing recipes with a cooking robot. In *Experimental Robotics*, volume 88 of *Springer Tracts in Advanced Robotics*, pages 481–495. Springer Int. Publishing, 2013. 1

[8] G. Csurka, C. R. Dance, L. Fan, J. Willamowski, and C. Bray. Visual categorization with bags of keypoints. In *Workshop on Statistical Learning in Computer Vision, ECCV*, pages 1–22, 2004. 1, 3

[9] R. Deits, S. Tellex, P. Thaker, D. Simeonov, T. Kollar, and N. Roy. Clarifying commands with information-theoretic human-robot dialog. *Journal of Human-Robot Interaction*, 2(2):58–79, 2013. 1

[10] R. Fergus, L. Fei-Fei, P. Perona, and A. Zisserman. Learning object categories from google's image search. In *IEEE Int. Conf. Computer Vision*, volume 2, pages 1816–1823, 2005. 1, 2, 4, 6

[11] R. Fergus, P. Perona, and A. Zisserman. A visual category filter for google images. In *Europ. Conf. Computer Vision*, pages 242–256, 2004. 1, 2

[12] G. Griffin, A. Holub, and P. Perona. Caltech-256 object category dataset. 2007. 1, 4, 6, 7

[13] S. Guadarrama, L. Riano, D. Golland, D. Gouhring, Y. Jia, D. Klein, P. Abbeel, and T. Darrell. Grounding spatial relations for human-robot interaction. In *IEEE/RSJ Int. Conf. on Intelligent Robots and System (IROS)*, pages 1640–1647, 2013. 1

[14] M. Guillaumin, J. Verbeek, and C. Schmid. Multimodal semi-supervised learning for image classification. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2010. 2

[15] W. Guo and M. Diab. Combining orthogonal monolingual and multilingual sources of evidence for all words wsd. Annual Meeting of the Association for Computational Linguistics, pages 1542–1551. Association for Computational Linguistics, 2010. 2

[16] G. E. Hinton and S. Osindero. A fast learning algorithm for deep belief nets. *Neural Computation*, 18, 2006. 1

[17] H. Holzapfel, D. Neubig, and A. Waibel. A dialogue approach to learning object descriptions and semantic categories. *Robotics and Autonomous Systems*, 56(11):1004–1013, 2008. 1

[18] P. Iravani, P. Hall, D. Beale, C. Charron, and Y. Hicks. Visual object classification by robots, using on-line, self-supervised learning. In *IEEE Int. Conf. on Computer Vision Workshops (ICCV Workshops)*, pages 1092–1099, 2011. 5

[19] P. Kaiser, M. Lewis, R. P. A. Petrick, T. Asfour, and M. Steedman. Extracting common sense knowledge from text for robot planning. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 3749 – 3756, 2014. 1

[20] T. Kulvicius, I. Markelic, M. Tamosiunaite, and F. Wörgötter. Semantic image search for robotic applications. In *Int. Workshop on Robotics in Alpe-Adria-Danube Region (RAAD)*, 2013. 2, 3

[21] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 60:91–110, Nov. 2004. 1, 3

[22] T. Miller, C. Biemann, T. Zesch, and I. Gurevych. Using distributional similarity for lexical expansion in knowledge-based word sense disambiguation. In *Proc. of COLING*, 2012. 2

[23] M. Muja, R. B. Rusu, G. Bradski, and D. G. Lowe. REIN-A fast, robust, scalable REcognition INfrastructure. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2011. 1

[24] R. Navigli and S. P. Ponzetto. Joining forces pays off: Multilingual joint word sense disambiguation. In *Joint Conf. on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1399–1410, 2012. 2

[25] F. Perronnin and C. Dance. Fisher kernels on visual vocabularies for image categorization. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, 2007. 1

[26] M. Schoeler, S. C. Stein, A. Abramov, J. Papon, and F. Wörgötter. Fast self-supervised on-line training for object recognition specifically for robotic applications. In *VISAPP*, 2014. 1

[27] F. Schroff, A. Criminisi, and A. Zisserman. Harvesting image databases from the web. In *IEEE Int. Conf. on Computer Vision (ICCV)*, pages 1–8, 2007. 2

[28] Y. Sun, L. Bo, and D. Fox. Attribute based object identification. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2013. 1

[29] M. Tenorth, U. Klank, D. Pangercic, and M. Beetz. Web-enabled robots. *Robotics Automation Magazine, IEEE*, 18(2):58–68, June 2011. 1